

PhD Thesis

Virtual Kidney

Full-scale reconstruction and simulation of the renal nephron-vascular network

Peidi Xu

Supervisors: Sune Darkner & Kenny Erleben

Co-supervisors: Olga Sosnovtseva & Niels-Henrik Holstein-Rathlou

Submitted: April 14, 2024

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen

大人者，不失其赤子之心者也。

— 《孟子·离娄下》

Abstract

The renal vasculature, functioning as a resource distribution network, plays an essential role in the kidney's physiology and pathophysiology. The diagnosis of vascular diseases such as renal artery stenosis can usually be achieved through medical scanning of the large renal arteries. However, the renal filtration and autoregulation mechanisms take place in the smallest terminal arteries in the renal arterial tree, also known as afferent arterioles, which are beyond the resolution of most existing medical imaging modalities. This limitation of medical imaging and the fundamental function of afferent arterioles makes data-driven generative models an important tool for studies of the regulation of renal blood flow. To better understand the regulatory mechanisms of renal blood flow, this PhD thesis presents a novel hybrid approach to reconstruct a full-scale renal arterial network. The hybrid approach reconstructs small arteries in the full-scale arterial tree using a physiologically based optimization method, while also integrating vascular geometries extracted from medical images. Then, models of nephrons, where the renal filtration and the control renal hemodynamics takes place, are attached to each afferent arteriole of the reconstructed full-scale arterial tree. With the full-scale nephron-vascular model, autoregulatory mechanisms are simulated to define how topological and hemodynamic profiles of microcirculatory networks are optimized and adapted to pathological changes. Finally, although most of the blood vessels in the full-scale model are reconstructed physiologically, segmenting visible large blood vessels from medical scans is still a crucial step for guiding the physiological reconstruction of the remaining small vessels. Therefore, the PhD thesis also includes medical image segmentation, and the related deep learning approaches, which have become state-of-the-art methods for automatic segmentation tasks.

Dansk Resumé

Nyrekarene, der fungerer som et ressourcefordelingsnetværk, spiller en afgørende rolle i nyrens fysiologi og patofysiologi. Diagnosen af vaskulære sygdomme såsom nyrearteriestenose kan som regel opnås ved scanning af de store nyrearterier. Men de renale filtrerings- og autoreguleringsmekanismer sker i de mindste terminalarterier i nyrearterietræet, de afferente arterioler, der ikke kan visualiseres af de eksisterende medicinske billedmodaliteter. Denne begrænsning sammen med de afferente arterioler fundamentale betydning for nyrefunktionen, gør datadrevne generative modeller til et vigtigt værktøj for undersøgelser af reguleringen af nyrens blodgennemstrømning. I denne sammenhæng præsenterer denne ph.d.-afhandling en ny hybrid tilgang til at rekonstruere en komplet model af nyrens arterielle kartræ. Denne hybrid tilgang rekonstruerer de små arterier i det arterielle netværk ved hjælp af en fysiologisk baseret optimeringsmetode, samtidig med at den integrerer kargeometrier hentet fra medicinske scanninger af nyren. Herefter tilknyttes der modeller af nefroner, hvor den renale filtrering sker og kontrollerer renal til hver afferent arteriole i det rekonstruerede arterielle kartræ. Det er nefronerne der er ansvarlige for dels filtrationen, og dels for reguleringen af nyrens blodgennemstrømning. Ved hjælp af det arterielle netværk og modellerne af nefronerne simuleres den renale autoregulering af gennemblødningen for at undersøge, hvordan topologiske og hæmodynamiske profiler i mikrocirkulatoriske netværk optimeres og tilpasses til patologiske ændringer. Selvom de fleste blodkar i karmodellen er rekonstrueret fysiologisk, er segmentering af de større, synlige blodkar fra medicinske scanninger stadig et afgørende skridt, som fundament for den fysiologiske rekonstruktion af de resterende kar. Denne ph.d.-afhandling omhandler derfor også segmentering af medicinske scanninger af nyrekar, og de relaterede maskinlæringsmetoder, der er blevet state-of-the-art metoder til automatiske segmenteringsopgaver.

Acknowledgements

The past three years have been a remarkable journey for me, both scientifically and personally. I would not say that it has been the most joyful or bountiful time of my life, but it is no exaggeration to say that it has been the most tranquil time in my life, a period where I could immerse myself in the topics of my interest, free from the cacophony of daily distractions. This tranquility I found in Denmark has infused me with renewed energy, optimism, and even a sense of youthful vigor that I can carry forward.

I would like to start by thanking my supervisors at DIKU (Department of Computer Science, University of Copenhagen), Sune Darkner and Kenny Erleben. Their expansive knowledge and astute perspectives have been the lighthouse guiding my research voyage. Their trust and support were the bulwark against the disheartenment of paper rejections. Beyond their technical prowess, they have imparted life lessons to me, a gift for which I am immensely grateful. My appreciation also extends to the other professors at DIKU, whose intellectual exchanges and support have been nothing short of stimulating.

My sincerest thanks also go to my medical advisors from SUND (Faculty of Health and Medical Sciences, University of Copenhagen), Niels-Henrik Holstein-Rathlou and Olga Sosnovtseva. They have provided tremendous help in my interdisciplinary research, offering indispensable insights into the complex weave of physiology and modeling. I would have never finished or even started this fascinating research without their expert insights into the physiological aspects.

Friendship is a sanctuary, more so in a land far from home. To my friends at DIKU, the warmth of shared hotpot meals is the cherished memory that dispels the chill of the prolonged Danish winters. I would also like to thank my friends at SUND, my friends from my hometown, and the friends I made during my master's at DIKU. They have each been a beacon of comfort and camaraderie in a foreign land.

Concluding these acknowledgments, I reserve my deepest gratitude for my parents in China. Their love, unbounded by conditions and unmatched in generosity, has been my guiding star. It is their unwavering support that fuels my resolve and propels me onward.

Contents

Abstract	5
Dansk Resumé	7
Acknowledgements	9
Contents	10
1 Introduction	13
1.1 Anatomy of the kidney	13
1.2 Mechanisms of renal autoregulation	15
1.3 Topology of renal arterial network	17
1.4 Coventional computational fluid dynamics	18
1.4.1 Navier-Stokes equation	19
1.4.2 Idealized general pipeline of CFDs	19
1.4.3 Limitation and challenges to our task	21
1.5 Deep-learning based image segmentation	22
1.5.1 Image segmentation and deep learning approach	24
1.5.2 Large language and vision models	25
1.6 Our approach	26
2 Modeling and reconstruction of renal arterial network	29
2.1 Skeleton-tree-based model of vasculature	29
2.2 Hagen–Poiseuille equation of a simplified flow	30
2.3 Constructive optimization algorithms	31
3 Full-scale model and simulation of renal autoregulation	35
3.1 Model formulation and validation	36
3.2 Simulation of pathologies	38
4 Segmentation of renal blood vessels	41
4.1 Reconstructed vascular tree as training data	42
4.1.1 Building segmentation maps via implicit surface	42
4.1.2 Building scans via domain adaptation	46

5	Segmentation of human hip joints and jaw bones	49
6	Final remarks	53
6.1	Conclusion and potential impact	53
6.2	Discussion: Limitations and future perspectives	54
6.2.1	Reconstruction vs experimental data	54
6.2.2	Segmentation of renal blood vessels	55
6.2.3	Dynamics of the model and interactions between nephrons	57
6.2.4	Construction of computational meshes for CFD studies	58
	Bibliography	61
	List of publications	71
A	Paper A	73
B	Manuscript B	89
C	Paper C	105
D	Paper D	117
E	Paper E	129

Chapter 1

Introduction

Chronic kidney disease (CKD) has become a major cause of mortality and morbidity in the 21st century, affecting more than 10% of the general population worldwide [1]. This is due in part to the increase in risk factors such as diabetes, obesity, and hypertension [1]. Although CKD may be caused by several mechanisms, dysfunction of and damage to the renal vasculature appears to play a prominent role in CKD associated with diabetes and hypertension [2, 3]. Current evidence suggests that especially an impaired autoregulation of renal blood flow is a culprit in the development of diabetic and hypertensive nephropathy [2, 4, 5]. Autoregulation minimizes the transmission of changes in the arterial pressure along the renal vascular network, thereby preventing pressure-induced damage to the renal microcirculation [2, 6]. In the face of rising CKD cases globally, understanding the intricacies of renal function is more crucial than ever.

Among all the components in the kidney, the vasculature, especially the arterial network, plays a unique role in the autoregulation mechanism. Not only is it the main effector of autoregulation, but it also functions as a resource distribution network, supplying the individual nephrons with blood and nutrients, and constitutes a communication network, allowing contiguous nephrons to interact through electric signaling along the vessels [7].

To better understand the regulatory mechanisms of renal blood flow, including autoregulation, this PhD thesis presents an innovative computational model that reconstructs and simulates the full-scale renal nephron-vascular network, offering detailed insights into the renal vascular topology and kidney autoregulation mechanisms under various physiological conditions.

1.1 Anatomy of the kidney

The kidney is a vital organ essential for maintaining fluid and electrolyte balance, filtration of plasma, and the elimination of waste products[8]. It exhibits a complex structure comprising distinct regions, each contributing to its over-

all function (Fig. 1.1A). The outermost layer is the renal cortex, which houses the major parts of the nephrons (Fig. 1.1A&B) – the functional units responsible for filtration, tubular reabsorption/secretion, urine production, and autoregulation of the blood flow. The human kidney contains 1 - 1.2 million nephrons, whereas the rat kidney, which is the target of the present work, contains approximately 30,000 nephrons. Deep within the kidney lies the renal medulla. The renal medulla contains the structures that are necessary for the creation of concentrated urine, including the loops of Henle and the medullary collecting ducts (Fig. 1.1A&C). Lastly, the renal pelvis serves as a collecting chamber where urine from the nephrons converges before reaching the urinary bladder via the ureter (Fig. 1.1A).

The vasculature of the kidney is integral to its function, ensuring proper perfusion of and filtration in the individual nephrons. The arterial supply of blood to the kidney originates from the renal artery that arises from the abdominal aorta. Inside the kidney it divides into segmental arteries, then into interlobar arteries traveling within the renal columns (Fig. 1.1A). From there, blood flows into arcuate arteries at the corticomedullary junction, which through the interlobular arteries supplies the cortex with blood. Afferent arterioles represent the smallest arteries in the network, and one afferent arteriole supplies a single glomerulus (Fig. 1.1A&B). Efferent arterioles emerge from glomeruli, forming peritubular capillaries in the cortex and vasa recta in the medulla. Venous drainage occurs through interlobular veins, arcuate veins, and ultimately, the renal vein, which exits the kidney to drain into the inferior vena cava [9] (Fig. 1.1B).

At the core of the renal vasculature is the afferent arterioles, which are the terminal blood vessels in the renal arterial tree that supply blood to each nephron. Because of their small diameter, they represent the major part of the hemodynamic resistance of the renal vascular network. The nephron, as the functional unit of the kidney, together with the afferent arteriole, is responsible for the autoregulation of blood flow.

A nephron consists of a renal corpuscle, including Bowman's capsule that surrounds the glomerulus, and renal tubules, such as the proximal convoluted tubule, loop of Henle, distal convoluted tubule, and collecting duct (Fig. 1.1C). Blood flows from the afferent arteriole into the glomerular capillaries, and leaves the glomerulus through the efferent arteriole (Fig. 1.1B). As blood flows through the capillaries, fluid is filtered through the capillary walls and enters the renal tubule from Bowman's capsule. The capillary walls are permeable to water and small solutes (predominantly NaCl), but impermeable to the plasma proteins. Consequently, the filtrate contains all the small solutes dissolved in the plasma, whereas the plasma proteins are retained within the capillaries. The rate at which blood is filtered in the glomeruli and enters the renal tubule, also known as the glomerular filtration rate (GFR), is an important measure of overall kidney function and is a function of the renal blood flow and arterial pressure. If too low the kidney will fail to eliminate waste products, and if too

high there will be a loss of essential compounds, like NaCl and water, through the kidneys.

In the tubule (Fig. 1.1C), the filtrate first traverses the proximal convoluted tubule (PCT), where cuboidal epithelial cells, rich in microvilli, actively reabsorb essential nutrients, ions, and water back into the bloodstream. As the filtrate progresses, it passes through the loop of Henle, a crucial segment featuring a descending limb permeable to water and an ascending limb impermeable to water but conducive to ion transport, establishing a concentration gradient essential for subsequent reabsorption of water in the collecting duct. Subsequently, the distal convoluted tubule (DCT) fine-tunes urine composition through selective ion reabsorption and acid-base balance regulation. Finally, the collecting duct adjusts urine concentration based on the hydration status, either reabsorbing water under the influence of antidiuretic hormone (ADH) or allowing for its excretion to produce dilute urine.

1.2 Mechanisms of renal autoregulation

The circulatory system, including the renal blood flow, is controlled by homeostatic mechanisms of autoregulation. The hemodynamic response continuously monitors and adapts to conditions in the body and its environment. In the kidney, autoregulation of renal blood flow and glomerular filtration rate (GFR) play a critical role in maintaining renal function despite fluctuations in systemic blood pressure [10]. This process involves the intrinsic ability of the afferent arterioles to adjust their diameter in response to changes in blood pressure, ensuring stable renal perfusion. Specifically, the myogenic mechanism and tubuloglomerular feedback mechanism contribute to autoregulation. The myogenic mechanism is intrinsic to the vessels, and induces vasoconstriction of afferent arterioles in response to increased blood pressure, limiting renal blood flow. Conversely, vasodilation occurs in response to decreased blood pressure, maintaining adequate renal perfusion. The tubuloglomerular feedback mechanism involves the release of vasoactive substances by the macula densa in the distal convoluted tubule in response to changes in sodium chloride (NaCl) concentration, regulating GFR by altering arteriolar resistance.

Despite intense research, the mechanisms that underlie the pathways to chronic kidney disease (CKD) remain poorly understood. However, it is known that impaired autoregulation may lead to vascular damage in the renal microcirculation, and thereby contribute to the development of CKD [2, 4, 5]. A major obstacle to the study of the role of autoregulation in CKD is the fact that the autoregulatory mechanisms act at the nephron level, and the complex interplays among the nephrons are poorly understood because modern experimental technologies cannot assess hemodynamics in the deeper regions of the kidney.

In the literature, there exists a strong foundation of mathematical mod-

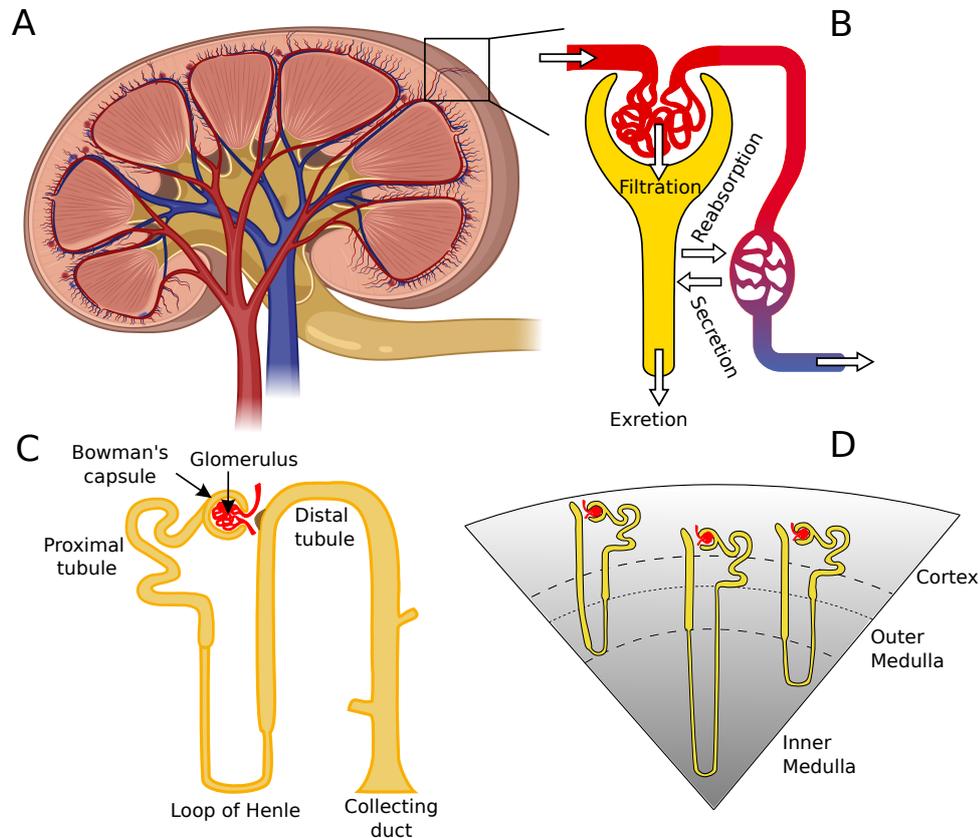


Figure 1.1: (A) Kidney anatomy: cortex and medulla exhibit distinct morphological and functional features (created with BioRender.com); (B) Renal microcirculation with afferent/efferent arterioles, glomerular capillaries, and vasa recta; (C) Nephron – the functional unit of the kidney; (D) Three populations of nephrons are distinguished: superficial, midcortical, or juxtamedullary based on the location of their glomerulus within the cortex. (D) is also in manuscript B.

eling and simulation of the autoregulation mechanisms of blood flows in a single nephron [11, 12, 13, 14, 15]. As the number of nephrons is known [16, 17], the single model can be scaled up to represent the whole kidney, assuming that the nephrons work independently. However, the afferent arterioles, as the feeding blood vessel to the nephron, are all connected in the renal arterial tree, meaning that the nephrons function together rather than independently.

As a result, a detailed description and modeling of renal vasculature topology that reaches and connects all the afferent arterioles and nephrons together is necessary for a complete simulation of the blood flow and autoregulation mechanisms in the kidney. This modeling and simulation will be crucial for

understanding both the physiology and pathophysiology of the kidney, such as early identification of progressive kidney disease and drug designs. Unfortunately, most medical scanning devices have limited spatial resolution, and can only resolve the first few large branches of the renal vasculature. Although the first few large branches can provide diagnostic information on renal conditions such as renal artery stenosis [18, 19, 20, 21, 19, 22], more detailed information on the renal microvasculature is necessary for the study of the mechanisms of renal autoregulation, which takes place at the level of the afferent arterioles and nephrons.

1.3 Topology of renal arterial network

Despite the lack of dynamical observations of the far-surface tissues of the kidney and a holistic observation of the renal arterial tree, there exists a large amount of statistical data retrieved from high-resolution scans of the far-surface patches [16, 7, 23, 24]. Nordsletten et al. [16] provided the hitherto most detailed and quantitative description of the rat renal vasculature. They combined high ($4 \mu m$) and low ($20 \mu m$) resolution micro-CT images obtained from a vascular cast of a rat kidney. They used the skeletonization method to trace the path of contiguous vessels and then applied the Strahler approach [25] to sort and interpret the data. Although they do not have a full-scale renal arterial tree structure, their main contribution is their results of detailed morphometric property, such as radius and length distribution and branching patterns, especially on the afferent arterioles.

Marsh et al. [23] used micro-CT scans with $2.5 \mu m$ resolution to examine the three-dimensional microvascular structure of the rat renal arterial tree. The cast revealed an arterial tree stemming from arcuate arteries, with as few as twice or as many as six branchings before reaching a terminal artery. These terminal arteries terminate in pairs, triplets, or quadruplets of afferent arterioles [23]. They also observed various patterns in the origins of afferent arterioles, either arising from nonterminal arteries at different branch orders or directly from terminal arteries that formed the apexes of the arterial tree. Other research groups have documented similar branching configurations through the analysis of microdissected arterial trees from four distinct mammalian species [26, 27, 28]. Postnov et al. [24] have demonstrated that the pressure drop in a simple bifurcating tree with the vessel dimensions reported by Nordsletten et al. [16] surpasses the experimentally observed value. This is expected since, in a simple bifurcating tree, afferent arterioles appear only at the terminal branch points of the tree, which is an assumption that maximizes the hemodynamic resistance between the renal artery and the glomerulus. Postnov et al. [24] and Marsh et al. [23] have also reported an exponential distribution of the distances between branch points for afferent arterioles throughout the vascular tree. This distribution, coupled with

the possibility of branching from any arterial segment, forms the foundation for the pressure within the glomerular capillaries to be significantly elevated, falling within a range conducive to normal nephron function, unlike in a simple bifurcating tree.

In summary, the lack of a full-scale observation of the renal arterial tree but with a large amount of statistical data retrieved from high-resolution scans of the far-surface patches [16, 7, 23, 24] motivates data-driven modeling and biosimulation methods that can reconstruct a full-scale renal vasculature by combining the extraction of tissue geometry from medical data with generative models that can extrapolate modeling to unseen parts of the kidney.

1.4 Coventional computational fluid dynamics

Computational Fluid Dynamics (CFDs) is the conventional method to simulate blood flows and dysfunctions over vasculatures, which have been widely used in Cardiovascular studies such as coronary artery disease (CAD) and aortic aneurysm [29, 30, 31, 32, 33]. As a branch of fluid mechanics, CFDs employ numerical methods and algorithms to analyze and solve complex fluid flow problems based on the fundamental principles of fluid dynamics such as the Navier-Stokes equation. The Navier-Stokes equation is a set of non-linear, partial differential equations that describe the motion of fluid substances, such as liquids and gases, and, in our case, blood flows. However, when dealing with complex three-dimensional geometries, these non-linear, partial differential equations cannot be solved analytically. Therefore, numerical methods such as the finite volume or finite element method are commonly employed to address the discretized version of these equations. In these numerical methods, the Navier-Stokes equations defined above serve as “governing equations”, which are integrated across all domain elements and subsequently transformed into a system of non-linear algebraic equations. This system of non-linear algebraic equations, often in the order of millions, is typically solved through iterative methods using computer workstations or high-performance computing clusters [34].

In the context of blood vessels, CFD simulations offer valuable insights into hemodynamics studies. In the renal vascular studies, CFDs have been mostly only focusing on the renal artery (the single root vessel in the renal arterial tree) to study renal arteries stenosis (RAS) [18, 19, 20, 21, 19, 22], which is the narrowing of renal arteries, most often caused by atherosclerosis or fibromuscular dysplasia. By employing CFD, it becomes possible to assess the impact of different physiological and pathological conditions on blood flow and help researchers and clinicians investigate various aspects of blood flow, including the formation of potential flow disturbances like thrombosis or atherosclerosis.

In this section, we briefly present the Navier-Stokes equations and detail

the steps in standard CFD studies of blood flow. We then explain in detail why the standard pipeline of CFDs cannot be adopted in our studies to simulate blood flow in the renal arterial network or the renal autoregulation mechanisms.

1.4.1 Navier-Stokes equation

CFD-based blood flow simulations are focused on using numerical methods to solve the Navier-Stokes equation, which describes the motion of fluid substances and consists of two components, namely the conservation of mass (continuity equation) and the conservation of momentum.

The first component of the Navier-Stokes equation is the conservation of mass equation, also known as the continuity equation. It states that the rate of change of mass within a fluid element is equal to the rate of mass flow (flux) into the volume.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (1.1)$$

where ρ is the fluid density, \mathbf{u} is the fluid velocity vector.

The conservation of momentum equation in the Navier-Stokes equation describes the relationship between acceleration, pressure, viscous forces, and body forces acting on a fluid element. It states that the rate of change of momentum within a fluid element equals the rate of momentum flow (flux) into the volume plus the momentum rate due to other forces.

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g} \quad (1.2)$$

where p is the pressure, μ is the dynamic viscosity and $\nabla^2 \mathbf{u}$ is the Laplacian of the velocity vector \mathbf{u} and \mathbf{g} is the acceleration due to gravity.

1.4.2 Idealized general pipeline of CFDs

Ideally, a CFD-based simulation of blood flow should, in principle, involve the following steps [35]. A toy illustration of applying CFD on our renal arterial vasculatures is shown in Fig. 1.2.

Medical imaging The initial input to perform CFDs is the medical imaging data of blood vessels, such as computed tomography (CT) scans, magnetic resonance imaging (MRI), ultrasound, and X-ray angiography. Most of them are non-invasive and are only used to extract geometries of blood vessels but do not measure any hemodynamic properties.

Geometry reconstruction (Vessel segmentation) The three-dimensional (3D) geometries of blood vessels need to be reconstructed from medical images. This usually involves a semantic segmentation of the blood vessels.

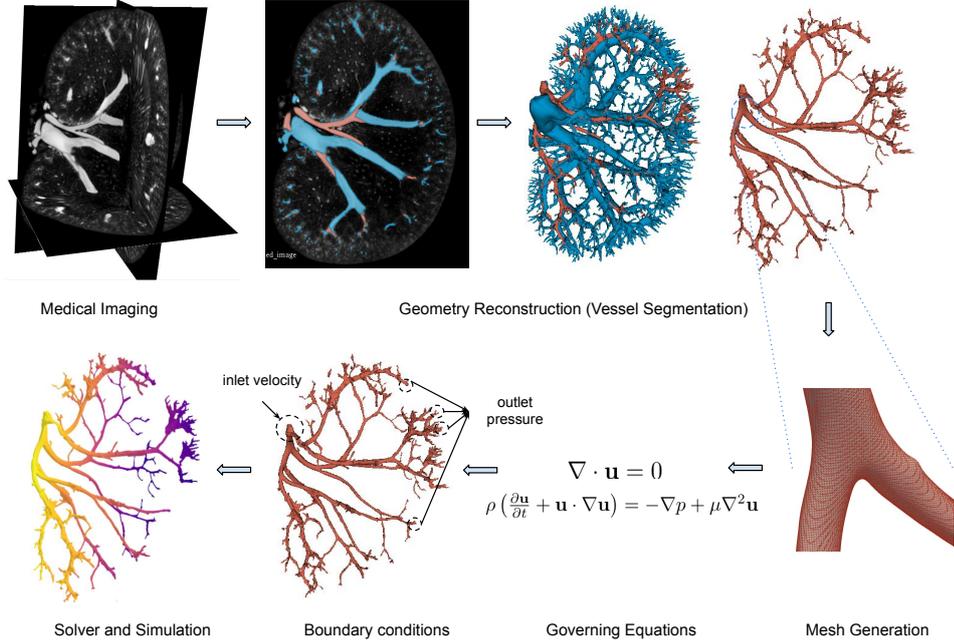


Figure 1.2: An toy illustration of the idealized CFD pipeline applied over the renal arterial network. It starts by segmenting blood vessels from a micro-CT scan of a rat kidney. While only arteries (colored red) are preserved and reconstructed for further blood flow simulations, veins (colored blue) are also shown in the segmentation for completeness. For boundary conditions, a common practice is to define a single inlet velocity profile and all the outlet pressure fields, of which only some are circled for illustration purposes. In the final result, the idealized pipeline should output the rendered pressure or flow field. Here, each node is simply colored by its coordinates for a toy illustration.

Sophisticated segmentation approaches such as level-set segmentation [36, 37] have been applied to reconstruct 2D and 3D patient-specific coronary models [38, 39]. The advance of deep learning also makes auto-segmentations of blood vessels possible, which most often requires adequate training data with pre-segmented label maps [40, 41, 42]. Nonetheless, it still remains challenging to integrate deep learning models into a complete pipeline of CFDs since their performance heavily depends on the available training data and image qualities.

Mesh generation A computational grid, or mesh, is generated over the reconstructed geometry to divide the domain into discrete elements. Ideally, surface meshes can be generated by applying marching cubes [43] over segmentation results, while tetrahedralization methods such as fTetWild [44] is

usually applied over the surface meshes to produce volume meshes. Certain surface corrections and mesh repairs are also necessary to capture the significant hemodynamic behavior of the modeled section [35].

Governing equations The Navier-Stokes equations introduced previously in Section 1.4.1 are used as the governing equations in blood flow simulations, which need to be solved numerically over the generated mesh from the previous step. These equations need to be modified to account for the non-Newtonian properties of blood, such as varying viscosity and elasticity, e.g., Carreau model [18, 20].

Boundary conditions To enable CFD analysis, the physiological conditions at the wall and inlet/outlet boundaries need to be delimited and specified. Appropriate boundary conditions are crucial to mimic the physiological conditions of the blood vessels being studied. They may be retrieved from patient-specific data, population data, physical models, or assumptions. In hemodynamics, CFD usually requires an inflow velocity and outlet pressure information as the boundary condition, which are often adopted as constant values from the population data and thus do not require patient-specific information [31, 32, 33, 29, 30].

Solver and simulation Numerical algorithms, such as finite element or finite volume methods, are employed to discretize the governing equations and numerically solve the Navier-Stokes and continuity equations. The numerical algorithms reach convergence by gradually approaching a final solution through incremental steps.

Post-processing and analysis Once the simulation is complete, the CFD solver produces hemodynamic features such as the pressure distribution and velocity field over all elements at each time step, which can then be used to study blood flow characteristics and detect abnormal blood flows [35].

1.4.3 Limitation and challenges to our task

In our problem, we are given ex-vivo micro-CT scans of rat kidneys with an isotropic voxel size of $22.6 \mu m$. Ideally, our task can be solved by following the steps defined previously in Section 1.4.2. Specifically, as a toy example shown in Fig. 1.2, we start by segmenting blood vessels (renal arterial trees) from the ex-vivo micro-CT scans, followed by mesh reconstruction from the segmentation results using methods such as marching cubes [43] and fTetWild [44]. We then run numerical solvers over the constructed mesh with predefined boundary conditions to solve the Navier-Stokes equation, which should output hemodynamic properties such as velocity and pressure field (currently colored by its coordinates for a toy illustration in Fig. 1.2).

However, the following problems challenge the direct adoption of CFDs to our problem.

- **Coarse segmentation:** Although the kidney cast is filled with contrast agents, it is challenging to separate arteries, veins, and capillaries, which share almost the same intensity values (Fig 1.3b). Therefore, although it is relatively easy to differentiate vessels from the background, extracting a clean arterial tree from other vessel components is extremely difficult, especially near the cortex region where it is hard to tell the origin of specific branches (Fig 1.3b). It thus remains challenging and requires extremely intense manual corrections to acquire a clean segmentation of the blood vessels.
- **Inconsistent radius appearance:** The radius of blood vessels is a large variable between living and dead bodies. Thus, the ex-vivo micro-CT scan does not reflect the actual thickness of the renal blood vessels in a living rat. A standard radius computation from Euclidean distance transformation of the vessel centerline to its surface (Fig 1.3d) is thus not trustworthy.
- **Low scan resolution:** Although the micro-CT scan (Fig 1.3a-c) has a much higher resolution than traditional CTs ($22.6 \mu m$ vs $1 mm$), the resolution is still too low to capture the afferent arterioles, which have a mean radius of $10 \mu m$ [16].
- **Computational resources:** In CFDs, the number of equations that need to be solved numerically depends on the number of elements of the constructed mesh. Notably, a typical 3D cardiovascular simulation involves only 16 vessel branches from the aorta, while its simulation encompasses over 1 million discrete elements, executed across several cardiac cycles [30, 35]. Typically, achieving the solution of these millions of non-linear partial differential equations demands several days of computational effort [30, 35]. Such computational complexity clearly does not generalize to a full-scale renal arterial model that involves approximately 50K branches [16].

1.5 Deep-learning based image segmentation

In Section 1.4.2, we have demonstrated that the first essential step of general CFD-based blood flow simulation is to segment blood vessels from medical imaging. On the other hand, we have emphasized in 1.4.3 that a direct segmentation from the micro-CT scans cannot be used to build computational meshes of the full-scale arterial tree. Nonetheless, the segmentation of blood vessels from medical scans can still be crucial, both for visualization purposes

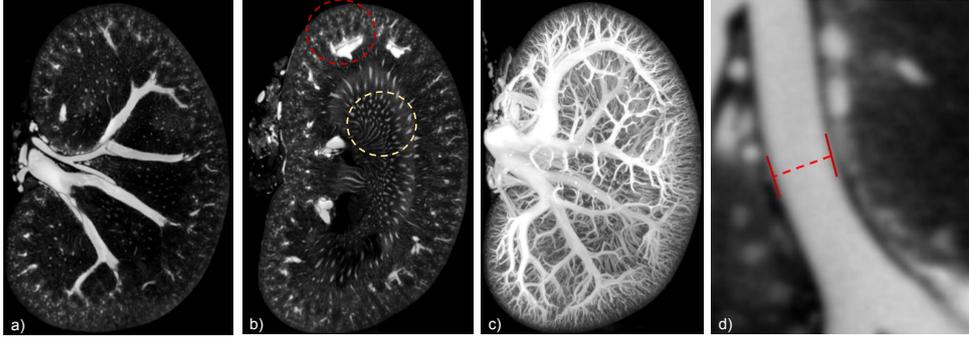


Figure 1.3: Example of our micro-CT scan of a rat kidney and related challenges. a) shows a particular slice where the large veins and arteries are mostly separable. But in most cases, as an example in b), the arteries and veins are extremely difficult to separate, especially near the cortex (red circle), and the segmentation of veins and arteries is further challenged by the existence of capillaries in renal medullar (yellow circle). c) is the maximum intensity projection (MIP) of the micro-CT scan, as the standard method to visualize blood vessels, which shows that although the micro-CT scan has a much higher resolution than traditional CTs, it is extremely noisy and insufficient to capture afferent arterioles. d) shows the standard radius estimation of the vessel from the scan, which is a large variable between living and dead bodies.

and for analyzing the general geometry of vascular trees and branching patterns, etc [16, 7, 45]. Moreover, as will be introduced in Chapter 2 and paper A, our proposed hybrid approach to reconstructing the renal arterial network integrates image priors by prebuilding an arterial tree, which requires the segmentation of the large visible vessels from the scan.

In general, deep learning has been widely used for automatic medical image segmentation, both on 3D blood vessels and many other organs, e.g., hip joints, teeth, and mandibles [42, 46, 47]. Therefore, part of the Ph.D. project is focused on automatic medical image segmentation with deep learning models. This section will briefly introduce medical image segmentation tasks and give a high-level overview of deep-learning approaches used to address such problems. Technical details about deep learning models such as the definition of convolution operation, loss function, and backpropagation will not be touched upon. Chapter 4 (paper C) will then present our approach of utilizing our reconstructed renal vascular trees from Chapter 2 (paper A) as training data to train a deep-learning renal vessel segmentation network. The following Chapter 5 (paper D and E) will then present our two related works on the deep-learning-based segmentation of human hip joints and human jaws, which involve an essential simulation-inspired component that is crucial for its downstream simulation tasks [48, 49].

1.5.1 Image segmentation and deep learning approach

Within the realm of medical imaging, image segmentation is a pivotal process offering a fundamental means of extracting precise information from complex visual data. At its core, segmentation involves the meticulous partitioning of a medical image into discrete regions or structures of interest, such as organs, lesions, or blood vessels. Accurate segmentation is paramount in medical diagnosis, treatment planning, and research, as it allows clinicians and researchers to precisely identify and analyze anatomical structures and pathological conditions. Traditional manual segmentation methods, though reliable, are often time-consuming and labor-intensive. In recent years, advancements in deep neural networks, especially convolutional neural networks, have facilitated the automation of this critical task, significantly enhancing efficiency and accuracy.

Convolutional Neural Networks (CNN) is the state-of-the-art deep learning model for visual imagery tasks. In standard CNN models, semantic information is gradually increased with hierarchies of features, while location information is gradually squeezed with smaller spatial dimensions, also known as downsampling [50, 51]. CNNs are powerful visual models and have been shown to be on par with human experts on whole-image classification [50] where the output is a single value. In the medical imaging field, this can be used to classify scans with tumor vs without tumor, fracture vs not fracture, or any kind of anomaly classification [52, 53].

Image segmentation is a particular task in that the model makes a pixel-wise prediction, which means that standard CNNs that operate downsampling need to be combined with its reverse process to increase feature map size. This upsampling process can be achieved by particular forms of convolution, such as transpose convolution. A fully convolutional network (FCN) [51] is such a network structure with downsampling and upsampling phases, which is an intuitive semantic segmentation model to make pixels-to-pixels output.

UNet [41] is a particular kind of FCN [51] used for pixel-wise prediction tasks, with the “U” name representing a symmetric architecture with two paths: encoding and decoding path. The encoding path is a traditional stack of convolutional and max pooling layers where the spatial information is reduced while feature information is increased, thus encoding deep features at each location. On the contrary, the decoding path is the symmetric expanding path, which is used to expand the feature map while merging the deep features generated from the encoding path. A distinctive feature of U-Net is the incorporation of skip connections that link layers in the encoding path with corresponding layers in the decoding path, which facilitates the transfer of spatial and contextual information.

UNet was initially proposed for 2D biomedical images where problems such as very few training images and touching borders of the instances need to be addressed [41]. Although proposed in 2015 [41], its simple yet effective

structure with skip connections still remains state-of-the-art architecture in the medical image segmentation field [54]. In [54], the authors showed that a correct and automated configuration for UNet can surpass most task-specific models on 23 public datasets used in international biomedical segmentation competitions.

1.5.2 Large language and vision models

Very recently, Large language models (LLMs) have revolutionized the Natural Language Processing community [55, 56, 57]. These models are primarily based on Transformers [58], which is a more recent neural network structure. These models are primarily designed for sequential data such as natural languages, which comprise a sequence of words. Conversely, the images are not sequential data by nature but can be regarded as sequences of smaller image patches. Based on this observation, researchers have designed Vision Transformers (ViTs) [59] which use the same principle on image patches to consider different parts of an image when encoding specific patterns.

Notably, besides the architecture, a more crucial factor that makes LLMs a huge success is that they are pre-trained on large online datasets. LLMs trained on large datasets have the generalization ability on tasks beyond those seen during training with zero or only a few new training data, also known as zero-shot or few-shot learning. This function often involves prompt engineering, where carefully designed text prompts are used to guide the language model to produce an appropriate textual response for the given task [60].

Inspired by LLMs, the Segmentation Anything Model (SAM) [61], based on vision transformers, has gained massive attention as a powerful and general vision segmentation model capable of generating various and fine-grained segmentation masks conditioned by the user prompt. With 1B masks and 11M images used to pretrain the model, it has shown impressive capabilities in all-around segmentation tasks in natural images [61]. However, applying the model to the medical domain is more complex since there exists a domain shift from natural to medical images. Many researchers have thus been trying to adapt SAM to the medical domain [62, 63, 64]. For example, MedSAM [64], which adopts pretrained SAM and applies few tricks in fine-tuning, has obtained state-of-the-art performance on 19 medical image segmentation tasks across five different image modalities.

In general, adopting a larger but more general model appears to be the trend in the deep learning field. Therefore, although deep-learning-based medical image segmentation models are an essential part of the current PhD research, our focus is on its applicability to downstream tasks rather than the model designs.

1.6 Our approach

Taking all the problems listed previously in Chapter 1.4.3 together, such as coarse segmentation, limited scan resolution, and computational resources, a skeleton-based tree structure is more appropriate modeling of the renal blood vessels, which has been largely adopted in the research of blood vessels [45, 16, 24, 65, 66]. Only this way can we get a full-scale representation of the renal vasculatures and get more control over the computational complexities. The skeleton-based tree structure will be explained in detail in Chapter 2 (paper A). Briefly, instead of a segmentation of the blood vessel or its corresponding computational meshes, blood vessel is more efficiently represented by its centerlines that connect every branch together.

Fig. 1.4 presents the overview of our full-scale nephron-vascular network modeling and renal autoregulation simulation. In general, the first and major part of the PhD thesis will be focused on incorporating statistical topological data, anatomical features of kidneys, and a real micro-CT scan of the kidney into reconstructing a full-scale 3D vascular tree model, which starts from the renal artery and ends in afferent arterioles, as will be introduced in Chapter 2 and in paper A.

The second part of the thesis will present potential applications of the vascular model. The most crucial application of the full-scale 3D vascular model is to simulate realistic blood flows and renal functions, as will be introduced in Chapter 3 and manuscript B. To implement such simulations, we need to integrate nephron models into the reconstructed 3D vascular model because this is where autoregulation occurs. We will then simulate the autoregulation mechanism with two feedbacks and measure the resulting renal blood flow from different levels of arterial blood pressure and different afferent arteriolar resistances induced by the two feedbacks. In addition, our simulated full-kidney-size 3D vascular model can be used to generate training data for deep-learning-based blood vessel segmentation models, as will be covered in Chapter 4 and in paper C.

The following Chapter 5 (paper D and E) presents two related medical image segmentation tasks that use deep learning approaches. Although not related to the renal vascular network directly, these studies further demonstrate the potential of deep learning in segmenting medical scans for biosimulation studies, even in a data-scarcity setting.

In the remaining part of the introduction chapter, we will briefly summarize and highlight our main pipeline.

- **3D reconstruction of full-scale renal arterial network (Chapter 2 and paper A).** In our innovative work, the full-scale arterial tree model is reconstructed by a physiologically based constructive optimization method while incorporating the anatomical structure from the micro-CT scan of a rat kidney. Notably, since the vessels at far-

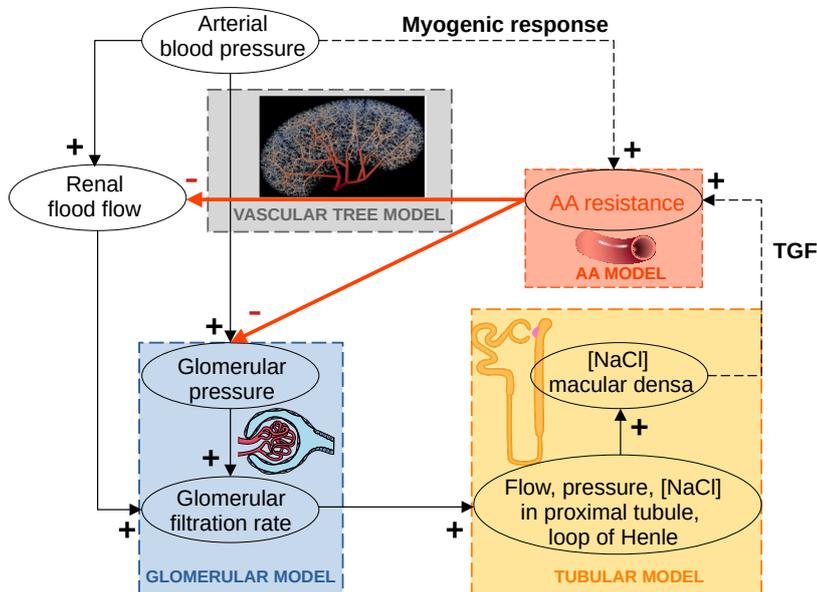


Figure 1.4: An overview of the full-scale modeling of nephron-vascular network and simulation of renal autoregulation. Figure from manuscript B.

surface regions in the kidney are beyond the experimental resolution, it is impossible to detect and reconstruct the small vessels solely from medical scans. In summary, the ex-vivo micro-CT scan only extracts the topology of large arteries consisting of the first few large branches and estimates the renal cortex. The small arteries, including arterioles, are physiologically reconstructed instead of being detected from the micro-CT scans. The topological tree structure of large arteries extracted from the micro-CT scan only involves the Cartesian coordinate associated with each node and the connectivity information (edges). On the other hand, the thickness (radius) of each vessel will not be inferred directly from the scan but will be modeled physiologically. For more details, please refer to Chapter 2 and paper A. The output tree model from this method is a skeletal representation of the vascular network, which consists of a set of nodes, where each node is associated with its three-dimensional Cartesian coordinate vector, and a set of directed edges, with a radius reflecting the cross-sectional extent associated with each edge. Such graph-based skeletal representation of a vascular tree has been proposed fundamentally for morphometric analysis of blood vessels and simulation of the renal blood flows [16, 67, 68, 24]. As a cru-

cial property of the reconstructed full-scale renal arterial network, the blood flow can be simulated in a much simpler form, because the Navier-Stokes equation can be solved analytically, giving the Hagen–Poiseuille Equation as discussed in Section 2.2.

- **Simulation of blood flow and autoregulatory mechanisms (Chapter 3 and manuscript B).** We develop a full-scale model of renal autoregulation that integrates the realistic vascular network topology generated from the previous step with a single nephron model that includes glomerular filtration, tubular reabsorption together with autoregulation of the afferent arteriolar resistances (Fig. 1.4). This allows for the simulation of full-scale blood flow along the renal arterial network and the autoregulation mechanisms as a complex interplay between the autoregulatory mechanisms (the myogenic response and the tubuloglomerular feedback). Validation of the model is achieved by comparing the simulation results of autoregulation to experimental measurements available in the literature. We further investigate how anatomical features of microcirculatory networks (i.e. different populations of nephrons as shown in Fig. 1.1 D) contribute to the optimization and adaptation of regulating blood flow. Our further pathological simulation of the effect of renal artery stenosis on blood flow shows good agreement with the experimental observations.
- **Synthesizing training data for deep-learning-based blood vessel segmentation models (Chapter 4 and paper C).** Although segmenting renal blood vessels is not the focus when building the full-kidney-size 3D vascular model because most of the small vessels are invisible from the scan and are reconstructed physiologically, segmenting visible blood vessels is still a crucial step, both for guiding the reconstruction of the remaining vessels and for estimating the remaining renal functions. While deep learning methods have demonstrated state-of-the-art performance in automatic blood vessel segmentations, they necessitate a substantial amount of labeled training data. Our physiologically constructed renal vascular trees from the previous step (Chapter 2 and paper A) can naturally provide synthesized training data that can be used to train deep segmentation networks. Although this application has less physiological impact, the ability to train neural networks demonstrates a crucial potential of the physiologically based constructive methods in the deep learning field by reducing intensive manual annotations.

Chapter 2

Modeling and reconstruction of renal arterial network

2.1 Skeleton-tree-based model of vasculature

There generally exist two main methods for modeling blood vessels. Model-free methods start from segmenting blood vessels from a medical scan followed by mesh reconstruction to build 3-D geometric models that capture the high-level structure of an individual's blood vessels from clinical images [40, 29, 30, 31, 32, 33], as mentioned in Section 1.4.2. It makes no assumption or approximation over the vascular structure. Therefore, it is more accurate and suitable for the diagnosis and surgery of vascular diseases. However, the result depends heavily on vessel segmentations and is usually less smooth. Therefore, for further simulations of blood flows, the output requires intensive post-processing to generate computational models that are suitable for numerical solvers to solve the Navier-Stokes equations. More importantly, as outlined earlier in Chapter 1.4.3, the small vessels in the kidney remain undetectable in the scan due to their size falling below the resolution of the image. However, these tiny vessels play a vital role in supplying individual nephrons and must therefore be accurately represented in the final model.

Instead, numerous researchers have found that cylindrical pipes can very well approximate and model blood vessels, especially the arteries [45, 16, 69, 65, 66]. Thus, a vascular network can be considered as a collection of connected cylindrical vessels, which is usually represented by a skeleton-tree-based model, as shown in Fig. 2.1. Specifically, the skeleton-tree-based vasculature is modeled by a directed acyclic graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$. Here, \mathcal{V} represents the set of nodes situated at the endpoint of each vessel centerline, associated with its coordinates in Euclidean space as the node feature. \mathcal{E} is a set of directed edges that form a connected tree structure. Each edge corresponds to a single vessel segment, conceptualized as a cylinder with its radius and flow as the edge feature.

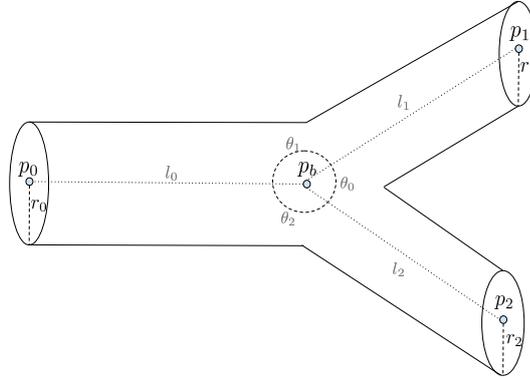


Figure 2.1: A typical vessel branching model. The branching vessels are uniquely defined by the locations of the three end nodes (p_0, p_1, p_2) , the location of the bifurcation node (p_b) , and the radii of the three incident edges (r_0, r_1, r_2) . Length is not explicitly associated with each edge but can be easily retrieved from the Euclidean distance between two nodes, e.g., $l_0 = \|p_0 - p_b\|_2$. Figure and caption from paper [A](#), reprinted with permission.

The tree-based model has implicitly resolved the computational size limitation issue because storing a graph structure of a vascular network is significantly more efficient than its corresponding mesh or volume. As an essential part of skeleton-tree-based models, the Navier-Stokes equation (Eq. (1.1) and Eq. (1.2)) can be solved analytically and simplified into a much simpler form, giving the Hagen–Poiseuille equation. This simplification allows for an easier modeling of blood flow throughout the vascular tree.

2.2 Hagen–Poiseuille equation of a simplified flow

With the assumption of simplified one-dimensional flows, such as in cylindrical pipes where the velocity varies only in the direction perpendicular to the flow, the Navier-Stokes equation can be solved analytically, giving the Hagen–Poiseuille equation. Specifically, the Hagen–Poiseuille equation requires the following assumptions in cylindrical pipes, which are not necessarily assumed in the Navier-Stokes equation:

- **Steady flow:** Poiseuille’s equation assumes that the flow is steady, meaning that the velocity does not change with time.
- **Incompressible flow:** Poiseuille’s equation assumes that the fluid density remains constant. Although the Navier-Stokes equation allows for compressible flow, the incompressible property is also usually assumed in CFDs on blood flows.

- Fully developed flow: Fully developed means that the velocity profile has reached a steady state and does not change along the length of the pipe.
- Laminar flow: Laminar flow has the behavior that the fluid layers move smoothly in parallel with no significant mixing between them. The Navier-Stokes equation, on the other hand, can account for both laminar and turbulent flow conditions, which involve chaotic mixing and eddies.
- Newtonian fluid behavior: Newtonian means viscosity remains constant regardless of the shear rate.
- Axisymmetric flow: Poiseuille’s equation assumes that the flow is axisymmetric, meaning that it remains consistent around a central axis.
- No body force: As in most hemodynamic studies, gravity is neglected as it has a minimal influence compared to other forces.
- No slip boundary condition: the velocity of fluid is assumed to be zero at the point where fluid touches the pipe.

Given these assumptions, the Navier-Stokes equation can be solved analytically, which provides the relation between the pressure drop Δp and flow Q along each vessel segment (a single tube), commonly known as Hagen–Poiseuille’s Equation.

$$\Delta p = \frac{8\mu l Q}{\pi r^4} \quad (2.1)$$

where μ is viscosity, l and r denotes the length and radius of corresponding pipe respectively.

With the analytical solution to the Navier-Stokes equation, there is no need for computational meshes or complex finite volume solvers as required in the CFD-based simulations introduced in Section 1.4.2. Instead, the main focus would be to build a skeleton-tree-based model of the renal arteries, which contains the branching patterns and the length and radius information of each blood vessel because these are the variables involved in the Hagen–Poiseuille’s equation (Eq. (2.1)).

2.3 Constructive optimization algorithms

With the fractal tree-based modeling of blood vessels, there exist some generative models to simulate the growth of vasculatures by considering the biological and physiological factors involved in the process [70, 71, 72]. For example, the Constrained Constructive Optimization (CCO) method proposed by Schreiner and Buxbaum [70], and its variant Global Constructive Optimization (GCO)

proposed by Georg and Hahn et al. [71, 72] model the vascular tree growing as an optimization problem following the assumption that the vasculature tries to achieve optimal topological and geometrical structure over the vascularized tissues from hemodynamic principles [70, 71, 72].

The Poiseuille's law in Section 2.2 is an essential part of our fractal tree-based modeling of renal arterial networks to define the cost function for the optimization problems. Specifically, numerous researches have found that the vascular tree is built to minimize the intravascular volume while ensuring efficient flow [73, 74, 65, 66]. Therefore, with the cylindrical tube assumption and fluid dynamics assumptions defined in Section 2.2, the cost at each branching point v is defined as a weighted combination of the volume, which is defined by

$$M_{loss}(v) = \sum_{e \in \mathcal{B}_v} \pi r_e^2 l_e \quad (2.2)$$

where \mathcal{B}_v denotes the set of all the incident edges of node v .

and the power dissipated during blood circulation, which, in analog to electric power, is defined as

$$P_{loss}(v) = \sum_{e \in \mathcal{B}_v} Q_e \Delta p_e = \sum_{e \in \mathcal{B}_v} \frac{8\mu l_e Q_e^2}{\pi r_e^4} \quad (2.3)$$

where the second equation follows by Poiseuille's law

Thus, the total cost at each branching v can be defined as

$$\begin{aligned} C_{local}(v) &\equiv w_c M_{loss}(v) + w_p P_{loss}(v) \\ &= w_c \left(\sum_{e \in \mathcal{B}_v} \pi r_e^2 l_e \right) + w_p \left(\sum_{e \in \mathcal{B}_v} Q_e^2 \frac{8\mu l_e}{\pi r_e^4} \right) \\ &= \sum_{e \in \mathcal{B}_v} \left(w_c \pi r_e^2 l_e + w_p Q_e^2 \frac{8\mu l_e}{\pi r_e^4} \right). \end{aligned} \quad (2.4)$$

Here, our work (paper A) is a GCO-based approach. Specifically, the GCO algorithm begins by selecting a single root node s and sampling n leaf nodes l_i in the perfusion territory of the organ. Given the position of a single root node s , and n leaf nodes l_i , the algorithm tries to simulate the growth of vasculature physiologically, with the goal to find a tree $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ that contains s and $l_i \in \mathcal{V}$ with minimum cost defined in Eq. (2.4) by adding new intermediate nodes to \mathcal{V} and connections (edges) to \mathcal{E} .

One of the main modifications we make here in paper A is that the medical scan can be used to extract the first few branches of the renal arterial tree, so that the GCO process does not start with only a single root node s , but with a prebuilt tree $\mathcal{G}_0 \equiv (\mathcal{V}_0, \mathcal{E}_0)$ where $s \in \mathcal{V}_0$.

Fig. 2.2 shows the whole pipeline of the reconstruction of the renal arterial network. In summary, a pre-built arterial tree $\mathcal{G}_0 \equiv (\mathcal{V}_0, \mathcal{E}_0)$ consisting of the

main arteries is extracted from the main artery segmentation from the micro-CT scan (Fig. 2.2 e - g), while the scan is also used to estimate renal cortex to determine where the afferent arterioles should be sampled (Fig. 2.2 a - d). In the GCO initialization step, these sampled terminal nodes (Fig. 2.2d) are then connected to the pre-built vascular tree $\mathcal{G}_0 \equiv (\mathcal{V}_0, \mathcal{E}_0)$ (Fig. 2.2 h). Therefore, the algorithm can initiate vasculature growth from an already partially constructed vascular tree, rather than beginning from a single root position. This approach ensures that the final full-scale vascular tree incorporates subject-specific details, maintaining the uniqueness and higher accuracy of individual vascular structures.

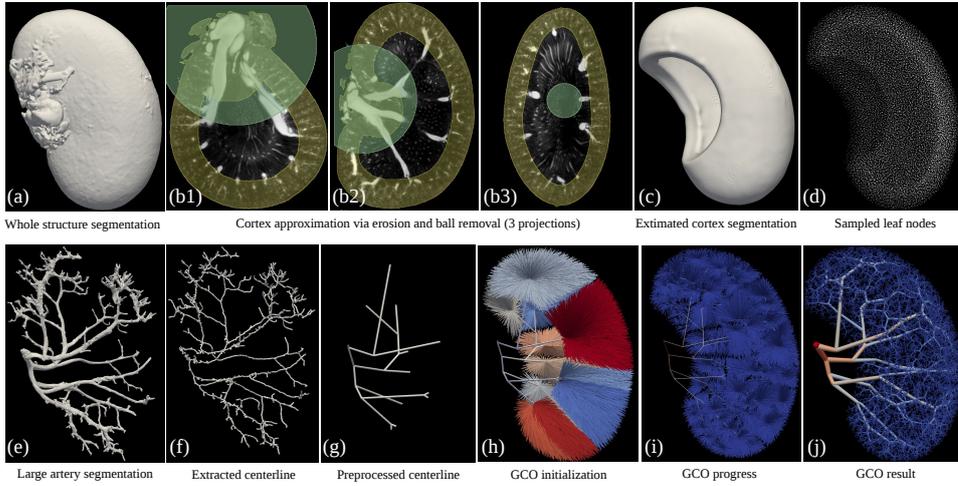


Figure 2.2: GCO Pipeline. The initial micro-CT scan is used to extract whole structure segmentation (a) and large artery segmentation (e). Top row: renal cortex (c) is approximated by a subtraction of erosion followed by a ball removal (b), where the leaf nodes (d) are sampled using Poisson disk sampling. Bottom row: extracted centerline (f) is pre-processed to pre-build a renal arterial tree consisting of only the first few large arteries (g). In GCO initialization (h), all the sampled leaf nodes (d) are connected to the nearest node in the pre-built tree (g), with color indicating the group of leaf nodes that are connected to the same node. Colors in the GCO progress and result (i and j) indicate the radius of each vessel: from $300 \mu m$ in renal artery to $10 \mu m$ in afferent arterioles (AA). Figure and caption from paper A, reprinted with permission.

For other details such as how GCO simulates the vascular tree, how our hybrid approach integrates image-priors to guide the GCO approach, as well as the validation of the simulated vascular tree with real anatomical data, please refer to the appended paper A.

Chapter 3

Full-scale model and simulation of renal autoregulation

In the previous chapter and in paper [A](#), a full-scale renal arterial tree structure that starts from the renal artery and ends in the afferent arterioles has been reconstructed and shown to be in good agreement with existing anatomical data [16]. However, this reconstructed renal arterial tree is only a topology of the renal arterial structure and is built under optimal assumptions in a normal, healthy, and steady-state scenario.

The major function of a kidney is to regulate blood flow and the glomerular filtration rate (GFR) to meet the needs of the organism. Although in paper [A](#), we have tried to simulate the hemodynamics in the constructed arterial tree and showed that it matches reality, such simulation is under the assumption that all afferent arterioles share identical blood flows, which is oversimplified and cannot be used to model realistic renal functions.

To model renal functions such as autoregulation, it is necessary to add functional units - nephrons - to the renal arterial tree, which are connected to the arterial tree from afferent arterioles as shown in Fig. 1.1 B. Nephrons are essential to properly define boundary conditions when simulating full-scale blood flows and to send signals back to the afferent arterioles, which causes them to contract or dilate. This contraction or dilation of the afferent arterioles will then change their resistances and allow the kidney to adjust its renal blood flow and GFR. Therefore, in manuscript [B](#), nephron models are attached to the afferent arterioles in the full-scale arterial tree generated in paper [A](#), which together form the full-scale nephron-vascular network. This full-scale nephron-vascular network model allows the evaluation of the operating conditions of around 30,000 nephrons in rat kidneys, and the efficiency of autoregulation in pathological conditions.

This chapter gives an overview of the computational modeling of the full-scale nephron-vascular network and autoregulatory mechanisms and the most crucial simulation results. For more details about the model formulation,

simulation experiments, and simulation results, please refer to manuscript [B](#).

3.1 Model formulation and validation

As shown in [Fig. 1.4](#), the full-scale model of the nephron-vascular network includes four compartments: a glomerular model, a tubular model, a model of the afferent arteriole, and the full-scale model of the renal arterial tree. The general objective is to integrate the vascular network topology (full-scale model of the renal arterial tree) and a single nephron model (glomerular model, a tubular model, afferent arteriolar model) that account for glomerular filtration, tubular reabsorption together with autoregulation of afferent arteriolar resistances. This autoregulation is determined by the tubuloglomerular feedback (TGF) and the myogenic response.

[Fig 3.1](#) illustrates the flowchart of our simulation process that integrates the four models and solves them consequently. The renal arterial blood pressure P_{art} serves as the boundary condition to the model of the renal vascular tree, and drives the single nephron blood flow $SNGBF$ and glomerular pressure P_{GC} at afferent arterioles (AA). These two parameters are the input to the Glomerular model, which will then give as output the glomerular filtration rate $Q_T(0)$ and the tubular pressure $P_T(0)$. These two variables serve as the initial flow and pressure values to the Tubular model. The Tubular model will give as output the NaCl concentration at the macula densa ($C_S(md)$), which is the stimulus for TGF feedback. The myogenic response is induced by the pressure at the beginning (P_A) and end (P_{GC}) of the afferent arteriole. TGF and the myogenic response, as the two inputs to the Afferent Arteriole (AA) model, will work together to adjust the afferent arteriolar resistance. With a new set of afferent arteriolar resistances, the full-scale arterial tree will provide a new set of single nephron blood flows $SNGBF$ and glomerular pressures P_{GC} . This process will be iterated until convergence. For details of how these variables are computed, please refer to manuscript [B](#).

In summary, the total renal blood flow (colored blue in [Fig 3.1](#)), which is the sum of the blood flows in the individual nephrons, is stabilized by the regulated afferent arteriolar resistances. The total renal blood flow does not directly participate in the autoregulation model itself but is the indicator of the autoregulation outcome and determines if the iteration reaches convergence.

To simulate autoregulation in the full-scale vascular model, we vary the inlet (arterial) pressure from 80 mm Hg to 200 mm Hg and calculate the resulting renal blood flow. The renal autoregulatory mechanisms will try to minimize the change in renal blood flow by adjusting the afferent arteriolar resistances by means of the two mechanisms.

[Fig. 3.2](#) (Left) shows the resulting renal blood flow with and without autoregulation. Without autoregulation (orange curve), the renal blood flow increases linearly with increasing renal arterial pressure. In contrast, the au-

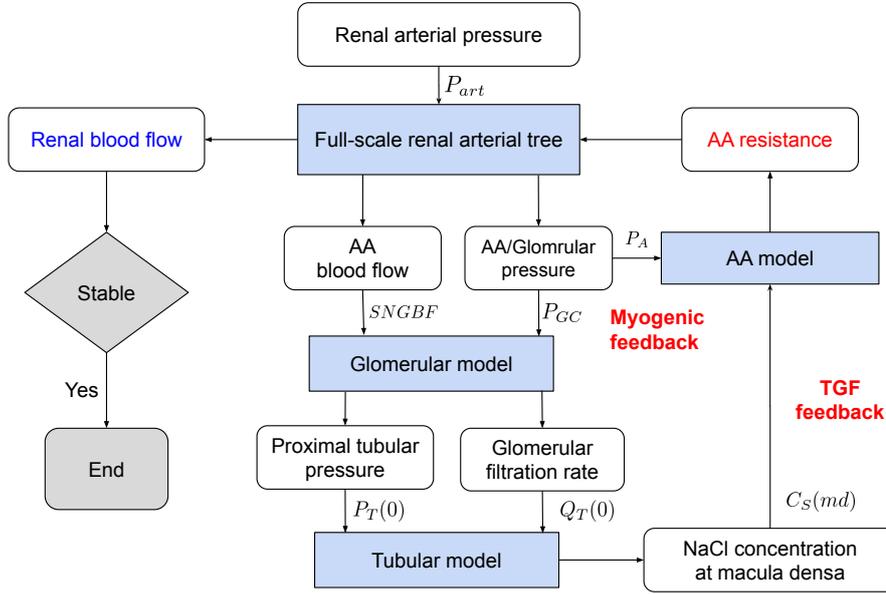


Figure 3.1: Flowchart for the full-scale model of renal autoregulation. Figure from manuscript B.

autoregulation provided by the TGF and the myogenic response is effective in minimizing the increase of renal blood flow when the inlet pressure increases from 80 mm Hg to 200 mm Hg (blue and red curve). The autoregulation degrades when the inlet pressure becomes too large, as can be seen from the larger slope at inlet pressures from 180 mm Hg to 200 mm Hg. To assess the efficiency of autoregulation quantitatively, Fig. 3.2 (Right) shows the Autoregulation index (AI), which is given as the ratio between the relative change of renal blood flow change and the relative change of renal arterial pressure [75]. Experimental studies in normal rats have found an AI between 0.2 and 0.5 [9]. As can be seen from Fig. 3.2 (Right), the model predicts similar values in the inlet pressure range from 100 mm Hg to 160 mm Hg.

In Fig. 3.2, the curve labeled “one population” denotes the simulation under the assumption that all nephrons are cortical nephrons, as was originally proposed in [11] for the simulation of autoregulation of a single nephron model. As shown in Fig. 1.1 D in Chapter 1, nephrons are better distinguished as superficial, midcortical, or juxtamedullary based upon the location of their glomerulus within the renal cortex [76]. They are also associated with different lengths of loops of Henle and thus cause slightly different autoregulation results, as labeled “three populations” in Fig. 3.2. For a detailed comparison of the modeling and results from three nephrons, please refer to manuscript B.

The green curve labeled “only myogenic response” in Fig. 3.2 (Left) de-

notes the simulation of blocking TGF (which can be achieved by giving a loop diuretic) so that only the myogenic response is active. The result shows that both TGF and myogenic response are important mechanisms of autoregulation: the contribution of only myogenic response is not enough for the blood flow stabilization.

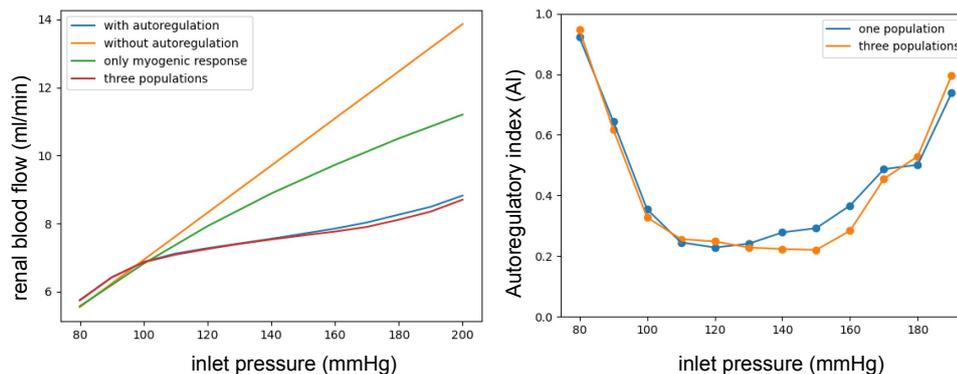


Figure 3.2: Left: Renal blood flow as a function of inlet (arterial) blood pressure. Right: Autoregulation Index at various inlet (arterial) blood pressure. Figure and caption from manuscript B.

3.2 Simulation of pathologies

In addition to simulating the role of autoregulation mechanisms in maintaining stable blood flow at varying inlet pressure levels, another perspective in the full-scale modeling of the nephron-vascular network is to simulate certain pathological conditions. One prevalent pathology in the renal arteries is renal artery stenosis (RAS), which involves the narrowing of the renal artery's lumen, typically due to atherosclerosis or fibromuscular dysplasia [21]. Computer simulation of RAS enables the exploration of the impacts of various stenosis severities and the kidney's capacity to adjust to these changes. By investigating different stenosis levels and analyzing the resultant effects on renal hemodynamics, more effective preventative and surgical interventions can be developed [21, 20, 22]. For example, the American Heart Association defined a RAS over 60% as the critical level requiring treatment [77].

We simulate RAS in the full-scale model of the nephron-vascular network in both a control case (Fig. 3.3 Left) with inlet pressure of $P_{art} = 100$ mmHg and a hypertensive case (Fig. 3.3 Right) using an inlet pressure of $P_{art} = 140$ mmHg, while gradually decreasing the diameter of renal artery (the root vessel in the full-scale tree model).

As shown in Fig. 3.3 Left, the renal blood flow stays relatively constant at the lower degrees of RAS but drops dramatically when stenosis exceeds

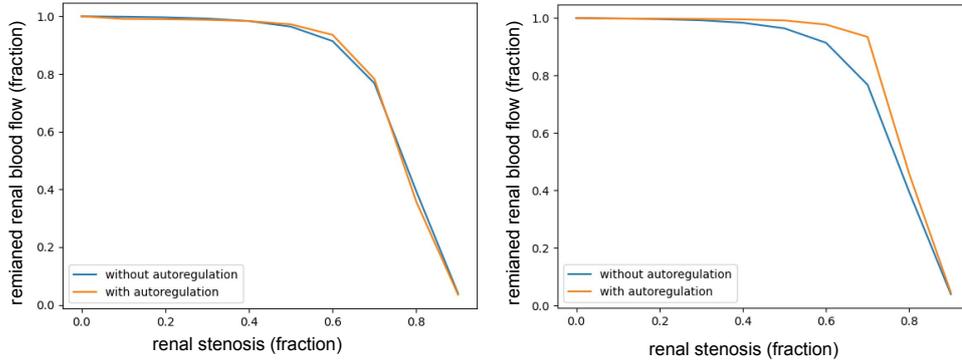


Figure 3.3: Renal blood flow at various stenosis levels in the control case (Left panel with inlet pressure $P_{art} = 100$ mmHg) and in the hypertensive case (Right panel with inlet pressure $P_{art} = 140$ mmHg). The control case shows a drastic decrease of renal blood flow with around 50% stenosis, while the hypertensive cases show that the autoregulation can postpone the drastic decrease of blood flow from around 50% stenosis to around 70% stenosis. Figure and caption from manuscript B.

50%, regardless of whether or not autoregulation is active. This behavior is in good agreement with the American Heart Association which defines a RAS of more than 60% as a critical stenosis [77] that needs treatment according to the arterial disease management guidelines [78]. Fig. 3.3 Right shows that in hypertension, the autoregulatory mechanism is effective in stabilizing the renal blood flow when the stenosis is up to 70%, while without autoregulation, the renal blood flow decreases earlier at a stenosis level of around 50%. But in general, the impact of autoregulation in preventing a drop in blood flow is minimal. This simulation result suggests that the flat curve in Fig. 3.3 when the stenosis $< 50\%$ is not primarily due to autoregulation, but rather stems from the fact that the hemodynamic resistance of the renal artery has little impact on renal blood flow until the stenosis is severe.

In addition, it is intriguing to compare our results to the CFD-based RAS simulations as mentioned in Chapter 1.4.2 [18, 19, 20, 21, 19, 22]. The crucial difference between our objective and these CFD-based studies is that they only reach the single renal artery while our research aims to simulate the blood flow along a full-scale renal vasculature with impact from the autoregulation mechanism. On the other hand, our simulation method is based on a tree structure without computational meshes or finite element or volume solvers. Nonetheless, our simulation results show close agreement with CFD-based methods, which also show a dramatic decrease of renal blood flow with stenosis of more than 50% and needs adequate treatment according to the arterial disease management guidelines [78]. This could indicate that complex computational meshes or finite element or volume solvers are not necessary

when simulating these pathologies in hemodynamics, as a tree-based model can generate very similar results and is much more efficient.

Chapter 4

Segmentation of renal blood vessels

The segmentation of blood vessels from the ex-vivo CT scans is still an essential task even though the segmentation results cannot be used to build computational meshes for simulations directly due to its limited resolution and computational complexity, as mentioned in Chapter 1.4.3. Instead of building computational meshes, blood vessel segmentation is usually post-processed and converted to a graph-based structure by a centerline extraction, also known as skeletonization, which can be used to analyze 3D morphologies of blood vessels such as radius and length distribution and branching patterns [16, 45]. In our hybrid approach to reconstructing a full-scale renal arterial tree (Chapter 2 and paper A), the ex-vivo micro-CT scan is integrated into the reconstruction process by extracting the centerline from the segmentation of large arteries with certain post-processings. Extracting 3D skeleton from vessel segmentations is a complicated task and remains a specific field of research [79, 80, 81], which is beyond the scope of this PhD project. We have focused on the method [79] in our framework because it directly turns a segmentation into a graph-based structure rather than a binary mask with a local width one, as discussed in paper A. This chapter (paper C) only focuses on the segmentation of blood vessels.

Manual segmentation of visible renal blood vessels from the ex-vivo CT scans of rat kidneys is highly time-consuming, given the vast volume size. It is estimated that the segmentation of visible blood vessels of a single ex-vivo CT scan would take half a year of a trained specialist (paper C). Therefore, there is a need for automatic segmentation of the blood vessels using deep learning approaches, as introduced in Chapter 1.5.1. On the other hand, deep-learning-based segmentation networks require manually labeled blood vessel segmentation maps as the ground truth training data. However, due to the challenging nature of manually annotating micro-CT scans, we encounter the issue where not even one definitive, accurately labeled map is available to

train a deep learning segmentation model.

4.1 Reconstructed vascular tree as training data

Motivated by the difficulty of manual labeling of blood vessels and lack of training data in ex-vivo micro-CT scans, this study focuses on the segmentation of blood vessels without any manually labeled segmentation maps. To accomplish this task, note that in the previous task (Chapter 2 and paper A), we have reconstructed full-scale vascular trees physiologically. These full-scale trees can, in turn, be used to synthesize training data for a deep-learning-based segmentation network. Moreover, since the reconstruction process involves several randomities, the result can vary for each run visually, resulting in some variety in the synthesized training data, which is essential to neural network training.

In order to train a neural network to segment blood vessels in micro-CT scans of rat kidneys, we need to have 1) blood vessel segmentation label maps (binary masks) and 2) their corresponding micro-CT scans. For 1), we need a voxelization method to convert the reconstructed vascular trees (a graph structure) into binary label maps of the blood vessels, which has been briefly mentioned in the supplementary material in paper A. For 2), the task of generating corresponding micro-CT scans of these simulated blood vessels is solved by training a deep generative model [82] over unlabeled micro-CT scans. The following two sections present more details on the two tasks, especially on the technical details in 1) that have not been given in paper A or C. For the details of segmentation results, please refer to paper C.

4.1.1 Building segmentation maps via implicit surface

The simulated fractal-tree-based modeling of the full-scale renal arterial network can be visualized by simply stacking the cylinder of each blood vessel together, colored-coded by properties like radius and flow of each vessel, as shown in paper A and in Fig. 4.1. Although such visualization is effective enough in visually inspecting the generated vascular structure, it does not generate smooth transitions in vessel junctions (Fig. 4.1 Right).

Producing smooth transitions at branchings is essential to generate organic-looking blood vessel segmentation maps. This is vital for the quality of the constructed artificial kidney dataset used to train deep-learning segmentation models [41, 82, 83]. To create such an image dataset, an inverse process is required to convert the reconstructed vascular tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ back to binary label maps.

The key aspect of this process is ensuring that the resultant maps appear as natural as possible. This includes, for instance, achieving smooth transitions at points where branches diverge, thereby mirroring the organic characteristics of vascular structures.

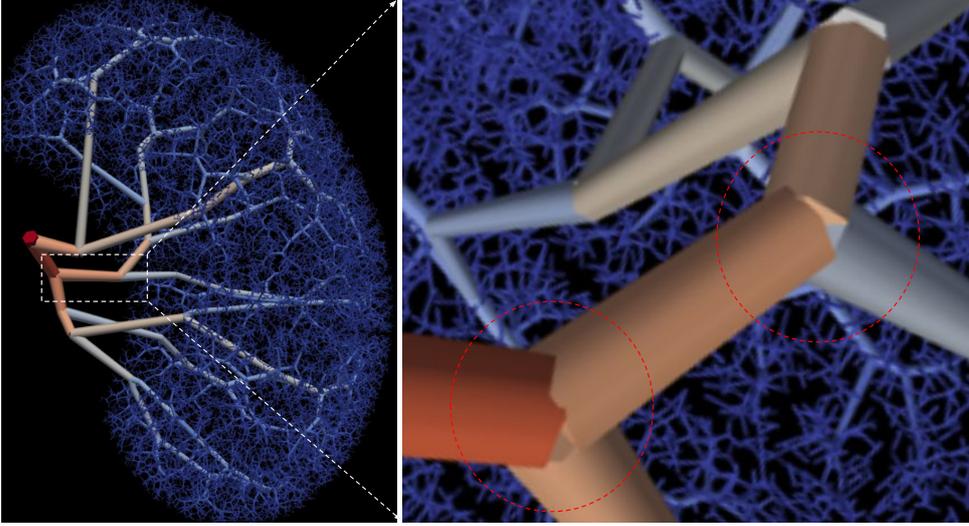


Figure 4.1: Visualization of the reconstructed vascular tree with a zoom-in view (and rotated for better illustration). Left image from paper A, reprinted with permission.

Implicit surface

In medical imaging, iso-contouring is the preferred technique for visualizing organic-looking blood vessels. These approaches seek an implicit function that generates tubular structures of skeletons [84, 85, 86]. The explicit surface can then be obtained by sampling the implicit function on a voxel grid and applying a contouring method such as marching cubes for visualization purposes [43, 87]. Similarly, binary segmentation masks can be easily generated by thresholding the implicit functions. Therefore, the focus is on properly generating implicit functions of the vessel structure [84, 85, 86].

A reconstructed vascular tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from our result in paper A hosts the centerline location of each node in Euclidean space as well as the radius information of each edge. A standard implicit tube function $T : \mathcal{R}^3 \rightarrow \mathcal{R}$ in Eq. (4.1) is a scalar field function that maps every voxel point $x \in \mathcal{R}^3$ to the tube surface along the centerline [86], defined by

$$T_{\mathcal{G}}(x) \equiv \min_i \{d(x, i) - r(x, i)\} \quad (4.1)$$

where $i \in \mathcal{E}$ denotes the line segment i of the centerline graph, $d(x, i)$ denotes the distance of the voxel point x to the line segment i and $r(x, i)$ denotes the corresponding radius of the line segment i at point x .

The resulting scalar value of $T_{\mathcal{G}}(x)$ facilitates point classification by simply checking the sign of it.

$$T_{\mathcal{G}}(x) - Iso = 0 \quad (4.2)$$

where Iso denotes the isovalue, which is used for creating an isosurface such that the implicit equation is zero. The specific tube function T defined in Eq.(4.1) that subtracts radius from distance has a natural property that points inside the tube have negative values. In contrast, points outside the vessel have positive values, with smooth transitions, indicating $Iso = 0$ in this case.

Note that $d(x, i)$ is the distance of a point to a line segment rather than to a line. Thus, it should be defined by either the perpendicular distance to the line segment or the distance to one of the two endpoints based on whether the projection is within the line segment.

In order to compute Eq. (4.1), we start with the definition of vector projection in Euclidean geometry. Suppose $i_0 \in \mathcal{R}^3$ and $i_1 \in \mathcal{R}^3$ denote the starting and ending node of the line segment i respectively. For each node x in the space, $a = x - i_0$ defines the vector starting from node i_0 and ending at x . We can then compute the projection of a at i by

$$\tau \equiv \text{proj}_i a = \frac{a \cdot i}{\|i\|} = \frac{(x - i_0) \cdot (i_1 - i_0)}{\|i_1 - i_0\|} \quad (4.3)$$

The distance $d(x, i)$ in Eq. (4.1) can then be computed by a simple Pythagorean theorem. However, note that the projection can lie outside the line segment i . Therefore, $d(x, i)$ is properly defined as

$$d(x, i) = \begin{cases} \|x - i_0\| & \text{if } \tau \leq 0 \\ \sqrt{\|x - i_0\|^2 - \tau^2} & \text{if } 0 < \tau < \|i_1 - i_0\| \\ \|x - i_1\| & \text{if } \tau \geq \|i_1 - i_0\| \end{cases} \quad (4.4)$$

Then, the corresponding radius $r(x, i)$ in Eq. (4.1) is defined as a function of both the line segment i and the location x . This is because in the standard pipelines of vascular modeling, the vascular tree structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is achieved by applying skeletonization of vessel segmentations from medical scans. The radius at each single point of the skeleton can then be computed by finding the maximum inscribed sphere to the original segmentation [45]. Therefore, radius information is associated with each node rather than each edge. In order to properly define $r(x, i)$, radii are usually assumed to be linearly interpolated from i_0 to i_1 [86]. Therefore, $r(x, i)$ is defined as

$$r(x, i) = \begin{cases} R(i_0) & \text{if } \tau \leq 0 \\ R(i_1) \left(\frac{\tau}{\|i_1 - i_0\|} \right) + R(i_0) \left(1 - \frac{\tau}{\|i_1 - i_0\|} \right) & \text{if } 0 < \tau < \|i_1 - i_0\| \\ R(i_1) & \text{if } \tau \geq \|i_1 - i_0\| \end{cases} \quad (4.5)$$

where $R(i_0)$ and $R(i_1)$ denote that radius at i_0 and i_1 respectively.

Implementation details and results

In implementation, the tube function is defined in a regular 3D grid T , with large positive numbers as its original values. The program then iterates all the vessel segments (edge) in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and updates the grid values by Eq. (4.1). After the process, surface meshes can be obtained by applying a contouring method such as marching cubes [43, 87] with a contour value 0, while the binary segmentation masks can be easily generated by thresholding of 0.

The process of computing the tube function over a large grid is extremely time-consuming because the tube function has to be evaluated over the grid on each line segment (vessel) $i = [i_0, i_1]$ such that the tube function is minimized. A crucial acceleration we make here is that for each line segment $i = [i_0, i_1]$, we only compute the tube function inside a bounding cube around it. Specifically, only voxel points within the bounding cube of $[\min(i_0, i_1) - \max(R(i_0), R(i_1)), \max(i_0, i_1) + \max(R(i_0), R(i_1))]$ is evaluated for each line segment, where $\min(i_0, i_1)$ and $\max(i_0, i_1)$ are both in \mathcal{R}^3 representing the minimum and maximum coordinates of the two endpoints i_0 and i_1 in the three dimensions separately. This simple modification reduces the computational time from weeks to only minutes over a generated vascular tree with around 50K vessel segments.

An example reconstructed blood vessel segmentation mask of renal arterial structure is shown in Fig. 4.2. Although with some zig-zag artifacts, it illustrates much smoother joints than just stacking cylinders together (Fig. 4.1) and is thus more suitable to be used as ground segmentation masks for training deep-learning-based blood vessel segmentation models.

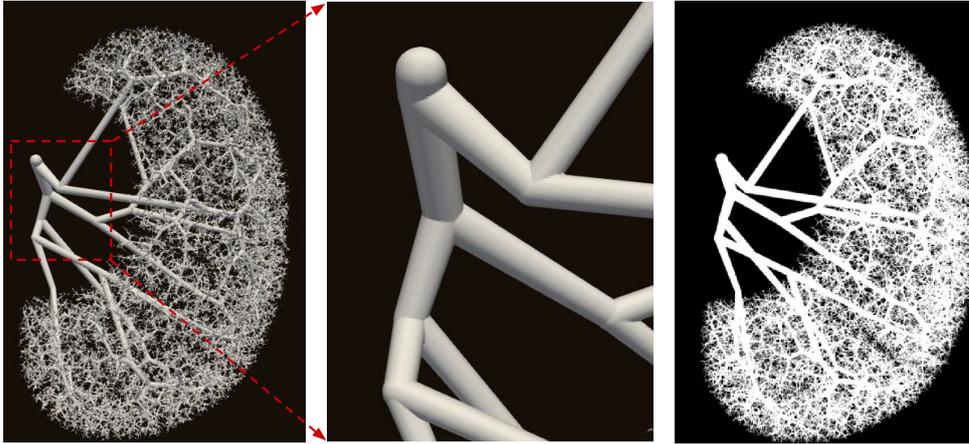


Figure 4.2: Visualization of the reconstructed vascular tree from implicit tube function (Left) with a zoom-in view (Middle), and the maximum intensity projection of the synthesized binary vessel mask by thresholding the implicit tube function (Right)

4.1.2 Building scans via domain adaptation

Deep-learning-based blood vessel segmentation networks require paired scans and vessel segmentation masks as training data. The previous step has synthesized blood vessel segmentation masks, while the next challenge is to generate their corresponding scans.

Medical scans are acquired from specific medical devices such as CT, MRI, or Optical coherence tomography angiography (OCTA), each involving different noises and artifacts in the image acquisition process. Menten et al. recently showed that it is possible to create corresponding 2D OCTA images from physiologically synthesized retinal blood vessels by applying physics-based image transformation functions that emulate the image acquisition process, such as flow projection and shear artifacts [88]. However, such functions are tailored only to OCTA and do not generalize to our micro-CT scans.

Without any expert knowledge of medical devices, deep generative models, e.g., Generative Adversarial Networks (GANs) [89], are one of the advances of neural networks that can automatically generate synthetic images closely resembling real distributions. Consisting of two neural networks: the generator G and the discriminator D , the generator learns to produce synthetic data samples from random noise inputs, while the discriminator learns to distinguish between real and synthetic samples. Through adversarial training, the generator can generate samples that are indistinguishable from real data, while the discriminator improves its ability to differentiate between real and fake samples [90]. This dynamic training approach incrementally improves the performance of both the generator and discriminator, leading to increasingly realistic generated data.

CycleGAN [82], an extension of the generative adversarial network framework, introduces a novel approach to unpaired image-to-image translation tasks by combining two generators and two discriminators. As shown in Fig. 4.3, given images from two domains (A and B), the generators ($G_{A \rightarrow B}$ and $G_{B \rightarrow A}$) aim to transform images from one domain to another and then back to the original domain, while the discriminators (D_A and D_B) differentiate between translated and real images in both domains. Especially in the medical domain, CycleGAN [82] has been used to perform domain adaptation from CT to MRIs and vice versa [91, 92], i.e., domain adaptation.

In our specific task, physiologically synthesized data from our pipeline (paper A) serve as the source domain (B as in Fig. 4.3), while unlabeled real micro-CT scans serve as the target domain (A as in Fig. 4.3). As shown in Fig. 4.3, although we only need the generator $G_{B \rightarrow A}$ to adapt the synthesized data (domain B) to real micro-CT scans (domain A), another generator $G_{A \rightarrow B}$ applied over the output of $G_{B \rightarrow A}$ is necessary to make sure that the critical semantics, i.e., blood vessels are retained during the process. This is explained more in detail in CycleGAN’s original paper as cycle consistency loss [82].

From Fig. 4.3, the adapted output from $G_{B \rightarrow A}$ (fake A) then provides

the corresponding scans of the synthesized blood vessel segmentation maps (Seg_{label} in Fig. 4.3) from the previous step, which together can be used to train a standard segmentation model, e.g., UNet [41, 83] (S in Fig. 4.3) to segment blood vessels in real micro-CT scans.

For more details about the pipeline and the segmentation results, please refer to our paper C. For more details about how CycleGAN is constructed and trained, please also refer to its original paper [82].

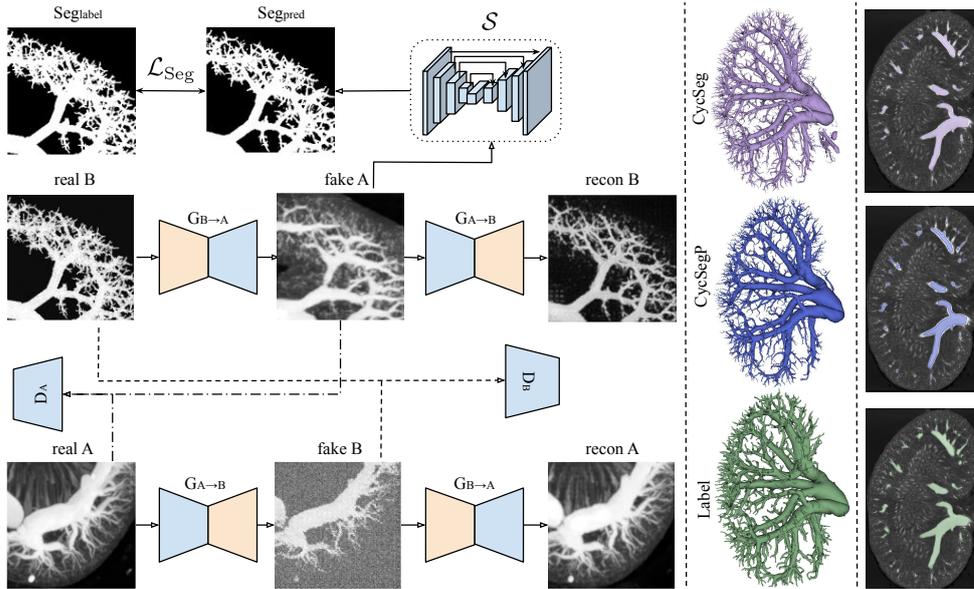


Figure 4.3: Left: An illustration of CycleGAN with an additional segmentation branch working on adapted (fake) domain A images by $G_{B \rightarrow A}$. All 3D patches are shown with Maximum Intensity Projection. Loss functions in CycleGAN are not shown for simplicity. Middle: An example result in 3D. Right: A sample slice overlaid with segmentation. Image and caption from paper C, reprinted with permission.

Chapter 5

Segmentation of human hip joints and jaw bones

Although segmentation from medical scans is an essential task, the segmentation output is only the first step for more complicated downstream tasks, e.g., simulations-based studies [48, 49]. Depending on the downstream task, the output from medical image segmentation needs to focus on certain anatomical correctness of specific structures. Although in our case, the scans of blood vessels cannot be used in a CFD-based pipeline as emphasized in Chapter 1.4.2, in many cases, the segmentation results from medical images are used for generating finite element models by building computational meshes, which can be further used for simulating patient-specific implants [93, 49]. For example, in the case of hip joint segmentation (paper D), the focus should not only be on generally partitioning the raw scan into regions of femur and hip bones, but also on detailing the bone boundaries and inter-bone cavities between them. This is because the soft tissues located in the cavities are essential to generate biomechanics models for stress analysis in human hips [48]. Similarly, when segmenting human jaws (paper E), our focus is not only on separating teeth and alveolar bones, but also on preserving the gaps between tooth-bone interfaces where the periodontal ligament (PDL) resides [49]. PDL layer is essential in orthodontic treatments due to its pivotal role in transferring load from the teeth to the bone [49]. This process, when activated by sufficient orthodontic forces, results in bone remodeling.

In the remainder of this section, we include a summary of the core parts of our two related deep-learning approaches to address the automatic segmentation of human hip joints and human jaws, which are the essential first step to building patient-specific finite element models for simulation studies [49, 48]. For more details on the two medical image segmentation tasks, please refer to our paper D and E. The segmentation results from paper D and E are successfully used to generate finite element models for simulation studies on human hip joints [48] and teeth [49]. For more details about the two

simulation studies, please refer to [49] and [48].

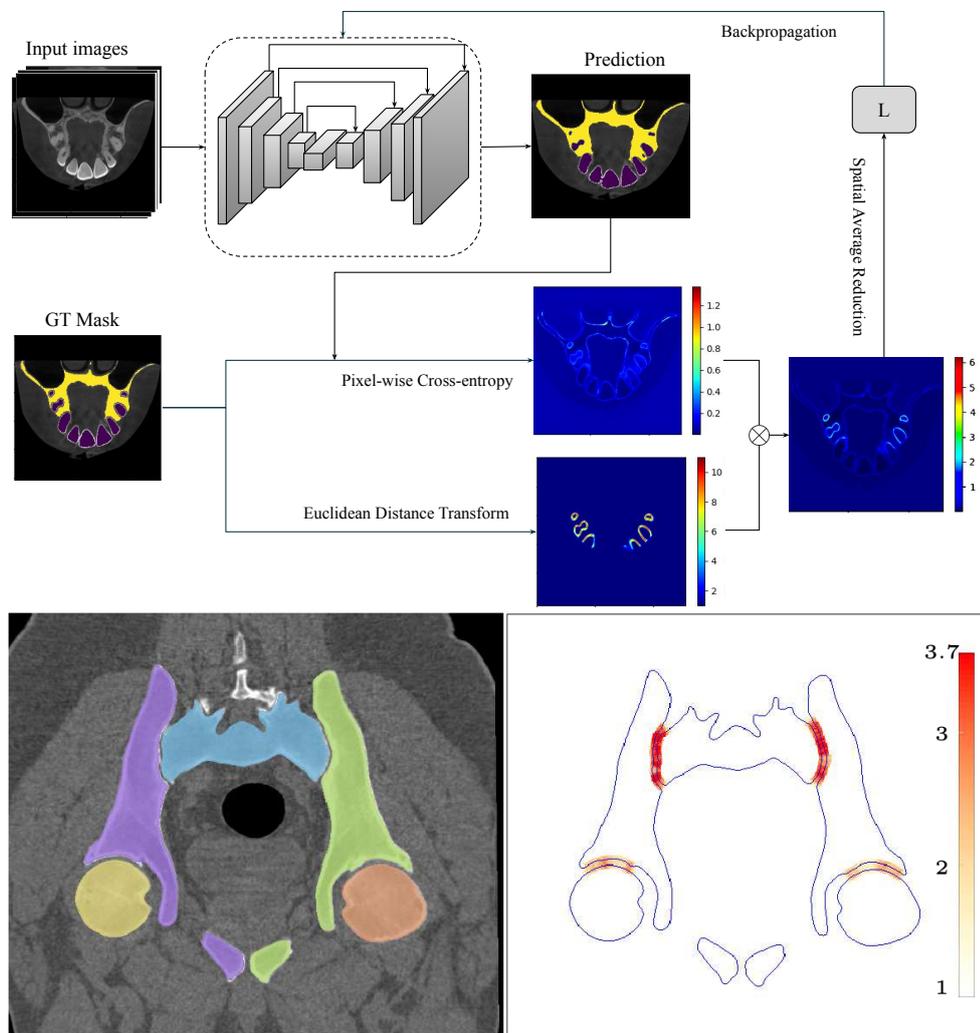


Figure 5.1: Model training pipeline of segmenting teeth (up) and hip joints (bottom), which both shows the weighted distance map calculated from Eq. (5.1). \otimes denotes the element-wise product, which suppresses the general boundary uncertainties while amplifying loss near the gaps. Note that the pixel-wise cross-entropy is visualized after averaging over all the classes. L is the final loss as a scalar after reduction. Image and caption from paper [D](#) and [E](#), reprinted with permission.

Backbone model Since we have very limited training data, and our focus is on the simulation of certain anatomical structures rather than outperforming state-of-the-art AI models, we adopt MultiPlanar UNet, an Efficient,

lightweight, and free of parameter-tuning UNet model as the backbone model [94]. The model segments the 3D medical scans using 2D UNet while preserving as much 3D information as possible by generating views from different perspectives [94].

Design choice 1: Transfer learning from inaccurate labelmaps One of the design choices we have found crucial is to pre-train the model using datasets with poor labeling. As mentioned above, our focus is on detailing the bone boundaries and inter-bone cavities in hip joints and on preserving the gaps between tooth-bone interfaces for PDL simulation in human jaws. However, acquiring such detailed ground truth labels is extremely time-consuming. As a common trick in deep learning, we pre-train the model using datasets with poor labeling, which ignores the cavities and PDLs. This enables the model to learn the general geometries for hip joints (paper D) and jaws (paper E) and requires less additional training data to detail the gaps afterward.

Design choice 2: Simulation-inspired weight-map in fine-tuning On the other hand, when subsequently fine-tuning the model with a small set of accurate data to detail the gaps, we would like the model to be focused on the gap regions, as shown in Fig. 5.1. We achieve this by enforcing a simulation-inspired voxel-wise weight-map $w(x)$ to the loss function based on the distances to the border of the foreground classes.

$$w(x) = w_c(x) + w_0 \cdot e^{-\frac{(d_1(x)+d_2(x))^2}{2\sigma^2}} \quad (5.1)$$

where d_1 and d_2 denote the distance to the border of the nearest foreground class and the second nearest foreground class, respectively. This strategy was originally adopted in the UNet paper [41] to separate 2D touching cells. In our 3D cases, the incorporation of the distance-based weights (Eq. (5.1)) into the training of the neural network is inspired by the expectation that in subsequent Finite Element (FE) simulations, a similar distance-based metric will be employed to delineate the space in the two areas: between the segmented teeth and bone geometries to locate the PDL, and between femur and hip bones to locate soft tissues in the cavities, as outlined in [49, 48].

Chapter 6

Final remarks

6.1 Conclusion and potential impact

This thesis introduces a novel, full-scale model of the renal nephron-vascular network, bridging gaps in current understanding and offering a tool for future renal pathology research. Specifically, we proposed a hybrid framework for reconstructing the arterial vascular network of a rat kidney in Chapter 2 (paper A). The framework generates a full-scale renal arterial tree using a physiologically constructive optimization method while taking image-based priors from the medical scan. We have shown that the reconstructed vascular tree has a good morphometric agreement with anatomical data from a real rat kidney [16].

With the reconstructed renal arterial tree, we have also integrated nephron models into the tree by attaching them to the afferent arterioles and simulated the active autoregulation of pressure and flow with two feedback mechanisms from nephrons in Chapter 3 (manuscript B). Our simulation results reveal a strong autoregulation of renal blood flow, especially in counteracting the increase of inlet renal arterial pressure by contracting afferent arterioles and thus increasing resistance, which is in good agreement with experimental studies [9]. Our further simulation of the effect of renal artery stenosis on the blood flow also shows close agreement with both the experimental observations [77] and existing CFD-based simulation studies [19, 20, 21].

Note that the pipeline of our research cannot be used for the diagnosis of vascular diseases because a significant portion of the vascular tree is reconstructed physiologically rather than detected from the medical scan. Instead, it is aimed at analyzing the mechanisms of renal functions and is crucial for therapy planning, drug design, and medical research. On the other hand, in Chapter 4 (paper C), we have also demonstrated that the physiologically reconstructed vascular trees from our pipeline can create a synthetic renal vessel dataset, which can be used to train a deep learning model for segmenting blood vessels from real medical scans. This can potentially reduce the intense

manual work involved in annotating the vessel structures from the medical scans and help the automatic diagnosis of vascular diseases.

Once the model of autoregulation is formulated and implemented, the model can be further extended to other mechanisms and signaling pathways to solve specific problems with urine concentration mechanisms or oxygen transport in the kidney. This mathematical modeling of the renal microcirculation is crucial because alterations in the renal microcirculation may cause local hypoxia leading to glomerular, tubular, and vascular damage or to renal dysfunction through interference with glomerular filtration and/or tubular secretion and reabsorption. Microvascular dysfunction may, therefore, serve as an early hallmark of kidney injury, implicating that non-invasive assessment of renal microvascular structure and function could be an improvement in the early identification of patients with progressive kidney disease [95]. This early non-invasive assessment can be used to predict better preventive strategies and therapeutic treatments, such as mechanism-driven drug development aimed at reducing renal damage.

6.2 Discussion: Limitations and future perspectives

6.2.1 Reconstruction vs experimental data

Of course, our full-scale renal arterial tree is reconstructed physiologically rather than extracted from real scans. Although our proposed hybrid approach integrates the real scan of a rat kidney into the process and has shown good morphometric agreement with anatomical data [16], it is still possible that the physiologically reconstructed vascular tree does not fully represent real renal arteries.

On the one hand, we can generate more realistic vascular trees by incorporating more vessels from the segmentation of the scans, or by having a more accurate estimate of the renal cortex. Currently in paper A, we have excluded a large portion of the visible arteries to preserve only the first few main arteries. This decision is based on the fact that the other parts are extremely susceptible to noise and present considerable challenges in segmentation from the scan. If we have scans with better qualities and better segmentations, a larger part of the full-scale arterial tree can be extracted and prebuilt from the actual scan, while fewer invisible small vessels need to be constructed physiologically.

On the other hand, a new imaging modality, Hierarchical Phase-Contrast Tomography (HiP-CT), was recently introduced in 2020 at the European Synchrotron Radiation Facility [96]. This new imaging modality can image the whole human organs at an unprecedented resolution by first scanning whole organs with $20\mu\text{m}/\text{voxel}$, then followed by high resolution (down to around

$2\mu\text{m}/\text{voxel}$) regions of interest within the sample without sectioning [96, 97]. However, high-resolution imaging technique only means it is theoretically possible to extract a full-scale vascular tree from medical scans, while technical problems such as imaging noises, artifacts, computational bottleneck, and intense manual works still pose great challenges to extracting vascular trees down to afferent arterioles from the scans [97, 98].

Nonetheless, if, in the future, a full-scale renal arterial tree structure can be extracted from HiP-CT scans, the renal blood flows and autoregulation could also be simulated over the extracted tree. These results can also be used to further validate the morphologies of our physiologically reconstructed renal arterial tree. From another perspective, in Chapter 3 (manuscript B), we have built the whole pipeline to simulate blood flow and autoregulation mechanisms from a full-scale renal arterial tree, regardless of how the full-scale renal arterial tree is built. Therefore, a full-scale renal arterial tree potentially extracted from HiP-CT scans can be easily integrated into our pipeline to simulate autoregulation mechanisms.

6.2.2 Segmentation of renal blood vessels

In paper C, we showed that the full-scale renal vasculature generated from paper A can be used to create a synthetic renal vessel dataset for training a deep-learning-based vessel segmentation network. However, we had to switch the task from segmenting arteries to veins because most of the micro-CT scans only have veins visible, while the arteries only consist of a tiny part, which we have thus considered as noise in the vessel segmentation task. Unfortunately, segmenting veins is physiologically uninteresting because the hemodynamics of renal blood vessels and autoregulation mechanisms are mostly related to the renal arterial structures. It is, however, non-trivial to modify our approach in paper A to generate both arteries and veins while modeling their pair-wise coupling. Right now, it is only possible to generate arteries and veins independently, which does not model their pair-wise coupling and thus cannot generate realistic segmentation masks for training an Artery/Vein multi-class segmentation model.

Moreover, segmenting veins from the micro-CT scans with contrast agency is a relatively easy task because the vessels usually have higher intensities than the background. Synthesizing training data from simulation-based vessel graphs is beneficial in the beginning when there is not a single labeled scan, as discussed in Chapter 4 (paper C). After the segmentation model has been trained on the synthesized data and applied over real micro-CT scans, the segmentation results can be manually corrected, providing real micro-CT scans with corresponding vessel segmentation masks. These labeled real data can then be used to fine-tune the segmentation model, which was previously trained only on synthesized data. This process can be iterated as an interactive learning pipeline to gradually output more accurate segmentation

predictions from the model. This interactive learning pipeline is also what we implemented to gradually get accurate segmentations for human hip-joints in paper D and [48].

In the deep generative model field, Diffusion models [99, 100] have been gradually replacing GANs due to their stable training process. Nonetheless, after and parallel to our work, there have been some more works on generating training image data from physiologically simulated vessel graphs, mainly by training a GAN model over OCTA images [101, 102, 103]. When doing domain adaptation to generate realistic scans, CycleGAN still seems to be the standard model due to its ability to achieve both image-to-image translation and image-to-label alignment. However, it can be elaborated for specific tasks. For example, in [102] on generating 2D OCTA images, the authors found that it is possible to directly use the segmentation network (S in Fig. 4.3) as the regularizer for image-to-label alignment and thus obviate the generator and discriminator over real images ($G_{A \rightarrow B}$ and D_A in Fig. 4.3). This simplification can be potentially more beneficial in our 3D setting as the training has been largely limited by the patch and batch size due to memory bottlenecks.

One main limitation of our work (paper C) is the limited number of validation data and the correctness of these data, which also hinders us from trying and evaluating different generative models. On the one hand, more renal blood vessel segmentations should be annotated to thoroughly validate model predictions, possibly using the interactive learning pipeline mentioned above. On the other hand, in paper C, we have qualitatively shown that the segmentation prediction from our model surpasses the level of detail in the manual annotations, which would falsely penalize the model during quantitative evaluation and result in an inaccurate assessment. This challenge has also been encountered in [102], where they decided to filter synthetic segmentation labels by vessel radius to match the human annotation details. However, they also note that manual annotations do not always correlate with vessel diameter, thus the challenge remains unless we have accurate and detailed manual labelings [102].

Finally, the vascular structures from constructive optimization methods are built under healthy and optimally functioning organs [70, 71]. Therefore, the synthesized training data only contain segmentation masks for healthy kidneys. To train a segmentation model that can also segment blood vessels of kidneys under pathological conditions, e.g., diabetes, we also need to synthesize training data with these pathologies. These data with pathologies can potentially be provided by first synthesizing vasculatures under healthy conditions and then simulating pathologies over the vasculatures before generating segmentation masks and scans. Specifically, a vascular tree model enables more non-standard augmentations, e.g., by analytically modifying the radii or lengths of some vessels or removing some vessels and their children to simulate pathological changes. For example, Brown et al. [103] simulated retinal vein occlusion (RVO) by randomly choosing a location of veins and decreasing the

vein's diameter. The blood flow is then recalculated to further simulate the change of flow in all the downstream vessels. This process will create more varieties of the synthesized training data, which can potentially train a better and more generalizable segmentation model.

6.2.3 Dynamics of the model and interactions between nephrons

In our simulation of blood flow and autoregulation mechanisms (manuscript B), we are currently only focused on steady-state solutions. Specifically, we are only interested in the final afferent arteriolar resistances and the resulting renal blood flows in a given pressure state. However, the adaptation of afferent arteriole to different pressure and flow patterns is dynamic. For example, the governing equations of the tubular model, as one-dimensional approximations of the Navier-Stokes equations describe the flow in the tubular model as coupled partial differential equations [11]

$$\begin{aligned}\frac{\partial P_T}{\partial z} &= -\frac{\rho}{\pi r^2} \frac{\partial Q_T}{\partial t} - \frac{8\eta}{\pi r^4} Q_T, \\ \frac{\partial Q_T}{\partial z} &= -2\pi r \frac{\partial r}{\partial P_T} \frac{\partial P_T}{\partial t} - J_V.\end{aligned}\tag{6.1}$$

In our steady-state solution, we assume there are no time derivatives, so the above partial differential equations can be simplified to ordinary differential equations, which is much easier to solve numerically.

$$\begin{aligned}\frac{dP_T}{dz} &= -\frac{8\eta}{\pi r^4} Q_T, \\ \frac{dQ_T}{dz} &= -J_V.\end{aligned}\tag{6.2}$$

However, without time variables, the transfer function [104, 11] for the renal blood flow cannot be calculated. The characteristic features of the transfer function can provide insights into how the system responds to different frequencies of input signals and ensure further validation of the model [11]. Unfortunately, solving partial differential equations in a full-scale vasculature model is extremely computationally expensive and requires more advances in numerical methods and parallel computing to be solved in a reasonable time, which is beyond the scope of the Ph.D. project.

Moreover, although our model connects all afferent arterioles and nephrons in a full-scale tree structure, we have not covered interactions between nephrons that communicate via electrical signals propagating along the vessel walls. In our model, the adjusted afferent arterioles work together to autoregulate the blood flow, but each nephron adjusts its afferent arteriole independently. It has been found experimentally that neighboring nephrons interact with each other

to reach synchronization. For example, Holstein-Rathlou found synchronization of tubular pressure oscillations at the TGF frequency in the neighboring pairs of nephrons on the renal surface [105]. The interaction occurs due to electrical signals propagating along the vascular walls of the vessels. This concept was developed later in the experiments with laser speckle flowmetry [106, 107]. Several theoretical studies have addressed the effects of nephron-nephron interaction [108, 109, 110]. Unfortunately, we did not cover such interactions between nephrons because the electrical signal propagating along the full-scale vasculature requires a lot of computer powers and optimization of the current models and algorithms.

6.2.4 Construction of computational meshes for CFD studies

In Chapter 4.1.1, we presented the implicit functions to remap the reconstructed vascular tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to binary label maps with smooth transitions at branchings. Besides producing an artificial kidney dataset for training deep-learning-based segmentation models, organic-looking blood vessels with smooth transitions at branching can also be utilized for contouring methods to produce smooth surface meshes. These surface meshes can then be used as the input to many volume mesh generation algorithms like TetGen [111], to produce computational meshes. As introduced in Chapter 1.4.2, these computational meshes are essential for CFD-based studies to solve the Navier-Stokes equation numerically.

CFDs are not covered in this Ph.D. project due to the problems outlined in Chapter 1.4.3 such as limited resolution, inconsistent radius appearance, and computational bottleneck. The proposed fractal tree-based model is currently the only feasible way to simulate hemodynamics in a full-scale renal arterial vasculature because a graph structure is a significantly more efficient representation of vascular networks than its corresponding mesh or volume, and the Navier-Stokes equation can be solved analytically and simplified into a much simpler form, giving Poiseuille's equation as discussed in Chapter 2.2.

However, as outlined at the beginning of Chapter 2.2, Poiseuille's equation requires several further assumptions than the standard Navier-Stokes equation. Although most of the assumptions are feasible and also assumed in many studies of renal blood flows [24, 11, 15], some of them could be oversimplified. For example, it cannot model turbulence since it requires laminar flow, and similarly, it cannot model changes in the velocity field as it assumes steady flow.

Therefore, another potential approach to applying CFD-based blood flow simulation over the full-scale renal arterial structure is to assume a simplified skeleton-tree model of the vasculature only during the vascular reconstruction process (Chapter 2 and paper A). Then, instead of applying on the blood vessel segmentation from medical scans, the computational mesh can also be built out of the reconstructed skeletal tree model. The reconstructed meshes

can then be integrated into standard CFD studies, which does not require the assumptions for Poiseuille's equation. Unfortunately, the task of constructing computational mesh from skeletal trees is non-trivial. Although the result from the implicit surface in Fig. 4.2 visually shows smoother branches than simply stacking cylinders, it is not guaranteed to be watertight to generate computational meshes suitable for CFD simulations [111]. Remapping the efficient tree modeling of vasculature back to its computational meshes would also significantly increase computational complexities, posing further challenges to the memory bottleneck.

Bibliography

- [1] Csaba P Kovesdy. “Epidemiology of chronic kidney disease: an update 2022”. In: *Kidney International Suppl.* (2022).
- [2] Anil K Bidani et al. “Renal microvascular dysfunction, hypertension and CKD progression”. In: *Curr Opin Nephrol Hypertens* 22 (2013), pp. 1–9.
- [3] Lennart Tonneijck et al. “Glomerular Hyperfiltration in Diabetes: Mechanisms, Clinical Significance, and Treatment”. In: *Journal of the American Society of Nephrology* 28 (2017), pp. 1013–1039.
- [4] Marilyn Burke et al. “Molecular mechanisms of renal blood flow autoregulation”. In: *Current vascular pharmacology* 12.6 (2014), pp. 845–858.
- [5] Ruisheng Liu. “The real culprit behind diabetic nephropathy: impaired renal autoregulation?” In: *Physiological Reports* 5.5 (2017).
- [6] Mattias Carlström, Christopher S. Wilcox, and William J. Arendshorst. “Renal Autoregulation in Health and Disease”. In: *Physiological Reviews* 95 (2012), pp. 405–511.
- [7] Donald J Marsh et al. “The nephron-arterial network and its interactions”. In: *American Journal of Physiology-Renal Physiology* 316.5 (2019), F769–F784.
- [8] Walter F Boron and Emile L Boulpaep. *Medical physiology E-book*. Elsevier Health Sciences, 2016.
- [9] E.J. Johns and A.F. Ahmeda. “Renal Circulation”. In: *Reference Module in Biomedical Sciences*. Elsevier, 2014. ISBN: 978-0-12-801238-3.
- [10] Armin Just and William J Arendshorst. “Nitric oxide blunts myogenic autoregulation in rat renal but not skeletal muscle circulation via tubuloglomerular feedback”. In: *Journal of Physiology* 569.3 (2005), pp. 959–974.
- [11] N -H Holstein-Rathlou and DJ Marsh. “A dynamic model of renal blood flow autoregulation”. In: *Bulletin of mathematical biology* 56 (1994), pp. 411–429.

- [12] Leon C Moore, Jurgen Schnermann, and Shiroh Yarimizu. “Feedback mediation of SNGFR autoregulation in hydropenic and DOCA-and salt-loaded rats”. In: *American Journal of Physiology-Renal Physiology* 237.1 (1979), F63–F74.
- [13] TENEI Sakai, Eileen Hallman, and Donald J Marsh. “Frequency domain analysis of renal autoregulation in the rat”. In: *American Journal of Physiology-Renal Physiology* 250.2 (1986), F364–F373.
- [14] Frederick H Daniels, William J Arendshorst, and Robert G Roberds. “Tubuloglomerular feedback and autoregulation in spontaneously hypertensive rats”. In: *American Journal of Physiology-Renal Physiology* 258.6 (1990), F1479–F1489.
- [15] NH Holstein-Rathlou, Anthony J Wagner, and Donald J Marsh. “Tubuloglomerular feedback dynamics and renal blood flow autoregulation in rats”. In: *American Journal of Physiology-Renal Physiology* 260.1 (1991), F53–F68.
- [16] David A Nordsletten et al. “Structural morphology of renal vasculature”. In: *American Journal of Physiology-Heart and Circulatory Physiology* 291.1 (2006), H296–H309.
- [17] Edwin J Baldelomar et al. “Measuring rat kidney glomerular number and size in vivo with MRI”. In: *American Journal of Physiology-Renal Physiology* 314.3 (2018), F399–F406.
- [18] Stephen C Textor. “Renal arterial disease and hypertension”. In: *Medical Clinics* 101.1 (2017), pp. 65–79.
- [19] Shah Mohammed Abdul Khader et al. “Haemodynamics behaviour in normal and stenosed renal artery using computational fluid dynamics”. In: *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences* 51.1 (2018), pp. 80–90.
- [20] Pooja Jhunjunwala et al. “Prediction of blood pressure and blood flow in stenosed renal arteries using CFD”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 346. IOP Publishing. 2018, p. 012066.
- [21] Mohammad Andayesh, Azadeh Shahidian, and Majid Ghassemi. “Numerical investigation of renal artery hemodynamics based on the physiological response to renal artery stenosis”. In: *Biocybernetics and Biomedical Engineering* 40.4 (2020), pp. 1458–1468.
- [22] Weisheng Zhang et al. “Hemodynamic analysis of renal artery stenosis using computational fluid dynamics technology based on unenhanced steady-state free precession magnetic resonance angiography: preliminary results”. In: *The international journal of cardiovascular imaging* 30 (2014), pp. 367–375.

- [23] Donald J Marsh et al. “Architecture of the rat nephron-arterial network: analysis with micro-computed tomography”. In: *American Journal of Physiology-Renal Physiology* 313.2 (2017), F351–F360.
- [24] Dmitry D Postnov et al. “Modeling of kidney hemodynamics: probability-based topology of an arterial network”. In: *PLoS computational biology* 12.7 (2016), e1004922.
- [25] Arthur N Strahler. “Hypsometric (area-altitude) analysis of erosional topography”. In: *Geological society of America bulletin* 63.11 (1952), pp. 1117–1142.
- [26] Robert H More and G Lyman Duff. “The renal arterial vasculature in man”. In: *The American Journal of Pathology* 27.1 (1951), p. 95.
- [27] Mark J Horacek, Alvin M Earle, and Joseph P Gilmore. “The renal vascular system of the monkey: a gross anatomical description.” In: *Journal of Anatomy* 153 (1987), p. 123.
- [28] Daniel Casellas et al. “Anatomic pairing of afferent arterioles and renin cell distribution in rat kidneys”. In: *American Journal of Physiology-Renal Physiology* 267.6 (1994), F931–F936.
- [29] Alin-Florin Totorean et al. “Computational Fluid Dynamics Applications in Cardiovascular Medicine—from Medical Image-Based Modeling to Simulation: Numerical Analysis of Blood Flow in Abdominal Aorta”. In: *Advances in Fluid Mechanics: Modelling and Simulations*. Springer, 2022, pp. 1–42.
- [30] Alin F Totorean et al. “Medical image-based numerical simulation of the abdominal aorta flow”. In: *2021 10th International Conference on ENERGY and ENVIRONMENT (CIEM)*. IEEE. 2021, pp. 1–5.
- [31] R Vinoth et al. “Steady and transient flow CFD simulations in an aorta model of normal and aortic aneurysm subjects”. In: *The Proceedings of the International Conference on Sensing and Imaging*. Springer. 2019, pp. 29–43.
- [32] John D Martin. “CFD Analysis comparing steady flow and pulsatile flow through the aorta and its main branches”. In: *ASME International Mechanical Engineering Congress and Exposition*. Vol. 50534. American Society of Mechanical Engineers. 2016, V003T04A064.
- [33] Gaoyang Li et al. “Prediction of 3D Cardiovascular hemodynamics before and after coronary artery bypass surgery via deep learning”. In: *Communications biology* 4.1 (2021), p. 99.
- [34] Gianluca Rigatelli et al. “Applications of Computational Fluid Dynamics in Cardiovascular Disease”. In: *TTU J Biomed Sci* (2022).
- [35] Paul D Morris et al. “Computational fluid dynamics modelling in cardiovascular medicine”. In: *Heart* 102.1 (2016), pp. 18–28.

- [36] Erik J Bekkers and Charles A Taylor. “Multiscale vascular surface model generation from medical imaging data using hierarchical features”. In: *IEEE Transactions on Medical Imaging* 27.3 (2008), pp. 331–341.
- [37] Thomas Deschamps and Laurent D Cohen. “Fast extraction of tubular and tree 3D surfaces with front propagation methods”. In: *2002 International Conference on Pattern Recognition*. Vol. 1. IEEE. 2002, pp. 731–734.
- [38] Michail I Papafaklis et al. “Association of endothelial shear stress with plaque thickness in a real three-dimensional left main coronary artery bifurcation model”. In: *International journal of cardiology* 115.2 (2007), pp. 276–278.
- [39] Ryo Torii et al. “Stress phase angle depicts differences in coronary artery hemodynamics due to changes in flow and geometry after percutaneous coronary intervention”. In: *American Journal of Physiology-Heart and Circulatory Physiology* 296.3 (2009), H765–H776.
- [40] Jiafa He et al. “Learning hybrid representations for automatic 3D vessel centerline extraction”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 24–34.
- [41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.
- [42] Yuting He et al. “Dense biased networks with deep priori anatomy and hard region adaptation: Semi-supervised learning for fine renal artery segmentation”. In: *Medical image analysis* 63 (2020), p. 101722.
- [43] William E. Lorensen and Harvey E. Cline. “Marching Cubes: A High Resolution 3D Surface Construction Algorithm”. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’87. New York, NY, USA: Association for Computing Machinery, 1987, pp. 163–169. ISBN: 0897912276.
- [44] Yixin Hu et al. “Fast tetrahedral meshing in the wild”. In: *ACM Transactions on Graphics (TOG)* 39.4 (2020), 117:1–117:18.
- [45] Zohaib Khan et al. “Three-dimensional morphometric analysis of the renal vasculature”. In: *American Journal of Physiology-Renal Physiology* 314.5 (2018), F715–F725.

- [46] Zhiming Cui, Changjian Li, and Wenping Wang. “ToothNet: automatic tooth instance segmentation and identification from cone beam CT images”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6368–6377.
- [47] Chunliang Wang et al. “Pelvis segmentation using multi-pass U-Net and iterative shape estimation”. In: *International Workshop on Computational Methods and Clinical Applications in Musculoskeletal Imaging*. Springer. 2018, pp. 49–57.
- [48] Faezeh Moshfeghifar et al. “LibHip: An open-access hip joint model repository suitable for finite element method simulation”. In: *Computer Methods and Programs in Biomedicine* 226 (2022), p. 107140.
- [49] Torkan Gholamalizadeh et al. “Open-Full-Jaw: An open-access dataset and pipeline for finite element models of human jaw”. In: *Computer Methods and Programs in Biomedicine* 224 (2022), p. 107009.
- [50] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).
- [51] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.
- [52] Justin S Paul et al. “Deep learning for brain tumor classification”. In: *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging*. Vol. 10137. SPIE. 2017, pp. 253–268.
- [53] Seok Won Chung et al. “Automated detection and classification of the proximal humerus fracture by using deep learning algorithm”. In: *Acta orthopaedica* 89.4 (2018), pp. 468–473.
- [54] Fabian Isensee et al. “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation”. In: *Nature methods* 18.2 (2021), pp. 203–211.
- [55] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.
- [56] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).
- [57] Alec Radford et al. “Improving language understanding by generative pre-training”. In: (2018).
- [58] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

- [59] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).
- [60] Jiaqi Wang et al. “Review of large vision models and visual prompt engineering”. In: *Meta-Radiology* (2023), p. 100047.
- [61] Alexander Kirillov et al. “Segment anything”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4015–4026.
- [62] Maciej A Mazurowski et al. “Segment anything model for medical image analysis: an experimental study”. In: *Medical Image Analysis* 89 (2023), p. 102918.
- [63] Yuhao Huang et al. “Segment anything model for medical images?” In: *Medical Image Analysis* (2023), p. 103061.
- [64] Jun Ma et al. “Segment anything in medical images”. In: *Nature Communications* 15.1 (2024), p. 654.
- [65] Cecil D Murray. “The physiological principle of minimum work: I. The vascular system and the cost of blood volume”. In: *Proceedings of the National Academy of Sciences* 12.3 (1926), pp. 207–214.
- [66] Cecil D Murray. “The physiological principle of minimum work: II. Oxygen exchange in capillaries”. In: *Proceedings of the National Academy of Sciences* 12.5 (1926), pp. 299–304.
- [67] Jan Kretschmer et al. “Interactive Patient-Specific Vascular Modeling with Sweep Surfaces”. In: *IEEE Transactions on Visualization and Computer Graphics* 19 (Dec. 2013), pp. 2828–37.
- [68] Marwan Abdellah et al. “Interactive visualization and analysis of morphological skeletons of brain vasculature networks with VessMorpho-Vis”. In: *Bioinformatics (Oxford, England)* 36 (July 2020), pp. i534–i541.
- [69] Peter J Yim et al. “Vessel surface reconstruction with a tubular deformable model”. In: *IEEE transactions on medical imaging* 20.12 (2001), pp. 1411–1421.
- [70] Wolfgang Schreiner et al. “Constrained constructive optimization of arterial tree models”. In: *Scaling in biology* 145 (2000), p. 65.
- [71] Manfred Georg, Tobias Preusser, and Horst K Hahn. “Global constructive optimization of vascular systems”. In: (2010).
- [72] Horst K Hahn, Manfred Georg, and Heinz-Otto Peitgen. “Fractal aspects of three-dimensional vascular constructive optimization”. In: *Fractals in biology and medicine*. Springer, 2005, pp. 55–66.

- [73] L.F.M. Cury et al. “Parallel generation of extensive vascular networks with application to an archetypal human kidney model”. In: *Royal Society open science* 8.12 (2021), p. 210973.
- [74] Elif Tekin et al. “Do vascular networks branch optimally or randomly across spatial scales?” In: *PLoS computational biology* 12.11 (2016), e1005223.
- [75] Frank P Tiecks et al. “Comparison of static and dynamic cerebral autoregulation measurements”. In: *Stroke* 26.6 (1995), pp. 1014–1019.
- [76] Charlene McQueen. *Comprehensive toxicology*. Elsevier, 2017.
- [77] Krishna J Rocha-Singh et al. “Atherosclerotic Peripheral Vascular Disease Symposium II: intervention for renal artery disease”. In: *Circulation* 118.25 (2008), pp. 2873–2878.
- [78] Alan T Hirsch et al. “ACC/AHA 2005 practice guidelines for the management of patients with peripheral arterial disease (lower extremity, renal, mesenteric, and abdominal aortic) a collaborative report from the American Association for Vascular Surgery/Society for Vascular Surgery,* Society for Cardiovascular Angiography and Interventions, Society for Vascular Medicine and Biology, Society of Interventional Radiology, and the ACC/AHA Task Force on Practice Guidelines (writing committee to develop guidelines for the management of patients with peripheral arterial disease): Endorsed by the American Association of Cardiovascular and Pulmonary Rehabilitation; National Heart, Lung, and Blood Institute; Society for Vascular Nursing; TransAtlantic Inter-Society Consensus; and Vascular Disease Foundation”. In: *circulation* 113.11 (2006), e463–e654.
- [79] Andreas Bærentzen and Eva Rotenberg. “Skeletonization via local separators”. In: *ACM Transactions on Graphics (TOG)* 40.5 (2021), pp. 1–18.
- [80] Ta-Chih Lee, Rangasami L Kashyap, and Chong-Nam Chu. “Building skeleton models via 3-D medial surface axis thinning algorithms”. In: *CVGIP: Graphical Models and Image Processing* 56.6 (1994), pp. 462–478.
- [81] Andrea Tagliasacchi et al. “3d skeletons: A state-of-the-art report”. In: *Computer Graphics Forum*. Vol. 35. Wiley Online Library. 2016, pp. 573–597.
- [82] Jun-Yan Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

- [83] Özgün Çiçek et al. “3D U-Net: learning dense volumetric segmentation from sparse annotation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17–21, 2016, Proceedings, Part II 19*. Springer. 2016, pp. 424–432.
- [84] Jules Bloomenthal and Ken Shoemake. “Convolution surfaces”. In: *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. 1991, pp. 251–256.
- [85] Steffen Oeltze and Bernhard Preim. “Visualization of vasculature with convolution surfaces: method, validation and evaluation”. In: *IEEE Transactions on Medical Imaging* 24.4 (2005), pp. 540–548.
- [86] Luca Antiga and David A Steinman. “Robust and objective decomposition and mapping of bifurcating vessels”. In: *IEEE transactions on medical imaging* 23.6 (2004), pp. 704–713.
- [87] Scott Schaefer, Tao Ju, and Joe Warren. “Manifold Dual Contouring”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.3 (2007), pp. 610–619.
- [88] Martin J Menten et al. “Physiology-based simulation of the retinal vasculature enables annotation-free segmentation of OCT angiographs”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*. Springer. 2022, pp. 330–340.
- [89] Antonia Creswell et al. “Generative adversarial networks: An overview”. In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.
- [90] Meenal R. Kale et al. “Enhancing Cryptojacking Detection Through Hybrid Black Widow Optimization and Generative Adversarial Networks”. In: *International Journal of Advanced Computer Science and Applications* 15.3 (2024).
- [91] Toan Duc Bui et al. “6-month infant brain MRI segmentation guided by 24-month data using cycle-consistent adversarial networks”. In: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2020, pp. 359–362.
- [92] Andrew Gilbert et al. “Generating synthetic labeled data from existing anatomical models: an example with echocardiography segmentation”. In: *IEEE Transactions on Medical Imaging* 40.10 (2021), pp. 2783–2794.
- [93] Xiaojun Chen et al. “Image-guided installation of 3D-printed patient-specific implant and its application in pelvic tumor resection and reconstruction surgery”. In: *Computer Methods and Programs in Biomedicine* 125 (2016), pp. 66–78.

- [94] Mathias Perslev et al. “One network to segment them all: A general, lightweight system for accurate 3d medical image segmentation”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22*. Springer. 2019, pp. 30–38.
- [95] Uwe Querfeld, Robert H Mak, and Axel Radlach Pries. “Microvascular disease in chronic kidney disease: the base of the iceberg in cardiovascular comorbidity”. In: *Clinical science* 134.12 (2020), pp. 1333–1356.
- [96] CL Walsh et al. “Imaging intact human organs with local resolution of cellular structures using hierarchical phase-contrast tomography”. In: *Nature methods* 18.12 (2021), pp. 1532–1541.
- [97] Ekin Yagis et al. “Deep Learning for Vascular Segmentation and Applications in Phase Contrast Tomography Imaging”. In: *arXiv preprint arXiv:2311.13319* (2023).
- [98] Shahrokh Rahmani et al. “Micro to macro scale analysis of the intact human renal arterial tree with Synchrotron Tomography”. In: *BioRxiv* (2023).
- [99] Robin Rombach et al. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.
- [100] Chen Henry Wu and Fernando De la Torre. “A latent space of stochastic diffusion models for zero-shot image editing and guidance”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 7378–7387.
- [101] Bastian Wittmann et al. “Simulation-Based Segmentation of Blood Vessels in Cerebral 3D OCTA Images”. In: *arXiv preprint arXiv:2403.07116* (2024).
- [102] Linus Kreitner et al. “Synthetic optical coherence tomography angiographs for detailed retinal vessel segmentation without human annotations”. In: *IEEE Transactions on Medical Imaging* (2024).
- [103] Emmeline Brown et al. “Physics-informed deep generative learning for quantitative assessment of the retina”. In: *bioRxiv* (2023), pp. 2023–07.
- [104] Julius S Bendat and Allan G Piersol. *Random data: analysis and measurement procedures*. John Wiley & Sons, 2011.
- [105] N -H Holstein-Rathlou. “Synchronization of proximal intratubular pressure oscillations: evidence for interaction between nephrons”. In: *Pflügers Archiv* 408 (1987), pp. 438–443.
- [106] Alexey R Brazhe et al. “Synchronized renal blood flow dynamics mapped with wavelet analysis of laser speckle flowmetry data”. In: *PloS one* 9.9 (2014), e105879.

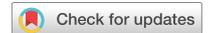
-
- [107] Dmitry Postnov et al. “Synchronization in renal microcirculation unveiled with high-resolution blood flow imaging”. In: *Elife* 11 (2022), e75284.
- [108] HE Layton, E Bruce Pitman, and Leon C Moore. “Bifurcation analysis of TGF-mediated oscillations in SNGFR”. In: *American Journal of Physiology-Renal Physiology* 261.5 (1991), F904–F919.
- [109] Donald J Marsh et al. “Nonlinear interactions in renal blood flow regulation”. In: *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology* 288.5 (2005), R1143–R1159.
- [110] Donald J Marsh et al. “Multinephron dynamics on the renal vascular network”. In: *American Journal of Physiology-Renal Physiology* 304.1 (2013), F88–F102.
- [111] Si Hang. “TetGen, a Delaunay-based quality tetrahedral mesh generator”. In: *ACM Trans. Math. Softw* 41.2 (2015), p. 11.

List of publications

- Appendix A** Peidi Xu, Niels-Henrik Holstein-Rathlou, Stinne Byrholdt Søggaard, Carsten Gundlach, Charlotte Mehlin Sørensen, Kenny Erleben, Olga Sosnovtseva, and Sune Darkner. “A hybrid approach to full-scale reconstruction of renal arterial network.” *Scientific Reports* 13(1), p.7569, 2023. <https://doi.org/10.1038/s41598-023-34739-y>
- Appendix B** Peidi Xu, Sune Darkner, Olga Sosnovtseva, and Niels-Henrik Holstein-Rathlou. “Full-scale model and simulation of renal autoregulation.” Manuscript to be submitted to *Function*, 2024.
- Appendix C** Peidi Xu, Blaire Lee, Olga Sosnovtseva, Charlotte Mehlin Sørensen, Kenny Erleben, and Sune Darkner. “Extremely weakly-supervised blood vessel segmentation with physiologically based synthesis and domain adaptation.” In *Workshop on Medical Image Learning with Limited and Noisy Data* (MICCAI Workshop), pp. 191-201. Cham: Springer Nature Switzerland, 2023. https://doi.org/10.1007/978-3-031-44917-8_18
- Appendix D** Peidi Xu, Faezeh Moshfeghifar, Torkan Gholamalizadeh, Michael Bachmann Nielsen, Kenny Erleben, and Sune Darkner. “Auto-segmentation of Hip Joints Using MultiPlanar UNet with Transfer Learning” In *Workshop on Medical Image Learning with Limited and Noisy Data* (MICCAI Workshop), pp. 153-162. Cham: Springer Nature Switzerland, 2022. https://doi.org/10.1007/978-3-031-16760-7_15
- Appendix E** Peidi Xu, Torkan Gholamalizadeh, Faezeh Moshfeghifar, Sune Darkner, and Kenny Erleben. “Deep-learning-based segmentation of individual tooth and bone with periodontal ligament interface details for simulation purposes.” *IEEE Access* 11, pp. 102460-102470, 2023. <https://doi.org/10.1109/ACCESS.2023.3317512>

Appendix A

Paper A



OPEN

A hybrid approach to full-scale reconstruction of renal arterial network

Peidi Xu¹✉, Niels-Henrik Holstein-Rathlou², Stinne Byrholdt Søgaard², Carsten Gundlach³, Charlotte Mehlin Sørensen², Kenny Erleben¹, Olga Sosnovtseva^{2,4} & Sune Darkner^{1,4}

The renal vasculature, acting as a resource distribution network, plays an important role in both the physiology and pathophysiology of the kidney. However, no imaging techniques allow an assessment of the structure and function of the renal vasculature due to limited spatial and temporal resolution. To develop realistic computer simulations of renal function, and to develop new image-based diagnostic methods based on artificial intelligence, it is necessary to have a realistic full-scale model of the renal vasculature. We propose a hybrid framework to build subject-specific models of the renal vascular network by using semi-automated segmentation of large arteries and estimation of cortex area from a micro-CT scan as a starting point, and by adopting the Global Constructive Optimization algorithm for generating smaller vessels. Our results show a close agreement between the reconstructed vasculature and existing anatomical data obtained from a rat kidney with respect to morphometric and hemodynamic parameters.

Computational models of vital organs play an increasing role in the understanding of both normal and diseased organ function. Realistic models of organs require not only a detailed description of the various biochemical and physiological processes but also an accurate representation of the essential anatomy of the organ. In the kidney, the vasculature plays a special role. Not only does it function as a resource distribution network, supplying the individual nephrons with blood and nutrients, but it also constitutes a communication network, allowing contiguous nephrons to interact through electric signaling along the vessels¹. However, no imaging techniques allow a full reconstruction of the structure of the renal vasculature due to limited spatial resolution.

The primary aim of the present work is the development of a hybrid framework that allows the reconstruction of a realistic model of the full-scale renal vasculature, which matches real anatomical data and can be used in advanced mathematical models of renal function. In addition, real-looking networks that respect the true geometric properties of the organs could allow the simulation of CT scans on the generated vascular trees^{2,3}. Such simulated CT scans could help train AI-based models for vascular segmentation on real CT scans, which is one of the long-range aims of the present work. Our approach utilizes existing scans to extract 3D geometrical priors and adopts physiologically based criteria in the construction of the vascular tree^{4–6}.

Biological background. In each organ, the vasculature has a characteristic structure adapted to meet the specific needs of the organ, and a detailed description of the specific vasculature of an organ is necessary for a full understanding of both its physiology and pathophysiology. Nordsletten et al.⁷ provided the hitherto most detailed and quantitative description of the rat renal vasculature. They combined high (4 μm) and low (20 μm) resolution micro-CT images obtained from a vascular cast of a rat kidney. They used the skeletonization method to trace the path of contiguous vessels and then applied the Strahler approach⁸ to sort and interpret the data. Strahler ordering sorts treelike networks by the diameter of the branches according to a bifurcating scheme (see “[Strahler ordering](#)”). The principal assumption required for its use is the existence of a diameter-based hierarchy of vessels, ending in the narrowest vessels, i.e., the afferent arterioles supplying the individual nephrons.

Marsh et al.⁹ used micro-CT with 2.5 μm resolution to assess the three-dimensional microvascular structure of the rat renal arterial tree. The cast revealed an arterial tree network originating in arcuate arteries, branching as few as twice or as many as six times before reaching a terminal artery that terminated in pairs, triplets, or quadruplets of afferent arterioles. Marsh et al. identified different motives for how afferent arterioles originated

¹Department of Computer Science, University of Copenhagen, Universitetsparken 1, Copenhagen 2100, Denmark. ²Department of Biomedical Sciences, University of Copenhagen, Blegdamsvej 3B, Copenhagen 2200, Denmark. ³Department of Physics, Technical University of Denmark, Kongens Lyngby, Copenhagen 2800, Denmark. ⁴These authors contributed equally: Olga Sosnovtseva and Sune Darkner. ✉email: peidi@di.ku.dk

from all branch orders of nonterminal arteries or from terminal arteries forming the tops of the arterial trees. Similar branching patterns have been reported by other groups using microdissected trees from four different mammalian species^{10–12}.

Postnov et al.¹³ have shown that the pressure drop in a simple bifurcating tree with the vessel dimensions reported by Norsletten et al.⁷ exceeds the value found experimentally. This is expected since in a simple bifurcating tree, afferent arterioles appear only at the terminal branch points of the tree—an assumption that maximizes the hemodynamic resistance between the renal artery and the glomerulus. Postnov et al.¹³ and Marsh et al.⁹ have reported an exponential distribution of the distances between branch points for afferent arterioles across the vascular tree. This distribution, and the possibility to branch from any arterial segment, is the basis for the pressure in the glomerular capillaries being significantly higher, and in a range compatible with normal nephron function, than in a simple bifurcating tree.

The number of nephrons in a kidney in a given species is variable and is thought to play an important role in renal health. Baldelomar et al. estimated nephron numbers from in vivo images and from high-resolution ex vivo images¹⁴, while Letts et al.¹⁵ assessed the location of glomeruli in the outer 30% of the cortex, midcortical nephrons (30–60%), and juxtamedullary nephrons of the inner 40% of the cortex. Both studies show high variability in the number and characteristics of nephrons. Taken together, the high variability, both in the structure of the renal vasculature and in the number of nephrons in a given kidney, suggests that a probabilistic-based approach to model nephron-vascular architecture and blood flow dynamics is the right choice.

Modeling outlook. Three major methods have been described in the literature to construct models of vascular networks. Pure rule-based models^{1,13,16} generate vascular trees analytically from a given root while completely ignoring the spatial structure of the network. The length of each vessel, the radius distribution to its children in a bifurcation, as well as the stopping criteria are all derived from given probability distributions obtained from experimental data. Although the hemodynamics can be simulated without information on the spatial structure, these methods cannot generate real-looking networks and ignore the subject-specific information, and thus cannot be utilized for individual analysis.

The image-based reconstruction methods build 3-D geometric models that capture the high-level structure of an individual's blood vessels from clinical images^{17–19}. These methods involve either a segmentation followed by a centerline extraction or a direct tracking of the blood vessels. Despite advances in deep learning models, learning from very thin structures is still challenging and will suffer from errors due to both vessel merging and discontinuity, resulting in extremely intense manual work afterward. More importantly, in the kidney the vessels at far-surface regions are beyond the experimental resolution, making it impossible to detect the small vessels from an image alone. These small vessels, however, are the ones supplying the individual nephrons and thus have to be resolved in the final model. Therefore, image-based reconstruction alone is unable to provide complete and detailed 3-D vasculatures in the kidney, making biosimulation the only tool available with generative models that can extrapolate modeling to unresolved parts of the kidney.

The angiogenesis-based methods simulate the growth of vasculatures by considering the biological and physiological factors involved in the process such as the size of branching vessels (Murray's law^{5,20} (cf. Eq. (9))) and the hemodynamics in the tree²¹ (cf. Eq. (5)). These algorithms model the vascular tree growing as an optimization problem following the assumption that the network achieves a topological and geometrical structure over the vascularized tissues from hemodynamic principles²², e.g., by minimizing the intravascular volume of the tree while ensuring efficient flow. The details of the optimization function are presented in “Physiologically based cost functions”.

There are two main methods to generate the vasculature based on the growing algorithm, namely, Constrained Constructive Optimization (CCO) method proposed by Schreiner and Buxbaum⁴, and its variant Global Constructive Optimization (GCO) proposed by Georg et al.⁶. Both the CCO and GCO algorithms grow the tree inside a pre-defined perfusion territory. In both algorithms, a single tree root location of the blood inlet is chosen manually. In addition, boundary conditions such as terminal radius and flow distributions are imposed to represent physiologic conditions.

These methods are able to generate real-looking vascular structures with both spatial location and connectivity information and have been applied in the liver, heart (left ventricle), and eye^{4,6,23}. Recently, Shen et al.²⁴ and Li et al.²⁵ incorporated GCO and CCO, respectively, to reconstruct vasculatures in the human brain. Although Cury et al.²² has recently applied an adaptive CCO on a prototypical human kidney model, no similar research exists on real renal vasculatures due to the complex non-convex geometry. Moreover, most of the studies produce homogeneous vascular networks that do not account for individual differences, so they cannot be used for individual analysis.

Both CCO and GCO require a convex structure since the connections between any two nodes should not leave the structure. This is one of the reasons why it is challenging to adopt these methods to an organ like the kidney with a complex internal structure (i.e., not convex). Some parts, like the renal pelvis and the pyramids, also pose intrinsic spatial restrictions on vessel construction, which are difficult to model when the tree grows from only a single root node.

Our work follows a similar idea of²⁴ by proposing a hybrid way to incorporate subject-specific image-based priors via a semi-automated segmentation of the main (large) arteries and an automatic cortex approximation from the ex-vivo micro-CT scan of a real rat kidney, which both are utilized in the GCO initialization step. In summary, a pre-built arterial tree consisting of the main arteries is extracted from the main artery segmentation, while we also propose a novel approach to sample terminal nodes (glomerulus) from the estimated renal cortex while maximizing the distance between any two neighboring nodes using Poisson disk sampling²⁶.

These sampled terminal nodes are then connected to the pre-built vascular tree. Instead of growing from a single root position, the algorithm can now start growing from a pre-built vascular tree and thus will retain subject-specific information in the final generated full-scale vascular tree. At the same time, this procedure avoids violating the intrinsic spatial constraints, because these connections naturally avoid penetrating the middle part of the kidney where the renal medulla resides and do not cross the kidney's outer boundary, thus making the complex structure piece-wise convex. In contrast to²⁴ which proposes forest growth, our algorithm consists of a single tree but with pre-built large branches because of the single inlet of the renal artery. Currently, the image prior starts with a semi-automated segmentation of the vessels, which is time-consuming work. However, recent advances in deep learning are likely in the future to make it possible to automatically segment large vessels given a decent amount of labeled training data, especially in the case of micro-CT with relatively low variation among the images. Apart from the segmentation retrieval and a single user-defined root position, our work is fully automatic, meaning that no software interface is involved in the process. In particular, the leaf nodes sampling and centerline extraction are both implemented in pure Python, unlike²⁴ which used BrainSuite for cortex extraction, and²⁵ which used Amira for brain hemispheres extraction and skeletonization. All the 3-D software packages, e.g. 3D Slicer²⁷ and ParaView²⁸ are only used for visualization. This has the advantage that it allows automatization of the process, does not require expert knowledge of the software packages, and provides for better flexibility, e.g., adjustment of parameters to adapt to other organs or image modalities.

Our results show that the structural and functional properties of the reconstructed vascular network are in good agreement with existing anatomical data, e.g., with respect to the radius, length, and pressure distributions^{7,29}. We expect the reconstructed full-scale model of the renal vasculature can be utilized to develop realistic computer simulations of renal function or model pathological changes in the kidney, and to develop new image-based diagnostic methods based on artificial intelligence.

Results

The output from the hybrid framework is a reconstructed renal arterial tree, which begins at the renal artery and ends in the afferent arterioles. In this section, we first present visualizations of the reconstructed tree. In order to quantitatively characterize the tree, we compute the statistics of various variables, e.g., vessel radius and frequency, with respect to the Strahler order (see “[Strahler ordering](#)”) of the vessels.

Model implementation and rendering. Our hybrid modeling approach to reconstruct a renal vascular network combines semi-automated segmentation of large arteries from micro-CT images and the Global Constructive Optimization algorithm for the generation of smaller microvessels (Fig. 1 and is described in detail in “[Methods](#)”). The raw scan has an isotropic voxel size of $22.6\ \mu\text{m}^3$. From Table 2 in Nordsletten's paper⁷, renal

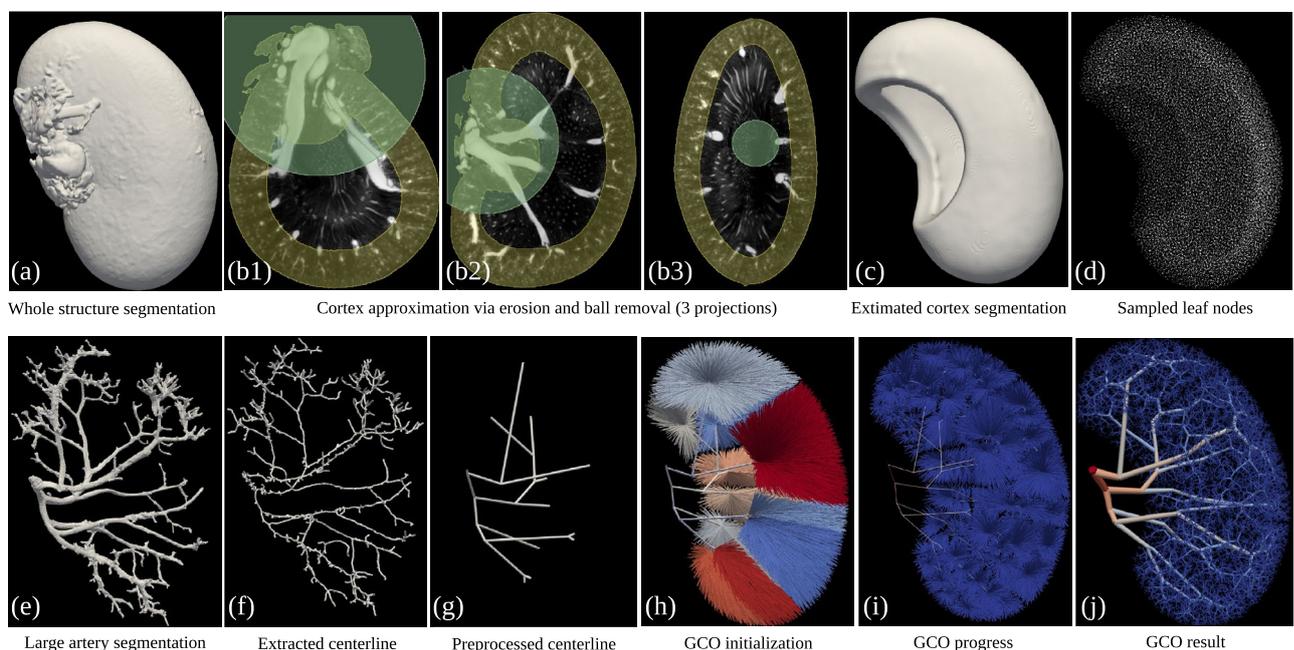


Figure 1. GCO Pipeline, visualized by 3D Slicer²⁷ and ParaView²⁸. The initial micro-CT scan is used to extract whole structure segmentation (a) and large artery segmentation (e). Top row: renal cortex (c) is approximated by a subtraction of erosion followed by a ball removal (b), where the leaf nodes (d) are sampled using Poisson disk sampling. Bottom row: extracted centerline (f) is pre-processed to pre-build a renal arterial tree consisting of only the first few large arteries (g). In GCO initialization (h), all the sampled leaf nodes (d) are connected to the nearest node in the pre-built tree (g) with color indicating the group of leaf nodes that are connected to the same node. Colors in the GCO progress and result (i,j) indicate the radius of each vessel: from $300\ \mu\text{m}$ in renal artery to $10\ \mu\text{m}$ in afferent arterioles (AA).

arteries with Strahler order from 0 to 2 have a mean radius of 10.08, 13.90, 20.06 μm respectively, making it impossible to detect those small arteries from the 22.6 μm^3 scans. Using our hybrid approach, the small arteries are successfully resolved and connect to the large branches in an anatomically correct manner, producing a full-scale renal arterial tree.

Figure 1j presents a visualization of the generated vascular tree. Each vessel is visualized as a separate cylinder with a thickness corresponding to its radius and color coded by the radius. Although more advanced visualization methods exist^{30,31}, this provides a sufficient rendering of the topology of the generated vascular tree. Clearly, if the tree is to be used for more advanced purposes, such as Navier–Stokes-based flow calculations, visualization methods that produce smooth surfaces at junctions and allow the construction of volume meshes will be needed. 3D gif animations with rotation are available at: <https://github.com/KidneyAnonymous/RenalArterialTree>.

Morphometric validation. Numerical validation is done by comparing the morphometric properties as shown in Fig. 2, such as the distribution of vessel radius, branch length, and Strahler order in the reconstructed network with data from a real renal arterial tree collected in a rat kidney by Nordsletten et al.⁷. The radius and length of the vessels in a kidney are variable and depend on many factors, including the strain, age, and size of the rat. Although Nordsletten et al.⁷ provided the most detailed and quantitative description of the rat renal vasculature, their collected data are from one kidney in one rat and, therefore, cannot be regarded as a “gold standard”. Instead, it represents one sample from the total population of rats. The purpose of the comparison between the simulated vascular tree and the experimental data is, therefore, not to demonstrate an absolute closeness or identity between the lengths and radii but rather to compare the topology and the distribution of vessel lengths, radii and Strahler orders between them.

In the rat kidney, the radius was found to increase exponentially with the Strahler order of the vessel. The same feature is present in the simulated tree as shown in Fig. 2a. This is contrary to²⁴ which shows that the result from GCO and from anatomical data on the brain vasculature both follow a linear increase of radius with Strahler order. This indicates that although the cost function and general process are similar among organs, GCO is able to adjust based on the distinct geometrical features of each organ. The exact values for the radii of the reconstructed vascular tree deviate somewhat from the values reported in⁷, as can be seen in Fig. 2a. This is especially evident for the vessel of the largest Strahler order (the root), which corresponds to the renal artery. In the reconstructed tree, all radii are calculated from the radii of the afferent arterioles using Murray’s law^{5,20} (cf. Eq. (9)). In our initialization process, we assume that the 30K afferent arterioles are derived from the distribution $r_0 \sim \mathcal{N}(10.08, 0.14)$ Ref.⁷ (first row of Table 2). Given strict compliance with Murray’s law, the root radius (radius at Strahler order 10) can be computed analytically by $r_{10} = \sqrt[3]{\sum_{i=1}^{n=30,000} r_{0,i}^3}$ regardless of the branching patterns,

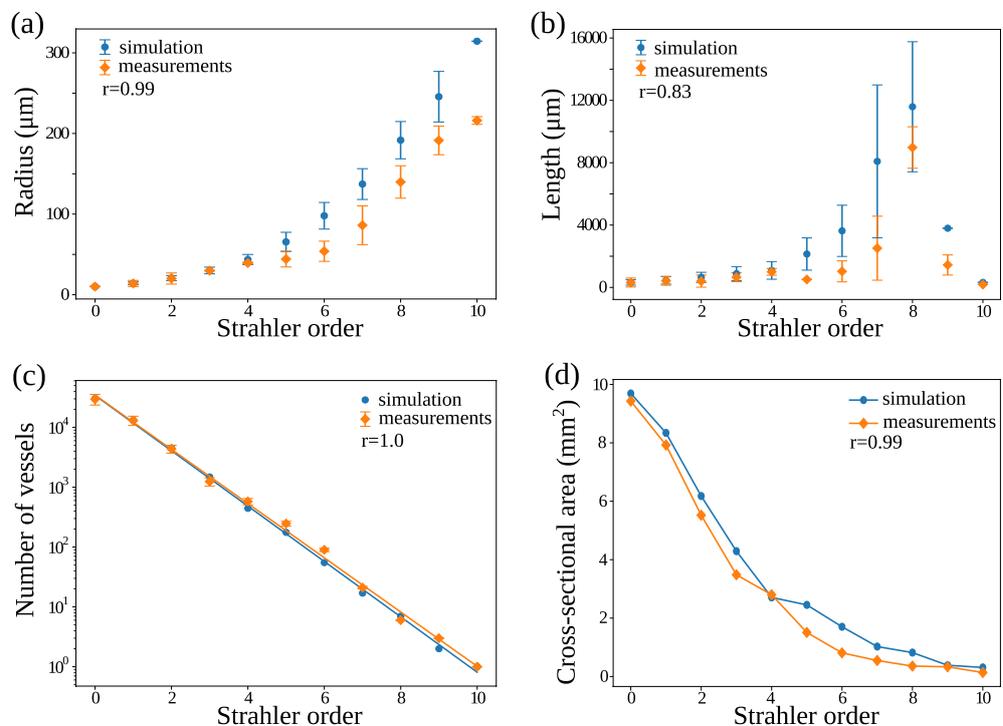


Figure 2. Morphometric features of the generated renal vascular arterial network (simulation) and the experimental data reported in the literature (measurements). In each subfigure, r indicates the Pearson correlation coefficient of the mean values with respect to Strahler order. (a) Vessel radius vs Strahler order. (b) Vessel length vs Strahler order. (c) Number of vessels (in log scale) of a particular Strahler order. (d) Total cross-sectional area vs Strahler order.

which will give a mean value around $\mu(r_{10}) \approx 313.21 \mu\text{m}$. This number matches our result but deviates from Ref.⁷ (last row of Table 2) where $r_{10} \sim \mathcal{N}(216.10, 4.74)$. Similarly, we plot the vessel length for each Strahler order, both from the literature⁷ and from our result, as shown in Fig. 2b. Both the data in the literature and our work show that vessel length has a poor correlation with Strahler order. However, they both show a maximum at order 8, indicating that the large arteries usually are longer, but also that they branch fast when being close to the root, meaning that starting from the renal artery (order 10), vessels only grow to a small length before they branch. This results in a decrease of length with increasing Strahler order from 8 to 10.

Moreover, we plot the number of vessels vs the Strahler order, both from the literature⁷ and from our result, as shown in Fig. 2c. The data from both the literature and our work show an exponential decrease in the vessel numbers vs the Strahler order. As a result, both the vessel numbers from the literature and from our generated tree fit very well to a straight line in log scale.

We further plot the total cross-sectional area vs the Strahler order from our GCO output and the experimental data⁷ in Fig. 2d. Note that the experimental data were extracted from a figure in Ref.⁷ (Fig. 12) and replotted here in Fig. 2d. In agreement with experimental observations, the total cross-sectional area in the generated vascular tree decreases, although the mean radius increases exponentially with Strahler order. Specifically, in both experimental data and our simulated vasculature, the cross-sectional area decreases from a value around 10 mm^2 , halves after order 2, and keeps decreasing to a value below 0.5 mm^2 at the renal artery (order 10). This indicates that the decrease in the number of vessels exceeds the exponential growth of the mean radius with Strahler order. A final interesting property is the Strahler order of the parent vessel of each afferent arteriole, shown in Fig. 4a. Specifically, a parent Strahler order 1 means that afferent arterioles (order 0) branch from terminal arteries (order 1). This case indeed consists of most of the scenarios, but it also demonstrates other possibilities, where afferent arterioles can branch from larger vessels. The parent vessels of the afferent arterioles have Strahler orders from 1 to 8, meaning that in our model, afferent arterioles can branch from any of the larger vessels, except the largest vessels with Strahler order 9 and 10. This characteristic has been shown to be crucial for the ability of the renal vascular tree to supply the glomeruli with blood at a sufficient pressure^{1,13}.

Figure 2 also shows the Pearson correlation coefficient of the mean values of the morphometric features between the simulated and experimental data. All the correlations are highly statistically significant and confirm a good agreement between the morphometric features of the two vascular trees.

The above results show that the distribution of e.g., radius and vessel frequency with respect to Strahler order in our generated arterial tree agrees with the literature⁷. Some further properties of the generated vascular tree are noteworthy. The number of Strahler orders in the generated tree was not specified in the optimization process, but notably, the process resulted in a tree with 11 Strahler orders, which match with⁷. Furthermore, given the assumption of around 30 K leaves, our generated tree produces around 51 K vessel segments (edges) in total, with 23 levels (depth) and 11 Strahler orders. On the contrary, if the algorithm simply builds a complete binary tree with perfect symmetries, the total number of vessel segments in the generated tree can be computed by $2 \times 30 \text{ K} = 60 \text{ K}$ analytically, given the assumption of 30 K leaves. Then, the total number of levels and Strahler orders are identical, and can be computed analytically by $\log_2 60 \text{ K} \approx 16$, which significantly deviates from⁷. This means that asymmetries are correctly inherited in our constructive optimization process, which agrees with the literature that vascular trees are not symmetrically balanced^{32–34}.

Physiological features. To examine the physiological properties of the generated tree, we plot the blood flow and pressure distribution in Fig. 3. The flow associated with each vessel is derived from the zero-addition rule (cf. Eq. (8)) and the assumption of equal flow distribution among the afferent arterioles. As shown in Fig. 3a, the blood flow over our generated renal arterial network ranges from $1.2 \times 10^{11} \mu\text{m}^3/\text{s}$ (7 ml/min) in the renal artery to around $4 \times 10^6 \mu\text{m}^3/\text{s}$ (240 nl/min) in afferent arterioles (AA).

We further plot the flow in each vessel vs the Strahler order in Fig. 3b (in log scale), which shows a clear exponential increase in flow with Strahler order. Although we have found no literature on such statistics in the rat kidney, this exponential increase is in close agreement with³⁵ which measures the coronary blood flow vs Strahler order.

The pressure drop Δp_i along each vessel i in the generated vascular tree can be computed by Hagen-Poiseuille's law, cf. Eq. (5). Therefore, given the boundary condition of the inlet pressure p_0 , the pressure value at every node along the generated tree can be computed by a simple breadth-first-search with $p_{i+1} = p_i - \Delta p_i$, where p_{i+1} and p_i denote the pressure at the outlet and inlet of vessel i respectively. From the literature²⁹, pressure in the renal artery is around 90–110 mmHg, we hereby assumed an inlet pressure $p_0 = 100 \text{ mmHg}$.

The pressure at each node in Fig. 3c shows a smooth decrease from 100 mmHg to a minimum of around 30 mmHg along the network without abrupt changes, indicating that the reconstructed vascular network produces physiologically feasible hemodynamic behaviors.

We plot node pressure (in mmHg) at the outlet of each vessel vs Strahler order of the generated renal arterial network in Fig. 3d, which shows a smooth and linear increase from around 55 mmHg at the end of afferent arterioles (Strahler order 0) to near 100 mmHg at the end of vessels with Strahler order 9, which then becomes flatter at the last order. This is expected because the root vessel has a short length l and a large radius r , resulting in a small pressure drop.

We further plot the histogram of the pressures at the end of the afferent arterioles in Fig. 4b. Experimental data²⁹ shows that the pressure at the end of the afferent arteriole is around 50–55 mmHg, which is in close agreement with the mean value from our result. However, since we only have a reasonable topological structure of the vascular tree but have not modeled the active regulation of pressure in the vascular tree, the histogram shows a wider distribution than found experimentally. In the future, such regulation and interaction among

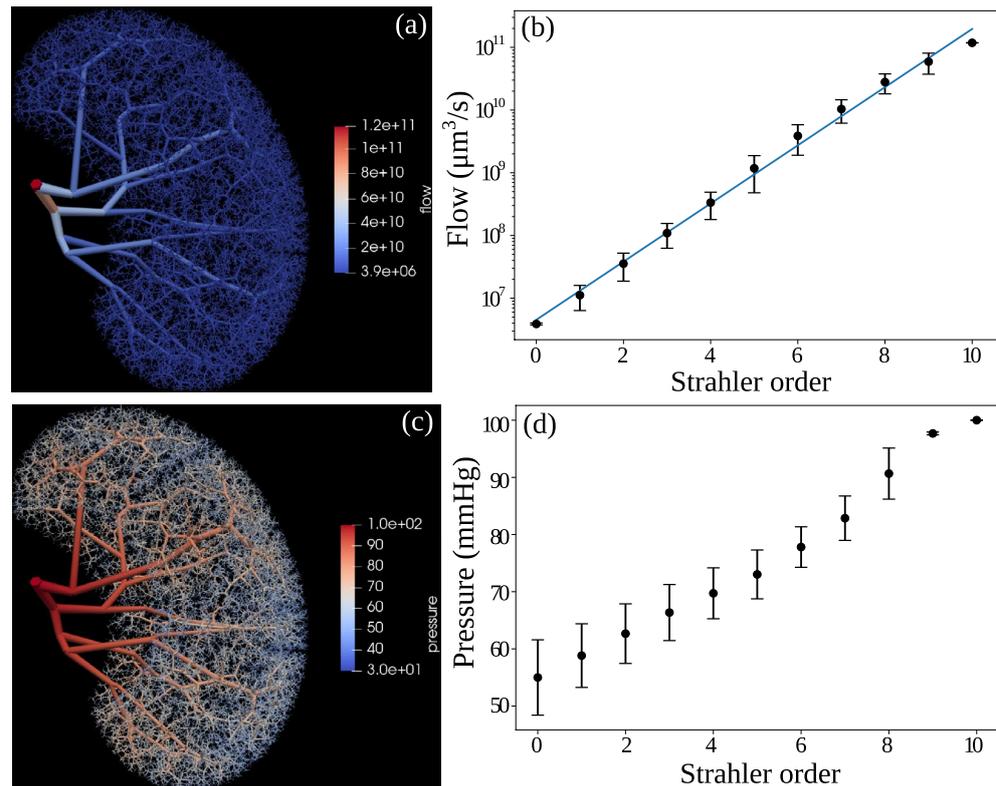


Figure 3. Physiological features of the generated renal vascular network of the rat kidney. (a) Visualization of blood flow distributed across the network and (b) its distribution vs Strahler order (in log scale) : from $1.2 \times 10^{11} \mu\text{m}^3/\text{s}$ (7 ml/min) in renal artery to $4 \times 10^6 \mu\text{m}^3/\text{s}$ (4 nl/s) in afferent arterioles (AA). (c) Visualization of pressure distributed across the network and (d) its distribution (at the outlet of each vessel) vs Strahler order, ensuring smooth pressure drop from 100 mmHg at the inlet to a minimum of 30 mmHg at the end of afferent arterioles (AA). In the left panels (a,c), each vessel is visualized by a separate cylinder with a thickness corresponding to its radius, and color coded by the flow (a) or pressure (c), visualized by ParaView²⁸.

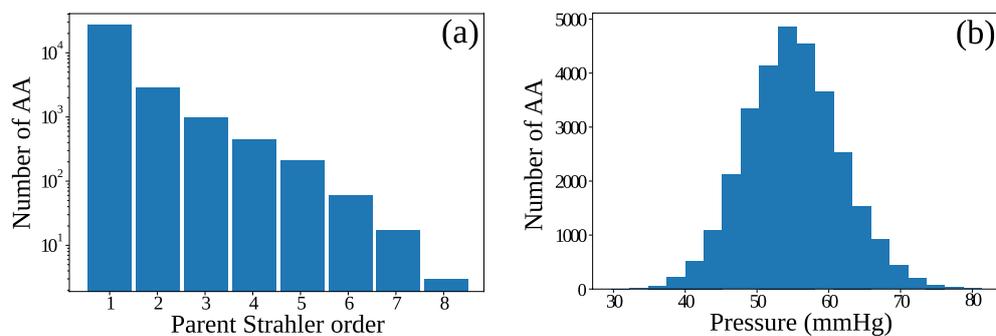


Figure 4. Morphometric and physiological features of afferent arterioles (AA) in the generated renal vascular network. (a) Number of afferent arterioles (in log scale) branching from the parent vessel of a given Strahler order in the generated tree. (b) Histogram of the pressure distribution among afferent arterioles.

contiguous afferent arterioles^{13,16} need to be modeled to include the fine-tuning of the pressures and radii of the afferent arterioles.

Discussion

We propose a hybrid framework for the reconstruction of the arterial vascular network of a rat kidney. The framework generates a full-scale 3-D vascular tree model based on a modified Global Constructive Optimization algorithm while taking image-based priors from a subject-specific scan. The hybrid method preserves subject-specific information by taking both the kidney's shape and the main artery segmentation from micro-CT images of a real rat into the initialization step.

The reconstructed vascular tree shows good morphometric agreement with anatomical data from a real rat kidney⁷. Furthermore, the calculated pressure distribution throughout the vascular tree is in good agreement with values found experimentally²⁹. Whereas the overall topology is in good agreement with the experimental observations, there are some deviations between the values, especially the vessel radii. In contrast to the topology of the vascular tree, vessel radius is a dynamic variable that is determined by the local conditions in the tissue. Thus, vessel radius may vary considerably from time to time in a given vessel, and the measurement process itself, e.g., the injection of casting material, may in itself cause changes in the radius. Nonetheless, the largest deviation was in the root vessel, which does not play a significant role in renal hemodynamics. Its resistance is negligible compared to the smaller vessels in the tree and, therefore, of little significance for total renal blood flow.

It could be argued that vessel radius should be part of the cost function, and thus be optimized in the construction of the tree. However, we found that incorporating the radius in the cost function defined in Eq. (7), in fact, deteriorated the outcome of the optimization process, see “[Physiologically based cost functions](#)”. One reason could be that the cost function is local, whereas the flow is determined not only by local factors but also by the global physiological demands on the organ. This suggests that an optimal procedure could be to construct a realistic vascular tree using an algorithm like the one proposed in the present paper, followed by an optimization of the vessel radii based on a global physiological target. In the kidney, this could be the resulting glomerular filtration rate or a similar global measure of renal function.

Importantly, the reconstructed structure is not a simple bifurcating tree, a structure which previously has been shown to be insufficient for supplying blood to the glomeruli at a sufficient pressure¹³. Instead, in agreement with previous work on the structure of the renal vascular tree¹, afferent arterioles arise not only from terminal arteries (Strahler order 1), but also from all arteries of higher order, except for the largest arteries (Strahler order 9 and 10) (cf. Fig. 4a). Therefore, the proposed method can generate both morphometrically correct and physiologically feasible vascular trees while respecting the prior information from the subject scan.

Modifications of subprocesses can easily be integrated into our framework. For example, although the current image prior requires a semi-automated segmentation of the main arteries, one can replace it with state-of-the-art deep learning models, e.g., UNet³⁶ to do the auto segmentation if one has accurate training data. Similarly, if renal cortex segmentation is feasible in a different scanning setting, one could skip the cortex extraction (Eq. 10) step and directly do the sampling of terminal nodes over the segmented cortex.

Future work will be to apply the reconstructed network in the areas described in the previous section, e.g., to model the active regulation of pressure and flow in our generated arterial tree^{13,16}. We have focused on the rat kidney since it is the only one for which detailed data on the vascular tree are available for validations^{7,29}. Nonetheless, we expect that our model can generalize to the human kidney as well, given similar micro-CT scans with similar resolution. The renal medulla imposes intrinsic constraints on artery growth, which must be addressed in the CCO or GCO process. However, since our network starts with a pre-built tree, the vessels will never pass these regions as long as the vessel segmentation is accurate to a certain extent, giving piece-wise convex regions in the initialization process.

With a full-scale tree structure, it is the ultimate goal to model pathological changes in the kidney. For example, at the structural level, it could be reducing the number of terminal vessels to mimic loss of nephrons, or modifying the radii of certain vessels while simulating the resulting changes of pressures and flow to mimic renal artery stenosis and atherosclerotic changes in the kidney. However, the pathological changes may be functional, e.g., a reduced glomerular filtration or tubular reabsorption rate. To include these types of pathology, it is necessary to expand the model with models of the nephrons. This will be part of the next step because, in addition to the structure of the vascular network, it will require an additional model of functioning nephrons attached to each afferent arteriole together with the relevant regulatory systems, e.g., the myogenic mechanism and tubuloglomerular feedback³⁷.

Another future direction is to create a synthetic renal vessel dataset by generating the ground truth segmentation labels corresponding to the generated tree. To create such an image dataset, we need to remap the reconstructed vascular tree back to a smooth surface mesh or binary label map. The details of such a process and an example of an image-label pair (Fig. S2) are given in the Supplementary material. We will test whether these artificially generated vessel images can be used to pre-train a deep-learning-based segmentation network for transfer learning or to train a Generative Adversarial Network (GAN) for domain transfer. As an example, Menten et al.² recently applied CCO to synthesize retinal vascular plexuses and generated corresponding Optical Coherence Tomography Angiography (OCTA) images by emulating the OCTA acquisition process. They showed that these simulated data can successfully pre-train a retinal vessel segmentation network to segment real OCTA retinal images.

Upon finishing our work, we also notice that each subtree inside each piece-wise convex region after the initialization step of our GCO method (cf. each colored subtree in Fig. 1h) is independent of each other. Specifically, the leaf nodes that are connected to a certain node of the pre-built tree in the initialization step will always belong to the successors of that node. Figure 5 shows an example of the initialization and the result of one subtree. Here, the resulting subtree will stay inside the region defined in the initialization step with the same group of leaves and is independent of other subtrees. Although they all belong to a larger tree with a single root vessel, each subtree can be optimized independently in parallel before being merged together in the end. This parallelization has not been implemented explicitly, which should also be a future direction to speed up the whole computation.

The subtrees created by the initialization process are reminiscent of the vascular dominant regions found in the kidney^{38,39}. When planning surgical removal of part of the kidney, surgeons identify the first few arterial branches from the renal artery to estimate the subregion supplied by each of the branches^{38,39}. Each subregion is assumed to only get blood supply from the closest branch. These independently supplied regions resemble the piece-wise convex regions in our initialization step for the growth of the subtrees (Fig. 1h). Similar to our

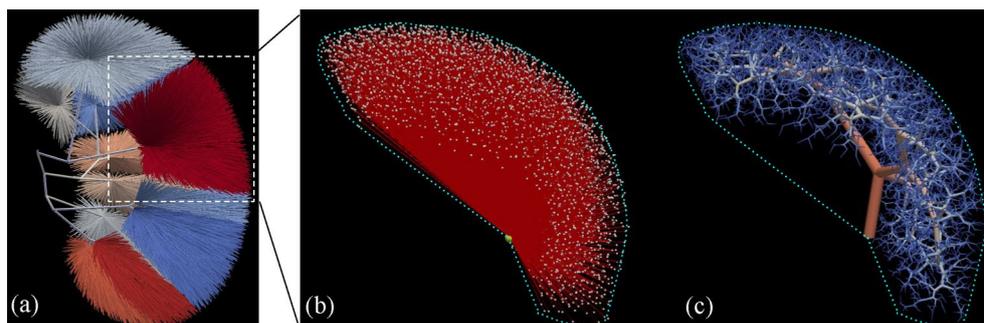


Figure 5. Illustration of the initialization and result of a subtree, visualized by ParaView²⁸. **(a)** The initialization of the whole tree from Fig. 1h. **(b)** The initialization of one subtree from (a), zoomed in and rotated. **(c)** The result of the subtree, which can only grow inside the outlined region defined in (b), thus it will satisfy the convex constraint, and will not penetrate the renal medulla or grow outside the kidney structure.

pre-processed arterial tree (Fig. 1g), these procedures only identify the first few (around 3) branches from the renal artery^{38,39}. Otherwise, the vascular dominant regions become less well-defined.

The pair-wise coupling of the arterial and venous systems is not trivial to integrate, when in the future extending the framework to cover both arteries and veins. Currently, it is only possible to independently generate two individual trees. This does not capture the pair-wise coupling of arteries and veins, nor does it avoid the early intersection of the two trees. Kretowski et al. applied a CCO-based approach to create complementary hepatic arterial and venous trees⁴⁰. They detect and avoid intersections between the two trees explicitly during the growing process by adjusting the radius of each vessel at the expense of violating Murray's law (Eq. (9)). This process is time-consuming and needs modification over the GCO process, where vessels are optimized on a larger scale.

Finally, we can generate more realistic trees by utilizing more vessels from the segmentation of the scans, or by having a better estimate of the renal cortex region in the initialization step. Currently, we are deliberately removing a large portion of the extracted centerline to only preserve the main arteries, since the other parts are prone to noise and difficult to detect by later auto-segmentation methods. If we have better segmentations, these small segments could also be used to guide the reconstruction of subject-specific vascular networks. On the other hand, one could also experiment with how small a portion of a pre-built tree we need in the initialization step of GCO before it will violate anatomical constraints, e.g., pass through the renal pyramid or outside the kidney structure without the piece-wise convex premise. Another potential future direction involves integrating large vessels in a totally different manner. Instead of a pre-built tree, the segmentation could also be incorporated in the cost function as gravitation to guide the whole process by “pulling” the intermediate nodes close to their positions. However, it is non-trivial to add such pull-force to Eq. (7) while balancing the other constraints. It will probably involve lots of testing of the parameters to find such a balance.

Methods

The original CCO algorithm works by iteratively adding a new edge (vascular segment). After each addition, the newly created bifurcation is locally remodeled and all tree radii are adjusted geometrically. Such remodeling is clearly inefficient when the vessel structure becomes large like in the kidney. In contrast, we adopt the alternative GCO algorithm as our backbone model. This method overcomes the problems of the CCO by starting with a fully connected tree, where the leaves are usually either defined on a regular grid or randomly positioned within the organ hull⁶. In our case, the leaves are sampled using Poisson disk sampling²⁶ from the estimated renal cortex (Fig. 1c) as detailed in “[The leaf node sampling](#)”. It performs a multi-scale optimization to find an optimal tree for all leaf nodes simultaneously and introduces a global pruning operation after each iteration to produce a new tree with better global branching structures.

In this section, we start by stating the assumptions and objectives of the constructive algorithm (GCO). We then explain the underlying cost function to be optimized by GCO and the general process of GCO, including the modifications we have made. We then present in detail how the image priors are integrated into our hybrid framework and end by introducing the utilized scan information and implementation details.

We do acknowledge that there are several approximations involved in the whole process, e.g., all the assumptions defined in “[Assumptions and objectives](#)”; the approximation algorithm involved in the splitting process in “[Global Constructive Optimization algorithm](#)”, as well as the cortex approximation in “[The image priors](#)”. We will explain the underlying rationale or why they are inevitable when introducing these approximations.

Assumptions and objectives. Several assumptions have to be made for the whole process of reconstructing vascular trees. The first assumption forms the basis of the mathematical modeling of any vascular tree and has been adopted in most of the angiogenesis-based methods, e.g., on liver, heart, and brain^{4,6,23,24}. The second assumption comes from structural and functional properties of the kidney that glomeruli and afferent arterioles of all nephrons are located in the cortex region^{41,42}. It is the rationale for a Poisson disk sampling of terminal nodes (glomerulus), which will be discussed in “[The leaf node sampling](#)”. The other assumptions relate to the

hemodynamics and are necessary to satisfy Poiseuille’s equation in Eq. (5), which is an integral part of the cost function Eq. (7).

- A vascular tree is modeled as a collection of connected, straight cylindrical tubes, indicating constant radius and no curvature between branch points.
- All the renal arteries end in the renal cortex with a certain perfusion territory.
- Blood is incompressible and Newtonian, and blood flow is laminar.
- Pressure drop due to branching is negligible.
- Flows are equally distributed among each terminal vessel.

Given such assumptions, a vascular tree is modeled by a directed acyclic graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ where \mathcal{V} is a set of nodes at the endpoints of each vessel centerline with node features being its coordinates in Euclidean space, and \mathcal{E} is a set of directed edges which form a connected tree structure. Each edge represents a single vessel segment as a cylinder with its radius and flow as the edge feature. Note that length is not modeled as an edge feature but rather derived from the Euclidean distance between the two end nodes of each edge. The goal is to find a tree that minimizes the system’s overall cost function while fulfilling the constraints. Specifically, given the position of a single root node s , and n leaf nodes l_i , the goal is to find a tree $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ that contains s and $l_i \in \mathcal{V}$ with minimum cost defined in the next subsection and fulfills constraints by introducing new intermediate nodes $v_i \notin \{s, l_1, \dots, l_n\}$ and connections (edges) \mathcal{E} .

In our work, we propose a novel way to integrate image priors into the initialization of GCO, so that the input is no longer a single root with n leaves, but a pre-built tree $\mathcal{G}_0 \equiv (\mathcal{V}_0, \mathcal{E}_0)$ with $s \in \mathcal{V}_0$ that already covers the main arteries.

Physiologically based cost functions. Our reconstruction method generates subject-specific arterial vascular networks $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ under the optimality assumption that the network structures will maintain adequate blood perfusion with minimal total expense along all its edges, which can be approximated by the total sum of the local cost at each branching node v :

$$C(\mathcal{G}) \equiv \sum_v C_{\text{local}}(v) \equiv \sum_v \sum_{e \in \mathcal{B}_v} C(e) \tag{1}$$

where \mathcal{B}_v denotes the set of all the incident edges of node v .

A typical branching model is shown in Fig. 6. Note that the GCO model does not enforce bifurcation explicitly and can indeed model any number of branches > 2 . Still, branches other than bifurcations or trifurcations are rarely seen in the final result, because they usually incur a higher cost. In each branching model, an optimal branching point is positioned with respect to fixed neighboring edge radii and neighboring node positions to minimize the cost function. Following the work of Tekin and Shen et al.^{24,32}, we incorporate both the material cost (M_{loss}) and power cost (P_{loss}), resembling the biological infrastructure cost to build the vessel and the power dissipated during blood circulation, respectively. Therefore, $C_{\text{local}}(v)$ is a weighted combination of the two costs:

$$C_{\text{local}}(v) \equiv w_c M_{\text{loss}}(v) + w_p P_{\text{loss}}(v) \tag{2}$$

where w_c and w_p are the weight factors to balance the two costs. $M_{\text{loss}}(v)$, which expresses the amount of materials that constitute the blood in the vessels, is only dependent on the intravascular volume of the arterial tree. It is given by

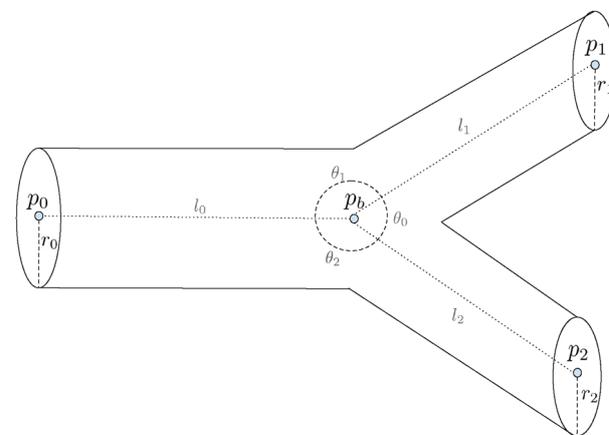


Figure 6. A typical vessel branching model. The branching vessels are uniquely defined by the locations of the three end nodes (p_0, p_1, p_2), the location of the bifurcation node (p_b), and the radii of the three incident edges (r_0, r_1, r_2). Length (l_s) and branching angles (θ_s) are not modeled explicitly but can be trivially derived.

$$M_{loss}(v) \equiv \sum_{e \in \mathcal{B}_v} M(e) \equiv \sum_{e \in \mathcal{B}_v} \pi r_e^2 l_e. \tag{3}$$

Analogous to electric power, the power dissipated during blood circulation ($P_{loss}(v)$) is defined by the product of flow Q_e (analog to electric current) and pressure drop Δp_e (analog to potential difference),

$$P_{loss}(v) \equiv \sum_{e \in \mathcal{B}_v} Q_e \Delta p_e \tag{4}$$

where Hagen-Poiseuille’s law gives the pressure drop Δp_e necessary to overcome the resistance to flow due to the viscosity μ , of the blood in an individual blood vessel,

$$\Delta p_e = \frac{8\mu l_e Q_e}{\pi r_e^4}. \tag{5}$$

Combining Eqs. (4) and (5) gives

$$P_{loss}(v) \equiv \sum_{e \in \mathcal{B}_v} Q_e^2 \frac{8\mu l_e}{\pi r_e^4}. \tag{6}$$

The total cost at each branching v is a weighted combination of material cost Eq. (3) and power cost Eq. (6),

$$\begin{aligned} C_{local}(v) &\equiv w_c M_{loss}(v) + w_p P_{loss}(v) \\ &= w_c \left(\sum_{e \in \mathcal{B}_v} \pi r_e^2 l_e \right) + w_p \left(\sum_{e \in \mathcal{B}_v} Q_e^2 \frac{8\mu l_e}{\pi r_e^4} \right) \\ &= \sum_{e \in \mathcal{B}_v} \left(w_c \pi r_e^2 l_e + w_p Q_e^2 \frac{8\mu l_e}{\pi r_e^4} \right). \end{aligned} \tag{7}$$

Blood flow follows the simple zero-addition rule from Kirchhoff’s first law, from which we derive the relation,

$$Q_p = \sum_{e \in \mathcal{B}_v/p} Q_e \tag{8}$$

where p denotes the parent edge in \mathcal{B}_v of node v . Note that radii are not optimized but assumed to follow Murray’s law at the branch points^{5,20},

$$r_p^3 = \sum_{e \in \mathcal{B}_v/p} r_e^3. \tag{9}$$

Although we recognize that Murray’s law is only an approximation, we found that optimizing radii, e.g., by integrating an equality constraint into the optimization process as proposed by²⁴ deteriorates the result by producing abrupt changes in the radii at branch points, and resulting in a final tree with fewer than the 11 Strahler orders found experimentally⁷.

In our experiment, we set the weight factors $w_c = 5 \times 10^4 \text{ J s}^{-1} \text{ m}^{-3} = 5 \times 10^{-8} \text{ N } \mu\text{m}^{-2} \text{ s}^{-1}$, and $w_p = 1$, as we have found that they result in the same scale of the two cost terms. We also adopt constant viscosity $\mu = 3.6 \times 10^{-3} \text{ Pa s} = 3.6 \times 10^{-15} \text{ N s } \mu\text{m}^{-2}$, and inlet flow $Q_0 = 7 \text{ ml/min} = 1.167 \times 10^{11} \text{ m}^3 \text{ s}^{-1}$ with values from the literature^{43,44}. Assuming equal flow distribution over terminal vessels, flow at afferent arterioles can be calculated as $Q_t = \frac{Q_0}{N}$ where N is the number of afferent arterioles. In our case, $Q_t = \frac{1.167 \times 10^{11} \text{ m}^3 \text{ s}^{-1}}{3 \times 10^4} = 3.89 \times 10^6 \text{ } \mu\text{m}^3 \text{ s}^{-1} = 3.89 \text{ nl s}^{-1}$, complying with the literature where $Q_t \approx 4 \text{ nl s}^{-1}$ ⁴⁵. Note that the physics units and voxel size need to be consistent with each other to ensure that M_{loss} and P_{loss} are on the same scale. The above units give both M_{loss} and P_{loss} in the scale of $\text{N } \mu\text{m}^{-2} \text{ s}^{-1}$ (μW).

The combination of the two losses is in correspondence with the derivation of Murray’s law^{5,20}, and is based on a compromise between the power required to drive flow through the vessel (P_{loss}) and the rate of expenditure of metabolic energy required to maintain the volume of blood filling the vessel (M_{loss}). Although the original CCO and GCO methods only consider M_{loss} , P_{loss} is vital for a vascular system that requires efficient flow³² such as in the kidney. Experimentally we have also found it necessary to integrate P_{loss} into the loss function, since using M_{loss} alone (or a too large weight for M_{loss}) does not generate anatomically correct tree topologies, e.g., giving only 9 Strahler orders instead of 11 as reported in⁷. Note that the blood supply cost, defined in⁴⁶ to produce evenly dispersed terminal nodes, is not needed in our model, since the terminal nodes have already been sampled by Poisson disk sampling²⁶, which maintains a minimum distance among them, as discussed in “The leaf node sampling”.

Strahler ordering. The Strahler ordering method is a common method for labeling trees with a hierarchical structure, e.g., a vascular tree^{7,8,47}. It begins at the top of the tree by labeling all the leaves (the afferent arterioles in this case) as having Strahler order 0. A 0 order vessel has no vessels branching from it. Following the tree upstream, the order of the parent vessel (edge) increases by one order to $j + 1$, if two or more of the daughter vessels are of order j , where j is the highest order among the daughter vessels. Otherwise, the parent vessel takes the highest order (j) among its daughter vessels. The process is continued until it reaches the root of the tree.

Note that this ordering method is not used in the reconstruction algorithm itself, but is used for the quantitative comparison of the reconstructed tree with the measurements of a renal vascular tree reported in⁷.

Global Constructive Optimization algorithm. The Global Constructive Optimization (GCO) algorithm includes the following steps, which are iterated multiple times before convergence except for the first initialization step. Please refer to⁶ for a more detailed explanation of the process.

1. *Initialization.* In the original GCO initialization, each sampled terminal node is connected to a single user-defined root node, thus completely ignoring subject-specific information. In our hybrid framework, the sampled terminal nodes are connected to the main arteries derived from the patient's scan. Details of the sampling process and main arteries retrieval are explained in “[The image priors](#)”.
2. *Relaxation.* The relaxation process finds the best location for each branching node through optimization by minimizing the overall cost function defined in “[Physiologically based cost functions](#)”.
3. *Merging.* Merging involves contracting the edge between two neighboring nodes. It is applied when the ratio between the shortest incident edge of a node and the second incident edge is within a threshold, which usually happens when relaxation places a node at the same location as one of its neighboring nodes.
4. *Splitting.* Splitting is done whenever creating a new intermediate node and reconnecting a subset of the original child neighbors $S \in \mathcal{B}_v/p$ introduces a lower cost. Usually, this condition is fulfilled at a node with too many edges, indicating that bifurcation is implicitly imposed on the modeling. However, finding the optimal subset is in $\mathcal{O}(n!)$ thus NP-hard. Instead, an approximation algorithm is applied by first finding a subset $S_1 \in S$ with two edges that introduce the lowest cost. A new edge is then iteratively added to S_n from $S - S_{n-1}$ if it introduces a lower cost. This approximation has a complexity of $\mathcal{O}(n^2)$ thus much more efficient. It is worth mentioning that this operation is still more computationally heavy than the actual optimization step in relaxation.
5. *Pruning.* A pruned tree $\mathcal{G}_l \equiv (\mathcal{V}_l, \mathcal{E}_l)$ is created from \mathcal{G} by removing all edges deeper than some threshold. This process will only keep the large branches generated from each iteration and produce a new tree with a better global branching structure in the next iteration. Here we start by keeping the first two branches of each subtree while keeping one more branch after every two iterations. All the leaf nodes that are removed in this operation are reconnected to the nearest node in the pruned tree. The modification we make here is that each leaf node can only be reconnected to the subtree that it belongs to in the initialization step.

The image priors. In general, our proposed hybrid way of utilizing image-based priors involves two segmentation maps from the kidney scans, to begin with, as shown in Fig. 1: segmentation of large arteries Y_a (Fig. 1e), and segmentation of the whole kidney structure Y_w (Fig. 1a). The segmentation of large arteries is obtained using a semi-automated approach⁴⁸. Whole kidney structure segmentation, however, is obtained by simple thresholding, since the ex-vivo micro-CT scan makes it easy to separate the kidney from the background. These two segmentation maps are used in the following two tasks for the initialization of the whole GCO process. Figure S1 in Supplementary shows the flowchart of the process.

The leaf node sampling. The segmentation of the whole kidney structure Y_w is used to sample terminal nodes where the arteries end (Fig. 1d). Since the arteries end in the cortex region rather than only on the surface, to mimic the anatomical rules, several more steps are necessary.

Cortex approximation via erosion. The renal arteries end in the renal cortex, which requires a cortex segmentation to sample from. Since the cortex is not visible from our micro-CT scan, it is approximated by a certain distance ($R_1 \approx 2 \text{ mm}$) away from the surface by assuming equal thickness across the kidney. The thickness of the rat kidney cortex depends on the age, sex, and size of the rat. A value of 2 mm is typical for a 12–18 week-old rat^{49,50}. This is the age group typically used experimentally. This process can be easily obtained by the subtraction of a mathematical erosion (\cdot) applied to Y_w , as shown in the yellow regions in Fig. 1b.

Inner region removal. To avoid sampling terminal artery nodes around the renal artery, all the regions near a certain distance to the root node v_r are removed. This is accomplished by imposing a ball centered at v_r ($R_2 \approx 5.65 \text{ mm}$), as shown in the green regions in Fig. 1b. The size of the region is based on our scans of the kidney and provides an automatic means to avoid sampling afferent arterioles near the hilus. Specifically, R_2 has to be large enough to cover the hilar region that belongs to the segmentation Y_w but is not part of the cortex. We note that it also removes parts of the kidney cortex near the hilus. Despite this, we find it an appropriate approximation for the cortex geometry, especially as the results are not critically dependent on this parameter. In summary, cortex segmentation is a set of points

$$Y_c \equiv \{ \mathbf{x} \mid \mathbf{x} \in Y_w - (Y_w \cdot R_1) \wedge \|\mathbf{x} - \mathbf{v}_r\|_2 > R_2 \} \quad (10)$$

of which the surface mesh is visualized in Fig. 1c.

The Poisson disk sampling. Vessels inside an organ follow an anatomical structure that the leaf nodes should cover the entire perfusion territory while avoiding being too close to each other to prevent competition or overlap between branches⁴⁶. This is important as one of the main purposes of the vascular tree is to supply blood to all the tissues of an organ⁴⁶. With a fixed perfusion territory, i.e., the extracted cortex Y_c from the previous step and a fixed number of terminal nodes, evenly spreading the terminal nodes inside the cortex volume is the most

straightforward way to mimic this anatomical property. Poisson disk sampling²⁶ maintains a minimum distance between sampled points by sampling from the spherical annulus of existing points and rejecting points that are too close to each other, which resembles such an anatomical rule very much. In the present model, the minimum distance value can be approximated from the cortex volume and the number of points we would like to sample from. We follow the work of Nordesletten⁷ by sampling 30K terminal vessels (number of arteries with Strahler order 0), which results in a minimum distance of around 270 μm . Note that this is the distance between the distal end of afferent arterioles, and not the gap between the glomeruli that originates from the terminal nodes. Monte Carlo sampling is also adopted to do Poisson disk sampling over the whole cubic volume before filtering out the points from non-cortex regions.

Large artery extraction. To integrate the large artery segmentation Y_a into the GCO process, vessels need to be modeled by a graph with nodes along the centerline and edges with connectivity information. Therefore, we start by extracting the centerline from Y_a using the Skeletonization method proposed by Bærentzen et al.⁵¹. Instead of a binary image with width 1 at each local foreground voxel, the algorithm outputs a graph data structure $C(Y_a) = \mathcal{G}(\mathcal{V}, \mathcal{E})$ suitable for our needs. However, extracting a tree structure via skeletonization of segmentations is difficult. Intense manual work must be involved afterward, even with accurate segmentation label maps. Therefore, when we have very coarse segmentation Y_a without clean resolution, only the first few branches from the root are trustworthy, while veins can be falsely segmented as arteries in the deeper branches. Moreover, the extracted centerline (Fig. 1f) is an undirected graph and may contain loops, which cannot be directly used. Several preprocessing operations are necessary before initialization.

Minimum spanning tree. This first operation removes potential loops by creating a subset of the edges with the minimum total edge weight from $\mathcal{G}(\mathcal{V}, \mathcal{E})$ that connects all the vertices without any cycles. Here the weight of each edge is the negative of its radius derived from a Euclidean distance transform over the derived centerline to the segmentation. This operation will remove the thinnest edge to break any loop. After this step, the acyclic graph can be converted to a tree (directed acyclic graph) with a simple depth-first-search.

Intermediate nodes removal. As stated in the assumption from “Assumptions and objectives” that each vessel is modeled by a straight cylindrical tube, any intermediate nodes along each single vessel will have to be removed. This will, of course, introduce artifacts to the length computation, but is assumed to be negligible within a reasonable curvature.

Degree pruning. For each node with more than 4 branches, we only keep a maximum of 4 longest paths, since branching into more than 4 children is not realistic. Specifically, Marsh et al.⁹ found at most 4 branchings, while Nordesletten et al.⁷ modeled up to trifurcations.

Depth pruning. For each node, we compute its cumulative distance to the root along the tree and only keep nodes up to a certain distance, which we set around 10 K μm (450 voxels with a 22.6 μm voxel size). Such threshold is experimentally determined to keep only the first few branches which are noise-free. The rationale is that even though some thin vessels far away from the root are visible from the current segmentation, only large vessel segmentation is trustworthy. Especially if we would like to further adapt deep learning for automatic vessel segmentation, we cannot assume the model to be able to detail the thin vessels.

Connected component decomposition. The two pruning operations may introduce smaller disconnected trees, we thus only keep the largest tree.

Final GCO initialization. For the initialization of GCO (Fig. 1h), all the sampled terminal nodes (Fig. 1d) are connected to the nearest ending node along the extracted and pre-processed large artery centerline graph $\mathcal{G}'(\mathcal{V}, \mathcal{E})$ (Fig. 1g). The radii associated with the terminal vessels are sampled from a Gaussian distribution $r_0 \sim \mathcal{N}(10.08, 0.14)$ derived from literature⁷, while radii of other vessels are derived from the radii of terminal vessels by Murray’s law (Eq. (9)).

Besides retaining subject-specific information from image priors, the connection to the pre-built tree also makes the complex structure piece-wise convex, making the later constructive algorithms applicable here. Specifically, the connection between any terminal node to the pre-built tree should not enter or cross the renal pyramid, which is hard to satisfy when the pre-built tree is only a single root node.

The pruning operations in the previous subsection remove a large portion of the deep branches, only preserving the main arteries. This is necessary due to the noisy input. Nonetheless, the remaining large branches are enough to satisfy the piece-wise convex constraint in the initialization step. Specifically, the connections between the sampled terminal nodes to the nearest node in the large artery centerline graph $\mathcal{G}'(\mathcal{V}, \mathcal{E})$ naturally avoid passing through the renal pyramid or going outside the kidney structure, as shown in Fig. 5. The GCO process afterward will remain in each convex shape created in the initialization step because moving outside will always enforce a larger cost as defined in “Physiologically based cost functions”.

Ex vivo micro-CT imaging dataset. The kidney cast was prepared as described in⁴⁸ in agreement with approved protocols (approval granted from the Danish Animal Experiments Inspectorate under the Ministry of Environment and Food, Denmark). The rat kidney was ex vivo scanned in a ZEISS XRadia 410 Versa μCT scanner (Carl Zeiss Microscopy GmbH, Jena, Germany) at the following settings: isotropic voxel size 22.6 μm , 50

kV tube voltage, 0.2 mA current, appertaining LE3 filter, 360° scan around the vertical axis with 3201 different projections (0.112° rotation steps)⁴⁸. The raw scan has a dimension of 1000 × 1024 × 1014 voxels. To ease the computational overhead, the scan is auto-cropped to 955 × 508 × 626 by an intersected bounding cube of the largest component from simple Otsu's thresholding over maximum intensity projections to three dimensions.

Implementation details. Shen et al.²⁴ and Keelan et al.⁴⁶ proposed to optimize the cost function using Simulated Annealing, which is a metaheuristic to approximate the global optimum of a given function. Because of its non-gradient-based nature, this method is usually preferable for problems where gradients are hard to compute. However, we note that the cost function defined in Eq. (7) is quite differentiable, meaning its gradient can be easily computed analytically, giving

$$\begin{aligned}\nabla C_{\text{local}}(v) &= \nabla \left(\sum_{e \in \mathcal{B}_v} \left(w_c \pi r_e^2 l_e + w_p Q_e^2 \frac{8\mu l_e}{\pi r_e^4} \right) \right) \\ &= \sum_{e \in \mathcal{B}_v} \left(w_c \pi r_e^2 + w_p Q_e^2 \frac{8\mu}{\pi r_e^4} \right) \nabla \|v - \mathbf{n}_{v,e}\|_2 \\ &= \sum_{e \in \mathcal{B}_v} \left(w_c \pi r_e^2 + w_p Q_e^2 \frac{8\mu}{\pi r_e^4} \right) \frac{v - \mathbf{n}_{v,e}}{\|v - \mathbf{n}_{v,e}\|_2}\end{aligned}\quad (11)$$

where $\mathbf{n}_{v,e}$ denotes the position in \mathcal{R}^3 of the neighboring node of v along edge e . Therefore, we apply the standard Broyden–Fletcher–Goldfarb–Shanno (BFGS) method which proves to perform as well and is much faster. Figure S3 in Supplementary shows the convergence plot of the GCO process.

All the backbones are pure NumPy and SciPy-based computation, with graph representation using NetworkX⁵². The computations were conducted in Ubuntu 22.04 with an Intel Core i7-8700 processor at 3.20 GHz and 24 GB RAM. Currently, there is no GPU acceleration. In fact, the optimization process in relaxation is not the bottleneck. As discussed in “Global Constructive Optimization algorithm”, splitting is usually the computational bottleneck and dominates the time complexity, especially in the first few iterations where there are a small number of intermediate nodes each with a large number of neighbors. Moreover, since each branching has to be optimized individually and consecutively, switching to PyTorch with GPU acceleration will not help. The process of integrating image priors, such as centerline extraction and Poisson disk sampling, takes approximately 1 h, while the GCO process after initialization takes approximately 10 h to reach convergence.

Ethics approval

The experiments were conducted in agreement with approved protocols (approval granted from the Danish Animal Experiments Inspectorate under the Ministry of Environment and Food, Denmark). All procedures agreed with the ethical standard of the university, which meets that of the EU Directive 2010/63/EU for animal experiments.

Data availability

Raw data and image processing algorithms can be exchanged through a collaboration agreement to Carsten Gundlach (cagu@fysik.dtu.dk). Processed data and analysis algorithms can be made available upon request to Peidi Xu (peidi@di.ku.dk).

Code availability

The source code for the paper is available at <https://github.com/diku-dk/RenalArterialRecon> under Apache License 2.0 open-source license.

Received: 28 April 2023; Accepted: 6 May 2023

Published online: 09 May 2023

References

1. Marsh, D. J., Postnov, D. D., Sosnovtseva, O. V. & Holstein-Rathlou, N.-H. The nephron-arterial network and its interactions. *Am. J. Physiol. Renal Physiol.* **316**(5), 769–784 (2019).
2. Menten, M. J., Paetzold, J. C., Dima, A., Menze, B. H., Knier, B. & Rueckert, D. Physiology-based simulation of the retinal vasculature enables annotation-free segmentation of oct angiographs. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* 330–340 (Springer, 2022).
3. Whitehead, J. F., Nikolau, E. P., Periyasamy, S., Torres, L. A., Laeseke, P. F., Speidel, M. A. & Wagner, M. G. Simulation of hepatic arteries and synthesis of 2d fluoroscopic images for interventional imaging studies. In *Medical Imaging 2020: Physics of Medical Imaging* Vol 11312 459–468 (SPIE, 2020).
4. Schreiner, W., Karch, R., Neumann, F. & Neumann, M. Constrained constructive optimization of arterial tree models. *Scal. Biol.* **145**, 65 (2000).
5. Murray, C. D. The physiological principle of minimum work: I the vascular system and the cost of blood volume. *Proc. Natl. Acad. Sci.* **12**(3), 207–214 (1926).
6. Georg, M., Preusser, T. & Hahn, H. K. Global constructive optimization of vascular systems (2010).
7. Nordsletten, D. A., Blackett, S., Bentley, M. D., Ritman, E. L. & Smith, N. P. Structural morphology of renal vasculature. *Am. J. Physiol. Heart Circ. Physiol.* **291**(1), 296–309 (2006).
8. Stahler, A. Hypsometric (area altitude) analysis of erosional topology. *GSA Bull.* **63**, 1117–1142 (1952).
9. Marsh, D. J. et al. Architecture of the rat nephron-arterial network: Analysis with micro-computed tomography. *Am. J. Physiol. Renal Physiol.* **313**, 351–360 (2017).

10. More, R. H. & Duff, G. L. The renal arterial vasculature in man. *Am. J. Pathol.* **27**, 27–95 (1951).
11. Horacek, M. J., Earle, A. M. & Gilmore, J. P. The renal vascular system of the monkey: A gross anatomical description. *J. Anat.* **153**, 123–137 (1987).
12. Casellas, D. *et al.* Anatomic pairing of afferent arterioles and renin cell distribution in rat kidneys. *Am. J. Physiol.* **267**, 951–963 (1994).
13. Postnov, D. D. *et al.* Modeling of kidney hemodynamics: Probability-based topology of an arterial network. *PLoS Comput. Biol.* **12**(7), 1004922 (2016).
14. Baldelomar, E. J., Charlton, J. R., Beeman, S. C. & Bennett, K. M. Measuring rat kidney glomerular number and size in vivo with MRI. *Am. J. Physiol. Renal Physiol.* **314**, 399–406 (2018).
15. Letts, R. F. R. *et al.* Nephron morphometry in mice and rats using tomographic microscopy. *Am. J. Physiol. Renal Physiol.* **312**, 210–229 (2017).
16. Marsh, D. J. *et al.* Multinephron dynamics on the renal vascular network. *Am. J. Physiol. Renal Physiol.* **304**(1), 88–102 (2013).
17. He, J., Pan, C., Yang, C., Zhang, M., Wang, Y., Zhou, X., & Yu, Y. Learning hybrid representations for automatic 3d vessel centerline extraction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 24–34 (Springer, 2020).
18. Dorobaniu, A., Ogrea, V. & Brad, R. Coronary centerline extraction from cta using 3d-unet. *Future Internet* **13**(4), 101 (2021).
19. Schneider, M., Hirsch, S., Weber, B., Székely, G. & Menze, B. H. Joint 3-d vessel segmentation and centerline extraction using oblique hough forests with steerable filters. *Med. Image Anal.* **19**(1), 220–249 (2015).
20. Murray, C. D. The physiological principle of minimum work: II oxygen exchange in capillaries. *Proc. Natl. Acad. Sci.* **12**(5), 299–304 (1926).
21. Pfizner, J. Poiseuille and his law. *Anaesthesia* **31**(2), 273–275 (1976).
22. Cury, L., Maso Talou, G., Younes-Ibrahim, M. & Blanco, P. Parallel generation of extensive vascular networks with application to an archetypal human kidney model. *R. Soc. Open Sci.* **8**(12), 210973 (2021).
23. Jaquet, C. *et al.* Generation of patient-specific cardiac vascular networks: A hybrid image-based and synthetic geometric model. *IEEE Trans. Biomed. Eng.* **66**(4), 946–955 (2018).
24. Shen, J., Faruqi, A. H., Jiang, Y. & Maftoon, N. Mathematical reconstruction of patient-specific vascular networks based on clinical images and global optimization. *IEEE Access* **9**, 20648–20661 (2021).
25. Li, S. *et al.* Multiscale modeling of human cerebrovasculature: A hybrid approach using image-based geometry and a mathematical algorithm. *PLoS Comput. Biol.* **16**(6), 1007943 (2020).
26. Bridson, R. Fast Poisson disk sampling in arbitrary dimensions. *SIGGRAPH sketches* **10**(1), 1 (2007).
27. Kikinis, R., Pieper, S. D. & Vosburgh, K. G. 3d slicer: A platform for subject-specific image analysis, visualization, and clinical support. In *Intraoperative Imaging and Image-Guided Therapy 277–289* (Springer, 2014).
28. Ahrens, J., Geveci, B. & Law, C. Paraview: An end-user tool for large data visualization. *Visual. Handb.* **717**, 8 (2005).
29. Casellas, D. & Moore, L. C. Autoregulation of intravascular pressure in preglomerular juxtamedullary vessels. *Am. J. Physiol. Renal Physiol.* **264**(2), 315–321 (1993).
30. Oeltze, S. & Preim, B. Visualization of vasculature with convolution surfaces: Method, validation and evaluation. *IEEE Trans. Med. Imaging* **24**(4), 540–548 (2005).
31. Hahn, H. K., Preim, B., Selle, D. & Peitgen, H.-O. Visualization and interaction techniques for the exploration of vascular structures. In *Proceedings Visualization, 2001. VIS'01*, 395–578 (IEEE, 2001).
32. Tekin, E., Hunt, D., Newberry, M. G. & Savage, V. M. Do vascular networks branch optimally or randomly across spatial scales?. *PLoS Comput. Biol.* **12**(11), 1005223 (2016).
33. Zamir, M. On fractal properties of arterial trees. *J. Theor. Biol.* **197**(4), 517–526 (1999).
34. Zamir, M. *Vascular System of the Human Heart: Some Branching and Scaling Issues. Scaling in Biology* 129–143 (Oxford University Press, 2000).
35. Kassab, G. S. Network analysis of coronary circulation: I steady-state flow. In *Coronary Circulation* 309–362 (Springer, 2019).
36. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, 234–241 (Springer, 2015).
37. Holstein-Rathlou, N.-H. & Marsh, D. J. Renal blood flow regulation and arterial pressure fluctuations: A case study in nonlinear dynamics. *Physiol. Rev.* **74**, 637–681 (1994).
38. Isotani, S. *et al.* Feasibility and accuracy of computational robot-assisted partial nephrectomy planning by virtual partial nephrectomy analysis. *Int. J. Urol.* **22**(5), 439–446 (2015).
39. Wang, C. *et al.* Precise estimation of renal vascular dominant regions using spatially aware fully convolutional networks, tensor-cut and voronoi diagrams. *Comput. Med. Imaging Graph.* **77**, 101642 (2019).
40. Kretowski, M., Rolland, Y., Bézy-Wendling, J. & Coatrieux, J.-L. Physiologically based modeling of 3-d vascular networks and ct scan angiography. *IEEE Trans. Med. Imaging* **22**(2), 248–257 (2003).
41. Kriz, W., Kaissling, B. & Sa, G. *Structural and Functional Organization of the Kidney. The Kidney* 479–563 (Elsevier Inc, 2008).
42. Letts, R. F. *et al.* Nephron morphometry in mice and rats using tomographic microscopy. *Am. J. Physiol. Renal Physiol.* **312**(1), 210–229 (2017).
43. Stannov, S. U., Brasen, J. C., Salomonsson, M., Holstein-Rathlou, N.-H. & Sorensen, C. M. Interactions between renal vascular resistance and endothelium-derived hyperpolarization in hypertensive rats in vivo. *Physiol. Rep.* **7**(15), 14168 (2019).
44. Ronn, J., Jensen, E. P., Wewer Albrechtsen, N. J., Holst, J. J. & Sorensen, C. M. Glucagon-like peptide-1 acutely affects renal blood flow and urinary flow rate in spontaneously hypertensive rats despite significantly reduced renal expression of glp-1 receptors. *Physiol. Rep.* **5**(23), 13503 (2017).
45. Marsh, D. J., Sosnovtseva, O. V., Pavlov, A. N., Yip, K.-P. & Holstein-Rathlou, N.-H. Frequency encoding in renal blood flow regulation. *Am. J. Physiol. Regul. Integrat. Comp. Physiol.* **288**(5), 1160–1167 (2005).
46. Keelan, J., Chung, E. M. & Hague, J. P. Simulated annealing approach to vascular structure with application to the coronary arteries. *R. Soc. Open Sci.* **3**(2), 150431 (2016).
47. Horton, R. E. Erosional development of streams and their drainage basins; hydrophysical approach to quantitative morphology. *Geol. Soc. Am. Bull.* **56**(3), 275–370 (1945).
48. Andersen, S. B. *et al.* Evaluation of 2d super-resolution ultrasound imaging of the rat renal vasculature using ex vivo micro-computed tomography. *Sci. Rep.* **11**(1), 1–13 (2021).
49. Missbach-Guentner, J. *et al.* 3d virtual histology of murine kidneys-high resolution visualization of pathological alterations by micro computed tomography. *Sci. Rep.* **8**(1), 1407 (2018).
50. Pinnick, R. V. & Savin, V. J. Filtration by superficial and deep glomeruli of normovolemic and volume-depleted rats. *Am. J. Physiol. Renal Physiol.* **250**(1), 86–91 (1986).
51. Bærentzen, A. & Rotenberg, E. Skeletonization via local separators. *ACM Trans. Graphics* **40**(5), 1–18 (2021).
52. Hagberg, A. & Conway, D. Networkx: Network analysis with python. <https://networkx.github.io> (2020).

Acknowledgements

This research was supported by UCPH Data+ pool (UCPH Strategy 2023) and by ERC Synergy Grant SURE, project no. 854796. We acknowledge discussions with Anders BJORHOLM Dahl and support from the 3D Imaging

Centre at DTU and the Danish National Facility for Imaging with X-rays, DANFIX, funded by the Danish Agency for Science and Higher Education (Grant ID 5072-00030B). PX would also like to thank Dr. Sufang Han from Children's Hospital of Nanjing Medical University for the discussion about renal anatomical structures.

Author contributions

O.S. and S.D. initiated and coordinated the project. C.M.S. and N.H.H.R. provided the physiological background and participated in the study design. P.X. designed the modeling approach, conducted the experiment, and performed validation. K.E. supervised the validation of computational approaches. S.B.S. made the cast of the rat kidney for CT scanning. C.G. performed CT scanning and vessel segmentations. All authors participated in the discussion of the research and writing the paper.

Funding

Mentioned in acknowledgments.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-023-34739-y>.

Correspondence and requests for materials should be addressed to P.X.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023

Appendix B

Manuscript B

PAPER

Full-scale model and simulation of renal autoregulation

Peidi Xu,^{1,*} Sune Darkner,¹ Olga Sosnovtseva² and Niels-Henrik Holstein-Rathlour ²¹Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100, Copenhagen, Denmark and ²Department of Biomedical Sciences, University of Copenhagen, Blegdamsvej 3B, 2200, Copenhagen, Denmark

*Peidi Xu. peidi@di.ku.dk

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

The kidney's vascular network stands out because (i) the microcirculation not only supplies the tissues with oxygen and nutrients, but also supports glomerular filtration in each nephron, (ii) tubuloglomerular feedback responsible for blood flow autoregulation, and (iii) the topology of the renal arterial network influences signaling along the vascular wall mediating nephron-nephron interactions. We developed a full-scale nephron-vascular model that integrates a realistic vascular network topology with a single nephron model that includes glomerular filtration, tubular reabsorption together with autoregulation of the afferent arteriolar resistances. The regulated afferent arteriolar resistances stabilize blood flow in the full-scale renal vascular network. Our model evaluates the operating conditions of 30,000 nephrons in rat kidneys and the efficiency of autoregulation in pathological conditions. This is a significant step towards creating a virtual kidney.

Key words: kidney, autoregulation, computational simulation, renal vasculature, nephron model

Introduction

Blood vessels in the kidney form a demand-driven, balanced, and globally connected resource-distribution network. The kidney, like other organs, faces two concurrent challenges with regard to its blood flow. The first is that arterial blood pressure fluctuates due predominantly to the daily variation in physical activity. The blood pressure variations cause changes in the blood perfusion unrelated to the metabolic demands of the organ, and can, if large, cause damage to the organ microcirculation. Autoregulation is the term used for the intrinsic mechanisms in every organ that minimizes the impact of pressure variations. The second challenge is to adjust organ perfusion to match the metabolic demands. This requires feedback from the cells of the organ to the upstream vascular segments to adjust their hemodynamic resistances. By adjusting the hemodynamic resistances, the organ can achieve a balance between blood flow and the metabolic demands of the tissues.

In the kidney, the arterial network originates from the renal artery, branches, and forms a rooted tree that reaches all regions of the renal cortex. Terminal arteries at the top of the tree end in pairs or triples of afferent arterioles, where each afferent arteriole supplies a single glomerulus [1, 2].

The renal microcirculation has been the subject of many mathematical models during the last decades [3, 4, 5, 6]. The models have investigated the interplay between vascular factors, such as blood pressure and flow, and nephron function, such as glomerular filtration rate and tubular function. The kidney consists of a large number of functional units, nephrons, where the human kidney has around 1 to 1.2 million nephrons,

and the rat kidney has around 30,000 nephrons. Despite this, most models have focused only on the function of a single nephron, assuming that overall kidney function can be estimated by scaling the results from a model of a single nephron [6, 7, 8, 9, 10, 11]. However, results in the later years have made it clear that the individual nephrons do not function as independent units [12]. Rather, there is a significant interaction between neighboring nephrons. This is due to both the exchange of information along the vessels and the effects caused by local changes in the pressure and flow in the vascular network [12].

This has led to an increased interest in developing mathematical models of renal function that include several nephrons, which interact through a vascular network [13, 14, 15, 16]. The results of these studies point to a clear functional significance of the interactions between nephrons. However, the models have only included a relatively small number of interacting nephrons (less than 32) [13, 14, 15, 16].

One reason for the small number of nephrons in the existing models has been the lack of computational resources. But also lack of detailed information on the topology of the renal vascular network has been a limiting factor. Methods for imaging the renal vasculature cannot resolve the smallest vessels in the arterial network, and therefore cannot provide the information necessary to construct a realistic model of the renal vascular tree.

Motivated by the lack of a full-scale model of the renal vascular tree, Xu et al. [17] have developed a novel hybrid approach to reconstruct a full-scale renal arterial network through a physiologically based constructive optimization method that incorporates anatomical data extracted from a

scan of a rat kidney. The output is a full-scale renal arterial tree that starts from the renal artery and ends in the afferent arterioles, and it has been shown to be in good agreement with existing anatomical data [18].

In the present paper, we construct a computational model of renal autoregulation of blood flow based on the reconstructed full-scale model of the renal vascular tree from [17]. To the leaves of the tree (the afferent arterioles) are added mathematical models of nephrons, which include the main mechanisms of renal autoregulation, i.e., the tubuloglomerular feedback mechanism and the myogenic response. With around 30,000 nephrons, the model is the first full-scale model of blood flow regulation in a rat kidney, and it represents a first step towards the creation of a virtual kidney. By integrating a realistic vascular network topology with individual models of nephron autoregulatory mechanisms, the model allows for an exploration of adaptation and optimization within the nephron-vascular network and enables thorough simulation, a study of both normal and pathological conditions within the vascular network.

Materials and methods

The mathematical model includes four compartments: a glomerular model, a tubular model, a model of the afferent arteriole and the full-scale model of the renal arterial tree, as shown in Fig. 1. The nephron model (glomerular, tubular, and afferent arteriolar models) is largely based on a previously published model [6]. This section provides an overview of the essential components of the model, with detailed explanations available in the Appendix.

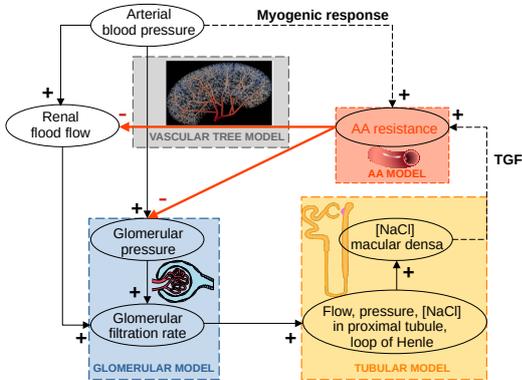


Fig. 1. An overview of the full-scale model of the nephron-vascular network and the simulation of renal autoregulation. The full model consists of four submodels: a glomerular model, a tubular model, an afferent arteriolar (AA) model, and a vascular tree model. The arterial blood pressure serves as the boundary condition in the model. The model integrates the mechanisms responsible for autoregulation renal blood flow: the myogenic response and the tubuloglomerular feedback (TGF).

Anatomical and physiological basis

The nephron consists of a glomerulus together with a tubule. The glomerulus is a network of capillaries enclosed by a cellular membrane, Bowman’s capsule. Blood flows from the

afferent arteriole into the glomerular capillaries and leaves the glomerulus through the efferent arteriole. As blood flows through the capillaries, fluid and small solutes are filtered through the capillary walls. Consequently, the filtrate contains all the small solutes dissolved in the plasma, whereas the plasma proteins are retained within the capillaries. The rate at which blood is filtered is known as the glomerular filtration rate (GFR) — an important measure of overall kidney function. GFR is the input to the tubule, which consists of several segments: the proximal tubule, the loop of Henle, the distal tubule, and the collecting duct. Most of the filtered water and NaCl are reabsorbed in the tubule, leaving waste products behind to be excreted with the urine. The GFR and, thereby the flow in the tubular system, is a function of the renal blood flow and the arterial pressure. If too low the kidney fails to eliminate waste products, and if too high there will be a loss of essential compounds like water and NaCl. Such effects are minimized through the operation of the tubuloglomerular feedback mechanism (TGF), which, together with the myogenic mechanism, accounts for renal autoregulation.

The anatomical foundation of tubuloglomerular feedback (TGF) lies in the return of the ascending limb of the loop of Henle to its own afferent arteriole [19]. At the point of contact, the macula densa, consisting of specialized epithelial cells along the wall of the ascending limb of the loop of Henle, serves as the sensing mechanism for TGF. Due to the dependence of NaCl reabsorption on tubular flow rate in the ascending limb of Henle’s loop, alterations in tubular flow rate, such as those induced by changes in arterial pressure, lead to variations in NaCl concentration at the macula densa. The macula densa detects this change and, through mechanisms that are still not fully understood, prompts adjustments in the radius of the afferent arteriole.

The myogenic mechanism operates within blood vessels and is thought to be an inherent characteristic of the smooth muscle cells comprising the vascular wall of the afferent arterioles [6]. The widely accepted explanation for the myogenic response is that it acts to minimize changes in the tension, T , in the vessel wall by adjusting vessel radius in response to changes in the transmural pressure [6]. The tension in the vessel wall is given by Laplace’s law:

$$P_a r_v = T \quad (1)$$

where P_a is the transmural pressure, and r_v is the vessel radius.

The mathematical model

Glomerular model

As blood flows from afferent arterioles to the glomerular through glomerular capillaries, fluid is filtered through membranes that are not permeable to proteins in the plasma. Protein is conserved within the capillaries, and the glomerular filtration rate in a single nephron ($SNGFR$) can be calculated from the product of the fractional change in protein concentration in the glomerular capillaries, $\frac{C_A}{C_E}$, and the plasma flow entering the capillaries, Q_A [6]:

$$SNGFR = \left(1 - \frac{C_A}{C_E}\right) Q_A \quad (2)$$

where C_A is the plasma protein concentration at the start, and C_E is the concentration at the end of the glomerular capillaries.

The filtration process that causes the change in the protein concentration, C , along the capillary is proportional to the local

pressure difference:

$$\frac{dC}{dx} = \frac{K_f}{LQ_A C_A} (C^2) [P_{GC} - P_T(0) - \pi(C)] \quad (3)$$

where x is fractional position along the glomerular capillary, K_f is the filtration coefficient, and L is the length of the capillary. P_{GC} , $P_T(0)$, and $\pi(C)$ are the capillary hydrostatic pressure, the tubular pressure, and the osmotic pressure due to plasma proteins, respectively. $\pi(C)$ is a nonlinear function that can be approximated by fitting a second-order polynomial to experimental data

$$\pi(C) = aC + bC^2 \quad (4)$$

where a and b are empirical constants.

The plasma flow entering the capillaries, Q_A , and the capillary hydrostatic pressure, P_{GC} , are retrieved from our full-scale renal arterial tree model, which will be discussed in the next section.

The boundary condition necessary to solve Eq. (3) is $C(0) = C_A$ at $x = 0$. From this, the concentration $C_E = C(L)$ at the end of the capillary ($x = L$) can be computed.

Tubular model

In the tubular model, the differential equations of the flow and pressure are based on the local conservation of mass. The NaCl concentration in the tubular fluid (C_S) changes because the tubule reabsorbs NaCl (J_S) and water (J_V). Since the flow of tubular fluid has a very low Reynold's number, it is assumed that the pressure drop within the tubule is given by Poiseuille's law. Thus, the total behavior is expressed by a system of three coupled differential equations consisting of pressure, P_T , flow, Q_T , and NaCl concentration, C_S :

$$\begin{cases} \frac{dP_T}{dz} = -\frac{8\mu}{\pi r^4} Q_T, & 0 < z < Z, t > 0 \\ \frac{dQ_T}{dz} = -J_V, & 0 < z < Z, t > 0 \\ \frac{dQ_T C_S}{dz} = Q_T \frac{dC_S}{dz} + C_S \frac{dQ_T}{dz} = -J_S, & 0 < z < Z, t > 0 \end{cases} \quad (5)$$

The tubule consists of two parts: the proximal tubule and the loop of Henle, where the latter can be divided into a descending and an ascending limb. The expression for J_V and J_S are defined differently based on the relevant tubular segment (see Appendix). The objective in solving the tubular model is to compute the NaCl concentration at the macula densa, $C_S(md)$, which is located in the ascending limb of the loop of Henle. The filtration rate from the glomerular model is equal to the tubular fluid flow at the start of the tubule, thus $SNGFR = Q_T(0)$, where $Q_T(0)$ serves as a boundary condition to Eq. (5), as shown in Fig. 1. Details of solving the system of three coupled differential equations are given in the appendix.

Afferent arteriole model

The tension in the wall of the afferent arteriole is given by Laplace's equation:

$$P_v r_v = T \quad (6)$$

where P_v is the transmural pressure, which is calculated as the average of the pressures at each end of the vessel segment, and r_v is the inner radius of the afferent arteriole.

The tension, T , is assumed to be a sum of two components: an elastic term, T_e , stemming from the passive stretch of the

elastic elements in the vessel wall, and an active tension, T_m , generated by the contraction of the vascular smooth muscle thus:

$$T = T_e + T_m \quad (7)$$

where

$$T_e = \sigma_e h(r_v) [\exp(k_e [r_v - r_e]) - 1] \quad (8)$$

and

$$T_m = \sigma_m h(r_v) f(x_{myo}) \left[\exp(-k_m [r_v - r_m]^2) \right] \quad (9)$$

where the term $f(x_{myo})$ (Eq. (11)) expresses the degree of activation of the vascular smooth muscle cells of the afferent arteriole as a function of the error signal from the myogenic mechanism, x_{myo} (Eq. (12)). The wall thickness, $h(r_v)$, varies with the radius, so that the cross-sectional area of the vessel wall remains constant. Details of the definition and computation of $h(r_v)$ are given in the Appendix. The remaining terms are empirical constants, whose values are given in Table 1 in the Appendix.

Given an intravascular pressure P_v . The afferent arteriole will adjust its radius so that

$$P_v r_v = T_e + T_m \quad (10)$$

$f(x_{myo})$ in T_m (Eq. (9)) is a softmax-like function defined as

$$f(x_{myo}) = \frac{3 \exp(x_{myo})}{\exp(x_{myo}) + 2 \exp(-0.5 x_{myo})} \quad (11)$$

Myogenic response.

The degree of activation of the vascular smooth muscle cells arising through the myogenic mechanism, x_{myo} , is given by

$$x_{myo} = G(T - T_0(1 - x_{tgf})) \quad (12)$$

where G is the gain of the myogenic mechanism, and T_0 is the set point. The set point is modified by the signal from the tubuloglomerular feedback mechanism, x_{tgf} (Eq. (13)).

Tubuloglomerular feedback.

The action of tubuloglomerular feedback was assumed to be mediated solely through the afferent arteriole [6] and to be described by a logistic equation that relates the feedback signal, x_{tgf} , to the NaCl concentration at the macula densa, $C_S(md)$:

$$x_{tgf} = \left(\xi_{\max} - \frac{\psi}{1 + \exp[k(C_S(md) - C_{1/2})]} \right) \quad (13)$$

Full-scale renal arterial tree model

The reconstructed full-scale renal arterial structure from [17] is a skeleton-tree-based structure, given as a directed acyclic graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ where \mathcal{V} is a set of nodes at the endpoints of each vessel centerline with node features being its coordinates in Euclidean space, and \mathcal{E} is a set of directed edges which form a connected tree structure. Each edge represents a single vessel segment as a cylinder with its radius as the edge feature.

With a full-scale vascular tree model that starts from a single renal artery and ends in the afferent arterioles [17], the hemodynamic resistance in each vessel segment connecting nodes i and j is computed by Poiseuille's law (assuming laminar flow):

$$R_{i,j} = \frac{8\mu l_{i,j}}{\pi r_{i,j}^4} \quad (14)$$

where $l_{i,j}$ is the length of the segment, and $r_{i,j}$ is the inner radius of the vessel.

For any intermediate node n in the tree, Kirchhoff's current law (1st Law) states that inflow equals the sum of outflows:

$$Q_{p,n} = \sum_{i \in B_v/p} Q_{n,i} \quad (15)$$

where B_n and p denotes the set of neighboring nodes of n and the single parent node of n , respectively.

Combined with Poiseuille's law

$$Q_{i,j} = \frac{P_i - P_j}{R_{i,j}} \quad (16)$$

Kirchhoff's current law can then be rewritten as

$$\frac{P_p - P_n}{R_{p,n}} = \sum_{i \in B_v/p} \frac{P_n - P_i}{R_{n,i}} \quad (17)$$

The above equation can be rearranged to separate P_n with other terms, which formulates the governing equation

$$\left(\frac{1}{R_{p,n}} + \sum_{i \in B_v/p} \frac{1}{R_{n,i}} \right) P_n - \left(\frac{P_p}{R_{p,n}} + \sum_{i \in B_v/p} \frac{P_i}{R_{n,i}} \right) = 0 \quad (18)$$

Note that boundary conditions need to be defined to solve the above equation. A single inlet pressure $P_{in} = P_{art}$ (the arterial pressure in the aorta) is provided to represent different input pressure levels. However, the outlet boundary condition needs more care since the pressure at the end of the afferent arteriole remains unknown before we solve the system. To properly define the outlet boundary condition, the afferent arteriole is extended with two parts in series: the efferent arteriole and the post-glomerular capillary bed, as shown in Fig. 2. The pressure at the end of the post-glomerular capillary bed is assumed to be 0 mm Hg, which naturally defines the outlet boundary condition: $P_{out} = P_{PC} = 0$ mm Hg (corresponding to the venous pressure). See the Appendix for details.

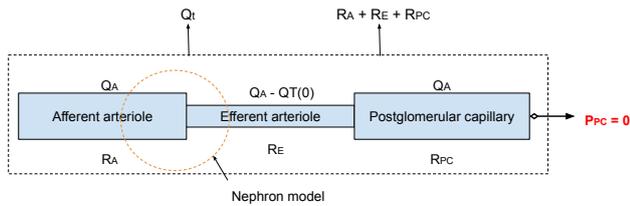


Fig. 2. Extended vascular model showing the afferent and efferent arterioles, and the post glomerular capillary bed, and their associated resistances, together with the nephron model.

After solving the equations associated with Kirchhoff's current law (Eq. (18)), the blood flows along the full-scale renal vascular tree, including the afferent arterioles, are updated.

Subpopulations of nephrons

The nephrons of the renal cortex are not identical. In general, the closer to the border between the cortex and the medulla, the longer their loop of Henle (see Fig. 3). Conventionally, three kinds of nephrons can be identified and are labeled as superficial, midcortical, or juxtamedullary based upon the location of their glomerulus within the renal cortex [20], as shown in Fig. 3. In general, superficial nephrons have glomeruli

located near the surface of the kidney with short loops of Henle. Juxtamedullary nephrons have a glomerulus near the border between the cortex and medulla and have long loops of Henle that extend deep into the medulla. Midcortical nephrons have loops of Henle with intermediate lengths. [20, 21, 22]. In the present model, the renal cortex is supposed to have a depth of 2 mm [17]. Nephrons located in the outer 1 mm of the cortex are supposed to be cortical nephrons, constituting 50 % of all nephrons. Midcortical nephrons have their glomeruli in a band of 1 mm to 1.4 mm from the surface of the kidney and constitute 20 % of all the nephrons. Finally, the juxtamedullary nephrons have their glomeruli located from 1.4 mm from the surface and down to the start of the medulla (2 mm from the surface). They constitute 30 % of the total nephron population.

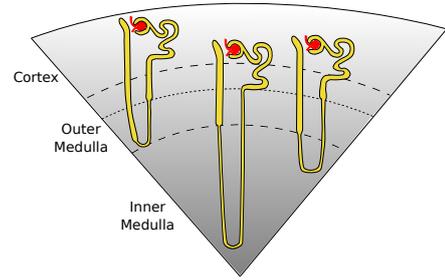


Fig. 3. Anatomical features of the three nephron populations. The glomerulus is indicated by the filled red circles. The dotted line indicates the border between the outer and inner stripe of the outer medulla.

It has been shown that the TGF is stronger in the juxtamedullary compared to the superficial nephrons [22]. Hence, we have adjusted the parameters of the TGF accordingly (see Table 5). The details on the morphology and parameters of the three nephron populations are given in Table 6 in the Appendix.

Simulation pipeline

Fig. 4 illustrates the flowchart of our simulation process that integrates the four models and solves them consequently.

With a prebuilt full-scale arterial tree model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and given the boundary condition of the arterial pressure P_{art} , we use the Kirchhoff's current law to calculate the pressure at every node and blood flow at every vessel as discussed in Section 2.2.4. The blood flow at each afferent arteriole defines its glomerular filtration rate $SNGFR$, which is then used to compute the plasma flow Q_A by

$$Q_A = (1 - Ht_A) SNGFR \quad (19)$$

where Ht_A is a constant representing the hematocrit.

With the plasma flow Q_A and the capillary hydrostatic pressure P_{GC} (equivalent to the pressure at the end of afferent arterioles), the only unknown in the Glomerular model in Section 2.2.1 is the tubular pressure $P_T(0)$ (pressure at the beginning of the tubule). Given an initial guess of $P_T(0)$, the filtration rate $SNGFR = Q_T(0)$ is calculated as discussed in the Glomerular model in Section 2.2.1.

As shown in Fig. 4, $P_T(0)$ and $Q_T(0)$ from the Glomerular model then serve as the boundary condition to the Tubular model, which will give the NaCl concentration at the macula densa $C_S(md)$ in the ascending limb, as well as a new tubular

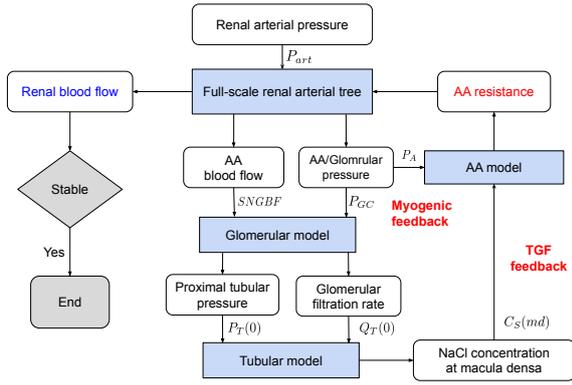


Fig. 4. Flowchart for full-scale model. The renal arterial blood pressure P_{art} serves as the boundary condition to the model of the renal vascular tree, which will provide the single nephron blood flow $SNGBF$ and glomerular pressure P_{GC} at afferent arterioles (AA) as the input to the Glomerular model. The Glomerular model will then give as output the glomerular filtration rate $SNGBF = Q_T(0)$ and the tubular pressure $P_T(0)$, which serve as the initial flow and pressure value to the Tubular Model. The Tubular Model will give as output the NaCl concentration at the macula densa ($C_S(md)$), which is the stimulus for TGF. The myogenic response is induced by the pressure at the beginning (P_A) and end (P_{GC}) of the afferent arteriole. TGF and the myogenic response, as the two inputs to the Afferent Arteriole (AA) model, will work together to adjust the afferent arteriolar resistance. With a new set of afferent arteriolar resistances, the full-scale arterial tree will provide a new set of single nephron blood flows $SNGBF$ and glomerular pressures P_{GC} . This process will be iterated until convergence. The total renal blood flow, which is the sum of the $SNGBF$ in the individual nephrons, does not directly participate in the autoregulation model itself but is the indicator of the autoregulation outcome and determines if the iteration reaches convergence.

pressure $P_T(0)$. This whole process is iterated until $P_T(0)$ converges.

The NaCl concentration at the macula densa $C_S(md)$ at convergence is then used in Eq. (13) to calculate the tubuloglomerular feedback, as shown in Fig. 4, which is then input to the Afferent arteriole model to calculate the active T_m and the elastic tension T_e . The calculated tension is then used in Eq. (10) to solve for the adjustment of afferent arteriole radius r_v .

As shown in Fig. 4, the adjusted afferent arteriole radii will induce a new set of resistances to the afferent arterioles, by Poiseuille's law, as defined in Eq. (14). We then re-apply Kirchhoff's law (Eq. (44)) on the full-scale vascular tree with these new resistances for the afferent arterioles, which results in a new set of flow $SNGBF$ and P_{GC} for the afferent arterioles. The whole process is then re-iterated until the flow and pressure converge.

Results

Single nephron

To illustrate the effects of the two regulatory mechanisms in the model, Fig. 5 shows the radius of the afferent arteriole for different afferent arteriolar pressures and NaCl concentrations at the macula densa, $C_S(md)$, in a single cortical nephron. The left panel shows the TGF response when the pressure in the afferent arteriole is held constant at a value of 57 mm Hg. As the $C_S(md)$ is increased, there is a decrease in the radius of the afferent arteriole (the TGF response). Likewise,

when the pressure in the afferent arteriole is increased, while the $C_S(md)$ is held constant at 45 mM (middle panel), there is also a decrease in the radius of the afferent arteriole (the myogenic response). The right panel shows that when both the pressure and the $C_S(md)$ are changed simultaneously the two regulatory mechanisms act additively in regulating the radius of the afferent arteriole.

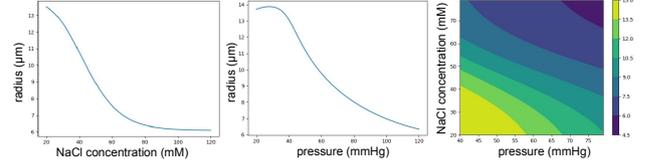


Fig. 5. The single cortical nephron model reproduces the well-known dependency of the radius with increasing NaCl concentration at macula densa, when the pressure in the afferent arteriole is held constant at 57 mm Hg (left panel) and with increasing arterial pressure, when the $C_S(md)$ is held constant at 45 mM (middle panel). The right panel shows the resulting radius (color bar) as a function of both the $C_S(md)$ and the pressure. Notice that the two regulatory mechanisms act additively.

Vascular tree: Control case

This section outlines the behavior of the full-scale vascular tree model with and without autoregulation in the case where all nephrons are of the cortical type. In the case without autoregulation, the radius of all the vessels in the tree is as generated by [17]. When the input pressure is $P_{art} = 100$ mm Hg as the control case, the full-scale arterial tree model will exhibit different radii, pressures, and flows in the afferent arterioles depending on their position in the arterial tree. Fig. 6 shows that without autoregulation (left panel), the pressure and flow in the afferent arterioles follow a normal distribution. Although their means (58 mm Hg and 3.9 nl/s) lie in the normal ranges for afferent arteriolar pressures and flows [23], the distributions are quite broad with too large or too small values associated with some afferent arterioles. On the other hand without autoregulation, the radius distribution is quite narrow with a mean of 10 μm .

When autoregulation is active, it is clear that the pressure and flow distributions become narrower, whereas the distribution for radii is broadened (Fig. 6). Thus, the nephron-vascular network adjusts the flow and pressure associated with each afferent arteriole by adjusting the radius of the afferent arteriole, through the action of the two feedback mechanisms, the TGF and the myogenic response.

Fig. 6 also shows that with active autoregulation (right panel), the radii of afferent arterioles will be adjusted, so that some afferent arterioles will constrict while others dilate, with only a minimal effect on the mean values of the flow, the pressure, and the radius. In other words, by adjusting the radii of the afferent arterioles, the autoregulatory mechanisms make the distribution in the network more uniform with respect to the variables that are essential for filtration in the individual nephrons, the blood flow and pressure in the individual glomeruli.

Fig. 7 shows scatter plots of various parameters with active autoregulation. The pressure-radius relationship (a) shows that in nephrons with a low filtration pressure, the afferent arterioles are dilated while the ones with a high pressure are contracted. The second panel (b) shows the TGF feedback

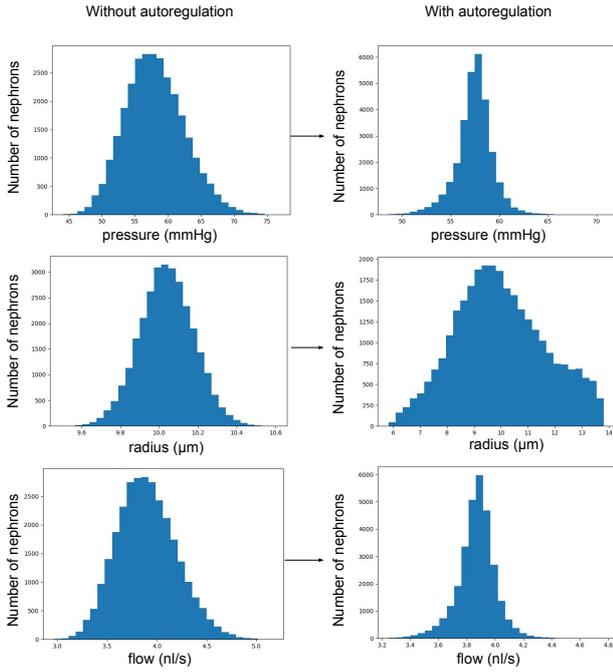


Fig. 6. The effect of autoregulation on the afferent arteriolar pressure (top), radius (middle), and flow (bottom) distributions in the control case (all nephrons are of cortical type). Left panel represents a situation without autoregulation; Right panel – with autoregulation. Renal blood flow is equal to the sum of the flows at afferent arterioles (bottom); without autoregulation, it is 6.92 ml/min, and with autoregulation, it is 6.87 ml/min.

signal (x_{TGF}) vs NaCl concentration at the macula densa $C_S(md)$. The relationship is sigmoidal, as would be expected from the TGF function in the individual nephrons (cf. Eq. 13). Some nephrons have reached the maximum value for the TGF mechanism. The tension-pressure relationship (c) shows that for low pressures, there is a direct positive relation between pressure and wall tension (T); at these low pressures, the vessels are maximally dilated and the wall tension will therefore be positively correlated with the pressure according to the Laplacian equation (Eq. (1)). At higher pressures, the vessel contracts, and the reduction in radius is relatively larger than the increase in pressure, so the net result is a decrease in wall tension with increasing pressure. The last panel (d) shows that the two mechanisms of autoregulation (TGF and myogenic response) are positively correlated, so that they act additively to provide effective autoregulation. The colored crosses in the four panels indicate five different nephrons operating under different combinations of flow and pressure in the vascular tree.

Vascular tree: Autoregulation

Mechanisms of autoregulation.

When the arterial pressure changes from the control value, the renal autoregulatory mechanisms will try to minimize the change in renal blood flow by adjusting the afferent arteriolar resistances by means of the two mechanisms described in Section 2.1. Here, we simulate autoregulation in the full-scale vascular model by varying the inlet (arterial) pressure from 80 mm Hg to 200 mm Hg and calculating the resulting renal blood flow, both with and without autoregulation. In these simulations, all nephrons are of the cortical type.

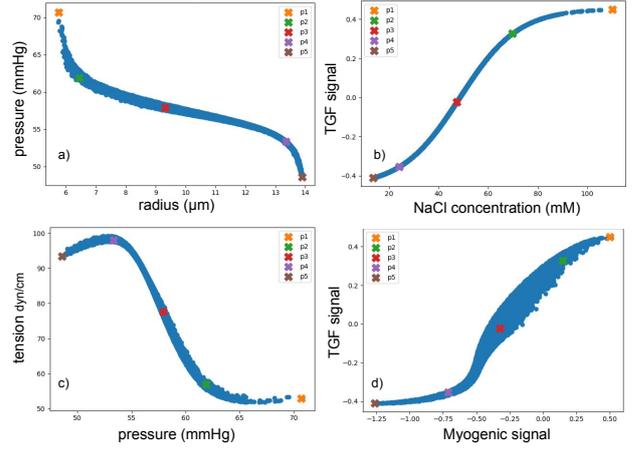


Fig. 7. Scatter plots in the control case (all nephrons are of cortical type) with activated autoregulation: pressure vs radius, TGF response vs NaCl concentration at macula densa ($C_S(md)$), tension vs pressure, and TGF response vs myogenic response.

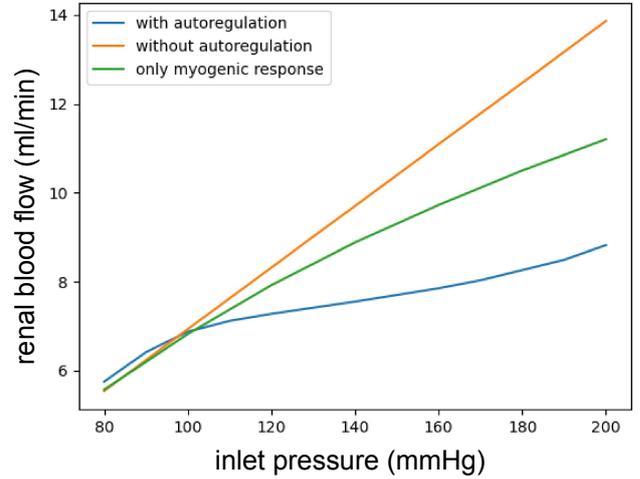


Fig. 8. Renal blood flow as a function of inlet (arterial) blood pressure increases linearly with pressure when there is no autoregulation (orange). When autoregulation is active, the blood flow reaches a plateau and stays relatively constant over a wide range of inlet pressures. (blue). If only the myogenic mechanism is active, the renal blood flow is intermediate between the two previous cases (green).

Fig. 8 shows the resulting renal blood flow with and without autoregulation. Without autoregulation (orange curve), the renal blood flow increases linearly with increasing renal arterial pressure. In contrast, the autoregulation provided by the TGF and the myogenic response is effective in minimizing the increase of renal blood flow when the inlet pressure increases from 80 mm Hg to 200 mm Hg (blue curve). The autoregulation degrades when the inlet pressure becomes too large, as can be seen from the larger slope at inlet pressures from 180 mm Hg to 200 mm Hg.

We simulate the effect of blocking TGF (which can be achieved by giving a loop diuretic) so that only the myogenic response is active. Note that TGF is integrated into the model by modifying the set point of the myogenic response (cf. Eq. (12)), so it is not possible to simulate the TGF response alone without the myogenic response. If TGF is inactivated, only the myogenic response contributes to autoregulation and

cannot fully compensate for the increasing blood flow (green curve). It is evident that both mechanisms are necessary for an efficient autoregulation of renal blood flow.

Fig. 9 shows the blood flow and radius distribution of the afferent arterioles without and with autoregulation at a renal arterial pressure of $P_{art} = 140$ mm Hg. Without autoregulation, there is a significant increase in renal blood flow (from 6.9 ml/min to 9.7 ml/min) due to a large inlet renal arterial pressure. After activation of autoregulation, the afferent arterioles contract, and the increase of their resistances counteracts the increase in inlet renal arterial pressure. As a result, the renal blood flow is reduced to 7.55 ml/min, which is closer to the control case with $P_{art} = 100$ mm Hg, giving RBF = 6.9 ml/min as shown in Fig. 6.

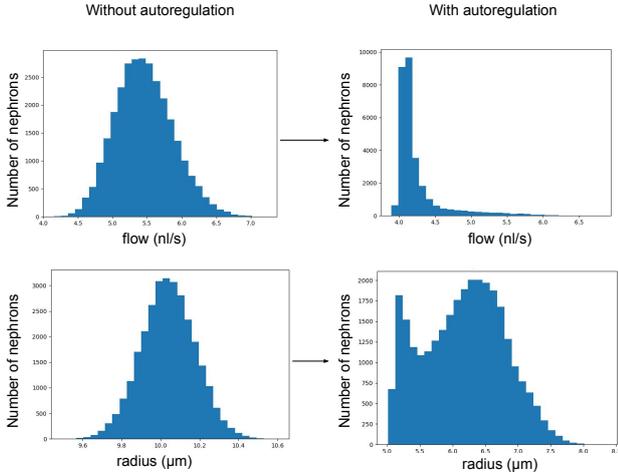


Fig. 9. The effect of autoregulation on afferent arteriolar flow (top) and radius (bottom) distributions when the inlet renal arterial pressure $P_{art} = 140$ mm Hg. Left panel without autoregulation; Right panel – with autoregulation. Renal blood flow is equal to the sum of the flows in afferent arterioles (top); without autoregulation, it is 9.7 ml/min, and with autoregulation, it is 7.55 ml/min.

To assess the efficiency of autoregulation quantitatively, we compute the autoregulation index (AI) at different levels of inlet arterial pressure [24]. The AI is given as the ratio between the relative change of renal blood flow (RBF) change and the relative change of renal arterial pressure (P_{art}):

$$AI = \frac{(RBF_2 - RBF_1) / RBF_1}{(P_{art2} - P_{art1}) / P_{art1}} \quad (20)$$

Fig. 10 shows the Autoregulation index (AI) in the full-scale nephron-vascular network at increasing levels of the inlet pressure. Experimental studies in normal rats have found an AI between 0.2 and 0.5 [25]. As can be seen from Fig. 10, the model predicts similar values in the inlet pressure range from 100 mm Hg to 160 mm Hg.

Nephron populations.

Fig. 10 and Fig. 11 show renal blood flow autoregulation after integrating three populations of nephrons, cortical, midcortical and juxtamedullary. It is clear, that using a more realistic scenario for the nephrons yields a slight improvement of autoregulatory efficiency, especially at the highest inlet pressures.

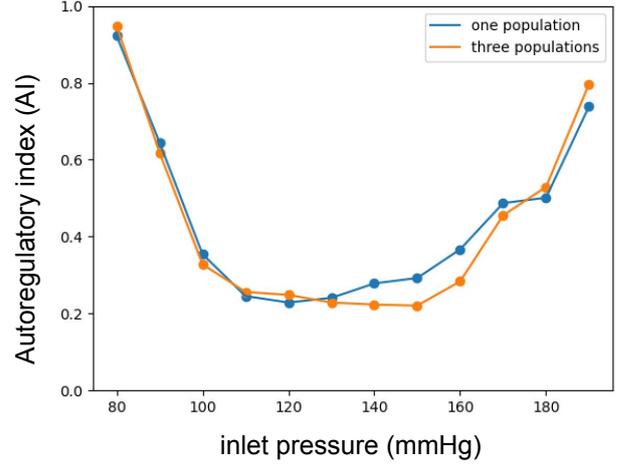


Fig. 10. Autoregulation Index at various inlet pressure levels shows a wider plateau when three populations of nephrons (orange) are involved in comparison with only one population of superficial nephrons (blue).

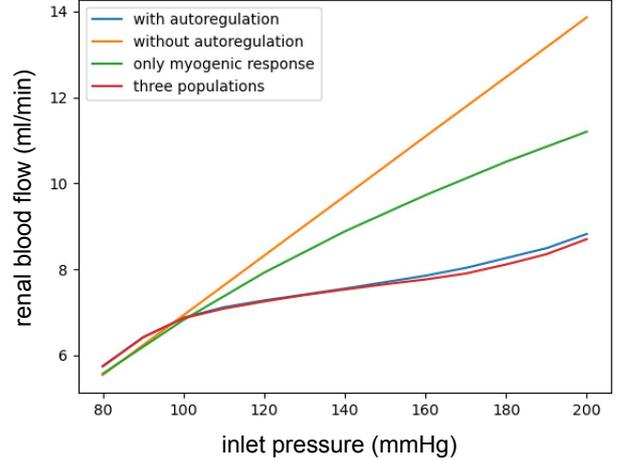


Fig. 11. Autoregulation of blood flow at various pressure levels including three populations of nephrons.

To take a closer look at the effect of having three nephron populations, Fig. 12 shows the behavior of the three types of nephrons at an arterial pressure of $P_{art} = 140$ mm Hg. From Fig. 12 top panel, it is clear that the NaCl concentrations at the macula densa ($C_S(md)$) are similar in the three types when the flows are around the control case (around 4 nl/min). However, when the flow becomes large, juxtamedullary nephrons with long loops of Henle will have larger NaCl concentrations at the macula densa ($C_S(md)$). Fig. 12 bottom panel shows the TGF response vs the NaCl concentration levels at the macula densa $C_S(md)$ in the three types of nephrons. As the sigmoid-like functions indicate, they all saturate when $C_S(md)$ becomes too large, but the juxtamedullary nephrons with long loops of Henle, saturate at a larger value for the TGF response. So, it can be concluded that the slight improvement in renal autoregulation seen when having three populations of nephrons, is due to the slightly stronger TGF response in the midcortical and the juxtamedullary nephrons compared to the cortical ones. However, as Fig. 11 shows, the nephrons having longer loops of

Henle will only have a moderate effect on the overall efficiency of renal blood flow autoregulation.

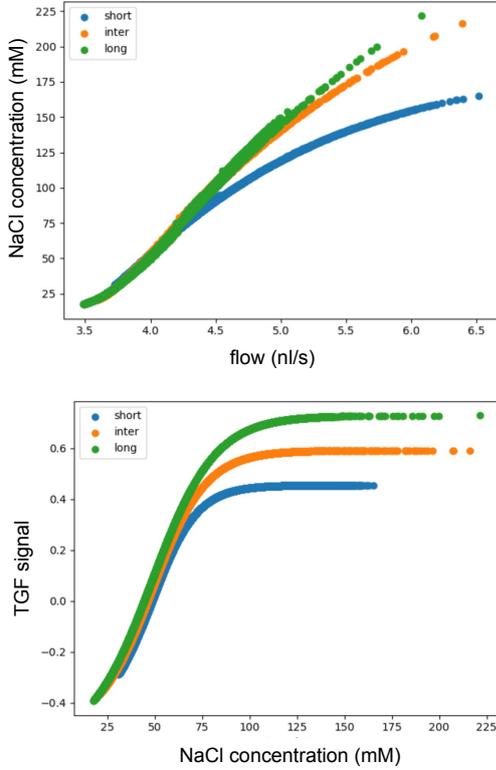


Fig. 12. Scatter plot shows distinct behavior of three populations of nephrons: NaCl concentration at macula densa ($C_S(md)$) vs flow (top), and TGF response vs $C_S md$ (bottom) at, $P_{art} = 140$ mmHg.

Renal artery stenosis.

In addition to simulating autoregulation's function in stabilizing blood flow under different input pressure levels, another perspective in the full-scale modeling of the nephron-vascular network is to simulate some pathological conditions. In the renal arteries, a common pathology is renal artery stenosis (RAS), which is a narrowing of the lumen of the renal artery, most often caused by atherosclerosis or fibromuscular dysplasia [26]. RAS is a serious condition that may cause hypertension and renal damage due to ischemia. For example, the American Heart Association defined a RAS of more than 60% to be the critical degree of stenosis demanding treatment [27].

One of the main purposes of simulating RAS is to examine the consequences of different degrees of RAS and the kidney's ability to compensate for the stenosis. By simulating different levels of stenosis and studying the corresponding outcome on renal hemodynamics, better preventative and surgical strategies can be made [26, 28, 29].

Initially, RAS is simulated in the full-scale model of the nephron-vascular network using the control case inlet pressure ($P_{art} = 100$ mmHg) while gradually decreasing the diameter of renal artery (the root vessel in the full-scale tree model).

As shown in Fig. 13 top panel, the renal blood flow stays relatively constant at the lower degrees of RAS but drops dramatically when stenosis exceeds 50%, regardless of whether or not autoregulation is active. This behavior is in

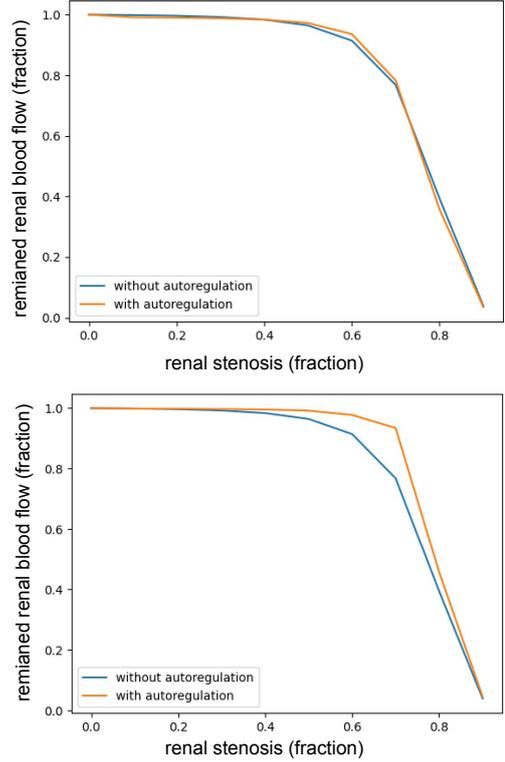


Fig. 13. Renal blood flow at various stenosis levels in the control case (Top panel with inlet pressure $P_{art} = 100$ mmHg) and in the hypertensive case (Bottom panel with inlet pressure $P_{art} = 140$ mmHg). The control case shows a drastic decrease of renal blood flow with around 50% stenosis, while the hypertensive cases show that the autoregulation can postpone the drastic decrease of blood flow from around 50% stenosis to around 70% stenosis.

good agreement with the American Heart Association, which defines a RAS of more than 60% as critical stenosis [27] that needs treatment according to the arterial disease management guidelines [30].

It is clear that the impact of autoregulation in preventing a drop in blood flow is minimal (cf. Fig. 13 top). This is to be expected because in the control case ($P_{art} = 100$ mmHg), autoregulation has limited power to compensate for a further decrease in the renal perfusion pressure, cf. Fig. 10 and Fig. 11. Consequently, as the stenosis becomes severe, the renal blood flow will be dramatically reduced.

The result suggests that the flat curve in Fig. 13 when the stenosis $< 50\%$ is not mainly attributable to autoregulation, but rather stems from the fact that the hemodynamic resistance of the renal artery has little impact on renal blood flow until the stenosis is severe.

The consequences of RAS were also simulated in hypertension using an inlet pressure of $P_{art} = 140$ mmHg, since RAS often occurs together with hypertension. Fig. 13 bottom panel shows that in hypertension, the autoregulatory mechanism is effective in stabilizing the renal blood flow when the stenosis is less than 70%. Without autoregulation, the renal blood flow already decreases at a stenosis level of around 50%. But it can also be seen that with a high level of stenosis, even at a pressure level where the autoregulatory mechanisms are highly effective, the impact on the overall renal blood flow remains limited.

Discussion

In this work, about 30000 nephron models were coupled to a full-scale renal arterial tree that was reconstructed by Xu et al. by a physiologically based constructive optimization method that incorporates anatomical data extracted from a scan of a rat kidney [17]. Such a nephron-vascular network enables the simulation of renal autoregulation at a full scale, while previous works were mostly based on one or a few nephrons [6, 7, 8, 9, 10, 11]. This is important because the nephrons and their afferent arterioles are inter-connected in a renal arterial tree and their resistances will jointly adjust the renal blood flow.

Our simulation results in Fig. 8, Fig. 10, and Fig. 11 show a strong autoregulation due to the joint feedback from the TGF and the myogenic response. In general, autoregulation is most effective in counteracting an increase in arterial blood pressure (from 100 mm Hg to 160 mm Hg), while there is little autoregulation when the pressure is reduced (< 100 mm Hg). The range of autoregulation is limited due to the saturation of especially TGF, and autoregulation degrades when the pressure becomes too high (> 160 mm Hg) or too low (< 100 mm Hg).

On the other hand, most textbooks assume perfect autoregulation [31, 32], i.e., the blood flow would stay constant within certain pressure ranges in Fig. 11. In most experiments and computer simulations, a well-functioning autoregulation usually has an AI between 0.2 and 0.5, while $AI < 0.2$ shows excellent autoregulation [25, 24], which matches our simulation results where the AI is very close to 0.2 with inlet pressure ranging from 110 to 160 mm Hg.

The relative roles of TGF and the myogenic response in renal autoregulation have been the subject of many studies over the last decades [33]. Today the consensus is that they both contribute in equal amounts to the response. This is in good agreement with the present model. When simulating autoregulation without the TGF response (Fig. 11), autoregulation is significantly impaired. This demonstrates that both mechanisms are necessary for an efficient autoregulation of renal blood flow.

Computer simulation of the full-scale nephron-vascular network can also be used to assess the impact of different physiological and pathological conditions in the kidney and help researchers and clinicians investigate various aspects of blood flow disruptions, e.g., the consequences of thrombosis or atherosclerosis in the renal vasculature. As an example, we have simulated renal artery stenosis to examine its effect on renal blood flow.

There have been several efforts in simulating renal artery stenosis (RAS) using Computational Fluid Dynamics (CFDs) [34, 35, 28, 26, 35, 29], which is the conventional method used to simulate blood flow and blood flow dysfunction in the vasculature. These methods involve constructing computational meshes by segmenting the medical images. This is then used to numerically solve the Navier-Stokes equations for the system under investigation. This approach has been widely used in studies of coronary artery disease and aortic aneurysm [36, 37, 38, 39, 40]. The difference between the approach in this paper, and the previous work is that whereas the latter has only focused on the effect of the stenosis in the renal artery [35, 28, 26], the present approach simulate the blood flow in a full-scale model of the renal vasculature together with the impact of renal autoregulation. Since RAS usually only affects the diameter of the renal artery, it is possible to use CFD-based methods to simulate renal artery stenosis with only one

segment of the renal arterial network. Nonetheless, our full-scale nephron-vascular network also accounts for the effects of autoregulation. Note that although RAS involves the decrease in diameter of only the renal artery, we need a full-scale structure to properly define the outlet boundary conditions.

Our simulation results show a significant decrease of renal blood flow with stenosis of more than 50%, which is in good agreement with both existing CFD-based simulations [34, 35, 28, 26, 35, 29] and the American Heart Association which defines RAS of more than 60% as a critical stenosis [27] that needs adequate treatment. This could indicate that complex computational meshes or finite element or volume solvers are not necessary, as a tree-based model can generate very similar results and is significantly more efficient.

The current model is a steady-state model. It is well known that, at the microcirculatory level, renal hemodynamics show significant dynamics, such as oscillations in single nephron blood flow and glomerular filtration rate [41]. However, incorporating dynamic elements will require solving partial differential equations within the full-scale vascular model, a task that is computationally intensive.

Although our model encompasses all afferent arterioles and nephrons within a complete tree structure, we have yet to address the interactions between nephrons that communicate via electrical signals along vessel walls [12, 13]. While our model includes afferent arterioles to collaborate in autoregulating blood flow throughout the arterial tree, each nephron independently regulates its afferent arteriole. The goal of future work is to include nephron-nephron interactions in the current full-scale model of the renal vascular network.

References

- Daniel Casellas, Madeleine Dupont, Nathalie Bouriquet, Leon C Moore, Annie Artuso, and Albert Mimran. Anatomic pairing of afferent arterioles and renin cell distribution in rat kidneys. *American Journal of Physiology-Renal Physiology*, 267(6):F931–F936, 1994.
- Donald J Marsh, Dmitry D Postnov, Douglas J Rowland, Anthony S Wexler, Olga V Sosnovtseva, and Niels-Henrik Holstein-Rathlou. Architecture of the rat nephron-arterial network: analysis with micro-computed tomography. *American Journal of Physiology-Renal Physiology*, 313(2):F351–F360, 2017.
- M Walker 3rd, LM Harrison-Bernard, AK Cook, and Navar LG. Dynamic interaction between myogenic and tgf mechanisms in afferent arteriolar blood flow autoregulation. *Am J Physiol Renal Physiol*, 279(5):F858–F865, 2000.
- F Kiil. Analysis of myogenic mechanisms in renal autoregulation. *Acta Physiol Scand*, 174(4):347–355, 2002.
- WA Cupples, AS Wexler, and DJ Marsh. Model of tgf-proximal tubule interactions in renal autoregulation. *Am J Physiol Renal Physiol*, 259(4):F715–F726, 1990.
- N H Holstein-Rathlou and DJ Marsh. A dynamic model of renal blood flow autoregulation. *Bulletin of mathematical biology*, 56:411–429, 1994.
- Leon C Moore, Jurgen Schnermann, and Shiroh Yarimizu. Feedback mediation of sngfr autoregulation in hydropenic and doca-and salt-loaded rats. *American Journal of Physiology-Renal Physiology*, 237(1):F63–F74, 1979.
- Tenei Sakai, Eileen Hallman, and Donald J Marsh. Frequency domain analysis of renal autoregulation in the rat. *American Journal of Physiology-Renal Physiology*, 250(2):F364–F373, 1986.
- Frederick H Daniels, William J Arendshorst, and Robert G Roberds. Tubuloglomerular feedback and autoregulation in spontaneously hypertensive rats. *American Journal of Physiology-Renal Physiology*, 258(6):F1479–F1489, 1990.
- NH Holstein-Rathlou, Anthony J Wagner, and Donald J Marsh. Tubuloglomerular feedback dynamics and renal blood flow autoregulation in rats. *American Journal of Physiology-Renal Physiology*, 260(1):F53–F68, 1991.
- HE Layton, E Bruce Pitman, and Leon C Moore. Bifurcation analysis of tgf-mediated oscillations in sngfr. *American Journal of Physiology-Renal Physiology*, 261(5):F904–F919, 1991.
- Donald J Marsh, Dmitry D Postnov, Olga V Sosnovtseva, and Niels-Henrik Holstein-Rathlou. The nephron-arterial network and its interactions. *American Journal of Physiology-Renal Physiology*, 316(5):F769–F784, 2019.
- Donald J Marsh, Olga V Sosnovtseva, Ki H Chon, and Niels-Henrik Holstein-Rathlou. Nonlinear interactions in renal blood flow regulation. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 288(5):R1143–R1159, 2005.
- Donald J Marsh, Anthony S Wexler, Alexey Brazhe, Dmitri E Postnov, Olga V Sosnovtseva, and Niels-Henrik Holstein-Rathlou. Multinephron dynamics on the renal vascular network. *American Journal of Physiology-Renal Physiology*, 304(1):F88–F102, 2013.
- Dmitry D Postnov, Donald J Marsh, Dmitri E Postnov, Thomas H Braunstein, Niels-Henrik Holstein-Rathlou, Erik A Martens, and Olga Sosnovtseva. Modeling of kidney hemodynamics: probability-based topology of an arterial network. *PLoS computational biology*, 12(7):e1004922, 2016.
- E Bruce Pitman, Roman M Zaritski, Kevin J Kessler, Leon C Moore, and Harold E Layton. Feedback-mediated dynamics in two coupled nephrons. *Bulletin of mathematical biology*, 66:1463–1492, 2004.
- Peidi Xu, Niels-Henrik Holstein-Rathlou, Stinne Byrholdt Sogaard, Carsten Gundlach, Charlotte Mehlin Sorensen, Kenny Erleben, Olga Sosnovtseva, and Sune Darkner. A hybrid approach to full-scale reconstruction of renal arterial network. *Scientific Reports*, 13(1):7569, 2023.
- David A Nordsletten, Shane Blackett, Michael D Bentley, Erik L Ritman, and Nicolas P Smith. Structural morphology of renal vasculature. *American Journal of Physiology-Heart and Circulatory Physiology*, 291(1):H296–H309, 2006.
- Ki H Chon, Niels-Henrik Holstein-Rathlou, Donald J Marsh, and Vasilis Z Marmarelis. Comparative nonlinear modeling of renal autoregulation in rats: Volterra approach versus artificial neural networks. *IEEE transactions on neural networks*, 9(3):430–435, 1998.
- Charlene McQueen. *Comprehensive toxicology*. Elsevier, 2017.
- Joseph Feher. 7.2-functional anatomy of the kidneys and overview of kidney function. *Quantitative Human Physiology. Boston, MA: Academic*, pages 626–632, 2012.
- Roland Muller-Suur, Hans R Ulfendahl, and AE Persson. Evidence for tubuloglomerular feedback in juxtamedullary nephrons of young rats. *American Journal of Physiology-Renal Physiology*, 244(4):F425–F431, 1983.
- Donald J Marsh, Olga V Sosnovtseva, Alexey N Pavlov, Kay-Pong Yip, and Niels-Henrik Holstein-Rathlou. Frequency encoding in renal blood flow regulation. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 288(5):R1160–R1167, 2005.
- Frank P Tiecks, Arthur M Lam, Rune Aaslid, and David W Newell. Comparison of static and dynamic cerebral autoregulation measurements. *Stroke*, 26(6):1014–1019, 1995.
- E.J. Johns and A.F. Ahmeda. Renal circulation. In *Reference Module in Biomedical Sciences*. Elsevier, 2014.
- Mohammad Andayesh, Azadeh Shahidian, and Majid Ghassemi. Numerical investigation of renal artery hemodynamics based on the physiological response to renal artery stenosis. *Biocybernetics and Biomedical Engineering*, 40(4):1458–1468, 2020.
- Krishna J Rocha-Singh, Andrew C Eisenhauer, Stephen C Textor, Christopher J Cooper, Walter A Tan, Alan H Matsumoto, Kenneth Rosenfield, and Writing Group 8. Atherosclerotic peripheral vascular disease symposium ii: intervention for renal artery disease. *Circulation*, 118(25):2873–2878, 2008.
- Pooja Jhunjhunwala, PM Padole, SB Thombre, and Atul Sane. Prediction of blood pressure and blood flow in stenosed renal arteries using cfd. In *IOP Conference Series: Materials Science and Engineering*, volume 346, page 012066. IOP Publishing, 2018.
- Weisheng Zhang, Yi Qian, Jiang Lin, Peng Lv, Kaavya Karunanithi, and Mengsu Zeng. Hemodynamic analysis of renal artery stenosis using computational fluid dynamics technology based on unenhanced steady-state free precession magnetic resonance angiography: preliminary results. *The international journal of cardiovascular imaging*, 30:367–375, 2014.

30. Alan T Hirsch, Ziv J Haskal, Norman R Hertzler, Curtis W Bakal, Mark A Creager, Jonathan L Halperin, Loren F Hiratzka, William RC Murphy, Jeffrey W Olin, Jules B Puschett, et al. Acc/aha 2005 practice guidelines for the management of patients with peripheral arterial disease (lower extremity, renal, mesenteric, and abdominal aortic) a collaborative report from the american association for vascular surgery/society for vascular surgery,* society for cardiovascular angiography and interventions, society for vascular medicine and biology, society of interventional radiology, and the acc/aha task force on practice guidelines (writing committee to develop guidelines for the management of patients with peripheral arterial disease): Endorsed by the american association of cardiovascular and pulmonary rehabilitation; national heart, lung, and blood institute; society for vascular nursing; transatlantic inter-society consensus; and vascular disease foundation. *circulation*, 113(11):e463–e654, 2006.
31. Matthew E. Cross and Emma V. E. Plunkett. Autoregulation and renal vascular resistance. In *Physics, Pharmacology and Physiology for Anaesthetists: Key Concepts for the FRCA*, page 291–292. Cambridge University Press, 2014.
32. Neil Turner, Karlhans Endlich, and Rodger Loutzenhiser. Tubuloglomerular feedback, renal autoregulation, and renal protection. In *Oxford Textbook of Clinical Nephrology: Three-Volume Pack*. Oxford University Press, 10 2015.
33. M Carlstrom, CS Wilcox, and WJ Arendshorst. Renal autoregulation in health and disease. *Physiol rev*, 95(2):405–511, 2015.
34. Stephen C Textor. Renal arterial disease and hypertension. *Medical Clinics*, 101(1):65–79, 2017.
35. Shah Mohammed Abdul Khader, Adi Azriff, Cherian Johny, Raghuvir Pai, Mohammad Zuber, Kamarul Arifin Ahmad, and Zanuldin Ahmad. Haemodynamics behaviour in normal and stenosed renal artery using computational fluid dynamics. *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*, 51(1):80–90, 2018.
36. Alin-Florin Totorean, Sandor Ianos Bernad, Tiberiu Ciocan, Iuliana-Claudia Totorean, and Elena Silvia Bernad. Computational fluid dynamics applications in cardiovascular medicine—from medical image-based modeling to simulation: Numerical analysis of blood flow in abdominal aorta. In *Advances in Fluid Mechanics: Modelling and Simulations*, pages 1–42. Springer, 2022.
37. Alin F Totorean, Maria C Ioncica, Tiberiu Ciocan, Sandor I Bernad, Claudia I Totorean, and Elena S Bernad. Medical image-based numerical simulation of the abdominal aorta flow. In *2021 10th International Conference on ENERGY and ENVIRONMENT (CIEM)*, pages 1–5. IEEE, 2021.
38. R Vinoth, D Kumar, Raviraja Adhikari, S Vijayapradeep, K Geetha, R Ilavarasi, and Saravanakumar Mahalingam. Steady and transient flow cfd simulations in an aorta model of normal and aortic aneurysm subjects. In *The Proceedings of the International Conference on Sensing and Imaging*, pages 29–43. Springer, 2019.
39. John D Martin. Cfd analysis comparing steady flow and pulsatile flow through the aorta and its main branches. In *ASME International Mechanical Engineering Congress and Exposition*, volume 50534, page V003T04A064. American Society of Mechanical Engineers, 2016.
40. Gaoyang Li, Haoran Wang, Mingzi Zhang, Simon Tupin, Aike Qiao, Youjun Liu, Makoto Ohta, and Hitomi Anzai. Prediction of 3d cardiovascular hemodynamics before and after coronary artery bypass surgery via deep learning. *Communications biology*, 4(1):99, 2021.
41. Niels-Henrik Holstein-Rathlou and Donald J. Marsh. Renal blood flow regulation and arterial pressure fluctuations: a case study in nonlinear dynamics. *Physiol Rev*, 74(3):637–681, 1994.

Tubular model details

Proximal part

In the proximal part, C_S stays constant due to the isosmotic transport such that the reabsorption rate of water and NaCl is identical, thus $C_S = C_S(0)$. J_V is defined as an exponentially decreasing function along the tubule.

$$J_V = \kappa e^{-\theta z} \quad (21)$$

Thus,

$$\frac{dQ_T}{dz} = -\kappa e^{-\theta z} \quad (22)$$

The flow Q_T can thus be computed analytically by solving the integral

$$Q_T = \int_z \frac{dQ_T}{dz} dz = \int_z -\kappa e^{-\theta z} dz = \frac{\kappa}{\theta} e^{-\theta z} + C \quad (23)$$

Given the boundary condition $Q_T|_{z=0} = Q_T(0)$, we can solve C , giving

$$\frac{\kappa}{\theta} + C = Q_T(0) \implies C = Q_T(0) - \frac{\kappa}{\theta} \quad (24)$$

Substitute into Eq. (23) gives the exact definition of flow,

$$Q_T = \frac{\kappa}{\theta} e^{-\theta z} + Q_T(0) - \frac{\kappa}{\theta} \quad (25)$$

Substitute into Eq. (5) gives the pressure definition:

$$\frac{dP_T}{dz} = -\frac{8\mu}{\pi r^4} \left(\frac{\kappa}{\theta} e^{-\theta z} + Q_T(0) - \frac{\kappa}{\theta} \right) \quad (26)$$

Thus, the pressure P_T can also be solved analytically by

$$P_T = \int_z \frac{dP_T}{dz} dz = \frac{8\mu\kappa}{\pi r^4 \theta^2} e^{-\theta z} - \frac{8\mu}{\pi r^4} \left(Q_T(0) - \frac{\kappa}{\theta} \right) z + C \quad (27)$$

Here, however, the boundary condition is defined at the proximal end where $z = Z = 1\text{cm}$ with the value $P_T(Z)$. Thus,

$$P_T(Z) = \frac{8\mu\kappa}{\pi r^4 \theta^2} e^{-\theta Z} - \frac{8\mu}{\pi r^4} \left(Q_T(0) - \frac{\kappa}{\theta} \right) Z + C \quad (28)$$

giving

$$C = P_T(Z) - \frac{8\mu\kappa}{\pi r^4 \theta^2} e^{-\theta Z} + \frac{8\mu}{\pi r^4} \left(Q_T(0) - \frac{\kappa}{\theta} \right) Z \quad (29)$$

Substitute into Eq. (27) gives the exact solution of pressure at the proximal part:

$$P_T(z) = \frac{8\mu\kappa}{\pi r^4 \theta^2} (e^{-\theta z} - e^{-\theta Z}) + \frac{8\mu}{\pi r^4} \left(Q_T(0) - \frac{\kappa}{\theta} \right) (Z - z) + P_T(Z) \quad (30)$$

Descending limb

In the Descending limb of the loop of Henle, water is reabsorbed, meaning that flow rate Q decreases while the NaCl concentration C_S increases

$$J_V = L_v n_s (C_I - C_S) \quad (31)$$

and

$$J_S = L_S (C_S - C_I) \quad (32)$$

Substitute into Eq. (5) gives a system of two coupled ordinary differential equations for flow and NaCl concentration

$$\begin{cases} \frac{dQ_T}{dz} = -L_v n_s (C_I - C_S) \\ Q_T \frac{dC_S}{dz} + C_S \frac{dQ_T}{dz} = -L_S (C_S - C_I) \end{cases} \quad (33)$$

where the second equation needs to be rearranged, giving the final system

$$\begin{cases} \frac{dQ_T}{dz} = -L_v n_s (C_I - C_S) \\ \frac{dC_S}{dz} = -\frac{L_S (C_S - C_I) + C_S \frac{dQ_T}{dz}}{Q_T} \end{cases} \quad (34)$$

which is solved numerically using the standard Runge-Kutta method.

The pressure is solved numerically after Q_T has been solved using finite difference.

Ascending limb

In the ascending limb, only NaCl is reabsorbed, meaning that NaCl concentration C_S decreases while the flow Q_T stays constant, which is equal to the flow value at the end of descending limb $Q_T(\text{Desc_end})$. Here

$$J_S = L_S (C_S - C_I) + \frac{V_{max} C_S}{K_m + C_S} \quad (35)$$

substitute into Eq. (5) gives the definition of C_S with the differential equation

$$Q_T \frac{dC_S}{dz} + C_S \frac{dQ_T}{dz} = -L_S (C_S - C_I) - \frac{V_{max} C_S}{K_m + C_S} \quad (36)$$

Since $Q_T = Q_T(\text{Desc_end})$ stays constant, $\frac{dQ_T}{dz} = 0$, the above differential equation can then be simplified to

$$\frac{dC_S}{dz} = \frac{-L_S (C_S - C_I) - \frac{V_{max} C_S}{K_m + C_S}}{Q_T(\text{Desc_end})} \quad (37)$$

which is solved numerically using the standard Runge-Kutta method.

Similarly, the pressure can be simplified to:

$$\frac{dP_T}{dz} = -\frac{8\mu}{\pi r^4} Q_T(\text{Desc_end}) \quad (38)$$

which can be solved analytically by

$$P_T = \int_z \frac{dP_T}{dz} dz = -\frac{8\mu}{\pi r^4} Q_T(\text{Desc_end}) z + C \quad (39)$$

The boundary condition to solve the above equation is the tubular pressure at the end of the loop of Henle at $z = Z = 0.65\text{cm}$ from the empirical relationship

$$P_T(Z) = \frac{Q_T(Z)}{(\alpha P_T(Z) + \beta)^4} \quad (40)$$

where $Q_T(Z) = Q_T(\text{Desc_end})$ since flow stays constant in the ascending limb.

Thus,

$$C = P_T(Z) + \frac{8\mu}{\pi r^4} Q_T(\text{Desc_end}) Z \quad (41)$$

Substitute gives the exact analytical solution for pressure at the ascending limb:

$$P_T(z) = \frac{8\mu}{\pi r^4} Q_T(\text{Desc_end}) (Z - z) + P_T(Z) \quad (42)$$

where $z \in [0, 0.65\text{cm}]$ and $Z = 0.65\text{cm}$

NaCl Concentration at macula densa

One of the main purposes of modeling NaCl concentration C_s is that, the tubuloglomerular feedback, which is a logistic equation of the vascular smooth muscle, is activated at the macula densa by different levels of NaCl concentration at the macula densa $C_S(md)$, as discussed later in Eq. (13). In our model, macula densa is located at $z = 0.5$ at the ascending limb.

Full-scale tree model boundary condition and simulation details

When solving the blood flow in the full-scale tree model given a new set of afferent arteriole radii, boundary conditions need to be defined to solve the above equation to solve Eq. (18). A single inlet pressure $P_{int} = P_{art}$ (the arterial pressure from the aorta) is manually given to represent different input pressure levels. However, the outlet boundary condition needs more care since the pressure at the end of the afferent arteriole remains unknown before we solve the system. To properly define the outlet boundary condition, the afferent arteriole is extended with two parts in a series with no branchings: the efferent arteriole and the Post-glomerular capillary, as shown in Fig. 2. This is because the pressure at the end of the Post-glomerular capillary is usually assumed to be 0, which naturally defines the outlet boundary condition: $P_{PC} = 0$ (pressure at the end of the post-glomerular capillary).

Note that for efferent arteriole and post-glomerular capillary, their resistances (R_E and R_{PC}) are assumed to be constant, so that their geometries (length and radius) are not required. Since the three parts are connected in a series, for each afferent arteriole (i, j) , its equivalent total resistance is thus.

$$R_t = R_{i,j} + R_E + R_{PC} \quad (43)$$

On the other hand, note that the flow at the afferent arteriole and efferent arteriole is different due to glomerular filtration. The amount of flow being filtered at afferent arteriole i, j corresponds to the filtration rate $Q_{i,j}^T(0)$, which is one of the outputs from our nephron model. To be solved efficiently, the three parts are first regarded as one vessel with equivalent flow Q_t and resistance R_t as shown in Fig. 2. The real flow at afferent arteriole ($Q_{i,j}$) will be recovered by the properties $Q_E = Q_{i,j} - Q_{i,j}^T(0)$ and $Q_{PC} = Q_{i,j}$. In the first round, $Q_{i,j}^T(0)$ is assumed to be $1/3$ of the plasma flow, where the plasma flow is assumed to be half of the blood flow from the afferent arteriole. Therefore, we assume $Q_{i,j}^T(0) = \frac{Q_{i,j}}{6}$ in the first round. From the second round, $Q_{i,j}^T(0)$ will be calculated from the nephron model in Eq. (2).

Given the above boundary conditions, the system of linear equations can be properly defined as:

$$\begin{cases} P_n = P_{art}, & n = n_{in} \\ P_n = 0, & n \in N_{terminal} \\ \left(\frac{1}{R_{p,n}} + \sum_{i \in B_{v/p}} \frac{1}{R_{n,i}} \right) P_n & \\ - \left(\frac{P_p}{R_{p,n}} + \sum_{i \in B_{v/p}} \frac{P_i}{R_{n,i}} \right) = 0, & \text{otherwise} \end{cases} \quad (44)$$

where n_{in} denotes the single root node and $N_{terminal}$ denotes the set of terminal nodes of the vascular tree. The resistance term in Eq. (44) is defined as

$$R_{i,j} = \begin{cases} \frac{8\mu l_{i,j}}{\pi r_{i,j}^4}, & j \notin N_{terminal} \\ \frac{8\mu l_{i,j}}{\pi r_{i,j}^4} + R_E + R_{PC}, & \text{otherwise} \end{cases} \quad (45)$$

where R_E and R_{PC} are constant.

The system of linear equations defined in Eq. (44) can be easily expressed in the matrix form

$$AV = b \quad (46)$$

where the values in the sparse matrix A and the vector b is determined by Eq. (44). V can thus be solved by inverting

$$V = A^{-1}b \quad (47)$$

The above equation solves the pressure at every node, and we can then compute the flow Q_e at every vessel segment $e = (i, j)$ by Poiseuille's law

$$Q_e = \frac{P_i - P_j}{R_e} \quad (48)$$

For $j \in N_{terminal}$, the flow Q_t computed by Kirchhoff's law represents the equivalent flow of the two parts, the real flow $Q_{i,j}$ associated with the afferent arteriole (i, j) can thus be recovered by equal pressure drop property

$$\begin{aligned} Q_t(R_{i,j} + R_E + R_{PC}) &= Q_{i,j}R_{i,j} + (Q_{i,j} - Q_{i,j}^T(0))R_E \\ &\quad + Q_{i,j}R_{PC} \\ \implies Q_{i,j} &= Q_t + Q_{i,j}^T(0) \frac{R_E}{R_{i,j} + R_E + R_{PC}} \end{aligned} \quad (49)$$

Parameters

The wall thickness, $h(r_v)$ in the elastic tension term T_e , is varied with the radius r_v so that the cross-sectional area of the vessel wall remains constant. Specifically, given the initial value of $r_0 = 10 \mu m$ and $h_0 = 2 \mu m$, $h(r_v)$ is expressed by a parabolic function

$$\begin{aligned} \pi(h(r_v) + r_v)^2 - \pi r_v^2 &= \pi(h_0 + r_0)^2 - \pi r_0^2 \\ \implies h(r_v)^2 + 2r_v h(r_v) - (h_0^2 + 2r_0 h_0) &= 0 \end{aligned} \quad (50)$$

which can be solved analytically by quadratic formula (omitting the negative root):

$$\begin{aligned} h(r_v) &= \frac{-2r_v + \sqrt{4r_v^2 + 4((r_0 + h_0)^2 - r_0^2)}}{2} \\ &= -r_v + \sqrt{r_v^2 + 2h_0 r_0 + h_0^2} \end{aligned} \quad (51)$$

Other parameter details are given in the tables below.

Competing interests

No competing interest is declared.

Author contributions statement

O.S. and S.D. initiated and coordinated the project. O.S. and N.H.H.R. provided the physiological background and participated in the study design. P.X. conducted the experiments. O.S. and N.H.H.R. analysed the results. All authors participated in the discussion of the research and writing the paper.

Table 1. Afferent arteriole parameters.

Parameter	Value	Explanation
σ_e	25.56 dyn cm ⁻²	elasticity coefficient
k_e	17,304 cm ⁻¹	elasticity coefficient
r_e	7 μ m	set point for elastic response
σ_m	7.23×10^5 dyn cm ⁻²	muscle contraction coefficient
k_m	8.75×10^6 cm ⁻²	muscle contraction coefficient
r_m	12.5 μ m	set point for muscle contraction
G	0.06 cm dyn ⁻¹	gain of the myogenic response
T_0	84.7 dyn cm ⁻¹	set point for tension
$C_{1/2}$	44 mM	inflection point

Table 2. Glomerulus parameters.

Parameter	Value	Explanation
C_A	57 g/l	Protein concentration in the beginning
Ht_A	0.5	hematocrit
K_f	2.5 nl min ⁻¹ mmHg ⁻¹	filtration coefficient
R_E	0.209 mmHg min nl ⁻¹	resistance of the efferent arteriole
R_{PC}	0.0702 mmHg min nl ⁻¹	resistance of postcapillary system
a	0.1631 mmHg · l · g ⁻¹	constant in computing osmotic pressure $\pi(c)$
b	0.00294 mmHg · l ² · g ⁻²	constant in computing osmotic pressure $\pi(c)$

Acknowledgments

The authors thank the anonymous reviewers for their valuable suggestions.

Table 3. Tubule parameters.

Parameter	Value	Explanation
K_m	20 mM	constant in Michaelis-Menten kinetics
L_V	$2 \times 10^{-5} \text{ cm}^2 \text{ l osmol}^{-1} \text{ sec}^{-1}$	water permeability
n_S	2 mol osmol^{-1}	number of osmoles per mole of solute
L_S	$3.4 \times 10^{-7} \text{ cm}^2 \text{ sec}^{-1}$	constant in diffusive term
V_{\max}	$0.65 \text{ and } 1.0 \times 10^{-7} \text{ mmol cm}^{-1} \text{ sec}^{-1}$	constant in Michaelis-Menten kinetics
α	$1.65 \times 10^{-2} (\text{nl min}^{-1} \text{ mmHg}^{-5})^{1/4}$	empirical constant in boundary pressure
β	$0.924 (\text{nl min}^{-1} \text{ mmHg}^{-1})^{1/4}$	empirical constant in boundary pressure
θ	1.3 cm^{-1}	constant in tubular fluid reabsorption
κ	$33.6 \text{ nl min}^{-1} \text{ cm}^{-1}$	constant in tubular fluid reabsorption
r_0	$12 \times 10^{-4} \text{ cm}$	Tubular radius in proximal
r_0	$10 \times 10^{-4} \text{ cm}$	Tubular radius in loop of Henle

Table 4. Initial conditions: Tree structure.

Parameter	Value	Explanation
r_v	$10.08 \pm 0.14 \mu\text{m}$	afferent arteriole radius
h	$2 \mu\text{m}$	afferent arteriole wall thickness
T	$81.13 \pm 5.83 \text{ dyn cm}^{-1}$	tension distribution
P_{art}	100 mmHg	inlet pressure
P_{GC}	$58.04 \pm 4.46 \text{ mmHg}$	pressure distribution at the end of afferent arteriole
Q_A	$117.02 \pm 8.97 \text{ nl min}^{-1}$	afferent arteriole plasma flow distribution
$Q_T(0)$	$39.01 \pm 2.99 \text{ nl min}^{-1}$	filtration rate distribution

Table 5. TGF response parameters for three populations of nephrons.

Parameter	Short	Intermediate	Long	Unit	Explanation
k	0.85	0.07	0.06	l/mmol	empirical constant
ψ	0.182	0.2583	0.2184	N/A	dynamic range of the response
L_S	3.4×10^{-7}	3.8×10^{-7}	5.8×10^{-7}	cm^2/sec	NaCl permeability
ξ_{\max}	0.091	0.1183	0.1456	N/A	maximal TGF response

Table 6. Position of tubule segments (in cm) for three populations of nephrons. In all cases, macula densa is located between the cortical and postmacular part of thick ascending limb.

Position	Short	Intermediate	Long
Proximal	1	1	1
Descending	0.3	0.55	0.8
Thin Ascending	0	0.25	0.5
Thick Ascending (Medullary)	0.3	0.3	0.3
Thick Ascending (Cortical)	0.2	0.1	0.02
Thick Ascending (Postmacular)	0.1	0.1	0.1
Transition zone	0.05	0.05	0.05

Appendix C

Paper C

Extremely weakly-supervised blood vessel segmentation with physiologically based synthesis and domain adaptation

Peidi Xu¹, Blaire Lee², Olga Sosnovtseva², Charlotte Mehlin Sørensen², Kenny Erleben¹, and Sune Darkner¹

¹ Department of Computer Science, University of Copenhagen, Denmark
peidi@di.ku.dk

² Department of Biomedical Sciences, University of Copenhagen, Denmark

Abstract. Accurate analysis and modeling of renal functions require a precise segmentation of the renal blood vessels. Micro-CT scans provide image data at higher resolutions, making deeper vessels near the renal cortex visible. Although deep-learning-based methods have shown state-of-the-art performance in automatic blood vessel segmentations, they require a large amount of labeled training data. However, voxel-wise labeling in micro-CT scans is extremely time-consuming, given the huge volume sizes. To mitigate the problem, we simulate synthetic renal vascular trees physiologically while generating corresponding scans of the simulated trees by training a generative model on unlabeled scans. This enables the generative model to learn the mapping implicitly without the need for explicit functions to emulate the image acquisition process. We further propose an additional segmentation branch over the generative model trained on the generated scans. We demonstrate that the model can directly segment blood vessels on real scans and validate our method on both 3D micro-CT scans of rat kidneys and a proof-of-concept experiment on 2D retinal images. Code and 3D results are available at ³

Keywords: Blood vessel · Renal vasculature · Semantic segmentation · Physiological simulation · Generative model · Domain adaptation.

1 Introduction

The vasculature in each organ has a characteristic structure tailored to fulfill the particular requirements of the organ. The renal vasculature serves as a resource distribution network and plays a significant part in the kidney’s physiology and pathophysiology. Not only does it distribute blood to individual nephrons and regulates the filtration of blood in the kidney, but it also allows neighboring nephrons to interact through electrical signals transmitted along the vessels as a communication network [16]. Automatic segmentation of renal blood vessels from medical scans is usually the essential first step for developing realistic computer simulations of renal functions.

³ <https://github.com/diku-dk/RenalVesselSeg>

General deep-learning based segmentation of blood vessels: Deep learning models have been widely used for automatic blood vessel segmentations and have shown state-of-the-art performances applied on lungs, liver, and eyes [11,5,12,27]. However, only a few efforts were made for renal blood segmentation. Recently, He et al. proposed Dense biased networks [9] to segment renal arteries from abdominal CT angiography (CTA) images by fusing multi-receptive fields and multi-resolution features for the adaptation of scale changes. However, the limited resolution of CTA images only allows the models to reach interlobar arteries and enables the estimation of blood-feeding regions, which is useful for laparoscopic partial nephrectomy but not for analyzing realistic renal functions. Therefore, there is a need for imaging with higher resolution, e.g., micro-computed tomography (micro-CT) scans.

Micro-computed tomography and related deep-learning works: Micro-CT shares the same imaging principle as conventional clinical CT, but with a much smaller field of view, such that microscale images of high spatial resolution can be obtained [20]. Micro-CT scans are commonly used to study various microstructures, including blood vessels [1]. Few existing research on the auto-segmentation of micro-CT scans focuses on segmenting internal organs of the heart, spinal cord, right and left lung [15], and blood vessels on colorectal tissue [19] using either nn-UNet [10] or variant 3DUNet [21,4]. There is, however, no prior work in segmenting vasculatures in organs like kidneys from micro-CT scans. Crucially, most of the above deep learning methods require a large number of label maps to train the segmentation network. Manual labeling of micro-CT scans is extremely time-consuming given the huge volume size. Therefore, in our case, we do not have any clean label maps to train a segmentation model.

Synthetic training data for blood vessel segmentation: Transfer learning from artificially generated data is one possible technique to train deep learning models in a data scarcity setting. The process involves pre-training models on synthetic data, which are then fine-tuned on a small set of labeled real data. In medical image segmentation, this strategy has been widely applied to tumor segmentations [13]. Since blood vessels do follow certain physiological and anatomical properties, e.g., Murray’s law [18], this approach has also been applied to train segmentation models for mouse brain vasculature with physiologically synthesized vessels [25]. However, these works only pre-train the models on synthetic data and still require real labeled data for fine-tuning. Recently, Menten et al. [17] synthesize retinal vasculature and then emulate the corresponding optical coherence tomography angiography (OCTA) images. They show that a UNet trained on these emulated image-label pairs can directly segment real OCTA images. However, the way they generate scans from synthesized labels is completely explicit, which includes a series of physics-based image transformation functions that emulate the image acquisition process (e.g., OCTA). These functions clearly require expert knowledge and do not translate to micro-CT settings.

Generative models for domain adaptation: In practice, a relatively large number of unlabeled scans are usually available. Thus, a more general way to generate scans without emulating image acquisition explicitly is to utilize these

unlabeled scans via a generative model, i.e., domain adaptation. Medical image segmentation with domain adaptation is an active research area, and a popular method is Generative Adversarial Networks (GANs). In particular, CycleGAN [29] has been used to perform domain adaptation for medical image segmentation such as liver and tumor segmentation in CT images, where the goal is to train a segmentation model on one domain and then apply it to another domain with missing or scarce annotated data [2,8]. Recently, Chen et al. [3] segment cerebral vessels in 2D Laser speckle contrast imaging (LSCI) images using public fundus images with segmentation labels as the source domain and a CycleGAN for domain adaptation. The ability of CycleGAN to perform translation without paired training data makes it a powerful tool for domain adaptation.

Our contribution: We propose a framework with two main components: 1) a physiology-based simulation that synthesizes renal vascular trees and 2) a generative model with an additional segmentation branch that adapts the synthesized vascular trees to real scans while performing segmentation simultaneously. For 1), we extend the work [26] from physiologically synthesizing renal arterial tree to venous tree. Since a small prebuilt tree needs to be manually provided in the initialization step of the process, we call our method *extremely weakly-supervised*. For 2), we aim to “emulate” corresponding scans using CycleGANs. Specifically, we follow the idea in [3] to train a vessel segmentation network over the output from CycleGAN while extending to 3D on kidney micro-CT images. Notably, although 3D CycleGAN has been adopted for segmenting brain tissues and heart chambers [2,8,28], no similar work exists on subtle structures like blood vessels in 3D. Moreover, these works still require scans or segmentations from other sources, modalities, or time points as the source domain. Instead, our source domain is purely physiologically synthesized vascular trees. We show that our combined model can directly segment blood vessels in real scans from the target domain and demonstrate the validity of our approach to segmenting vasculatures both on 3D micro-CT scans of rat kidneys and 2D retinal images.

2 Method

2.1 Physiologically-based simulation of renal blood vessels

Constraint Constructive Optimization (CCO) [22] and its variant Global Constructive Optimization (GCO) [7] are widely used fractal-based methods that simulate the growth of vascular trees. These methods turn tree growth into an optimization problem based on the biological, physiological, and hemodynamic factors involved in the process. Here, the vascular tree is modeled by a directed acyclic graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ where \mathcal{V} is a set of nodes in the two endpoints of each vessel centerline with its coordinates in Euclidean space as node features, and \mathcal{E} is a set of directed edges representing each vessel segment as a cylindrical tube with its radius as edge features. Boundary conditions such as terminal radius and flow distributions are imposed to represent physiologic conditions. The algorithms then find a vascular tree that minimizes the system’s overall cost function while fulfilling several constraints.

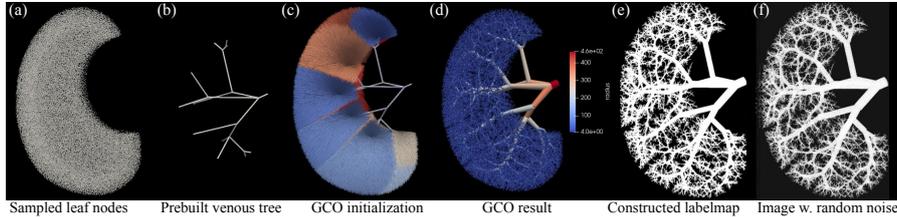


Fig. 1: Physiologically based vessel synthesizing pipeline. Details are given in [26]. The last two subfigures are shown with maximum intensity projection (MIP).

Here we follow [26], which adopts GCO as the backbone model for generating the renal vascular trees with optimal branching structures by performing multi-scale optimizations through iterating several operations such as splitting and relaxation [7]. However, instead of the arterial tree presented in [26], we only focus on the venous tree because it constitutes most of the vessel foreground. In summary, veins follow a similar pattern to arteries but are thicker, which is accomplished by sampling more terminal nodes with large radii (Fig. 1a). Detailed modifications over boundary conditions from arterial to venous trees are given in supplementary material, including tuning the weighting factors of the loss defined in [26]. Together with inherent randomnesses in the GCO process itself, the generated tree (Fig. 1d) will look different each run. This enables a variety of synthesized vascular trees to train the later deep-learning model.

Note that though a prebuilt tree $\mathcal{G}_0 \equiv (\mathcal{V}_0, \mathcal{E}_0)$ is required to guide the GCO process as noted in [26], \mathcal{G}_0 involves less than 20 nodes, as shown in Fig. 1b, which can be manually selected. This node selection process should take much less time than a voxel-wise manual annotation of the whole blood vessels. Therefore, we call our pipeline *extremely weakly-supervised* with a partially annotated tree structure but without any real segmentation label maps.

2.2 CycSeg: CycleGAN with additional segmentation branch

To create a synthetic image dataset, the reconstructed vascular tree structures $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ (Fig. 1d) are then remapped to 3D binary label maps (Fig. 1e) by voxelization, the detail of which is given in [26]. We then generate the corresponding gray-scale synthetic images (Fig. 1f) by simply assigning vessel foreground and background with random integers in $[128, 255]$ and $[0, 127]$ respectively. Of course, a segmentation model trained on these image-label pairs (Fig. 1f & e) will not work on real scans because of this oversimplified scan construction.

In order to adapt corresponding scans that emulate the micro-CT acquisition process out of the label maps from the previous step, unlabeled real micro-CT scans are utilized to train a generative model. Our backbone model is CycleGAN [29], which we extend to 3D while integrating an additional segmentation branch with segmenter S using a standard 3D UNet[4], as shown in Fig. 2(Left). We refer to our model as CycSeg later. The only modification we make in 3D is that

32 filters are used in the first layer instead of 64 to ease the computational load. Please refer to the supplementary material for detailed model architectures.

We strictly follow the original design [29] while extending to 3D for the loss functions definition and training procedure. Briefly, the two-way GAN loss $\mathcal{L}_{GAN}(G_{A \rightarrow B}, D_B)$ and $\mathcal{L}_{GAN}(G_{B \rightarrow A}, D_A)$ is to encourage the generator network to produce samples that are indistinguishable from real samples. Besides, cycle-consistency loss $\mathcal{L}_{cyc}(G_{A \rightarrow B}, G_{B \rightarrow A})$ is to assure the property $a \sim p_{\text{data}}(a) : G_{B \rightarrow A}(G_{A \rightarrow B}(a)) = a$ and $b \sim p_{\text{data}}(b) : G_{A \rightarrow B}(G_{B \rightarrow A}(b)) = b$. In our case, $p_{\text{data}}(a)$ and $p_{\text{data}}(b)$ denotes the real micro-CT scans distribution and the synthetic scans distribution from the physiologically generated trees, respectively. A final identity loss $\mathcal{L}_{id}(G_{A \rightarrow B}, G_{B \rightarrow A})$ is to stabilize the two generators. Please refer to [29] for the exact definition and computation of the above losses.

As shown in Fig. 2, instead of being trained on the synthesized scans (real B) directly, the segmenter (S) is trained on the adapted output (fake A) from the generator $G_{B \rightarrow A}$. Because the generator $G_{B \rightarrow A}$ learns to adapt realistic noise to the synthesized scans from unlabeled real scans (real A in Fig. 2), the segmenter (S) trained on the output from $G_{B \rightarrow A}$ will be able to segment unseen real scans. This introduces the segmentation loss $\mathcal{L}_{\text{seg}}(S, G_{B \rightarrow A})$. Thus, the final loss \mathcal{L}_{tot} is defined as

$$\begin{aligned} \mathcal{L}_{\text{tot}} &= \mathcal{L}(G_{A \rightarrow B}, G_{B \rightarrow A}, D_A, D_B, S) \\ &= \mathcal{L}_{GAN}(G_{A \rightarrow B}, D_B) + \mathcal{L}_{GAN}(G_{B \rightarrow A}, D_A) \\ &\quad + \lambda_1 \mathcal{L}_{cyc}(G_{A \rightarrow B}, G_{B \rightarrow A}) + \lambda_2 \mathcal{L}_{id}(G_{A \rightarrow B}, G_{B \rightarrow A}) + \lambda_3 \mathcal{L}_{\text{seg}}(S, G_{B \rightarrow A}) \end{aligned} \quad (1)$$

Here $\mathcal{L}_{\text{seg}}(S, G_{B \rightarrow A})$ is an unweighted combination of a dice loss and standard cross-entropy loss. Specifically, the segmenter S takes the output from the generator $G_{B \rightarrow A}$ as input. Thus, given the physiologically generated label map g and its corresponding synthetic gray-scale image $x \sim p_{\text{data}}(b)$, the segmenter outputs $p = S(G_{B \rightarrow A}(x))$, and $\mathcal{L}_{\text{seg}}(S, G_{B \rightarrow A})$ is defined as

$$\mathcal{L}_{\text{seg}}(S, G_{B \rightarrow A}) = \frac{1}{N} \left(- \sum_i^N g_i \log p_i + 1 - \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i + \sum_i^N g_i} \right) \quad (2)$$

where N is the total number of voxels in each sampled 3D patch.

We follow [29] by setting the weights $\lambda_1 = 10$ and $\lambda_2 = 5$ while setting $\lambda_3 = 3$ experimentally. Note that although $G_{B \rightarrow A}$ is one of the input models to compute $\mathcal{L}_{\text{seg}}(S, G_{B \rightarrow A})$, all the CycleGAN components including $G_{B \rightarrow A}$ are frozen when training the segmenter S by blocking backpropagation. Moreover, since the generator, discriminator, and segmenter are trained alternately, λ_3 does not strongly impact the training and only affects early stopping.

During inference, all CycleGAN components are discarded, while real scans (domain A) are directly passed to segmenter S to output segmentation maps.

3 Experiments and Results

Dataset: The kidney cast was prepared as described in [1] in agreement with approved protocols (approval granted from the Danish Animal Experiments Inspec-

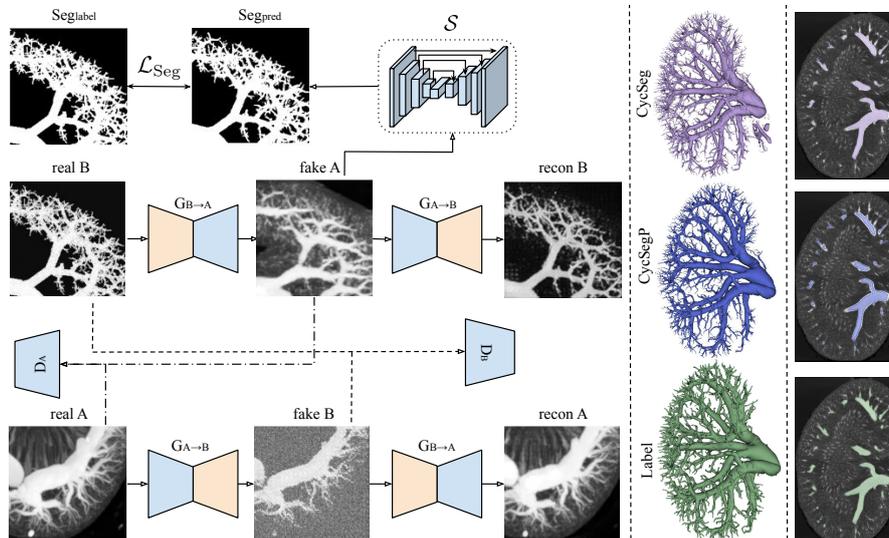


Fig. 2: Left: An illustration of CycleGAN with an additional segmentation branch working on adapted (fake) domain A images by $G_{B \rightarrow A}$. All 3D patches are shown with MIP. Computations for \mathcal{L}_{GAN} , \mathcal{L}_{id} , \mathcal{L}_{cyc} are not shown for simplicity. Middle: An example result in 3D. Right: A sample slice overlaid with segmentation.

torate under the Ministry of Environment and Food, Denmark). The rat kidneys were *ex vivo* scanned in a ZEISS XRadia 410 Versa μ CT scanner with an isotropic voxel size of $22.6 \mu\text{m}$ [1], with a fixed dimension of $1000 \times 1024 \times 1014$. To ease the computational load, scans are auto-cropped to $(692 \pm 33) \times (542 \pm 28) \times (917 \pm 28)$ by intersected bounding cubes from Otsu’s thresholding in each dimension [26]. Here, we use 7 unlabeled scans (domain A) for training and 4 labeled scans for testing. The synthesized dataset (domain B) has 15 image-label pairs for tuning parameters used in GCO and running multiple times as discussed in Section 2.1. **Pre-processing:** Each generated patch is only preprocessed by simple min-max normalization $X_{scale} = (x_i - x_{min}) / (x_{max} - x_{min})$.

Experimental Setup and Training process: The network is implemented in PyTorch and trained on NVIDIA A100 with a batch size of 1 and patch size of 208 for 200 epochs. All three components are optimized using the Adam optimizer with the same learning rate of 2×10^{-4} and reduced by 1% for every epoch. We apply early stopping if \mathcal{L}_{tot} (cf. Eq.(1)) of ten consecutive epochs does not decrease. Training takes approximately three days to reach convergence, while segmentation during inference takes only around two minutes per scan.

3.1 Results

As shown in Fig. 2, the CycleGAN successfully adapts realistic noise during micro-CT acquisition to the synthesized images (from real B to fake A), while a

Table 1: Segmentation result on private renal and public retina dataset (in %).

Training data		Model	Acc	DICE	clDICE
Renal	Synthetic label + Raw scan	CycSeg	99.1±0.1	74.3±1.7	64.7±5.0
		CycSegP	99.1±0.1	74.9±1.9	71.3±4.4
Retinal	CHASE label + DRIVE image	CycSeg	96.1±0.1	74.8±0.4	76.3±0.1
	DRIVE label + DRIVE image	UNet2D	96.5±0.1	79.6±0.5	78.9±0.8

UNet trained over the adapted image with the corresponding synthesized label maps can segment real micro-CT scans. Although the ex-vivo scan separates the organ of interest from other parts, the segmentation of venous trees is still challenging due to various noises during the micro-CT acquisition and efficacy of the contrast media. Despite of few false positives, the 3D results in Fig. 2 show a smooth and clear venous tree structure, which indicates that the segmenter is trained to recognize veins from noises in real scans by the adapted images from $G_{B \rightarrow A}$. Since the result should anatomically be a connected tree, we also apply a simple connected component post-processing to the model’s output (CycSegP), which removes the floating points and produces visually better results.

Table 1 shows the quantitative evaluation of the segmentation performance on the test set of four labeled scans, including accuracy, DICE score, and the topology-aware centerline DICE (clDICE) [23]. Although the test data size is too small to do a train-test split for training fully-supervised segmentation models, we believe that the visual inspection in Fig. 2 and relatively high quantitative results in Table 1 over 3D vessel segmentation task demonstrate the potential of our method in segmenting and building 3D renal models from micro-CT scans.

As a further qualitative evaluation, Fig. 3 shows one test sample with ground truth and our segmentation result (CycSeg) with MIP. These small vessels near the renal cortex are not labeled by human experts since manually reaching such details is too challenging and can take months to label a single scan. On the contrary, the small vessels are successfully recovered in our results since the model is trained on physiologically generated vascular trees that can reach as deep as possible. This is a major rationale behind creating a synthetic renal vascular dataset. Therefore, our results have potentially significantly better performance than those reported in Table 1 because these small vessels are currently regarded as false positives. Similar behavior has also been noticed in [17].

3.2 Proof-of-concept on retinal blood vessel segmentation

We acknowledge that the quantitative analysis above may not be thorough with four test data. Thus, we conduct a proof-of-concept on a 2D retinal blood vessel segmentation task. We follow the previous experimental setups but with a patch size of 256 and batch size of 32 due to the ease of computational load in 2D.

Dataset and domain construction: We adopt the DRIVE dataset [24] as the target domain, which includes 40 digital fundus images captured by a Canon CR5 3CCD camera with a resolution of 584×565 pixels. However, the 3D vascular

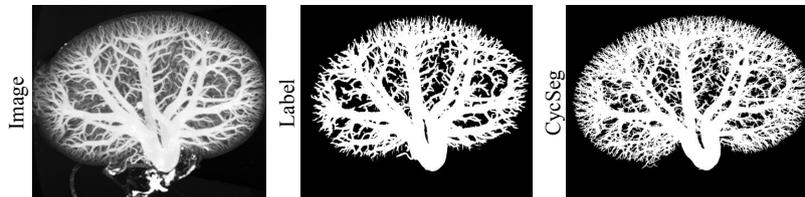


Fig. 3: An example test sample shown with MIP. Small vessels near the cortex are not labeled but correctly detected by CycSeg.

structure will not be physiologically correct when projecting to 2D images, and some works argue against compliance with Murray’s law in retinal blood vessels [14]. Thus, we do not physiologically synthesize retinal blood vessels but only focus on the validity of the segmentation power with domain adaptation. Thus, we directly adopt label maps from another dataset (CHASE [6]) as the source domain. Specifically, we use 20 from 40 images of the DRIVE dataset without label maps to form the target domain (real A), together with 28 label maps from the CHASE dataset to form the source domain (real B) in analog to Fig. 2. These unpaired data are used to train the CycSeg jointly. Note that DRIVE label maps and CHASE images are discarded. The remaining 20 labeled images from DRIVE are used for testing.

An illustration of retinal domain adaptation and visual results in analog to Fig. 2 is in the supplementary material. As an ablation study, we adopt the same train-test split to directly train a fully-supervised UNet [21] using real image-label pairs from the DRIVE training set. From Table 1, the fully-supervised UNet still outperforms domain adaptation, but the difference is acceptable, as our goal is not to outperform the state-of-the-art but to propose a pipeline that can do segmentation without any labeled images from the target domain.

4 Conclusion

We have presented a pipeline that segments blood vessels from real scans without any manually segmented training data. The pipeline first synthesizes label maps of renal vasculatures using a physiologically based model and then generates corresponding scans implicitly via a 3D CycleGAN from unlabeled scans. Simultaneously, an additional segmentation branch on top of CycleGAN enables the segmentation of blood vessels on real scans. This removes any need for expert knowledge of scanning settings compared to [17]. We believe our pipeline can crucially reduce annotations needed to segment blood vessels and easily adapt to other organs or modalities.

Since segmentation is the final objective, the intermediate image adaptation task is only visually inspected. Future work could include numerical tests like image structure clustering and Amazon Mechanical Turk perceptual studies [29]. A modification over [26] is necessary to model the pair-wise coupling of veins

and arteries and to train an Artery/Vein multi-class segmentation model in the future, which is far beyond the scope of this work. Nonetheless, more clean ground truth labels should enable a more thorough validation and benchmark test with standard segmentation models.

References

1. Andersen, S.B., Taghavi, I., Kjer, H.M., Sogaard, S.B., Gundlach, C., Dahl, V.A., Nielsen, M.B., Dahl, A.B., Jensen, J.A., Sørensen, C.M.: Evaluation of 2d super-resolution ultrasound imaging of the rat renal vasculature using ex vivo micro-computed tomography. *Scientific reports* **11**(1), 1–13 (2021)
2. Bui, T.D., Wang, L., Lin, W., Li, G., Shen, D.: 6-month infant brain mri segmentation guided by 24-month data using cycle-consistent adversarial networks. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). pp. 359–362. IEEE (2020)
3. Chen, H., Shi, Y., Bo, B., Zhao, D., Miao, P., Tong, S., Wang, C.: Real-time cerebral vessel segmentation in laser speckle contrast image based on unsupervised domain adaptation. *Frontiers in Neuroscience* p. 1523 (2021)
4. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. pp. 424–432. Springer (2016)
5. Cui, H., Liu, X., Huang, N.: Pulmonary vessel segmentation based on orthogonal fused u-net++ of chest ct images. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part VI 22. pp. 293–300. Springer (2019)
6. Fraz, M.M., Remagnino, P., Hoppe, A., Uyyanonvara, B., Rudnicka, A.R., Owen, C.G., Barman, S.A.: An ensemble classification-based approach applied to retinal blood vessel segmentation. *IEEE Transactions on Biomedical Engineering* **59**(9), 2538–2548 (2012)
7. Georg, M., Preusser, T., Hahn, H.K.: Global constructive optimization of vascular systems (2010), https://openscholarship.wustl.edu/cse_research/36
8. Gilbert, A., Marciniak, M., Rodero, C., Lamata, P., Samset, E., Mcleod, K.: Generating synthetic labeled data from existing anatomical models: an example with echocardiography segmentation. *IEEE Transactions on Medical Imaging* **40**(10), 2783–2794 (2021)
9. He, Y., Yang, G., Yang, J., Chen, Y., Kong, Y., Wu, J., Tang, L., Zhu, X., Dillenseger, J.L., Shao, P., et al.: Dense biased networks with deep priori anatomy and hard region adaptation: Semi-supervised learning for fine renal artery segmentation. *Medical image analysis* **63**, 101722 (2020)
10. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods* **18**(2), 203–211 (2021)
11. Jia, D., Zhuang, X.: Learning-based algorithms for vessel tracking: A review. *Computerized Medical Imaging and Graphics* **89**, 101840 (2021)
12. Keshwani, D., Kitamura, Y., Ihara, S., Iizuka, S., Simo-Serra, E.: Topnet: Topology preserving metric learning for vessel tree reconstruction and labelling. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VI 23. pp. 14–23. Springer (2020)

13. Lindner, L., Narnhofer, D., Weber, M., Gsaxner, C., Kolodziej, M., Egger, J.: Using synthetic training data for deep learning-based gbm segmentation. In: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 6724–6729. IEEE (2019)
14. Luo, T., Gast, T.J., Vermeer, T.J., Burns, S.A.: Retinal vascular branching in healthy and diabetic subjects. *Investigative ophthalmology & visual science* **58**(5), 2685–2694 (2017)
15. Malimban, J., Lathouwers, D., Qian, H., Verhaegen, F., Wiedemann, J., Brandenburg, S., Staring, M.: Deep learning-based segmentation of the thorax in mouse micro-ct scans. *Scientific Reports* **12**(1), 1822 (2022)
16. Marsh, D.J., Postnov, D.D., Sosnovtseva, O.V., Holstein-Rathlou, N.H.: The nephron-arterial network and its interactions. *American Journal of Physiology-Renal Physiology* **316**(5), F769–F784 (2019)
17. Menten, M.J., Paetzold, J.C., Dima, A., Menze, B.H., Knier, B., Rueckert, D.: Physiology-based simulation of the retinal vasculature enables annotation-free segmentation of oct angiographs. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*. pp. 330–340. Springer (2022)
18. Murray, C.D.: The physiological principle of minimum work: I. the vascular system and the cost of blood volume. *Proceedings of the National Academy of Sciences* **12**(3), 207–214 (1926)
19. Ohnishi, T., Teplov, A., Kawata, N., Ibrahim, K., Ntiamoah, P., Firat, C., Haneishi, H., Hameed, M., Shia, J., Yagi, Y.: Three-dimensional vessel segmentation in whole-tissue and whole-block imaging using a deep neural network: Proof-of-concept study. *The American Journal of Pathology* **191**(3), 463–474 (2021)
20. Ritman, E.L.: Current status of developments and applications of micro-ct. *Annual review of biomedical engineering* **13**, 531–552 (2011)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*. pp. 234–241. Springer (2015)
22. Schreiner, W., Karch, R., Neumann, F., Neumann, M.: Constrained constructive optimization of arterial tree models. *Scaling in biology* **145**, 65 (2000)
23. Shit, S., Paetzold, J.C., Sekuboyina, A., Ezhov, I., Unger, A., Zhylyka, A., Pluim, J.P., Bauer, U., Menze, B.H.: cldice-a novel topology-preserving loss function for tubular structure segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 16560–16569 (2021)
24. Staal, J., Abramoff, M.D., Niemeijer, M., Viergever, M.A., Van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging* **23**(4), 501–509 (2004)
25. Todorov, M.I., Paetzold, J.C., Schoppe, O., Tetteh, G., Shit, S., Efremov, V., Todorov-Völgyi, K., Düring, M., Dichgans, M., Piraud, M., et al.: Machine learning analysis of whole mouse brain vasculature. *Nature methods* **17**(4), 442–449 (2020)
26. Xu, P., Holstein-Rathlou, N.H., Søgaard, S.B., Gundlach, C., Sørensen, C.M., Erleben, K., Sosnovtseva, O., Darkner, S.: A hybrid approach to full-scale reconstruction of renal arterial network. *Scientific Reports* **13**(1), 7569 (2023)
27. Zhang, J., Zhang, Y., Xu, X.: Pyramid u-net for retinal vessel segmentation. In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 1125–1129. IEEE (2021)
28. Zhang, Z., Yang, L., Zheng, Y.: Translating and segmenting multimodal medical volumes with cycle-and shape-consistency generative adversarial network. In: Pro-

- ceedings of the IEEE conference on computer vision and pattern Recognition. pp. 9242–9251 (2018)
29. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)

Appendix D

Paper D

Auto-segmentation of Hip Joints using MultiPlanar UNet with Transfer learning

Peidi Xu¹, Faezeh Moshfeghifar¹, Torkan Gholamalizadeh^{1,3}, Michael Bachmann Nielsen², Kenny Erleben¹, and Sune Darkner¹

¹ Department of Computer Science, University of Copenhagen, Denmark
peidi@di.ku.dk

² Department of Diagnostic Radiology, Copenhagen University Hospital, Denmark

³ 3Shape A/S, Copenhagen, Denmark

Abstract. Accurate geometry representation is essential in developing finite element models. Although generally good, deep-learning segmentation approaches with only few data have difficulties in accurately segmenting fine features, e.g., gaps and thin structures. Subsequently, segmented geometries need labor-intensive manual modifications to reach a quality where they can be used for simulation purposes. We propose a strategy that uses transfer learning to reuse datasets with poor segmentation combined with an interactive learning step where fine-tuning of the data results in anatomically accurate segmentations suitable for simulations. We use a modified MultiPlanar UNet that is pre-trained using inferior hip joint segmentation combined with a dedicated loss function to learn the gap regions and post-processing to correct tiny inaccuracies on symmetric classes due to rotational invariance. We demonstrate this robust yet conceptually simple approach applied with clinically validated results on publicly available computed tomography scans of hip joints. Code and resulting 3D models are available at: <https://github.com/MICCAI2022-155/AuToSeg>

Keywords: Segmentation · Finite Element modeling · Transfer learning

1 Introduction

Precise segmentation of medical images such as computed tomography (CT) scans, is widely used for generating finite element (FE) models of humans for patient-specific implants [2]. A requirement in generating FE models is a proper geometrical representation of the anatomical structures [9]. In our case, an *accurate* segmentation of the hip joint (HJ) should essentially detail the shape and boundaries of the femur and hip bones and identify the inter-bone cavities. The segmented geometries should be closed, non-intersecting, and without spikes. As manual segmentation is labor-intensive and time-consuming [9], automated segmentation tools are usually necessary to generate accurate FE models.

Convolutional Neural Networks with encoder-decoder structures are widely used for auto semantic segmentation, among which the most successful one is

the UNet structure [10]. The architecture uses skip connection on high-resolution feature maps in the encoding path to include more fine-grained information. Although more recent models are proposed on segmenting natural images, e.g., DeepLabV3+, UNet still provides some of the best segmentation results in medical images [1]. Therefore, the variation of UNet, e.g., 3D UNet, is a straightforward way to segment 3D medical data like CT scans and has shown its state-of-the-art performance [3]. Applying 3D convolutions directly to large 3D images may overflow memory. Therefore, 3D models are usually trained on small patches, which results in a limited field of view and subsequent loss of global information. As an alternative with far less memory usage, the MultiPlanar UNet (MPUNet) model was proposed by Perslev et al. [8] which uses a 2D UNet to learn representative semantic information.

Most studies on auto-segmentation of the HJs focus on designing more powerful neural networks that separate anatomical structures with little manual intervention [12,13]. These studies focus primarily on the bone morphology and not on the inter-bone gaps. The consequence is that although they reach fairly high Dice scores, the segmentation results are anatomically inaccurate and are unsuitable for generating HJ 3D models. This limits the usability of the existing deep learning models for FE simulations [7].

We require the deep learning models to provide anatomically correct segmentation of the bones and the existing gap in the HJ as shown in Fig. 1 [12,13]. Due to the limited number of accurate training data, we propose a deep learning-based strategy for enhancing publicly available poorly annotated scans using only a few accurately segmented data to learn an accurate model and in our case the gap regions in HJ. Besides using the idea of MultiPlanar, our backbone model is a standard UNet with batch normalization. Therefore, the proposed pipeline is both parameter and memory efficient.

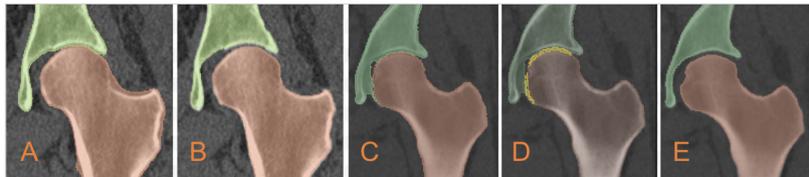


Fig. 1: Illustration of gap generation: Inferior ground truth of a training image from public dataset (A) and results by fine-tuned model (B). Results on a test image with model trained only on public dataset (C) with erroneous prediction detection (D) and fine-tuned (E)

To enforce the cartilage gap with few annotated data, we apply MPUNet with a dedicated loss function penetrated more on the gap regions combined with transfer learning and a post-processing step. Our framework uses an interactive learning pipeline involving pre-training MPUNet on a public dataset with inferior

HJ segmentation to learn general semantic features of the bones [5]. The model is then fine-tuned using a few highly accurate segmentation to learn the correct labeling of the gaps. We show that our proposed approach allows the model to learn the gap and generate anatomically accurate segmentation, using the pre-trained model and only four accurate segmentations for fine-tuning. Our work is validated on a set of HJs from which we construct FE models and report the Dice with the manually corrected segments used for biomechanical models.

2 Method

Our strategy for accurate HJ segmentation with very few accurate training images relies on the following: (i) we use the idea of MPUNet that segments 3D medical images using 2D models while preserving as much spatial information as possible by segmenting different views of the data. (ii) we use a relatively simple yet powerful backbone model for performing the segmentation to avoid overfitting and memory issues. (iii) we pre-train the model using publicly available datasets with poor labels, which are then fine-tuned with a very small set of accurately annotated data. (iv) we use a dedicated weighted distance loss to enforce the gap between the bones. (v) we introduce a post-processing step that solves the internal problem of MPUNet on images with symmetric features.

Model: As a baseline model, we use the MPUNet proposed by Perslev et al. [8] to segment the 3D HJs using 2D UNet while preserving as much 3D information as possible by generating views from different perspectives. During training, the model $f(x; \theta)$ takes a set 2D image slices of size $w \times h$, from different views, and outputs a probabilistic segmentation map $P \in R^{w \times h \times K}$ for K classes for each slice. Standard pixel-wise loss function is then applied for back-propagation. Our experiment uses a standard categorical cross-entropy loss augmented by the weighted distance map. We found no improvement using a class-wise weighted cross-entropy loss or the dice loss. In the inference phase, we run 3D reconstruction in each view separately over the segmentation results on all the parallel slices to get the volume back. This results in a volume probability map of size $m \times w \times h \times K$ for each view. Unlike original MPUNet [8] which suggests training another fusion model using validation data, we simply sum over the results (P) from different views followed by an argmax over last dimension to get the final label map. This strategy achieves good results on the validation data.

Transfer Learning: The accurate segmentation and fast convergence rely partially on pre-training the model using publicly available datasets with poor labeling, which is subsequently fine-tuned with a small set of accurate data. We detail two modifications that differ from standard transfer learning settings. First, we also transfer the weight in the last softmax layer for a much faster convergence because we work on exactly the same classes as before. Then, instead of freezing encoder and only fine-tuning decoder, it is necessary to explicitly learn encoder to detect the gap, as the gap must be encoded correctly first.

Weighted Distance Map: For the model to be fine-tuned to learn the gap between the bones, we enforce a voxel-wise weight-map $w(x)$ to the loss function

based on the distances to the border of the foreground classes. This strategy was initially suggested in the original UNet paper, which we employ in a modified version for 3D data [10,8]. We define $w(x)$ as follows,

$$w(x) = w_c(x) + w_0 \cdot e^{-\frac{(d_1(x)+d_2(x))^2}{2\sigma^2}} \quad (1)$$

where d_1 and d_2 denotes the distance to the border of the nearest foreground class and the second nearest foreground class respectively. We follow original UNet paper and set $w_0 = 10$ and $\sigma = 5$. $w_c : \Omega \rightarrow \mathcal{R}$ is used to balance the class frequencies, which we do not enforce, thus we set $w_c = 1$ for every c .

During fine-tuning, the corresponding slice of the 3D weight map is sampled together with the images and labels. We apply an element-wise multiplication of the weight map with the cross-entropy loss of predictions and labels on each pixel before reduction. Fig. 2 (left & middle) shows an example training slice. Note that we do not plot the prediction since it consists of multi-class probabilities.

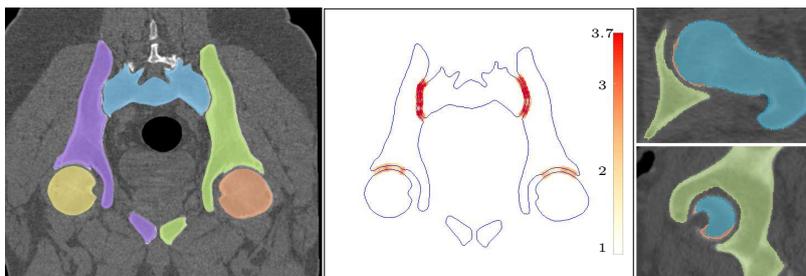


Fig. 2: (left) A sample training slice of true labels overlaid on top of raw image. (middle) Corresponding weight map computed with Eq. (1) overlaid on top of label boundaries. (right) Results of training with weight map calculated over eroded labels (orange contour), which shows a smoother and more complete contour near the boundaries than the results trained without erosion (blue contour).

We also notice that the model is prone to overfitting to the gap, producing a broader gap than usual if we assign higher weights only to the gap regions in Eq. (1). Instead, we would like to assign more weight to the boundaries around the gap to avoid false negatives. This is accomplished by applying a mathematical erosion to the labels over a ball with a radius of 3 voxels before calculating the weight map, as demonstrated in Fig. 2 (middle). To compensate for the increased value of $d_1 + d_2$ introduced by erosion, we double w_0 to 20.

Sampling Strategies: Sampling and interpolation are necessary to retrieve corresponding 2D slices from a 3D medical image viewed from a random orientation other than the standard RAS axes. We follow the idea in [8] by sampling pixel with dimension $d \in \mathcal{Z}^+$ on isotropic grids within a sphere of diameter $m \in \mathcal{R}^+$ centered at the origin of the scanner coordinate system in the physical scanner

space. We differ in that these two numbers are chosen as the 75 percentile across all axes and images during training but as maximum value during inference. This ensures both efficient training and complete predictions near the boundaries.

Post-Processing: Although MPUNet is both parameter and memory efficient, the model is trained on 2D slices with a possibly limited field of view near the boundaries. Furthermore, it is trained to segment the input viewed from different perspectives by sampling from planes of various orientations. This introduces some rotational invariance but makes it hard to distinguish between symmetric classes with very similar semantic features. For example, it is hard to be consistent with the left and right femurs when viewing the input from various perspectives. Therefore, some part of the left femur near the boundaries is mis-classified as the right femur respectively, and vice versa, as shown in Fig. 3.

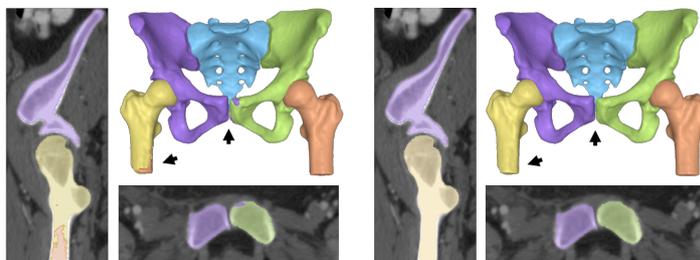


Fig. 3: Segmentation (left) with post-processing (right) where falsely predicted symmetric groups are recovered.

In order to solve this problem automatically, we propose a symmetric connected component decomposition. We only keep the largest connected component for each symmetric class pair while assigning the corresponding symmetric class value to all the other components. By doing this instead of just removing small components, those parts predicted as the left femur on the right femur are mapped correctly to the right femur, and vice versa. We then apply a standard connected component decomposition while keeping only the largest connected component for each foreground class to remove floating points (false positives).

We acknowledge that our post-processing is highly task-specific but could also be generalized to other segmentation tasks with symmetric classes that share similar semantic features and are disconnected from each other.

3 Data and Experiments

We use 35 CT scans from The Cancer Imaging Archive and crop the region of interest on the images to roughly cover the area around the HJs, including the sacrum, both hip bones, and both proximal femurs [4]. Each scan comprises $(415 \pm 47) \times (244 \pm 30) \times (266 \pm 29)$ voxels, with a voxel size of $(0.78 \pm 0.11) \times (0.77 \pm$

$0.1) \times (0.96 \pm 0.17) \text{ mm}^3$. For the pre-training step, we use 10 scans and their associated inferior segmentations from a publicly available dataset of *segmented CT pelvis scans with annotated anatomical landmarks* (CTPEL) [5,12]. We only use two scans with accurate segmentation to fine-tune the model in the first place. In the next step, two other unseen scans are used to get the segmentation results of the model. Then, we manually correct these two results and fine-tune the model again. The second fine-tuning process could be re-iterated, but four images is sufficient to obtain accurate results. We evaluate the segmentation results of our approach with minimal required fine-tuning data. A clinical expert evaluated the segmentation results of the 21 test cases.

Interactive Learning Setup: Using prior anatomical knowledge that each class should be disconnected by at least a certain distance, contradicting cases in the model output indicate false positives (collisions) on at least one of the classes. We thus apply another Euclidean transform over the output segmentation P such that each point in a predicted foreground class is mapped to the nearest distance to other foreground classes. We can then find those collision points set E by applying a threshold ϵ to the distance map, as shown in Fig. 1 (D).

$$E = \{x | P(x) \neq 0 \wedge d(x) \leq \epsilon\} \quad (2)$$

Since E only roughly captures the collision points, directly setting them to background will not be accurate and may introduce false negatives. However, the size of it ($|E|$) can be used as a metric for model performance without ground truth to decide when to terminate the interactive learning process. In our experiment where $\epsilon = 2$, the model without fine-tuning gives $|E| \approx 24803$, while fine-tuning with two and another two accurate data reduces $|E|$ to 1000 and 200 respectively.

Pre-processing: We pre-process the data by first filtering out all negative values in the volume because both bones and cartilages should have positive Hounsfield unit values. We then apply a standardization based on the equation $X_{\text{scale}} = (x_i - x_{\text{mean}}) / (x_{75} - x_{25})$, where x_{25} and x_{75} are the 1st and 3rd quartiles respectively. This removes the median and scales the intensity based on quartiles and is more robust to outliers. No other pre-processing is applied to avoid any manual errors that can easily propagate in a neural network.

Experimental Setup: The network is trained on NVIDIA GeForce RTX 3090 with a batch size of 10 using the Adam optimizer for 40 epochs with a learning rate of $1e-5$ and reduced by 10% for every two consecutive epochs without performance improvements. We apply early stopping if the performance of five consecutive epochs does not improve. Pre-training takes approximately one day, while fine-tuning takes about six hours to reach convergence.

Augmentations: We follow MPUNet by applying Random Elastic Deformations to generate images with deformed strength and smoothness [11] and assign a weight value of $1/3$ for the deformed samples during training [8].

4 Results

To have suitable geometries for FE models, the auto-segmentation framework must separate bones and generate accurate results near the boundaries, which

is essential for generating cartilage layers for HJ. Therefore, any standard evaluation metric such as the Dice score could be misleading. Hence, our results, including the bone outlines and the existing gap in the joints, are first validated by a senior consultant radiologist as our clinical expert.

The clinical expert initially scrolls through all the segmented slices to verify the bone contours and the gaps between the hip and femoral bones. Then, he verifies the anatomical shape and smoothness of the reconstructed 3D model. This procedure justifies our method in obtaining precise HJ geometries.

Fig. 3 illustrates the results of the fined-tuned model on the test set and demonstrates the effect of post-processing, where it shows that the misclassified regions in symmetric classes are successfully recovered by the post-processing step. With the distance weight applied to loss, the model can detail the gap accurately. The final result is accurate and requires little or no human intervention for subsequent simulation experiments, e.g., FE analysis. Results in 3D are available at GitHub Repo. As an example, we have generated the cartilage geometry on the segmented HJ with a method proposed by [6] to analyze the stress distributions as shown in Fig. 4. The results show a smooth stress pattern indicating that our method’s output is suitable for use in FE simulations.

	Dice \uparrow	GapDice \uparrow	HD(<i>#voxels</i>) \downarrow
Ours	98.63 \pm 0.56	96.47 \pm 1.60	3.67 \pm 1.13
NoPretrain	97.82 \pm 0.59	95.13 \pm 1.42	5.26 \pm 2.10
NoWeight	98.12 \pm 0.47	94.35 \pm 2.19	4.58 \pm 1.50
3DUNet	93.36 \pm 1.84	87.48 \pm 3.01	7.02 \pm 1.09
Ours(2)	97.59 \pm 0.74	95.19 \pm 1.14	5.18 \pm 2.08
NoPretrain(2)	90.80 \pm 9.29	91.13 \pm 8.53	11.20 \pm 7.19
NoWeight(2)	96.28 \pm 2.91	93.91 \pm 1.74	6.30 \pm 2.95

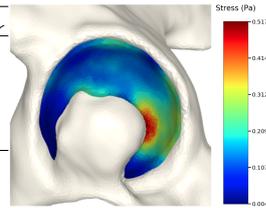


Fig. 4: Smooth von Mises stress pattern

Table 1: Test Results with various design choices

4.1 Numerical Validation and Ablation Study

Although numerical results could be misleading regarding the final FE simulations, we include them as a validation and ablation study of our several design choices. Table 1 shows the numerical validations on the test set, including nine images with manually corrected ground truth segmentations. We test the performance by varying one of the design choices each time while keeping the others fixed. (i) The strategy mentioned in Section 2 (ours), (ii) Training without using ten inaccurate public data (NoPretrain), (iii) Training without enforcing distance weight map (NoWeight), (iv) Using 3D UNet as the backbone (3DUNet). We also test and report the performance in the first stage when fine-tuned with only two manually corrected data except for (iv) because of its poor performance.

Besides the standard Dice score, we are especially interested in the surface and gap regions. Therefore, two more evaluation metrics are introduced. We

use Hausdorff distance (HD) as surface measurement by computing the largest distance between the result and the nearest point on the ground truth.

$$\text{HD}(P, Y) = \max(\max_{p \in P} \min_{y \in Y} \|p - y\|_2, \max_{y \in Y} \min_{p \in P} \|p - y\|_2) \quad (3)$$

We also propose a GapDice in Eq. (4) to measure the average Dice score between the segmentation result and the ground truth only around the gap regions. Given the segmentation results P and ground truth segmentation Y , we compute the Euclidean distance transformation map Y_d of Y , corresponding to the $d_1 + d_2$ term from Eq. (1). The gap region G is defined as the locations where $Y_d < \epsilon$. Dice score between P and Y is calculated in the standard way inside G . Here we choose $\epsilon = 10$ as we found it to be a good indicator of both the gap and boundary regions. Fig. 2 (middle) shows the region computed by eroded labels, which is also an indication of G . Please refer to GitHub Repo for generated G .

$$\text{GapDice}(P, Y) = \frac{2 * |P \cap Y \cap G|}{|P \cap G| + |Y \cap G|} \quad (4)$$

The results show that MPUNet (all the first three models) works significantly better than 3D UNet in a data scarcity setting. Our pipeline outperforms in all three metrics. Especially, although the difference of the Dice score is not significant in our fine tuned model with four manually corrected data, pretraining on inaccurate data and enforcing the weight map shows a significantly better GapDice score and HD, which is vital for further simulation. The benefit of pretraining is much clearer in the first round when fine-tuned with only two accurate data, which is crucial to have minimal manual work to be fine tuned again. We acknowledge that the ground truth for test data is manually modified over the results from our pipeline, giving a bias when comparing multiple models, but the general goal is to show that our pipeline suits well for further simulation.

5 Conclusion

We presented an auto-segmentation framework for accurate segmentation from CT scans considering the bone boundaries and inter-bone cavities. Our framework uses a modified MPUNet pre-trained on a public dataset with coarse segmentation and fine-tunes with very few data with accurate segmentation in a transfer and interactive learning setup. We demonstrate that our simple yet robust model can detail crucial features such as the gap where the cartilage resides.

This work is tested out on HJ CT scans and provides anatomically accurate segmentation, which has both been verified by a clinical expert and shown superior numerical results, reaching an overall Dice score above 98% and above 96% around gap regions. Our method can be used to enhance anatomically incorrect and poorly annotated datasets with a few accurately annotated scans. The FE analysis shows that the generated models produce smooth stress patterns without any geometry-related artifacts. Thereby, the segmentation result of this work can be used for generating FE models with little or no manual modifications.

Acknowledgements. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764644. This paper only contains the author’s views, and the Research Executive Agency and the Commission are not responsible for any use that may be made of the information it contains.

References

1. Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.
2. Xiaojun Chen, Lu Xu, Yiping Wang, Yongqiang Hao, and Liao Wang. Image-guided installation of 3d-printed patient-specific implant and its application in pelvic tumor resection and reconstruction surgery. *Computer Methods and Programs in Biomedicine*, 125:66–78, 2016.
3. Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.
4. Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26(6):1045–1057, 2013.
5. Bryan Connolly and Chunliang Wang. Segmented CT pelvis scans with annotated anatomical landmarks, 2019.
6. Faezeh Moshfeghifar, Max Kragballe Nielsen, José D. Tascón-Vidarte, Sune Darkner, and Kenny Erleben. A direct geometry processing cartilage generation method using segmented bone models from datasets with poor cartilage visibility, 2022.
7. Takashi Nishii, Nobuhiko Sugano, Yoshinobu Sato, Hisashi Tanaka, Hidenobu Miki, and Hideki Yoshikawa. Three-dimensional distribution of acetabular cartilage thickness in patients with hip dysplasia: a fully automated computational analysis of mr imaging. *Osteoarthritis and Cartilage*, 12(8):650–657, 2004.
8. Mathias Perslev, Erik Bjørnager Dam, Akshay Pai, and Christian Igel. One network to segment them all: A general, lightweight system for accurate 3d medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 30–38. Springer, 2019.
9. Sander Poelert, Edward Valstar, Harrie Weinans, and Amir A Zadpoor. Patient-specific finite element modeling of bones. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 227(4):464–478, 2013.
10. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
11. Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3, 2003.

12. Chunliang Wang, Bryan Connolly, Pedro Filipe de Oliveira Lopes, Alejandro F Frangi, and Örjan Smedby. Pelvis segmentation using multi-pass u-net and iterative shape estimation. In *International Workshop on Computational Methods and Clinical Applications in Musculoskeletal Imaging*, pages 49–57. Springer, 2018.
13. Alexander D Weston, Panagiotis Korfiatis, Kenneth A Philbrick, Gian Marco Conte, Petro Kostandy, Thomas Sakinis, Atefeh Zeinoddini, Arunnit Boonrod, Michael Moynagh, Naoki Takahashi, et al. Complete abdomen and pelvis segmentation using u-net variant architecture. *Medical physics*, 47(11):5609–5618, 2020.

Appendix E

Paper E

Received 28 August 2023, accepted 16 September 2023, date of publication 20 September 2023,
date of current version 25 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3317512

RESEARCH ARTICLE

Deep-Learning-Based Segmentation of Individual Tooth and Bone With Periodontal Ligament Interface Details for Simulation Purposes

PEIDI XU¹, TORKAN GHOLAMALIZADEH², FAEZEH MOSHFEGHIFAR¹, SUNE DARKNER¹,
AND KENNY ERLEBEN¹

¹Department of Computer Science, University of Copenhagen, 2100 Copenhagen, Denmark

²3Shape A/S, 1059 Copenhagen, Denmark

Corresponding author: Peidi Xu (peidi@di.ku.dk)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Center for Regional Development, The Scientific Ethics Committee, under Reference No. 21063693.

ABSTRACT The process of constructing precise geometry of human jaws from cone beam computed tomography (CBCT) scans is crucial for building finite element models and treatment planning. Despite the success of deep learning techniques, they struggle to accurately identify delicate features such as thin structures and gaps between the tooth-bone interfaces where periodontal ligament resides, especially when trained on limited data. Therefore, segmented geometries obtained through automated methods still require extensive manual adjustment to achieve a smooth and organic 3D geometry that is suitable for simulations. In this work, we require the model to provide anatomically correct segmentation of teeth and bones which preserves the space for the periodontal ligament layers. To accomplish the task with few accurate labels, we pre-train a modified MultiPlanar UNet as the backbone model using inferior segmentations, i.e., tooth-bone segmentation with no space in the tooth-bone interfaces, and fine-tune the model with a dedicated loss function over accurate delineations that considers the space. We demonstrate that our approach can produce proper tooth-bone segmentations with gap interfaces that are fit for simulations when applied to human jaw CBCT scans. Furthermore, we propose a marker-based watershed segmentation applied on the MultiPlanar UNet probability map to separate individual tooth. This has advantages when the segmentation task is challenged by common artifacts caused by restorative materials or similar intensities in the teeth-teeth interfaces in occurrence of crowded teeth phenomenon. Code and segmentation results are available at <https://github.com/diku-dk/AutoJawSegment>.

INDEX TERMS Cone-beam computed tomography, deep learning, finite element modeling, human jaws, instance segmentation, learning with limited data, semantic segmentation, transfer learning.

I. INTRODUCTION

Accurate segmentation of medical images of a human jaw, such as cone beam computed tomography (CBCT) scans, is crucial in creating patient-specific preoperative and predictive finite element (FE) models that improve the design of implants and treatments [30]. A key aspect in the

The associate editor coordinating the review of this manuscript and approving it for publication was Yi Zhang¹.

development of FE models is having a precise geometric representation of anatomical structures [21].

In the case of developing FE models of the human jaw, in addition to teeth and bone geometries, it is essential to model the connective tissue between them, called periodontal ligament (PDL). In general, PDL layer plays an important role in transferring load from teeth to the bone in orthodontic treatments, and when triggered with enough orthodontic forces, it results in bone remodeling [12]. As a result, accurate segmentations of human jaws must not only depict the shape

and boundaries of the involved teeth and bone structures but also preserve the space between them (see Fig. 1B and D) for the further modeling of PDLs [12].

Manual segmentation of the CBCT scans with the accurate geometrical representation of a human jaw's anatomies is labor-intensive and extremely time-consuming and depends on the scans' resolution and the annotator's expertise. In addition, it is especially challenging to accurately delineate the teeth and bone boundaries with relatively similar intensities to preserve the PDL from the CBCT scans. Hence, there is a need for automated segmentation tools that can generate accurate geometries for developing FE models.

Automatic segmentation methods commonly utilize Convolutional Neural Networks with an encoder-decoder architecture, of which the most effective is the UNet structure [22] which incorporates skip connections on high-resolution feature maps in the encoding stage to include more fine-grained information. Despite the development of newer models for natural image segmentation such as DeepLabV3+ [3] and transformer-based models [1], [25], [28], UNet remains one of the top performers in 3D medical image segmentation [14]. As a result, using a variation of UNet, such as 3D UNet, for segmenting 3D medical data like computed tomography scans is a straightforward approach that has demonstrated state-of-the-art performance [6], [14]. However, applying 3D convolutions directly to large 3D images may result in memory overflow. To mitigate this, 3D models are typically trained on small patches, which limits their field of view and causes the loss of global information. An alternative with lower memory usage is the MultiPlanar UNet (MPUNet) model proposed by Perslev et al. [20]. This model utilizes a 2D UNet to learn representative 3D semantic information by sampling slices from various orientations.

Most research in the field of auto-segmentation of human jaws aims on the neural networks designs that can accurately separate certain anatomical structures with minimal manual input [4], [5], [7], [9], [10], [15], [26], [27], [29], [31], [33]. Most of these works only focus on teeth segmentation [4], [5], [8], [9], [10], [11], [15], [29], [31], especially on the separation of individual tooth, while others only focus on the bone segmentation [26], [32], [33]. Although individual tooth segmentation is critical for computer-aided analysis towards clinical decision support and treatment planning, PDL layers cannot be retrieved from either tooth or bone segmentation alone, thus cannot be used to model the transferring load from teeth to the bone in orthodontic treatments [12]. Recently, Wang et al. [27] and Cui et al. [7] work on the multiclass segmentation of human jaws to simultaneously segment the bones (i.e., mandible and maxilla) and the teeth. Their models are either trained only over axial slices of CBCT scans [27] or trained on thousands of scans to reach a Dice score above 90% [7]. More crucially, their segmentations ignore the inter-bone gaps and thus are anatomically inaccurate. These anatomically inaccurate segmentations cannot be used to generate 3D models and limits the application in finite element simulations [12].

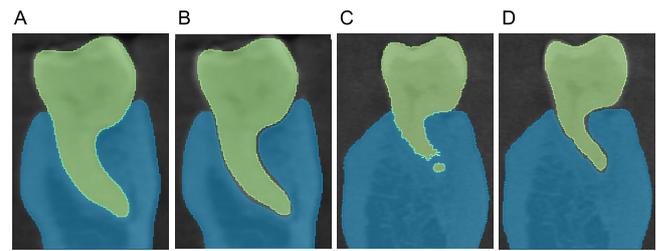


FIGURE 1. Illustration of gap generation. **A:** Inferior ground truth labels ignoring the space where the periodontal ligament resides. **B:** The accurate labels of the same patient that considers space for the periodontal ligament. **C:** Results of the proposed method on a test scan with model trained only on inferior dataset with no gap. **D:** Fine-tuned model with gap information.

Analog to Xu et al. [30], which accurately delineates the gap in hip joint segmentation for further cartilage simulation studies, we require the deep learning models to provide anatomically correct segmentation of human jaws which preserves the space for the PDL layers between teeth and mandibles as shown in Fig. 1. Our approach leverages a standard UNet with batch normalization as the backbone model and incorporates the concept of MultiPlanar [20] to integrate more volumetric features into the model and increase the model's efficiency. In addition, due to the difficulty of manual delineation of the PDL layers, we have very few anatomically accurate teeth-bone label maps of the CBCT scans that detail the gap where PDL resides. Our framework utilizes an interactive learning process to reinforce such gap with limited annotated data by pre-training the MPUNet on a dataset with subpar segmentation to gain a general understanding of the tooth and bone structures. Subsequently, the model is fine-tuned using just a few highly accurate segmentations with a specific loss function that penalizes more on the gap regions.

By combining these techniques, our proposed pipeline is capable of achieving accurate segmentation results that fill in the missing gaps between the tooth-bone interfaces where the PDL is located with few accurate training data, while being both memory and computationally efficient. Our findings are verified using a test set of CBCT scans, where we construct finite element models and numerically evaluate the segmentation performance with the manually corrected segments utilized in biomechanical models.

In addition to the gap generation process, a further task is to separate individual tooth. The UNet output consists of the segmentation of a single class of jaw bones, as well as a single class of teeth that contains all the teeth. Automatic segmentation of individual tooth is critical for computer-aided analysis towards clinical decision support and treatment planning, but this segmentation is further challenged by blurring the boundaries of neighboring teeth and metal artifacts. Therefore, a simple post-processing with Connected Component Decomposition (CCD) over the UNet output will not correctly separate the adjacent teeth, especially if the subject has crowded teeth or is in a biting position.

Deep-learning-based instance segmentation methods, e.g., Mask R-CNN, have shown state-of-the-art performance on 2D natural images [13]. These networks involve region proposals to generate bounding boxes around each instance, with one branch for box regression and object detection and another for semantic segmentation. Cui et al. exploited 3D Mask R-CNN as a base network to realize automatic tooth segmentation and identification from CBCT images [9]. However, region proposals in 3D are extremely time and memory-consuming and require a larger training set than semantic segmentation methods that only deal with voxel labeling. Many of the modifications by Cui et al. that make region proposal work on 3D cases rely on the teeth having similar structure and orientation, thus will fail with, for instance, wisdom teeth and, more fatally, after adding jaw bone classes. Moreover, a threshold of the confidence level on each proposed region needs to be selected manually during inference, which may completely miss an object or generate overlapping instances and hinder biomechanical modeling afterward.

Instead, since there is no occlusion in 3D images, a common way to accomplish instance segmentation in practice is to apply post-processing over semantic segmentation output. For example, Chen et al. proposed to apply watershed on the raw probability map of the output of semantic segmentation models [4]. Besides, they proposed to train a multi-task 3D VNet that learns both the teeth region and the teeth surfaces to gather more information about teeth and better separate neighboring teeth [17]. However, the dense skip connections in VNet and multi-task learning severely increase computational overhead. We follow the same idea of separating individual tooth by applying watershed over UNet probability map that fits into our pipeline. However, we keep a simple single-task problem with MPUNet as the backbone model while enforcing the gap regions for better separation of teeth through a dedicated loss function. To our best knowledge, our work is the first on automatic segmentation of human jaws that separates both individual tooth and bones (maxilla and mandible) while accurately detailing the gaps between them with very few data.

II. MATERIALS AND METHODS

In order to achieve accurate segmentation with a limited number of annotated training images, our strategy involves several key components: (i) We use the MPUNet approach, which segments 3D medical images by breaking them down into 2D views while maintaining as much spatial information as possible. (ii) To prevent overfitting and memory issues, we use a simple yet effective backbone model for segmentation. (iii) The model is pre-trained using data without the gap to learn general semantic features and then fine-tuned using a small set of highly accurate annotated data for gap generation. (iv) A dedicated weighted distance loss is used to emphasize the gap between the teeth and bones and between neighboring teeth. (v) We separate individual tooth

by applying marker based watershed segmentation over the UNet output probability map.

Here we divide the pipeline into two parts; the first part focuses on model construction and training, while the second deals with individual tooth segmentation over the model output, corresponding to strategy (v).

A. GENERAL PIPELINE

As a general pipeline, we are inspired by the method proposed by Xu et al. in regard to neural network training [30].

1) MODEL

As a baseline model, we use the MPUNet proposed by Perslev et al. to segment the 3D jaws using 2D UNet while preserving as much 3D spatial information as possible by generating views from different perspectives [20].

2) TRANSFER LEARNING

The accurate segmentation and efficient convergence with limited data rely partially on pre-training the model using a dataset with inferior annotation, followed by fine-tuning with a smaller set of precisely labeled data. Here we also follow [30] by transferring the weight in the last softmax layer and explicitly learning the encoder, which results in much faster convergence and correct encoding of the gap respectively.

3) LOSS FUNCTION WITH WEIGHTED DISTANCE MAP

In the pre-training step, the model is trained using a standard categorical cross-entropy loss, as we observed no improvement using a class-wise weighted cross-entropy loss or the Dice loss. During fine-tuning, to guide the model in learning the space between the teeth and bones where the PDL is located, a voxel-wise weight map $w(x)$ is applied to the loss function based on the distances from the foreground class borders. This approach was first proposed in the original UNet paper, which we have adapted for use with 3D data in a modified form [22], [30]. We define $w(x)$ as follows,

$$w(x) \equiv w_c(x) + w_0 e^{-\left(\frac{(d_1(x)+d_2(x))^2}{2\sigma^2}\right)}. \quad (1)$$

where d_1 and d_2 represent the distance to the border of the nearest foreground class and the second nearest foreground class respectively. We follow the original UNet paper and set $w_0 = 10$ and $\sigma = 5$. $w_c : \Omega \rightarrow R$ was originally proposed to balance the class frequencies, which we do not enforce; thus, w_c is set to 1 for every class c . During the fine-tuning process, the corresponding slice of the 3D weight map is sampled in conjunction with the images and labels. Then, the weight map is multiplied element-wisely with the cross-entropy loss between predictions and labels on each pixel before reduction and backpropagation. The whole process is illustrated in Fig. 2.

The incorporation of the distance-based weights (Eq. (1)) into the training of the neural network is inspired by the anticipation that in further FE simulations, a similar distance-based metric will be employed to generate space between the

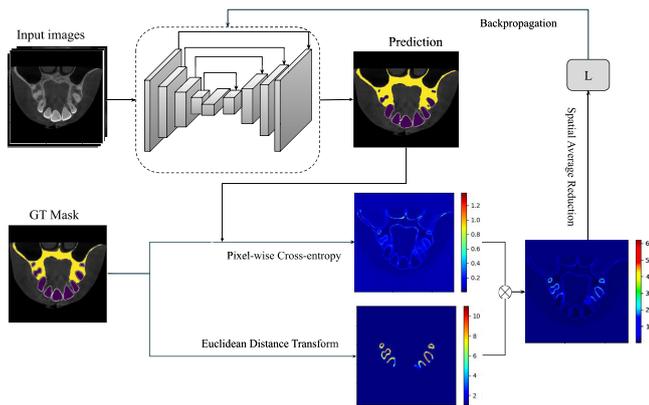


FIGURE 2. Model training pipeline weighted by distance map calculated from Eq. (1). \otimes denotes element-wise product, which suppresses the general boundary uncertainties while amplifying loss near the gaps. Note that the pixel-wise cross-entropy is visualized after averaging over all the classes. L is the final loss as a scalar after reduction.

segmented teeth and bone geometries to locate the PDL. This methodology is outlined in [12].

4) SAMPLING STRATEGIES

Careful sampling and interpolation are crucial for obtaining corresponding 2D slices from a 3D medical image viewed from a random orientation different from the standard RAS axes, which our initial test runs showed evidence of being the actual key to success. Here we follow the same idea of Perslev et al. to sample on isotropic grids within a sphere of diameter m centered at the origin of the scanner coordinate system in the physical scanner space [20]. Pixel dimension $d \in \mathbb{Z}^+$ of the grid and the actual size (diameter) $m \in \mathbb{R}^+$ (controlled by voxel size) in millimeters of the sphere need to be settled before the model training to decide the input size to UNet and its field of view. We differ from [20] by following the modification made by [30] in that these two numbers are chosen differently during training and inference. Briefly, d and m are computed heuristically as the 75 percentile across all axes and images during training but as the maximum value across all axes and all training images together with the current test image during testing. Please refer to [30] for the justification of this modification.

B. INDIVIDUAL TOOTH SEGMENTATION

The segmentation of the CBCT scans is conducted in two steps; first, proper teeth-bone segmentation is performed using the strategy discussed in the previous section with MPUNet; second, the teeth segments are decomposed into individual tooth segments.

Based on our observation, the model is less confident in the contacting interface of the two adjacent teeth in the output of the MPUNet before the argmax step, meaning a lower value in the probability map. Therefore, we apply a Marker Based Watershed Segmentation (MBWS) algorithm over the teeth probability map to separate the wrongly merged neighboring teeth [4]. Watershed is an unsupervised instance segmentation

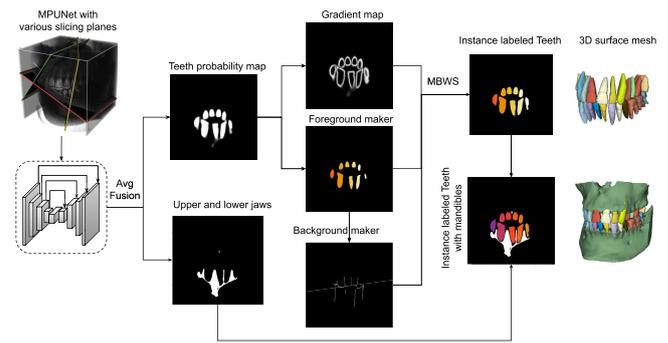


FIGURE 3. Individual tooth segmentation pipeline. MPUNet output from various views are first fused together. The probability map of teeth class is used by MBWS to generate individual tooth segmentation, which is then combined with the segmentation of bone class (maxilla and mandible). Coloring is random.

model that refers metaphorically to a geological watershed that separates adjacent drainage basins. Fig. 3 illustrates the whole process, where foreground and background markers are generated to guide the watershed operation based on the output probability map of MPUNet after averaging different views. Details of MBWS with foreground and background generation are explained in Appendix B.

The final result is the union of segmented tooth instances and the bone classes as shown in Fig 3, while the bone class has a higher priority in the intersecting/overlapping regions. Note that the upper and lower bone classes, i.e., mandible and maxilla can be trivially separated by a simple CCD because, unlike teeth, the upper and lower bone classes are always disconnected by a large gap. Here the bone class is labeled in one color for simplicity.

C. DATA AND EXPERIMENTS

We use 13 CBCT scans in this study, where 12 scans belong to 3Shape A/S in-house CBCT dataset, and one scan (P12 in Table 5) is obtained from 3DSlicer’s “Sample Data” module, titled “CBCT-MRI Head”. In all scans, the teeth and bone are annotated in both the upper and lower jaw. The scans were acquired from multiple resources from the typical age group of adult male and female ranging between 34 to 64 years old. Further details on sex, age, manufacturer details, and scanner settings are presented in Table 5 in Appendix A. Most of the patients have various dental problems such as dental implants and missing teeth. Besides, the dataset comprised of scans with different voxel sizes and various levels of artifacts such as metal filling artifacts or double contouring artifacts due to the movement of patients in the image acquisition step [16], [18]. Details about the utilized scans and artifacts are listed in Table 4 in Appendix A.

1) TRAIN-TEST SPLIT

Due to the difficulty of concise manual labeling to ensure the gap between teeth and bones, we only have eight cases with an accurate label map detailing the teeth-bone gaps whereas the rest five labeled data are inaccurate due to the missing gaps. These five scans are used for pre-training the

network, while the eight scans with accurate label maps are split equally for the train-test, meaning that only four scans are used to train the network to detail the gaps. Specifications about the data split are listed in Table 4 in Appendix A.

2) PRE-PROCESSING

As presented in Table 4, the original scans are with various voxel sizes and different dimensions. Therefore, all the scans are first upsampled to the smallest voxel size (0.15 mm) in the dataset with a B-Spline interpolation and cropped to an identical dimension of 512³. We then pre-process the data by applying an intensity standardization based on the equation $X_{scale} \equiv (x_i - x_{mean}) / (x_{75} - x_{25})$, where x_{25} and x_{75} are the 1st and 3rd quartiles respectively. This transformation scales the intensity based on quartiles and is more robust to outliers, which is especially crucial when working with data involving metal artifacts, in some cases resulting as outliers with extremely high-intensity values. We apply this standardization in two steps: over the 3D volume and then over each sampled slice to MPUNet. No other pre-processing is used to avoid potential errors that can easily propagate in the neural network.

3) EXPERIMENTAL SETUP

The network is trained on NVIDIA GeForce RTX 3090 with a batch size of 10 using the Adam optimizer for 60 epochs with a learning rate of 10⁻⁵ and reduced by 10% for every two consecutive epochs without performance improvements. We stop training if the performance of five consecutive epochs does not improve. Pre-training takes approximately one day, while fine-tuning takes about 10 hours to converge.

4) AUGMENTATIONS

We apply Random Elastic Deformations to generate images with deformed strength and smoothness [24]. The augmentations are generated on the fly during the training process, and following MPUNet we assign a weight value of 1/3 for the deformed samples [20].

D. ETHICS STATEMENT

The requirement for the ethical committee’s approval was waived from “Center for Regional Development, The Scientific Ethics Committee” with a reference number 21063693, with the following statement: “It has been assessed that this is not a health science research project as defined in section II of the committee act, but that it is a non-invasive study containing 3D scan images of jaws and teeth”. Note that, this work only uses an available dataset that already had been collected by 3Shape, and no new scans has been collected just for use of this study. All scans had been acquired as part of a patient’s treatment and had already been thoroughly studied by patient’s dentist/orthodontist, which is a legal requirement when performing a CBCT scan. Hence, here is no possibility that we can discover additional diseases etc. that the patient

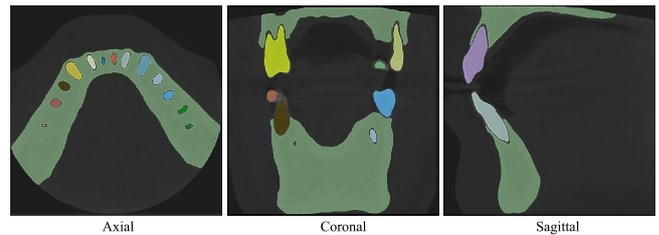


FIGURE 4. Generated gaps for one of the scans in the test set displayed from different views.

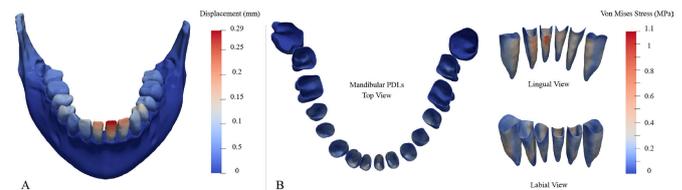


FIGURE 5. Finite element analysis of a tipping scenario. A: Displacement filed of the teeth. B: smooth von Mises stress pattern on the periodontal ligaments.

had not already been informed about. The patients and the dentists have given written consents for using the scans.

III. RESULTS AND DISCUSSIONS

In order to produce geometries suitable for finite element (FE) models, the auto-segmentation framework must accurately separate teeth and jaw bones and produce precise results near the boundaries, which are crucial for creating the PDL layers in the jaw. We evaluate the performance of generating the general tooth-bone semantic structures and the gaps between the teeth-bone interfaces in the first subsection. In the second subsection, we evaluate our further task of individual teeth segmentation.

As shown in Figure 4, the enforcement of a distance weight to the loss allows the model to accurately capture the gap. The final result is anatomically accurate and requires minimal manual intervention for subsequent simulations, such as finite element analysis. Segmentation results in 3D are available at <https://github.com/diku-dk/AutoJawSegment>. As an example, we have generated the PDL geometries on the reconstructed geometries obtained from the segmented CBCT scans with a method proposed in [12] to analyze the stress distributions in a tipping scenario as shown in Fig 5. The results demonstrate a smooth stress pattern, indicating that the output from our method is suitable for finite element (FE) simulations.

A. PERFORMANCE METRICS

Although the commonly adopted measurements of voxel-wise correspondence, e.g., Dice Score, could be misleading regarding the final FE simulations, we still include these measurements as part of the quantitative validation and an ablation study of our several design choices. The Dice Score is defined as

$$Dice(P, Y) \equiv \frac{2|P \cap Y|}{|P| + |Y|} \tag{2}$$

where P and Y denote the predicted result and ground truth segmentation respectively.

In addition to the standard Dice score, we are particularly interested in evaluating the performance of the model in the surface and gap regions. To assess this, we adopt two additional evaluation metrics. The first metric is the Hausdorff distance (HD), which measures the surface accuracy by calculating the largest distance between the predicted result P and the nearest point on the ground truth Y .

$$HD(P, Y) \equiv \max_{p \in P} (\min_{y \in Y} \|p - y\|_2), \quad (3)$$

$$\max_{y \in Y} (\min_{p \in P} \|p - y\|_2)$$

Secondly, Average Segmentation Surface Distance (ASSD) measures the average distance between the estimated segmentation surface S_P and the ground truth surface S_Y . The surface is computed by subtracting erosion from dilation.

$$ASSD(P, Y) \equiv \text{mean}(\text{mean}_{d \in S_P} (\text{dist}(d, S_Y))) \times \text{mean}_{g \in S_Y} (\text{dist}(g, S_P)). \quad (4)$$

where $\text{dist}(d, S_Y) \equiv \min_{y \in S_Y} \|d - y\|_2$ denotes the nearest Euclidean distance from point d to surface S_Y .

Although the above two surface measurements better capture the segmentation stability and conciseness than Dice score, they are based on the whole structure with parts that are not that critical for later simulation studies, e.g., the upper surface of the maxilla and lower surface of the mandible. Instead, we are only interested in the parts where two instances meet, i.e., the teeth-bone interfaces. Therefore, we also adopt GapDice proposed in [30] in Eq (6) to measure the average Dice score only around the gap regions.

Given the segmentation results P and the ground truth segmentation Y , the gap region G is defined by thresholding the Euclidean distance transformation map of Y

$$G = \{x | d_1(x) + d_2(x) < \epsilon\} \quad (5)$$

where as defined in Eq (1), d_1 and d_2 represent the distance to the border of the nearest foreground class and the second nearest foreground class in Y , respectively. ϵ is the threshold value, which we set $\epsilon = 5$ as we found it to effectively capture both the gap and boundary regions.

The Dice score between P and Y is then calculated in the standard manner, but only inside G , as defined in Eq (6). Fig 6 shows an indication of such regions.

$$\text{GapDice}(P, Y) \equiv \frac{2|P \cap Y \cap G|}{|P \cap G| + |Y \cap G|}. \quad (6)$$

B. QUANTITATIVE RESULTS AND ABLATION STUDY

Table 1 presents the aforementioned performance metrics on the test set, including four images with accurate ground truth segmentations. This experiment is implemented by modifying one of the design choices each time while fixing the others. (i) The strategy described in the Materials and

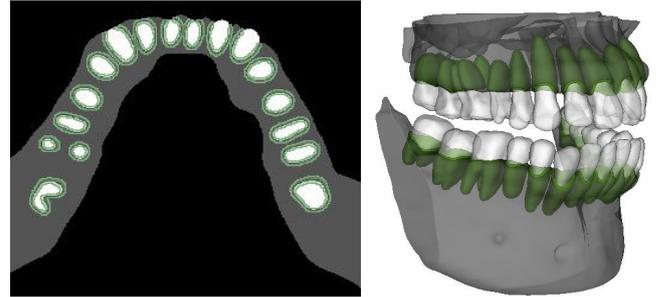


FIGURE 6. The estimated gap region (green) when calculating GapDice for a patient, illustrated in an axial slice (left) and in 3D (right).

TABLE 1. Test results of our model compared with various design choices and other models from the literature.

	Dice (%) ↑	GapDice (%) ↑	ASSD (mm) ↓	HD (mm) ↓
Ours	95.14 ± 1.21	64.28 ± 4.65	0.118 ± 0.03	8.538 ± 3.65
NoPretrain	94.05 ± 1.35	59.68 ± 4.69	0.139 ± 0.04	11.91 ± 6.93
NoWeight	94.85 ± 1.19	61.87 ± 4.49	0.127 ± 0.03	8.183 ± 1.62
NoFineTune	95.09 ± 1.34	58.24 ± 3.20	0.115 ± 0.03	7.098 ± 1.48
3DUNet [6]	64.07 ± 0.52	36.36 ± 4.51	2.010 ± 0.87	45.41 ± 11.4
MSDNet-3 [19], [27]	49.80 ± 2.77	21.85 ± 2.74	1.943 ± 0.599	29.41 ± 6.56
MSDNet-100 [19], [27]	88.91 ± 4.33	59.01 ± 1.89	0.838 ± 0.86	26.95 ± 7.48
MSDNet-200 [19], [27]	89.69 ± 3.74	58.97 ± 2.56	0.800 ± 0.90	28.09 ± 6.32

methods Section (ours), (ii) Training the model without pre-training inaccurate data with no gap (NoPretrain), (iii) Training the model without enforcing distance-based weight map (NoWeight), (iv) Using only inaccurate data without fine-tuning the model (NoFineTune), (v) Using a 3D UNet [6] as the backbone model (3DUNet).

In addition, we also compare our results with 2D mixed-scale dense CNN (MSDNet) [19] adopted by Wang et al. [27] for the segmentation of human jaws as mentioned in the Introduction. The model is trained only over the extracted axial slices of CBCT scans from the training set, as proposed in [27]. During inference, the 2D prediction results of all the slices of each test scan are concatenated back to 3D for validation. The MSDNet employed by Wang et al. [27] has only a depth of 3, which we found extremely insufficient for such task. We have thus also considered a depth of 100 and 200, as adopted in the original MSDNet paper [19].

The results indicate that the MPUNet (all the first four models) performs significantly better than the standard 3D UNet when dealing with limited data. The 3D UNet fails to learn the general semantic features of tooth-bones with few data compared with MPUNet. Similarly, our model significantly outperforms the MSDNet [19] adopted by Wang et al. [27], even with more depth adjustment. We speculate the poor performance of MSDNet is due to the model's oversimplified structure without downsampling and upsampling phases like in UNet and its insufficiency in learning 3D data from axial slices alone. These drawbacks prevent it from learning appropriate features, especially on data with high noise ratios like the scans in our dataset, compared with the dataset in [27] where the CBCT scans are free of metal artifacts.

Among ablation studies using MPUNet as the backbone model (all the first four models), it is very interesting to

notice that the model without being fine-tuned (NoFineTune) gives a high Dice score and best surface measurements (HD and ASSD). However, since it is anatomically incorrect in that it fails to detail the gap between tooth-bones (cf. Fig. 1c), it has significantly worse performance in the proposed task-specific measurement, i.e., GapDice. This is an indication of why the standard performance metrics that measure voxel-wise correspondence or surface closeness can be misleading regarding the final FE simulation and needs to be resolved for future segmentation works.

Apart from this, our pipeline outperforms in almost all four metrics. Especially, although the difference in the Dice score is not significant (95.14 ± 1.21 vs 94.05 ± 1.35), pre-training on inaccurate data and enforcing the weight map during fine-tuning shows a significantly better GapDice score (64.28 ± 4.65 vs 59.68 ± 4.65) and ASSD (0.118 ± 0.03 vs 0.139 ± 0.04), which is vital for further simulation. Nonetheless, we notice that the GapDice score is significantly lower than the standard Dice score even in our pipeline which has the highest GapDice. Such segmentation errors are mostly due to various artifacts in the scan, as listed in Table 4, which influences the segmentation results. In general, our results in Table 1 have shown good segmentation performance and robustness to the aforementioned artifacts by producing high Dice scores and low surface deviations. However, the concise modeling of details such as PDL layers in noisy scans can be challenging even with our model adaptations to penalize more on the gap regions. A future direction for providing an even more robust network against the mentioned artifacts would be including more data that capture various kinds of artifacts or adding synthetic artifacts to the scans to verify if the model can be trained to learn invariance to the artifacts. Alternatively, deep learning models have been proposed to reduce artifacts as a preprocessing step for the auto-segmentation task [2], [34]. This means we would need to have more data from the scan with artifacts along with the scan of the same patient without artifacts, which is difficult to obtain.

1) RESULTS WITH CROSS-VALIDATION

The aforementioned experiments and results in Table 1 are based on a specific train-test split of the eight scans with accurate label maps. This choice is to preserve a similar level of noise/artifacts in the training data (used for fine-tuning to learn the gap) and test data, as listed in Table 4 in Appendix A. As another common practice in machine learning, here we also conduct a 5-fold cross-validation by randomly dividing the eight scans into training and test sets, ensuring an equal split of four scans in each set as before to analyze performance variations. Table 2 shows the mean and standard deviation of the results with various design choices in correspondence to those in Table 1. Note that the methods with different backbone models (3DUNet and MSDNet) are not included for cross-validation as they have shown to have significantly poorer performance in Table 1. Table 2 generally

TABLE 2. Cross-validation with various design choices.

	Dice (%) \uparrow	GapDice (%) \uparrow	ASSD (mm) \downarrow	HD (mm) \downarrow
Ours	95.13 ± 1.35	65.75 ± 4.64	0.117 ± 0.06	7.76 ± 0.99
NoPretrain	91.63 ± 4.2	58.23 ± 1.35	0.174 ± 0.05	11.93 ± 0.07
NoWeight	95.27 ± 2.83	63.95 ± 6.11	0.117 ± 0.06	8.183 ± 1.00
NoFineTune	94.48 ± 0.59	58.07 ± 3.02	0.139 ± 0.02	7.46 ± 0.4

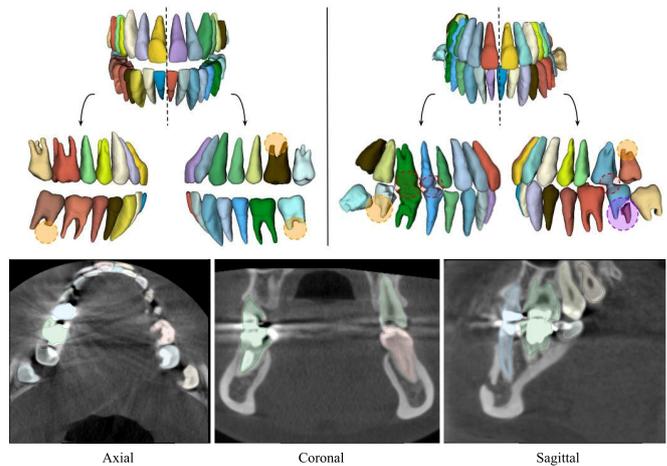


FIGURE 7. Illustration of failure cases. Top: The segmentation results on two different test scans. Different failure cases are illustrated with different colored circles, i.e., orange: inaccurate segmentation of the root apices; red: connected teeth problem; purple: a single tooth is wrongly segmented with different labels. Colors are randomly assigned to different teeth. Bottom: The scan with overlaid individual tooth labels from the top right case, displayed from different views and showing various artifacts that explain the failures.

shows a similar pattern with Table 1 in that our pipeline is able to give significantly better GapDice which is vital for further simulation studies [12].

C. PERFORMANCE OF INDIVIDUAL TEETH SEGMENTATION

We further evaluate the performance of individual tooth segmentation of our pipeline with the watershed method mentioned above. Fig 3 illustrates that our pipeline can generate visually accurate surface meshes of each tooth and bone even in cases where the CBCT had been acquired in the natural biting position, making the individual tooth segmentation complex as the maxillary and mandibular teeth are touching each other in most of the occlusal surfaces. On the other hand, Fig 7 illustrates several failure cases in one test scan. This test case is filled with various artifacts such as crowded teeth, metal fillings, or dental bridges, as indicated in Fig 7 Bottom. As mentioned in the previous subsection, such artifacts can influence the segmentation in fine detail, e.g., the gap between neighboring teeth and roots, which results in connected teeth and missing root apices.

Numerical evaluation of the individual tooth and bone segmentation is tricky because, unlike deep-learning-based instance segmentation methods, no soft region proposals are involved in the proposed method, making it impossible to compute a mean Average Precision (mAP). Therefore, our result of individual tooth segmentation is evaluated using the

TABLE 3. Test results of individual teeth segmentation compared with semantic segmentation.

	Dice (%) \uparrow	GapDice (%) \uparrow	ASSD (mm) \downarrow	HD (mm) \downarrow
Teeth	95.14 \pm 1.21	64.28 \pm 4.65	0.118 \pm 0.03	8.538 \pm 3.65
Individual Tooth	95.05 \pm 1.00	64.49 \pm 3.93	0.116 \pm 0.03	7.840 \pm 1.94

same metrics in the previous section as shown in Table 3. In this case, all the predicted teeth instances are mapped back to a binary case and then combined with the bone class. This is an unfair comparison since those metrics cannot reflect its ability to distinguish individual teeth. However, it is interesting to investigate if the further separation of individual teeth does not harm the overall performance, even in this unfair setting. In particular, Table 3 shows that the result after applying individual tooth segmentation gives almost identical results, with a surprising improvement of GapDice and HD. This evaluation scheme could provide insight that the teeth segments deviate negligibly from the prior segmented single tooth class.

We acknowledge that the watershed method involves several hyper-parameters, e.g., the threshold of the UNet probability map and the opening and erosion sizes in Eq (7). The values of these parameters must be tuned beforehand to ensure that neighboring teeth do not share the same foreground marker while avoiding creating multiple markers for the same tooth. Furthermore, one might need to tune these parameters when applying the same pipeline to other problems with different scales or resolutions. Therefore, a future work direction would be to infer those numbers automatically from the studied dataset.

IV. CONCLUSION

Our proposed auto-segmentation framework successfully segments both individual tooth and bones (maxilla and mandible) from CBCT scans of human jaws, with accurate tooth-bone boundaries and the gaps between the teeth roots and sockets. The framework employs a modified version of MPUNet, which is pre-trained on a dataset that does not consider the presence of the PDL layer to learn the general features of tooth-bone geometries. The model is then fine-tuned using a small set of highly accurate segmentations with a dedicated loss function that penalizes the gap regions. This allows the model to better understand the gap where the PDL layer resides and generate anatomically accurate segmentations. We further separate individual teeth by applying watershed segmentation over the MPUNet output. The results of our experiments demonstrate the effectiveness of our framework in detailing critical features, such as the gap between the teeth-bone interfaces and the interproximal regions of the teeth.

A trained segmentation professional has verified our work, and the results show improved numerical results, reaching an overall Dice score above 95% and a significantly higher GapDice than other methods. Our approach can improve anatomically incorrect and poorly annotated datasets with a few accurate labels. One ablation study indicates that the

standard performance metrics can be misleading regarding the final FE simulation by producing high-performance metrics but anatomically incorrect results. On the other hand, our results from the finite element (FE) analysis performance test indicate that the models generated produce stress patterns that are smooth and free of artifacts caused by missing gaps in the geometry. As a result, the segmentation outcomes from this study can be applied to generate FE models with minimal adjustments.

APPENDIX A UTILIZED SCAN DETAILS AND CBCT ARTIFACTS

Image artifacts can be broadly defined as visual effects in reconstructed data that are absent in the real-world object being studied. These artifacts may be the result of various factors, such as subject movement, hardware limitations, the simplified mathematical assumptions used for 3D reconstruction, or their combination. These artifacts, their severity, and voxel sizes can play a significant role in the segmentation task's complexity. Therefore, to provide an overview of observed artifacts in the dataset, we assessed the existence of the common CBCT artifacts, i.e., metal artifacts, noise, blurriness, motion, and aliasing artifacts [16], [18], [23].

The noise artifact can be observed as inconsistent voxel intensities in regions where similar intensities should be present. In addition, double contours can be observed in the CBCT scans that are typically caused due to the patient's movement during the image acquisition process, making it difficult to accurately identify boundaries and delicate structures. Another common effect is the aliasing pattern, which can be seen as lines diverging from the center toward the periphery [16], [18], [23]. Moreover, metal artifacts can be seen as regions with high intensities followed by streaks diverging from the center of the metallic restoration/crown, making it difficult to precisely identify the studied tooth's boundaries. Furthermore, such metal artifacts can cause inaccurate grayscale values in areas not immediately adjacent to the metallic restoration [16], which we refer to as the ghosting effect here.

Table 4 provides details of the utilized scans in this study and represents an overview of the train-test split in this study, as well as the involved artifacts in each scan ranked between zero to two, specifying the artifact's severity level. Further details on sex, age, manufacturer details, and scanner settings are presented in Table 5.

APPENDIX B INDIVIDUAL TOOTH SEGMENTATION DETAILS

Individual tooth is separated by applying the watershed method over the MPUNet probability map of the teeth class. The watershed method considers the intensity value of each voxel as the height, where a high value denotes spikes/hills and a low value denotes valleys. It fills every isolated valley (local minima) with different colored water (labels). As the water rises, depending on the peaks (gradients) nearby, water from other valleys with different colors will merge. The

TABLE 4. Specification of the utilized scans, including details on the original voxel size, number of missing teeth (including the wisdom teeth), different artifact types, and data split for model training. The included artifacts ranked between zero, one, and two to specify the artifact's level in each scan. This variety of artifacts indicates a challenging task for learning an auto-segmentation network. The last column represents the data split for the pre-training (PreT), fine-tuning (FineT), and testing (Test) steps.

Scan ID	Original scan		Cropped resampled ROI		Missing Teeth	Artifacts					Involved set	
	Voxel Size	Dimension	Voxel Size	Dimension		Metal	Noise	Motion	Ghosting	Aliasing		Total
P1	0.3	400x400x280	0.15	512x512x512	3	2	2	1	2	1	8	Test
P2	0.15	532x532x540	0.15	512x512x512	2	1	1	0	1	1	4	Test
P3	0.15	532x532x540	0.15	512x512x512	4	2	1	0	1	2	6	FineT
P4	0.15	532x532x540	0.15	512x512x512	4	1	1	0	0	1	3	PreT
P5	0.15	400x400x280	0.15	512x512x512	4	1	2	0	1	1	5	FineT
P6	0.3	400x400x280	0.15	512x512x512	3	2	0	1	2	2	7	Test
P7	0.3	400x400x280	0.15	512x512x512	4	1	1	0	1	1	4	FineT
P8	0.3	400x400x280	0.15	512x512x512	0	2	2	0	2	2	8	PreT
P9	0.3	400x400x280	0.15	512x512x512	1	2	0	0	2	2	6	PreT
P10	0.3	400x400x280	0.15	512x512x512	4	2	0	1	2	2	7	PreT
P11	0.2	752x750x400	0.15	512x512x512	1	1	1	0	2	1	5	PreT
P12	0.25	520x406x340	0.15	512x512x512	3	2	2	0	2	2	8	FineT
P13	0.2	501x501x501	0.15	512x512x512	1	2	2	0	2	2	8	Test

TABLE 5. Details of studied cohort and utilized devices for image acquisition including manufacturer information and device settings.

Scan ID	Sex	Age	Tube voltage (kvp)	Tube current (mA)	Manufacturer	Manufacturer model name
P1	F	55	85	4.8	Vatech Company Limited	Implagraphy
P2	N/A	N/A	N/A	N/A	3Shape Medical A/S	X1
P3	N/A	N/A	N/A	N/A	3Shape Medical A/S	X1
P4	N/A	N/A	N/A	N/A	N/A	N/A
P5	F	35	120	18	Xoran Technologies	i-CAT 3D Dental Imaging System
P6	F	64	89	4	Vatech Company Limited	PaX-Flex3D
P7	M	64	85	4.8	Vatech Company Limited	Implagraphy
P8	F	55	85	4.8	Vatech Company Limited	Implagraphy
P9	F	46	89	4	Vatech Company Limited	PaX-Flex3D
P10	M	36	89	4	Vatech Company Limited	PaX-Flex3D
P11	N/A	N/A	90	12	3Shape Medical A/S	X1
P12	M	N/A	N/A	N/A	N/A	N/A
P13	F	34	90	7	Planmeca	Planmeca ProMax

algorithm then tries to prevent the merging by building “barriers” locations where water merges until all the peaks are underwater. The barriers then naturally mark the boundary for each instance, which results in instance segmentation of the teeth.

In practice, the primary watershed method usually produces over-segmented results due to its sensitivity to noise or other irregularities in the image, like many local minima. Instead, Marker-Based Watershed Segmentation (MBWS) alleviates this problem by specifying the valley points (*foreground markers*) that are to be merged and barriers (*background markers*) to the model. The whole process is shown in Fig 3, which is explained in the following paragraphs.

1) FOREGROUND MARKERS GENERATION

Instead of working directly on the image here, the *foreground markers* are determined by thresholding the UNet output probability map since the probability map naturally represents how confident the model is in predicting the foreground class, in this case, teeth. Eq (7) below indicates the foreground regions where we first apply a threshold of 0.8 over the probability map on teeth class $P(x)$. We then remove isolated false positives and shrink the thresholded foreground regions by applying an opening, \circ , with a structural ball element $E_{5 \times 5 \times 5}$ followed by an erosion, \bullet , with a structural ball element $E_{3 \times 3 \times 3}$ to provide disconnected teeth. Note that the radii should be determined based on the general shape of the instance, in this case, a tooth, to separate neighboring teeth

while avoiding introducing undesired disconnectivity inside each tooth.

$$M_f \equiv \{x \mid (P(x) > 0.8) \circ E_{5 \times 5 \times 5} \bullet E_{3 \times 3 \times 3}\}. \quad (7)$$

2) BACKGROUND MARKERS GENERATION

The *background markers* are generated based on the *foreground markers* M_f generated from the previous step by first applying a distance transform over M_f , which corresponds to the terms of d_1 and d_2 in Eq (1). The final background region is generated by thresholding both the difference and the sum of the two distances, which is indicated in Eq (8). This choice of background markers corresponds to the trimmed perpendicular bisector plane between any two neighboring teeth, thus ensuring neighboring teeth do not get merged by the watershed. The threshold of $d_1 + d_2$ is necessary to ensure that the background marker does not penetrate other foreground regions. The value of 20 is experimental and will need to be tuned for other datasets or voxel sizes.

$$M_b \equiv \{x \mid |d_1(x) - d_2(x)| \leq 1 \wedge |d_1(x) + d_2(x)| \leq 20\}. \quad (8)$$

3) MARKER-BASED WATERSHED SEGMENTATION

As shown in Fig 3, with the selected foreground and background markers, the final MBWS is conducted on the gradient of the UNet probability map due to its good response to weak edge information [4]. The gradient is computed by convolving Gaussian derivative kernel with $\sigma = 2$. Our experience indicates that this preserves root structures better than directly working on the probability map.

4) TRAINING STRATEGY WITH ADDITIONAL WEIGHT-MAP

Although this MBWS to separate individual tooth has shown to be effective, its performance largely depends on the quality of the UNet probability map. More specifically, if the model gives inaccurate results (high probability of being foreground) near the gap between some neighboring teeth, these teeth will share a common foreground marker. Increasing the utilized thresholding value for the foreground or the erosion/opening kernel sizes in Eq (7) can provide different foreground markers for the adjacent teeth. However, this

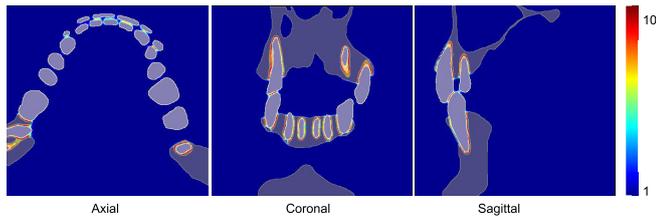


FIGURE 8. The values of the weight map presented as colormap along with the labeled teeth and bone. The proposed weight map enforces gaps not only between the teeth and bone segments where the periodontal ligament lies, but also between nearby teeth.

may also introduce fractions (several foreground markers) inside the same tooth, causing the watershed algorithm to assign several labels to the different parts of the same tooth. Therefore, keeping the morphological and thresholding level is crucial while providing a more accurate result near the interproximal gaps.

As the distance weight map is very effective in learning the gaps in the teeth-bone interfaces, we use this same strategy to learn the interproximal gaps. After applying connected component decomposition over ground truth to get a different label for each tooth, we can follow the same strategy in Eq (1) to enforce a higher weight on both the gap in the teeth-bone interface and in the interproximal regions of the teeth to better separate the adjacent teeth from each other. As shown in Fig 8, the weight map calculated by Eq (1) has a higher value not only between tooth-bone gaps but also between neighboring teeth. Note that such Euclidean transformation in Eq (1) is highly time-consuming because it involves the distance computation to every class and sorting the values afterward. Hence, the time complexity increases with at least $\mathcal{O}(n \log n)$ where n denotes the number of classes. For example, the gap modeling in the teeth-bone interfaces involved only two classes (bone and teeth), but modeling the gaps in the interproximal regions involves approximately 30 classes (number of teeth). Therefore, it is crucial that the weight map is computed before the model training and then sampled together with the corresponding images and labels.

ACKNOWLEDGMENT

The authors would like to thank 3Shape A/S, especially Peter Lempel Søndergaard, for providing this study's CBCT scans, and for their supports in enhancing the segmentation results. The mentioned data was originally acquired for diagnostic purposes unrelated to this study. No other aspect of this work triggered ethical issues.

REFERENCES

- [1] H. Bao, L. Dong, S. Piao, and F. Wei, "BEiT: BERT pre-training of image transformers," 2021, *arXiv:2106.08254*.
- [2] D. F. Bauer, C. Ulrich, T. Russ, A.-K. Golla, L. R. Schad, and F. G. Zöllner, "End-to-end deep learning CT image reconstruction for metal artifact reduction," *Appl. Sci.*, vol. 12, no. 1, p. 404, Dec. 2021.
- [3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

- [4] Y. Chen, H. Du, Z. Yun, S. Yang, Z. Dai, L. Zhong, Q. Feng, and W. Yang, "Automatic segmentation of individual tooth in dental CBCT images from tooth surface map by a multi-task FCN," *IEEE Access*, vol. 8, pp. 97296–97309, 2020.
- [5] M. Chung, M. Lee, J. Hong, S. Park, J. Lee, J. Lee, I.-H. Yang, J. Lee, and Y.-G. Shin, "Pose-aware instance segmentation framework from cone beam CT images for tooth segmentation," *Comput. Biol. Med.*, vol. 120, May 2020, Art. no. 103720.
- [6] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2016, pp. 424–432.
- [7] Z. Cui, Y. Fang, L. Mei, B. Zhang, B. Yu, J. Liu, C. Jiang, Y. Sun, L. Ma, J. Huang, Y. Liu, Y. Zhao, C. Lian, Z. Ding, M. Zhu, and D. Shen, "A fully automatic AI system for tooth and alveolar bone segmentation from cone-beam CT images," *Nature Commun.*, vol. 13, no. 1, p. 2096, Apr. 2022.
- [8] Z. Cui, C. Li, N. Chen, G. Wei, R. Chen, Y. Zhou, D. Shen, and W. Wang, "TSegNet: An efficient and accurate tooth segmentation network on 3D dental model," *Med. Image Anal.*, vol. 69, Apr. 2021, Art. no. 101949.
- [9] Z. Cui, C. Li, and W. Wang, "ToothNet: Automatic tooth instance segmentation and identification from cone beam CT images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2019, pp. 6368–6377.
- [10] Z. Cui, B. Zhang, C. Lian, C. Li, L. Yang, W. Wang, M. Zhu, and D. Shen, "Hierarchical morphology-guided tooth instance segmentation from CBCT images," in *Information Processing in Medical Imaging*. Cham, Switzerland: Springer, 2021, pp. 150–162.
- [11] E. Gardiyanoglu, G. Ünsal, N. Akkaya, S. Aksoy, and K. Orhan, "Automatic segmentation of teeth, crown-bridge restorations, dental implants, restorative fillings, dental caries, residual roots, and root canal fillings on orthopantomographs: Convenience and pitfalls," *Diagnostics*, vol. 13, no. 8, p. 1487, Apr. 2023.
- [12] T. Gholamalazadeh, F. Moshfeghifar, Z. Ferguson, T. Schneider, D. Panozzo, S. Darkner, M. Makaremi, F. Chan, P. L. Søndergaard, and K. Erleben, "Open-full-jaw: An open-access dataset and pipeline for finite element models of human jaw," *Comput. Methods Programs Biomed.*, vol. 224, Sep. 2022, Art. no. 107009.
- [13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.
- [14] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "NnU-net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, Feb. 2021.
- [15] T. J. Jang, K. C. Kim, H. C. Cho, and J. K. Seo, "A fully automated method for 3D individual tooth identification and segmentation in dental CBCT," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6562–6568, Oct. 2022.
- [16] S. R. Makins, "Artifacts interfering with interpretation of cone beam computed tomography images," *Dental Clinics North Amer.*, vol. 58, no. 3, pp. 485–495, Jul. 2014.
- [17] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.
- [18] N. Dwivedi, A. Nagarajappa, and R. Tiwari, "Artifacts: The downturn of CBCT image," *J. Int. Soc. Preventive Community Dentistry*, vol. 5, no. 6, p. 440, 2015.
- [19] D. M. Pelt and J. A. Sethian, "A mixed-scale dense convolutional neural network for image analysis," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 2, pp. 254–259, Jan. 2018.
- [20] M. Perslev, E. B. Dam, A. Pai, and C. Igel, "One network to segment them all: A general, lightweight system for accurate 3D medical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2019, pp. 30–38.
- [21] S. Poelert, E. Valstar, H. Weinans, and A. A. Zadpoor, "Patient-specific finite element modeling of bones," *Proc. Inst. Mech. Eng., H, J. Eng. Med.*, vol. 227, no. 4, pp. 464–478, Apr. 2013.
- [22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
- [23] R. Schulze, U. Heil, D. Groß, D. Bruellmann, E. Dranischnikow, U. Schwanecke, and E. Schoemer, "Artefacts in CBCT: A review," *Dentomaxillofacial Radiol.*, vol. 40, no. 5, pp. 265–273, Jul. 2011.

- [24] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. ICDAR*, vol. 3, 2003, pp. 958–963.
- [25] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segformer: Transformer for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 7262–7272.
- [26] P.-J. Verhelst, A. Smolders, T. Beznik, J. Meewis, A. Vandemeulebroucke, E. Shaheen, A. Van Gerven, H. Willems, C. Politis, and R. Jacobs, "Layered deep learning for automatic mandibular segmentation in cone-beam computed tomography," *J. Dentistry*, vol. 114, Nov. 2021, Art. no. 103786.
- [27] H. Wang, J. Minnema, K. J. Batenburg, T. Forouzanfar, F. Hu, and G. Wu, "Multiclass CBCT image segmentation for orthodontics with deep learning," *J. Dental Res.*, vol. 100, no. 9, pp. 943–949, 2021.
- [28] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, and F. Wei, "Image as a foreign language: BEiT pretraining for all vision and vision-language tasks," 2022, *arXiv:2208.10442*.
- [29] X. Wu, H. Chen, Y. Huang, H. Guo, T. Qiu, and L. Wang, "Center-sensitive and boundary-aware tooth instance segmentation and classification from cone-beam CT," in *Proc. IEEE 17th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2020, pp. 939–942.
- [30] P. Xu, F. Moshfeghifar, T. Gholamalizadeh, M. B. Nielsen, K. Erleben, and S. Darkner, "Auto-segmentation of hip joints using multiplanar UNet with transfer learning," in *Proc. Workshop Med. Image Learn. Ltd. Noisy Data*. Cham, Switzerland: Springer, 2022, pp. 153–162.
- [31] Y. Yang, R. Xie, W. Jia, Z. Chen, Y. Yang, L. Xie, and B. Jiang, "Accurate and automatic tooth image segmentation model with deep convolutional neural networks and level set method," *Neurocomputing*, vol. 419, pp. 108–125, Jan. 2021.
- [32] T. Yeshua, S. Ladyzhensky, A. Abu-Nasser, R. Abdalla-Aslan, T. Boharon, A. Itzhak-Pur, A. Alexander, A. Chaurasia, A. Cohen, J. Sosna, I. Leichter, and C. Nadler, "Deep learning for detection and 3D segmentation of maxillofacial bone lesions in cone beam CT," *Eur. Radiol.*, pp. 1–12, May 2023, doi: [10.1007/s00330-023-09726-6](https://doi.org/10.1007/s00330-023-09726-6).
- [33] J. Zhang, M. Liu, L. Wang, S. Chen, P. Yuan, J. Li, S. G.-F. Shen, Z. Tang, K.-C. Chen, J. J. Xia, and D. Shen, "Context-guided fully convolutional networks for joint craniomaxillofacial bone segmentation and landmark digitization," *Med. Image Anal.*, vol. 60, Feb. 2020, Art. no. 101621.
- [34] Y. Zhang and H. Yu, "Convolutional neural network based metal artifact reduction in X-ray computed tomography," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1370–1381, Jun. 2018.



PEIDI XU received the B.Eng. degree in computer science and technology from the Nanjing University of Posts and Telecommunications, in 2018, and the M.Sc. degree in computer science from the Department of Computer Science, University of Copenhagen (DIKU), in 2020, where he is currently pursuing the Ph.D. degree. His research interests include medical image analysis, including image segmentation and the computational modeling of renal blood vessels and flows simulation.



interests include computational modeling, medical image analysis, and machine learning applied to healthcare.

TORKAN GHOLAMALIZADEH received the Ph.D. degree in computer science from the University of Copenhagen, in 2022. During the Ph.D. degree, she focused on developing and analyzing patient-specific computational models obtained from medical images using simulation methods and machine learning. Since 2022, she has been working on AI-based solutions applied to medical images as a Machine Learning Researcher with 3Shape A/S, Copenhagen, Denmark. Her research



FAEZEH MOSHFEGHIFAR received the Ph.D. degree, in 2022. She is currently a Postdoctoral Researcher with the Department of Computer Science, University of Copenhagen. Her research interests include the application of state-of-the-art knowledge from the medical and computer science domains to assist medical research.



SUNE DARKNER received the Ph.D. degree, in 2009. He is currently an Associate Professor with the Department of Computer Science, University of Copenhagen. His research interests include density estimation and image similarity, histograms and scale space, geometry, and simulation with applications to digital twins and digital populations.



interface tracking meshing.

KENNY ERLEBEN received the Ph.D. degree, in 2005. He is currently a Full Professor with the Department of Computer Science, University of Copenhagen. His research interests include computer simulation and numerical optimization with particular interests in the computational contact mechanics of rigid and deformable objects, inverse kinematics for computer graphics and robotics, computational fluid dynamics, computational biomechanics, foam simulation, and

...