UNIVERSITÀ DEGLI STUDI FIRENZE

UNIVERSITÉ DE LYON

UNIVERSITÉ FRANCO ITALIENNE

UNIVERSITÀ ITALO FRANCESE

UNIVERSITÀ DEGLI STUDI DI FIRENZE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE (DINFO)

CORSO DI DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE

CURRICULUM: ELECTRONICS, ELECTROMAGNETICS AND ELECTRICAL SYSTEMS

DEVELOPMENT AND IMPLEMENTATION OF INNOVATIVE ALGORITHMS FOR HIGH FRAME RATE ULTRASOUND

ACADEMIC DISCIPLINE (SSD) ING-INF/01

Candidate
Giulio Bonciani

Supervisors
Prof. Enrico Boni
Prof. Alessandro Ramalli
Prof. Piero Tortoli
Prof. François Varray
Prof. Damien Garcia

Ph.D. Coordinator
Prof. Stefano Ricci

CICLO XXXVIII, 2022/2025

Université Claude Bernard Lyon 1 CREATIS

THÈSE de DOCTORAT DE L'UNIVERSITÉ CLAUDE BERNARD LYON 1

En cotutelle avec

UNIVERSITÀ DEGLI STUDI DI FIRENZE

Ecole Doctorale N° 162

MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE

Discipline: Acoustique

Soutenue publiquement le 07/05/2026, par:

Giulio BONCIANI

DÉVELOPPEMENT ET MISE EN ŒUVRE D'ALGORITHMES INNOVANTS POUR L'IMAGERIE ULTRASONORE À HAUTE FRÉQUENCE

Devant le jury composé de :

Hervé Liebgott	Professeur / Université Claude Bernard Lyon 1	**Président**
Fadnes Solveig	Chercheure / Norwegian University of Science and Technology	**Rapporteure**
Simona Turco	Professeur assistant / Eindhoven University of Technology	**Rapporteure**
Massimiliano Pieraccini	Professeur des universités / Università degli Studi di Firenze	**Examinateur**
Damien Garcia	Directeur de recherche / Université Claude Bernard Lyon 1	**Directeur de thèse**
Alessandro Ramalli	Professeur associé / Università degli Studi di Firenze	**Directeur de thèse**
François Varray	Professeur associé / Université Claude Bernard Lyon 1	**Invité**
Enrico Boni	Professeur associé / Università degli Studi di Firenze	**Invité**

Università degli Studi di Firenze,
Dipartimento di Ingegneria dell'Informazione (DINFO).

Thesis submitted in partial fulfilment of the requirements for the degree
of Doctor of Philosophy in Information Engineering.

Abstract

High-frame-rate (HFR) ultrasound imaging has revolutionized the way dynamic phenomena in tissues and blood flow are observed, enabling the reconstruction of thousands of images per second through unfocused transmissions such as plane or diverging waves. However, the real-time implementation of advanced HFR techniques, such as Color Flow Mapping (CFM), Power Doppler (PD), and Vector Doppler Imaging (VDI), requires high computational power and efficient data handling, which are difficult to achieve with conventional ultrasound systems.

The aim of this Ph.D. work was the development of a heterogeneous ultrasound platform capable of combining the advantages of hardware-oriented systems (based on FPGAs and DSPs) and software-oriented systems (based on GPUs), in order to enable the real-time implementation of complex ultrasound imaging algorithms. To this end, a GPU module was integrated into the hardware-oriented ULA-OP 256 research system, developed at the MSDLab of the University of Florence, resulting in the heterogeneous system ULA-OP 256G.

As a case study, the system was first validated by implementing the HFR CFM method on the GPU, which exhibited real-time performance superior to DSP-based implementations.

Subsequently, the Vector Doppler Imaging method was implemented in real time on the ULA-OP 256G platform, enabling the reconstruction of bidimensional velocity fields by combining information from two different observation directions.

In addition, an alias-free Doppler technique based on staggered pulse repetition frequency sequences (Staggered Pulse Repetition Frequency, S-PRF) was implemented, extending the Nyquist limit and correcting aliasing artifacts in real time.

Finally, a real-time Singular Value Decomposition (SVD) filtering framework was developed to improve clutter suppression and enhance sensitivity to weak blood signals, demonstrating the feasibility of advanced spatiotemporal filtering directly on the embedded GPU.

Experimental validations, performed on flow phantoms using linear probes, confirmed the ability of the developed methods to produce accurate and alias-free velocity maps. The proposed solutions demonstrate the

potential of heterogeneous architectures for advanced ultrasound imaging and represent a significant step toward the clinical translation of real-time HFR Doppler techniques.

Sommario

L'imaging ecografico ad alto frame rate (High Frame Rate, HFR) ha rivoluzionato il modo di osservare i fenomeni dinamici nei tessuti e nel flusso sanguigno, consentendo la ricostruzione di migliaia di immagini al secondo mediante trasmissioni non focalizzate, come onde piane o divergenti. Tuttavia, la realizzazione in tempo reale di tecniche HFR avanzate, come il Color Flow Mapping (CFM), il Power Doppler (PD) e il Vector Doppler Imaging (VDI), richiede un'elevata potenza di calcolo e una gestione efficiente dei dati, difficilmente ottenibili con i sistemi ecografici convenzionali.

L'obiettivo di questo lavoro di dottorato è stato lo sviluppo di una piattaforma ecografica eterogenea capace di combinare i vantaggi dei sistemi hardware-oriented (basati su FPGA e DSP) e software-oriented (basati su GPU), al fine di consentire l'implementazione in tempo reale di algoritmi complessi di imaging ad ultrasuoni. A tal fine, è stato integrato un modulo GPU nel sistema di ricerca hardware-oriented ULA-OP 256, sviluppato presso il laboratorio MSDLab dell'Università di Firenze, realizzando la versione eterogenea ULA-OP 256G.

Come caso di studio, il sistema è stato inizialmente validato implementando su GPU il metodo HFR CFM, che ha mostrato prestazioni in tempo reale superiori rispetto alle versioni basate su DSP.

Successivamente, sulla piattaforma ULA-OP 256G è stato realizzato il metodo Vector Doppler Imaging in tempo reale, che ricostruisce il campo di velocità bidimensionale combinando le informazioni provenienti da due direzioni di osservazione.

Inoltre, è stata implementata una tecnica Doppler alias-free basata su sequenze staggered di frequenze di ripetizione d'impulso (Staggered Pulse Repetition Frequency, S-PRF), in grado di estendere il limite di Nyquist e correggere gli artefatti di aliasing in tempo reale.

Infine, è stato sviluppato un framework di filtraggio in tempo reale basato sulla decomposizione ai valori singolari (SVD), con l'obiettivo di migliorare la soppressione del clutter e aumentare la sensibilità ai segnali

ematici deboli, dimostrando la fattibilità dell'elaborazione spaziotemporale avanzata direttamente sulla GPU integrata.

Le validazioni sperimentali, condotte su phantom di flusso utilizzando sonde lineari, hanno confermato la capacità dei metodi sviluppati di produrre mappe di velocità accurate e prive di aliasing, con frame rate superiori al kilohertz. Le soluzioni proposte dimostrano il potenziale delle architetture eterogenee per l'imaging ecografico avanzato e rappresentano un passo significativo verso la traslazione clinica delle tecniche Doppler HFR in tempo reale.

Résumé

L'imagerie ultrasonore à haut taux d'images (High Frame Rate, HFR) a révolutionné la manière d'observer les phénomènes dynamiques dans les tissus et dans le flux sanguin, en permettant la reconstruction de milliers d'images par seconde à partir de transmissions non focalisées, telles que les ondes planes ou divergentes. Cependant, la mise en œuvre en temps réel de techniques HFR avancées, telles que le Color Flow Mapping (CFM), le Power Doppler (PD) et le Vector Doppler Imaging (VDI), nécessite une puissance de calcul élevée et une gestion efficace des données, difficiles à atteindre avec les systèmes échographiques conventionnels.

L'objectif de ce travail de doctorat a été le développement d'une plateforme échographique hétérogène capable de combiner les avantages des systèmes orientés matériel (basés sur FPGA et DSP) et des systèmes orientés logiciel (basés sur GPU), afin de permettre l'implémentation en temps réel d'algorithmes complexes d'imagerie ultrasonore. À cet effet, un module GPU a été intégré au système de recherche orienté matériel ULA-OP 256, développé au laboratoire MSDLab de l'Université de Florence, donnant naissance à la version hétérogène ULA-OP 256G.

À titre d'étude de cas, le système a d'abord été validé en implémentant sur GPU la méthode HFR CFM, qui a montré des performances en temps réel supérieures à celles des versions basées sur DSP.

Par la suite, la méthode de Vector Doppler Imaging en temps réel a été réalisée sur la plateforme ULA-OP 256G, permettant la reconstruction du

champ de vitesse bidimensionnel en combinant les informations provenant de deux directions d'observation.

De plus, une technique Doppler sans repliement spectral (alias-free) basée sur des séquences staggered de fréquences de répétition d'impulsions (Staggered Pulse Repetition Frequency, S-PRF) a été implémentée, capable d'étendre la limite de Nyquist et de corriger en temps réel les artéfacts de repliement.

Enfin, un cadre de filtrage en temps réel basé sur la décomposition en valeurs singulières (SVD) a été développé afin d'améliorer la suppression du clutter et d'accroître la sensibilité aux signaux sanguins faibles, démontrant la faisabilité d'un traitement spatio-temporel avancé directement sur le GPU intégré.

Les validations expérimentales, menées sur des fantômes de flux à l'aide de sondes linéaires, ont confirmé la capacité des méthodes développées à produire des cartes de vitesse précises et exemptes de repliement, avec des taux d'images supérieurs au kilohertz. Les solutions proposées démontrent le potentiel des architectures hétérogènes pour l'imagerie ultrasonore avancée et représentent une étape significative vers la translation clinique des techniques Doppler HFR en temps réel.

Alla mia famiglia

October 2025

Acknowledgments

As I conclude this thesis, I am deeply aware of the many people who have made this work possible, and I would like to express my sincere gratitude to all of them.

To my professors at the University of Florence: Alessandro Ramalli, Piero Tortoli, Francesco Guidi, and Enrico Boni. More than professors, you have been true mentors. You taught me not only the rigor of research but also the perseverance required to succeed, combining hard work with lighthearted and cheerful moments. Thank you for your unwavering support throughout these three years, for always keeping your doors open, and for allowing me to discover the world of research.

To my professors at the University of Lyon: François Varray and Damien Garcia. Arriving in Lyon alone was not easy, but you made me feel at home from the very first day. Thank you for welcoming me, for your patience with my shaky French, and for your kindness.

To all the friends at CREATIS: thank you for all the moments shared together and for making those months away from home much lighter.

To my adventure companions at the MSDLab: I have shared some of the best moments of these years with you. Your constant help and support, the countless trays of pastries eaten in the lab, the conferences where we seemed more on a trip than at work, the dinners and the aperitifs, you are the proof that work can also be fun. Thank you for the laughs, for the advice, and for being more than colleagues but true friends.

To Mum and Dad: What can I say? You have always believed in me and allowed me to choose freely, even when the path seemed uncertain. Thank you for making me feel that, no matter what happened, I would always have a home to return to. I love you.

Thanks to Luisa's family, my second family, and to all my own family, for always supporting me with their love and affection.

A special thought goes to my friends Lorenzo, Riccardo, and Matteo, who brought balance to the hardest days and joy to the brightest ones.

And to you, Luisa, my love. You have endured and supported me unfailingly. Thank you for being my rock, my joy, and my certainty. Without you, none of this would have made sense.

To all of you: thank you.

Ringraziamenti

Mentre concludo questa tesi, mi rendo conto di quante persone hanno reso possibile tutto questo. E voglio dirvi grazie.

Ai miei professori dell'università di Firenze: Alessandro Ramalli, Piero Tortoli, Francesco Guidi, ed Enrico Boni. Ma più che professori, siete stati delle guide. Mi avete insegnato a fare ricerca, ma anche a non mollare, combinando duro lavoro a momenti leggeri e scherzosi. Grazie per avermi supportato in questi tre anni, per le vostre porte sempre aperte e per avermi permesso di conoscere il mondo della ricerca.

Ai miei professori dell'università di Lione: François Varray e Damien Garcia. Arrivare a Lione da solo non è stato facile, ma voi mi avete fatto sentire a casa dal primo giorno. Grazie per avermi accolto, per la pazienza con il mio francese traballante e per la vostra gentilezza.

A tutti gli amici del CREATIS: grazie per i momenti passati insieme e per aver reso quei mesi lontano da casa più leggeri.

Ai miei compagni di avventura del MSDLab: con voi ho vissuto i momenti più belli di questi anni. L'instancabile sostegno e supporto che mi avete dedicato, gli innumerevoli vassoi di paste mangiati in laboratorio, i congressi dove sembravamo più in gita che al lavoro, le cene e gli aperitivi. Siete la dimostrazione che il lavoro può essere anche divertimento. Grazie per le risate, per i consigli e per essere stati, oltre che dei colleghi, veri amici.

Mamma e babbo, cosa posso dirvi? Avete sempre creduto in me. Mi avete lasciato libero di scegliere, anche quando la strada sembrava incerta. Grazie per avermi fatto sentire che, qualunque cosa succedesse, avevo sempre casa dove tornare. Vi voglio bene.

Un grazie alla famiglia di Luisa, la mia seconda famiglia, e a tutti i miei parenti, per il loro costante sostegno, il loro affetto e il loro amore.

Un pensiero speciale ai miei amici Lorenzo, Riccardo e Matteo, che hanno portato equilibrio nei giorni più difficili e gioia in quelli più sereni.

E a te, Luisa, amore mio. Tu che mi hai sempre sopportato e sostenuto. Grazie per essere stata la mia roccia, la mia gioia, la mia certezza. Senza di te, niente di tutto questo avrebbe avuto senso.

A tutti voi: Grazie.

Contents

List of abbreviations

1-D	One-dimensional
2-D	Two-dimensional
3-D	Three-dimensional
CFM	Color flow mapping
CMUT	Capacitive micromachined US transducer
CNR	Contrast-to-noise ratio
CP	Co-processing
CPU	Central processing unit
DAS	Delay-and-sum
DDR3	Double Data Rate 3
DSP	Digital signal processor
DW	Diverging wave
FE	Front end
FIFO	First-in-first-out
FIR	Finite impulse response
FPGA	Field-programmable gate array
FR	Frame rate
GPU	graphics processing unit
HFR	High-frame-rate
HPF	High-pass filter
HS	Hop size
IIR	Infinite impulse response
InSAR	Interferometric synthetic aperture radar
IQ	In-phase and quadrature
LCM	Least common multiple
MC	Main control
MSDLab	Microelectronics systems design laboratory
NF	Not feasible
NRMSE	Normalized root mean square error
OS	Operating system
PC	Personal computer
PCIe	Peripheral component interconnect express
PD	Power Doppler
PMUT	Piezoelectric micromachined ultrasonic transducer

PRF	Pulse repetition frequency
PRI	Pulse repetition interval
PS	Packet size
PW	Plane wave
RF	Radiofrequency
RMS	Root-mean-square
RMSE	Root-mean-square error
ROI	Region of interest
RPM	Revolution per minute
rSVD	Randomized singular value decomposition
RX	Reception
SDRAM	Synchronous Dynamic Random-Access Memory
SF	Sliding factor
SoC	System-on-chip
SoM	System-on-module
S-PRF	Staggered pulse repetition frequency
SRIO	SerialRapidIO
SSM	Spatial similarity matrix
sSVD	Simplified singular value decomposition
SVD	Singular value decomposition
TX	Transmission
ULA-OP 256	256-channel ultrasound advanced open platform
ULM	Ultrasound localization microscopy
UPD	Ultrafast power Doppler
US	Ultrasound
USB	Universal serial bus
VDI	Vector Doppler imaging

Introduction

Ultrasound (US) imaging is widely used as a diagnostic modality [1] for medical applications, including cardiology, vascular imaging, obstetrics, and emergency medicine, because it is non-invasive, and it does not involve ionizing radiation.

The last two decades have been characterized by the introduction of an impressive number of novel US methods capable of increasing the attractiveness of this imaging modality. Among these methods, a major milestone was high-frame-rate (HFR) imaging, which employs unfocused transmissions, such as plane waves or diverging waves [2,3], and parallelized reception strategies, to reconstruct thousands of images per second. Such a high temporal resolution, in turn, has enabled the development of a variety of advanced imaging techniques that provide deeper insights into dynamic biological processes, including fast blood flow events and rapid tissue motion. Among them, HFR color flow mapping (CFM) enhances sensitivity to low-velocity flows and improves the temporal characterization of hemodynamics. Power Doppler (PD) imaging, building upon HFR acquisition, increases the sensitivity to weak vascular signals and has become particularly relevant for microvascular studies. Vector Doppler imaging (VDI) extends conventional Doppler by reconstructing two-dimensional velocity vectors, thus overcoming the limitations of angle dependence and enabling a more comprehensive assessment of blood flow patterns. More recently, ultrafast Doppler imaging combined with advanced filtering strategies, such as singular value decomposition (SVD), has opened the possibility of visualizing the microvasculature at a level of detail comparable to other high-resolution modalities.

Although the advantages of these methods are consolidated, their use in clinical practice remains limited. The real-time implementation of these methods requires high computational power to process large volumes of echo data. Furthermore, sophisticated filtering and processing algorithms require levels of flexibility and computing power that exceed the capabilities of conventional scanners. Addressing this challenge represents an important research direction in ultrasound imaging.

Open research scanners [4], i.e., platforms that allow researchers to program arbitrary transmission sequences, access raw data at multiple points in the processing chain, and implement custom algorithms, are ideal for validation and development of novel US methods. These systems are generally grouped into hardware- and software-oriented open scanners. The *hardware-oriented open scanners* [5–10] embed field-programmable gate arrays (FPGAs), digital signal processors (DSPs), or systems-on-chip (SoCs), optimal for real-time data processing, with high throughput and deterministic performance. However, they require expertise in low-level programming for efficient implementation of computationally demanding algorithms.

On the other hand, *software-oriented open scanners* minimize the embedded hardware, and transfer raw echo data directly to a host computer equipped with graphics processing units (GPUs) [11–14]. These systems benefit from massive parallelization and ease of programming through high-level languages such as C++ and CUDA [15–18]. Although their flexibility makes them particularly attractive for algorithm prototyping, the transfer of the large amounts of data generated by HFR acquisitions to an external computer introduces bandwidth and latency bottlenecks, limiting real-time performance [19,20].

Both approaches have significantly contributed to the field. Hardware-oriented platforms have enabled real-time demonstrations of several HFR techniques, while software-oriented platforms have facilitated the exploration of innovative methods offline.

The main objective of this PhD project is the development of a heterogeneous architecture that combines the advantages of both paradigms. Through the introduction of a GPU module into the ULA-OP 256 scanner [9], interfacing with FPGAs and DSPs, it becomes possible to perform the most time-critical tasks in dedicated hardware while leveraging the GPU for computationally demanding algorithms. This integration allows for real-time implementation of processing methods such as SVD filtering and vector Doppler imaging, which would otherwise remain confined to offline research.

The manuscript is organized as follows:

- *Chapter 1* presents the physics of ultrasound propagation, the characteristics of transducers and probes, and the fundamentals of US imaging. It also reviews Doppler methods, with emphasis on color flow mapping, power Doppler, and vector Doppler imaging.
- *Chapter 2* describes the development of the heterogeneous ULA-OP 256G system, derived from previous ULA-OP 256. Hardware, firmware, and software modifications that were required to integrate a GPU module, are reported. The platform was validated through a proof-of-concept implementation of HFR color flow mapping.
- *Chapter 3* describes the real-time implementation of a vector Doppler imaging method on the ULA-OP 256G, and its performance is evaluated through experimental tests.
- *Chapter 4* presents real-time implementations of alias-free color and vector Doppler imaging based on the staggered PRF concept.
- *Chapter 5* addresses computationally efficient SVD filtering for ultrafast Doppler imaging and its GPU-based implementation.
- *Chapter 6* concludes the thesis and gives the perspectives for future work.

Contributions

Conference Proceedings

- ***G. Bonciani**, A. Ramalli, A. Bernard, F. Guidi, P. Tortoli, E. Boni, D. Garcia, and F. Varray, "Real-time, high-frame-rate, vector Doppler ultrasound imaging by a hybrid open platform," 2024 IEEE Ultrasonics, Ferroelectrics, and Frequency Control Joint Symposium (UFFC-JS), Taipei, Taiwan, 2024, pp. 1-3, doi: 10.1109/UFFC-JS60046.2024.10793839.*
- ***G. Bonciani**, F. Guidi, C. Giangrossi, F. Varray, D. Garcia, E. Boni, and A. Ramalli, "Real-Time High Frame Rate Color Doppler Imaging with Staggered Pulse Repetition Frequency," 2025 IEEE International Ultrasonics Symposium (IUS), Utrecht, Netherlands, 2025.*

Journal papers

- ***G. Bonciani**, F. Guidi, P. Tortoli, C. Giangrossi, A. Dallai, E. Boni, and A. Ramalli, "A Heterogeneous Ultrasound Open Scanner for the Real-Time Implementation of Computationally Demanding Imaging Methods," in IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 72, no. 1, pp. 100-108, Jan. 2025, doi: 10.1109/TUFFC.2024.3474091.*
- *B. Pialot, F. Guidi, **G. Bonciani**, F. Varray, T. Loupas, P. Tortoli, and A. Ramalli, "Computationally Efficient SVD Filtering for Ultrasound Flow Imaging and Real-Time Application to Ultrafast Doppler," in IEEE Transactions on Biomedical Engineering, vol. 72, no. 3, pp. 921-929, March 2025, doi: 10.1109/TBME.2024.3479414.*
- *P. Tortoli, F. Lagonigro, A. Ramalli, **G. Bonciani**, B. Y. S. Yiu, and A. C. H. Yu, "Design of a Versatile Flow Phantom for Multi-Angle Multi-Depth Calibration", to be submitted to IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 2025.*

Chapter 1. Ultrasound Basics

This Chapter presents the fundamental physical principles of ultrasound imaging, the main characteristics of transducers and probes, and the evolution of imaging strategies from conventional (focused beam) methods to modern high frame rate techniques. The objective is to provide a comprehensive background to support the methods and implementations discussed in subsequent chapters.

_____

1.1 Ultrasound propagation

US refers to mechanical waves generated by the vibration of a material at frequencies above 20 kHz, exceeding the upper threshold of human hearing. These waves propagate through elastic media, and their propagation depends on the physical properties of the medium, such as density and elasticity, and the characteristics of the wave, as frequency, wavelength, amplitude, and propagation speed.

Among the various types of mechanical waves, longitudinal waves are the most relevant for medical US applications. Also known as compression waves, they are characterized by particle motion that occurs in the same direction as wave propagation, producing alternating regions of compression and rarefaction. Longitudinal waves can propagate in solids, liquids, and gases, but the speed of sound varies significantly depending on the medium's elastic and inertial properties. The speed of sound c is defined as:

$$c = \sqrt{\frac{K}{\rho}} \tag{1}$$

where K $[N/m^2]$ is the bulk modulus (elastic constant) and ρ $[kg/m^3]$ is the volumetric density of the medium.

In a homogeneous and lossless medium, the propagation of a longitudinal wave along the axial direction z can be described by:

$$T(z,t) = T_0 \cdot e^{j(2\pi f t \pm kz)} \tag{2}$$

where T $[N/m^2]$ is the acoustic stress, T_0 is the maximum value of pressure (modulus), f $[Hz]$ is the wave frequency, and k is the wave number. The frequency is related to the speed of sound and the wavelength λ $[m]$ by:

$$f = \frac{c}{\lambda} \tag{3}$$

This formulation assumes a linear, isotropic, and lossless medium, and it is derived from the conservation of momentum:

$$\rho \frac{\partial \vec{u}}{\partial t} + \nabla p = 0 \tag{4}$$

where $\vec{u}$ is the particle velocity and p is the acoustic pressure.

When an US wave propagates through a medium composed of different materials, its behavior is influenced by the acoustic impedance ζ, defined as:

$$\zeta = \rho \cdot c \tag{5}$$

Acoustic impedance governs the transmission and reflection of waves at interfaces and is expressed in $Rayl$ ($1\,Rayl = 1\ kg/(m^2 s)$). In homogeneous media, wavefronts propagate linearly; however, in the presence of an impedance discontinuity smaller than the wavelength, wave energy is scattered isotropically, this is known as Rayleigh scattering. The scattering cross-section σ is defined as:

$$\sigma = \frac{S}{I} \tag{6}$$

where S is the total scattered power and I is the intensity of the incident wave.

If the interface is smooth (i.e., its roughness is larger than the wavelength), part of the wave is transmitted into the second medium while the rest is reflected (Fig. 1.1). The angles of incidence and transmission follow Snell's law:

$$\frac{\sin \theta_I}{\sin \theta_T} = \frac{c_1}{c_2} \tag{7}$$

where c_1 and c_2 are the sound speeds in the respective media, and θ_I and θ_T are the angles of incidence and transmission.

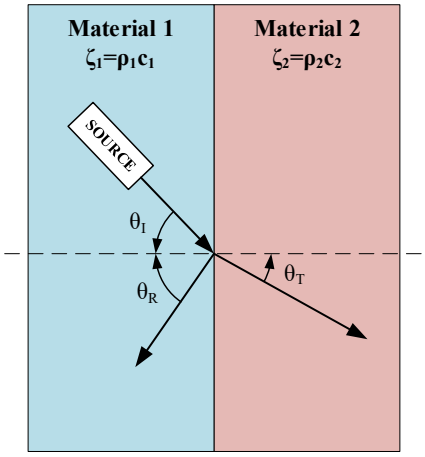


Fig. 1.1: Phenomenon of reflection and transmission of ultrasonic waves at the interface between two different materials.

The energy partition between the reflected and transmitted waves is quantified by the reflection R and transmission T coefficients:

$$R = \frac{(\zeta_2 \cos\theta_I - \zeta_1 \cos\theta_T)^2}{(\zeta_1 \cos\theta_T + \zeta_2 \cos\theta_I)^2} \tag{8}$$

$$T = 1 - R = \frac{(4\zeta_1\zeta_2 \cos\theta_I \cos\theta_T)^2}{(\zeta_1 \cos\theta_T + \zeta_2 \cos\theta_I)^2} \tag{9}$$

where ζ_1 and ζ_2 are the acoustic impedances of the two media.

Another important factor in US propagation is attenuation, which causes a reduction in wave intensity as it travels through the medium. The main mechanisms of attenuation are:

- *Absorption*, where mechanical energy is converted into heat;
- *Dispersion*, caused by multiple scattering, reflection, and refraction events.

The attenuation coefficient μ_A acts on the propagation of the signal and its intensity decreases exponentially as follows:

$$I = I_0 e^{-2\mu_A z} \tag{10}$$

where z is the propagation distance in centimeters, I_0 is the initial intensity.

The attenuation coefficient $\mu_{A_{dB}}$, expressed in $dB/(cm \cdot MHz)$, depends on both the propagation depth and the wave frequency, and is typically defined as:

$$\mu_{A_{dB}} = \frac{10}{z} \log \frac{I}{I_0} \tag{11}$$

1.2 Transducers and probes

US transducers are devices that convert electrical energy into acoustic energy and vice versa. Transducers can be realized using different technologies, including:

- *piezoelectric transducers*, based on the piezoelectric effect;
- *capacitive micromachined US transducers* (CMUTs), which use electrostatic actuation and sensing;
- *piezoelectric micromachined ultrasonic transducers* (PMUTs), which utilize thin-film piezoelectric materials on micromachined membranes.

Despite the emergence of CMUTs and PMUTs, piezoelectric transducers remain the most prevalent technology for constructing US probes. The following sections provide a detailed explanation of the operating principles of piezoelectric transducers.

The term "probe" usually identifies the complete assembly that houses one or more transducers together with the backing, matching, and protective layers, as well as the electrical wiring, cable, and connector required to interface with the ultrasound system. Probes can be designed as single-element devices, typically used in one-dimensional (1-D) applications, or as multi-element arrays, which enable electronic beam steering and focusing for two-dimensional (2-D) applications.

1.2.1 Piezoelectric transducers

Piezoelectric transducers operate based on the piezoelectric effect, which enables bidirectional energy conversion between electrical and mechanical domains. The most common used material is the lead zirconate titanate, known as PZT. The piezoelectric effect includes:

- The *direct piezoelectric effect*, in which mechanical stress applied to a piezoelectric material generates an electric charge proportional to the deformation.
- The *inverse piezoelectric effect*, in which an applied electric field induces mechanical deformation in the material.

These two effects are exploited in US imaging where pulses of ultrasonic waves are transmitted into the medium. During the transmission (TX), the pulse generator emits a brief electrical pulse to excite the transducer which, with the *inverse piezoelectric effect*, generates acoustic waves, while during reception (RX), the *direct piezoelectric effect* allows the transducer to detect returning echoes.

The acoustic performance of a piezoelectric element is significantly influenced by its geometry and physical properties. In particular, the element thickness d determines its fundamental resonance frequency f_R, given by:

$$f_R = \frac{c_{piezo}}{2d} \tag{12}$$

where c_{piezo} is the speed of sound in the piezoelectric material. Thinner elements resonate at higher frequencies, which is desirable for high-resolution imaging of superficial structures.

Although the transducer operates most efficiently at f_R, it can transmit and receive over a broader frequency range. This operational frequency range is defined by the −3 dB points bandwidth, and the *relative bandwidth* is expressed as:

$$\Delta B_{\%} = 100 \cdot \frac{f_{max} - f_{min}}{f_R} \tag{13}$$

where $f_{min} < f_R < f_{max}$.

The transducer's bandwidth represents a critical parameter, as it not only defines the operative frequency range, but also determines its capability to generate and detect short-duration pulses. This capability is intrinsically linked to *axial resolution*, i.e., the ability to resolve closely

spaced reflectors along the US propagation axis. This parameter is directly proportional to the transmit frequency and to the transducer bandwidth.

Another important parameter of the transducer is the *lateral resolution*, defined as the minimum distance between two objects located on planes orthogonal to the transducer surface for which the individual echoes can still be distinguished. This parameter is strongly influenced by beam focusing characteristics, transducer geometry, and transmit frequency.

To improve acoustic transmission between the transducer and biological tissues, a *matching layer* is inserted between the piezoelectric element and the tissue. This layer has an intermediate acoustic impedance to reduce reflection losses caused by the large impedance mismatch between the ceramic material (typically 25–35 MRayl) and soft tissue (around 1.5–2 MRayl). The layer's thickness and acoustic properties are designed to maximize energy transfer and optimize frequency response.

On the opposite side, a *backing* layer is used to absorb backward-propagating waves and damp transducer ringing. The choice of backing material, with appropriate impedance and attenuation characteristics, enhances the temporal response and widens the transducer bandwidth.

1.2.2 Transducer arrays

Modern US systems no longer rely on single-element transducers. Instead, they commonly employ array probes, which represent a fundamental advancement in US technology thanks to their more sophisticated architecture and significantly greater versatility. By arranging multiple piezoelectric elements in a regular geometry, array probes allow independent electronic control of each element, enabling a wide range of imaging strategies without mechanical motion.

1-D linear arrays are the most widely used in medical US because of their simplicity, versatility, and ability to deliver high-resolution images of superficial anatomical structures. A linear array consists of a large number of small rectangular piezoelectric elements aligned along a single axis, with a regular center-to-center spacing referred to as the *pitch*. Each element is acoustically and electrically isolated to minimize cross-talk and improve sensitivity. In the elevation plane, as in single-element probes, an acoustic lens can be employed to fix the focal depth. The main design

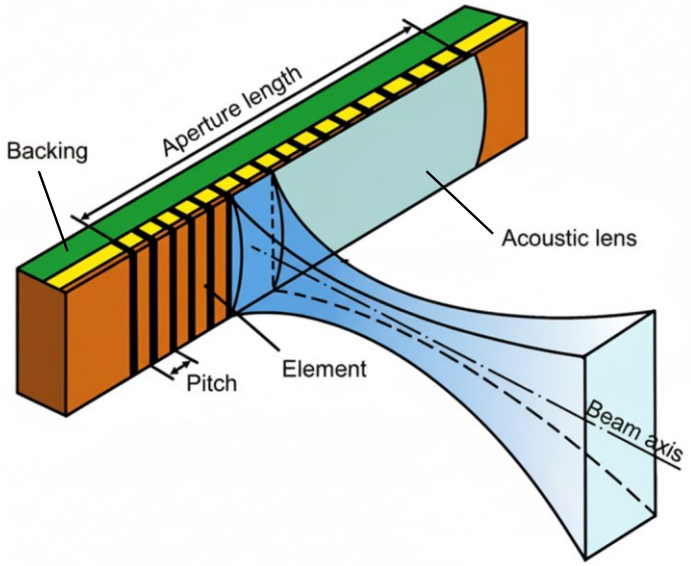


Fig. 1.2: Linear array transducer basic structure.

parameters of an array probe include the number of elements, the aperture size, the pitch, and the central frequency, which together determine the resolution, field of view, and penetration depth of the system. A schematic representation of a linear array is shown in Fig. 1.2.

1-D probes can be equipped with movement systems to extend the field of view, but scanning volumes without moving the probe is preferable. Multi-plane imaging at different angles is important for large organs. 2-D probes, with thousands of elements arranged in square or rectangular arrays, have been developed to enhance scanning capabilities.

1.3 Fundamentals of ultrasound imaging

US imaging relies on the transmission of short acoustic pulses into biological tissues and the reception of backscattered echoes. These echoes encode information about the acoustic properties of the medium and can be used to reconstruct images or extract functional parameters. Fig. 1.3 shows the general blocks to generate US images. The following sections describe the progression of US imaging, starting from 1-D techniques up to 2-D imaging.

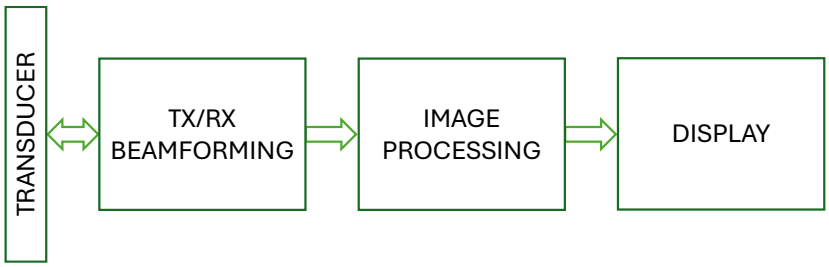


Fig. 1.3: General scheme of US imaging.

1.3.1 One-dimensional ultrasound imaging

In 1-D US imaging, the system interrogates tissue along a single line of sight. This modality forms the basis of several diagnostic techniques and relies on both transmission and reception beamforming.

During transmission, a focused acoustic beam is generated by exciting a subset of transducer elements with appropriately delayed signals. This electronic focusing concentrates the acoustic energy at a selected depth. In particular, in *pulsed-wave* mode the same subset of elements is used for alternating transmission and reception, transmitting short bursts of ultrasound at regular intervals. On reception, the echoes returning from tissue interfaces are detected by the array elements. These signals are then delayed and coherently summed using delay-and-sum (DAS) beamforming, with *dynamic receive focusing* applied to maintain optimal resolution across different depths.

For clarity, Fig. 1.4 illustrates an example in which three scatterers are positioned along the probe axis at different depths. The echo signals received by each transducer element are stored in rows of a matrix (shown on the right), with each colored cell representing the signal originating from the corresponding scatterer. To accurately reconstruct the image, it is necessary to time-align the signals and sum them coherently across all elements. This step ensures that the contributions from all elements combine constructively at each depth, compensating for the differences in echo time of flight and enhancing the overall signal quality [21].

A single scan line corresponds to an A-mode signal, in which the echoes received by the probe are displayed as relative amplitudes along the line of

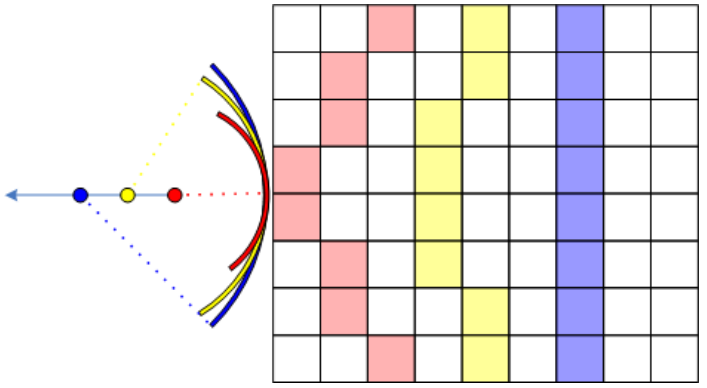


Fig. 1.4: Dynamic focalization in reception.

sight. The amplitude of each echo reflects the relative echogenicity of the structures encountered and is recorded as a function of time. Extending this concept, the same line can be visualized in M-mode, where the echoes are plotted over time to observe motion along a single line of investigation. In M-mode, the position of echoes shifts as structures move, and successive lines are acquired at intervals defined by the *pulse repetition interval* (PRI). The depth of each reflector is estimated from the *time-of-flight*, that is, the delay between transmission and reception of the corresponding echo, assuming a constant speed of sound (c_m) in tissue (typically 1540 m/s). The frequency at which pulses are transmitted, known as the *pulse repetition frequency* (PRF) which is the inverse of the PRI, governs both the maximum imaging depth (d_M) and the temporal resolution. Higher PRF allows faster imaging but reduces the depth range, as echoes must be received before the next pulse is emitted. The PRI depends on c_m and d_M as in the following formula:

$$PRI = \frac{2d_M}{c_m} \tag{14}$$

These 1-D imaging modalities provide the foundation for more advanced techniques. By acquiring multiple scan lines across a region of interest (ROI), the system can reconstruct 2-D images, which form the basis for the transition to 2-D imaging, described in the following section.

1.3.2 Two-dimensional imaging

While 1-D ultrasound imaging provides valuable information along a single line of investigation, most clinical applications require visualization of anatomical structures across a plane. This is achieved by acquiring multiple adjacent scan lines with array transducers, composed of multiple piezoelectric elements arranged linearly or curvilinearly. By electronically controlling the excitation and reception of these elements, the system can steer and dynamically focus the acoustic beam across the imaging plane.

Three main scanning geometries are commonly used:

- *Linear Scanning*: Parallel scan lines are generated using a linear array. The field of view is rectangular, ideal for superficial structures like vessels and muscles.
- *Convex Scanning*: A curved array produces diverging scan lines, resulting in a sector-shaped image. This configuration is suited for abdominal and obstetric imaging.
- *Phased Array Scanning*: A small footprint array steers the beam electronically to produce a fan-shaped image. It is particularly useful for cardiac imaging through narrow acoustic windows.

In 2-D imaging, dynamic transmit and receive focusing improve lateral resolution and sensitivity throughout the field of view. However, the sequential nature of line-by-line scanning inherently limits the temporal resolution. The maximum achievable *frame rate* (FR) is constrained by the number of lines N and the PRI:

$$FR = \frac{1}{N \cdot PRI} \tag{15}$$

As a consequence, the frame rate typically ranges from a tens to a few hundreds of frames per second, which may be insufficient for tracking fast physiological dynamics such as blood flow, tissue motion, or cardiac activity.

B-mode imaging

Brightness mode (B-mode) is the most widely used ultrasound imaging modality. In this mode, echoes from tissue interfaces are detected, and their amplitude is mapped to pixel brightness in a 2-D image. The intensity of each pixel is proportional to the echo amplitude, which depends on acoustic impedance mismatches and scattering properties of tissue. This allows clinicians to visualize anatomical structures such as vessel walls, organ boundaries, and pathological changes. B-mode provides the morphological background upon which functional imaging methods, such as Doppler techniques, are superimposed. In clinical practice, Doppler information is usually displayed in combination with B-mode, enabling simultaneous assessment of both anatomy and blood flow.

1.4 Doppler imaging

The so-called Doppler techniques are used to assess the velocity of blood flow within vessels. During such examinations, an ultrasound probe is typically placed on the skin near the vessel of interest. The transmitted ultrasound waves are backscattered by moving red blood cells, which act as isotropic scatterers. The returning echoes are collected by the probe, suitably processed, and ultimately displayed by the ultrasound system, allowing clinicians to extract quantitative information about blood flow dynamics.

1.4.1 Pulsed wave Doppler systems

As already introduced, pulsed wave systems emit ultrasound bursts at intervals defined by the PRI. In *single-gate Doppler* (Fig. 1.5), the velocity at a specific location is estimated by transmitting a sequence of short pulses and analyzing the echoes from moving scatterers. The velocity estimation is encoded in the *phase shift* of the received signal over successive pulses, rather than in a direct frequency shift, which can be represented as an equivalent Doppler frequency f_D. In this approach, a transducer (acting both as transmitter and receiver, Fig. 1.6) emits an ultrasound burst toward a target located at a distance D. The target, moving with velocity v and insonified at an angle θ, scatters the acoustic wave, which is then received

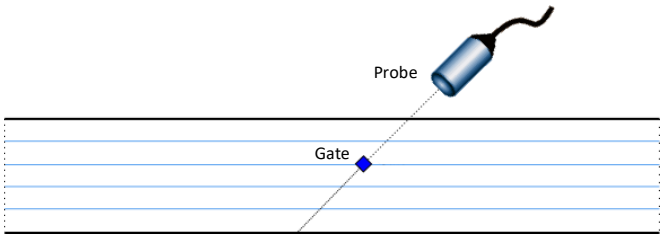


Fig. 1.5: Single gate US system.

by the transducer with a delay directly proportional to the distance D and inversely proportional to the speed of sound *c*.

Since the transmission and reception sequences are repeated with the same modality and at a cadence defined by the PRF, the signal backscattered from the target at the first transmission cycle can be expressed as:

$$s_1 = A\sin(2\pi f_{tx} t_1) \tag{16}$$

where A is the echo amplitude, f_{tx} is the transmitted frequency, and t_1 denotes the time instant corresponding to the first PRI.

In pulsed wave applications, the time interval Δt during which the target is insonified by the burst is extremely short. As result, the displacement of the target within this interval is negligible. The target can therefore be considered stationary during Δt, since its velocity is much lower than c and the burst duration is typically less than a few microseconds. Considering

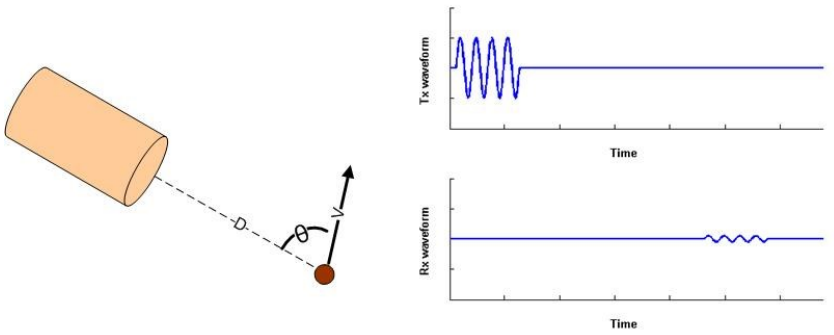


Fig. 1.6: pulsed wave approach.

these hypotheses, the instantaneous phase at generic *i*-th PRI changes and can be expressed as:

$$s_i = A\sin(2\pi f_{tx} t_i) \tag{17}$$

where t_i is the sampling time at the *i*-th PRI.

The sequences of values t_i depends on the velocity *v*, the PRI interval, the Doppler angle *θ,* and the speed of sound *c*, according to the relation:

$$t_i = t_1 - \Delta t_i = t_1 + \frac{2 \cdot i \cdot PRI \cdot v \cdot cos\theta}{c} \tag{18}$$

Indeed, due to the displacement of the target between two successive PRIs, the received signal will be advanced or delayed by the time that the ultrasound wave would have taken to travel, back and forth, across the *gap* between the two successive positions of the scatterer. The generic *i*-th sample corresponding to the depth of interest is therefore:

$$\begin{aligned} s_i &= A\sin\left[2\pi f_{tx}\left(t_1 + \frac{2 \cdot i \cdot PRI \cdot v \cdot cos\theta}{c}\right)\right] \\ &= A\sin\left(2\pi \frac{2 f_{tx} v cos\theta}{c} t_{slow} + 2\pi f_{tx} t_1\right) \end{aligned} \tag{19}$$

The temporal sequence of samples corresponding to the same depth at successive PRIs is indexed by the discrete variable $t_{slow} = i \cdot PRI$, and consists of a sinusoid whose frequency is:

$$f_D = \frac{2 f_{tx} v cos\theta}{c} \tag{20}$$

1.4.2 Nyquist limit and aliasing

In Doppler ultrasound, the maximum detectable velocity without aliasing is determined by the PRF through the Nyquist criterion. According to this principle, the highest frequency that can be accurately measured is

half the sampling rate. In the context of pulsed-wave Doppler, the sampling rate is set by the PRF, and the Nyquist frequency f_N is given by:

$$f_N = \frac{PRF}{2} \tag{21}$$

The Doppler information, extracted from the phase variation of echoes at successive pulses, can be represented as an equivalent frequency shift f_D, which is proportional to the axial velocity of moving scatterers. To avoid aliasing, this frequency must remain below f_N. When $f_D > f_N$, the measured frequency is folded back into the Nyquist interval, resulting in incorrect estimation of flow direction and velocity.

Furthermore, the Nyquist velocity V_N is inversely related to the imaging depth *d*. In fact, considering that the maximum achievable PRF depends on the time required for echoes to travel back from depth *d*, the following relation holds:

$$V_N \cdot d = \frac{c^2}{8 f_{tx}} \tag{22}$$

This expression highlights the trade-off between maximum measurable velocity and penetration depth: increasing one necessarily reduces the other.

1.4.3 Doppler methods

Doppler US has evolved significantly over the years. It is now an established tool for the assessment and monitoring of vascular diseases [22,23]. Building upon pulsed-wave Doppler, several approaches have been developed to extract velocity information from blood flow, ranging from single-point measurements to advanced two-dimensional imaging modalities.

Multi-gate Doppler (1-D Doppler, Fig. 1.7*)* [24] extends this concept by simultaneously sampling echoes at multiple depths along the same beam axis, thereby reconstructing the velocity distribution across a vessel segment. This enables visualization of flow profiles and detection of

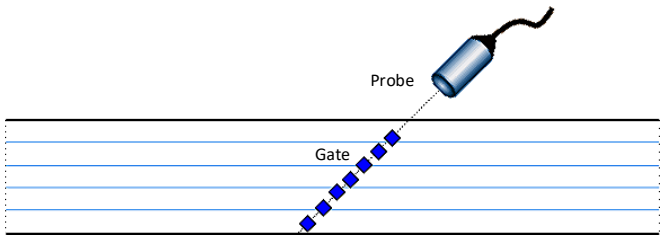


Fig. 1.7: Multi-gate US system.

localized variations, offering greater insight than single-point measurements. The temporal evolution of these depth-resolved velocities can then be displayed as a spatiotemporal map, which is particularly useful for visualizing pulsatility and transient flow phenomena.

2-D Doppler imaging generalizes velocity estimation to an entire plane, acquiring multiple scan lines to generate maps of blood flow within a ROI. The main 2-D Doppler modalities include *Power Doppler (PD), Color Flow Mapping (CFM),* and *Vector Doppler Imaging (VDI).*

Power Doppler

Fig. 1.8: Image processing scheme of PD Doppler method.

PD displays the total power of the Doppler signal rather than its mean frequency shift. Since the Doppler power is proportional to the concentration of moving scatterers, PD provides a highly sensitive representation of blood flow, especially in small vessels or in regions with slow flow. PD does not encode flow direction or velocity but instead highlights the presence and relative strength of flow.

The typical PD processing chain is illustrated in Fig. 1.8. After N PRIs, the complete set of echo data is demodulated and prepared for further processing. The demodulated in-phase and quadrature (IQ) data are first passed through a clutter filter to suppress strong low-frequency components from tissue motion. The Doppler power is then computed from

the filtered ensemble, yielding a measure of flow signal energy independent of phase. Finally, temporal persistence and spatial filtering are applied to improve image stability and reduce noise, producing the final PD image.

Due to its robustness to aliasing and angle independence, PD is widely used in clinical practice for the visualization of small vessels, perfusion imaging, and the assessment of low-velocity flow.

Clutter filter

In Doppler imaging, strong low-frequency echoes originating from tissue motion can dominate the received signal, masking the weaker blood flow components. To suppress this clutter, conventional approaches rely on temporal high-pass filters, often implemented as finite impulse response (FIR) or infinite impulse response (IIR) filters, which attenuate low-frequency signals while preserving the Doppler shifts associated with blood flow [25]. However, these methods may struggle in situations involving non-stationary tissue motion or in high-frame-rate acquisitions, where the spectral overlap between tissue and blood signals becomes more pronounced.

More advanced techniques exploit the spatiotemporal structure of the acquired data. Among them, SVD filtering has gained increasing attention. By decomposing the data matrix into singular values and vectors, tissue-related components, which are highly coherent across elements and frames, are associated with the largest singular values, while blood flow signals appear in the lower singular spectrum. By discarding the dominant singular components, tissue clutter can be effectively removed, improving sensitivity to blood flow without significantly compromising temporal resolution.

Color flow mapping

CFM imaging is an established pulsed-wave Doppler 2-D technique, widely employed in clinical diagnostics for the qualitative visualization of blood flow [26]. Each scan line is processed to extract the Doppler frequency shift, which is then mapped to a color scale, where red and blue

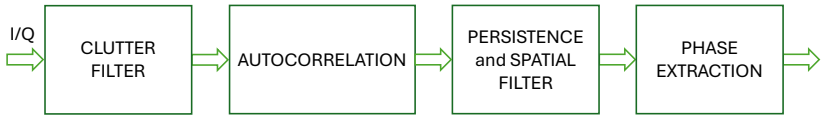


Fig. 1.9: Image processing scheme of CFM Doppler method.

represents flow toward and away from/to the transducer, while the color intensity reflects the velocity magnitude.

In traditional clinical practice, CFM primarily provides visual representations of blood flow patterns, enabling clinicians to identify abnormal flow characteristics, such as turbulent or high velocity jets, typically associated with pathologies like stenoses or valvular regurgitations [27]. These conditions often involve aliasing artifacts, which, while limiting quantitative interpretation, can qualitatively highlight the presence and severity of abnormal flow. However, conventional implementations of CFM are generally limited to qualitative or semi-quantitative assessments due to their low temporal resolution and frame rate constraints.

The processing pipeline of a typical CFM implementation is illustrated in Fig. 1.9. Starting from the demodulated IQ data, clutter filters are applied to suppress low-frequency components originating from tissue motion. The filtered signals are then processed using autocorrelation techniques to estimate the mean Doppler phase shift at each pixel. A key factor in this process is the *packet size* (PS), defined as the number of pulses transmitted per scan line within the color box. The PS directly impacts the temporal resolution: larger PS values improve velocity estimation accuracy but reduce the achievable FR. Temporal persistence and spatial filtering are finally applied to improve stability and reduce noise, followed by phase extraction and mapping to the final color display.

Autocorrelation-based velocity estimation

After clutter suppression, the estimation of blood flow velocity is typically performed using the autocorrelation method introduced by Kasai *et al.* [28]. This technique computes the lag-one autocorrelation of the demodulated IQ signals to estimate the mean phase shift between

consecutive pulses, which is directly related to the mean Doppler frequency and thus to the axial blood velocity:

$$\hat{f}_D = \frac{PRF}{2\pi} \cdot arg\left(\sum_{n=1}^{PS-1} s_n \cdot s_{n+1}^*\right) \tag{23}$$

where s_n denotes the complex IQ signal of the *n*-th pulse, and arg () extracts the phase angle of the complex sum and * is the complex conjugate. The corresponding axial velocity is then derived from the estimated Doppler frequency using the Doppler equation:

$$\hat{v} = \frac{c}{2f_{TX}} \cdot \hat{f}_D \tag{24}$$

The autocorrelation method is computationally efficient and well suited for real-time imaging, making it standard in clinical CFM systems. Its main limitations lie in its sensitivity to aliasing and noise, as well as in providing only mean velocity estimates rather than the full Doppler spectrum. Nonetheless, its combination of simplicity, robustness, and efficiency has made it the reference technique for CFM velocity estimation.

High-Frame-Rate Doppler imaging

Conventional Doppler imaging is inherently limited by the sequential acquisition of *focused beams*, which restricts the achievable frame rate and reduces sensitivity to fast or transient flow dynamics. To overcome this limitation, modern systems exploit *defocused transmissions*, in which the entire field of view is sonicated with one or few (n_{TX}) *plane waves* (PW) or *diverging waves* (DW). Since only one transmit event is required per image, the frame rate can reach several thousands of frames per second, as expressed by:

$$FR = \frac{n_{TX}}{PRI} \tag{25}$$

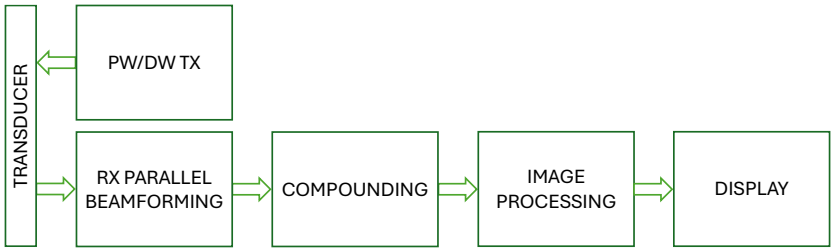


Fig. 1.10: General scheme of HFR US imaging.

To obtain an effective FR increase, the echo signals are processed in parallel, with RX parallel beamforming, and the echoes of the same scatterer are summed coherently. The Fig. 1.10 shows the general blocks to generate an HFR US imaging.

This capability has led to the development of HFR imaging, which represents a major technological advancement for Doppler applications. HFR imaging provides unprecedented temporal resolution, enabling detailed evaluation of flow dynamics, velocity profiles, and tissue motion.

For Doppler applications, high frame rate acquisition offers several benefits:

- Increased sensitivity to flow, as larger Doppler ensembles can be acquired within short observation times;
- Improved velocity estimation accuracy, due to reduced temporal decorrelation;
- Enhanced clutter filtering, with the possibility of applying advanced spatiotemporal methods such as adaptive filters or SVD;
- Feasibility of advanced Doppler techniques, including power Doppler and vector Doppler, which require high temporal sampling to achieve robust real-time performance.

Thus, HFR imaging based on defocused transmissions and coherent compounding represents the enabling technology for modern Doppler modalities. HFR-CFM [29] has been successfully applied to characterize complex cardiac flow dynamics [30,31], to measure arterial stiffness [32], and to assess myocardial motion [33], thereby extending its clinical applicability beyond purely qualitative flow assessment.

An important trade-off must however be considered. In HFR imaging, the PRF is typically increased to support fast imaging sequences. While a high PRF helps reduce aliasing, it also limits the maximum imaging depth, since the round-trip time of each pulse must remain within the pulse interval. Consequently, a balance must be struck between maximum unambiguous velocity, penetration depth, and frame rate.

Vector Doppler imaging

VDI techniques are designed to capture both the axial and transverse components of blood flow velocity, providing a complete characterization of flow direction and magnitude. Unlike conventional Doppler, which measures velocity only along the beam axis, VDI methods [34] provide 2-D velocity information over entire planes, enabling accurate representation of blood flow and a comprehensive understanding of flow dynamics at high spatial and temporal resolution. Implementations of VDI are typically based on either speckle tracking [35–37], or Doppler phase shift analysis [38–40].

HFR VDI exploits defocused transmissions, with Doppler phase-shift analysis applied in reception to reconstruct vector velocity information.

1.5 Conclusion

This chapter provided the theoretical background necessary to understand the subsequent developments of this work. It reviewed the fundamental physical principles governing ultrasound propagation, the structure and operation of piezoelectric and array transducers, and the basics of image formation. Doppler-based methods, including CFM, PD and VDI, were introduced as key tools for blood flow assessment.

The concepts presented here establish the groundwork for the implementation of HFR imaging and advanced Doppler techniques discussed in the following chapters. Chapter 2 describes the development of the heterogeneous ULA-OP 256G system, designed to provide the computational resources required for the real-time execution of these demanding algorithms.

Chapter 2. A Heterogeneous Ultrasound Open Scanner for the Real-Time Implementation of Computationally Demanding Imaging Methods

This chapter introduces the heterogeneous ultrasound research platform developed during this Ph.D. work, the ULA-OP 256G. Hardware and firmware modifications were carried out to integrate an embedded GPU into the ULA-OP 256 system, enabling efficient interaction with the existing FPGA- and DSP-based architecture. The capabilities of the platform were validated through the real-time implementation of a high-frame-rate color flow mapping method.

The PhD candidate was primarily responsible for all stages of the work, including hardware and firmware integration, real-time software development, and experimental validation on the ULA-OP 256G system.

This chapter is based on the following publication:

- ***G. Bonciani**, F. Guidi, P. Tortoli, C. Giangrossi, A. Dallai, E. Boni, and A. Ramalli, "A Heterogeneous Ultrasound Open Scanner for the Real-Time Implementation of Computationally Demanding Imaging Methods," in IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 72, no. 1, pp. 100-108, Jan. 2025, doi: 10.1109/TUFFC.2024.3474091.*

2.1 Introduction

As previously stated, open ultrasound research scanners can be broadly classified into hardware-oriented and software-oriented systems. The former rely on embedded processing devices such as FPGAs, DSPs, or SoCs to perform time-critical tasks with high throughput and deterministic performance, while the latter minimize embedded hardware and delegate computationally demanding tasks to external GPUs. Each approach has advantages and limitations, which motivated the development of heterogeneous platforms that combine both paradigms.

The objective of this work was to realize a heterogeneous open scanner leveraging the strengths of both hardware and software-oriented systems, which could enable the real-time implementation of advanced imaging techniques [41–43,3,34]. The development was based on the 256-channel ultrasound advanced open platform (ULA-OP 256), which is a hardware-oriented scanner, developed by the Department of Information Engineering of the University of Florence [9]. The ULA-OP 256 was designed to permit the collection of huge amounts of raw data and high-speed DAS beamforming [44], digital filtering, and demodulation. These tasks are efficiently performed by the FPGAs and the DSPs embedded on the front end of the system. Throughput rates of up to 600 mega samples per second have been achieved, sufficient for most advanced applications that involve imaging of 2-D [3] or small 3-D ROIs [9,45,46].

Further real-time processing of baseband data can be assigned to one DSP present on the back-end Main Control (MC) board of the scanner. This organization was demonstrated capable of implementing, in real-time, advanced techniques such as 2-D and 3-D HFR imaging [45,47,48], HFR tissue and blood motion mapping [49,29], multi-probe imaging [50], and vector Doppler [51]. However, the DSPs have fast memory caches with small dimensions (typically a few hundred kilobytes). This limitation, together with the few physical processing cores (8, in this case), hinders the implementation of highly parallel algorithms such as advanced clutter filtering schemes, multi-dimensional Fourier transforms, or multi-dimensional cross-correlations.

Starting from the outcome of a preliminary study [52], which demonstrated the feasibility of embedding a GPU System-on-Module (SoM) into the system, in this work, the firmware and software architecture of the ULA-OP 256 were further upgraded to embed the GPU processing capabilities in the processing chain. In the upgraded scanner, hereinafter identified by the acronym ULA-OP 256G, the GPU works as a co-processor of baseband data. This solution avoids the transfer bottlenecks typical of GPU cards while allowing flexible sharing of the computational load between the DSP and the GPU. Hence, the GPU would enable the implementation or further enhance the speed performance of real-time processing algorithms that require a large amount of fast-access memory or develop large computational loads.

In addition, since the GPUs are typically programmed with high-level programming languages, their use can speed up the implementation of novel advanced methods by users who are not firmware experts.

The effectiveness of the proposed solution was verified in a proof of concept: the real-time implementation of HFR CFM [53]. The achieved performance is reported and compared with the results obtained by using only the MC DSP for baseband data processing [29].

2.2 The ULA-OP 256G system

2.2.1 Architecture

The architecture of ULA-OP 256 [9] can be split into three main functional blocks: front end (FE), MC, and co-processing (CP).

The FE functionality is shared among eight FE boards, each managing 32 channels connected to 32 probe elements. As shown in Fig. 2.1:, for each board, an FPGA (ARRIA V GX Family, Altera, San Jose, CA, USA) produces the delayed, sigma-delta-converted bit streams of the TX waveforms, which are analogically low-pass filtered and transferred to high-voltage linear amplifiers. In RX, the echo signals gathered by the active elements are low noise amplified, band-pass filtered, time gain compensated, and digitized by an analog FE chip (AFE5807, Texas Instruments, Austin, TX, USA6) at a sampling frequency of 78.125 MHz

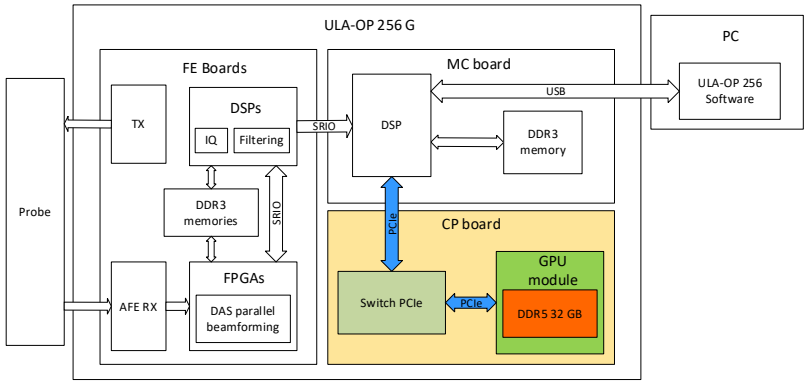


Fig. 2.1: Block diagram of ULA-OP 256G with, highlighted, the novel Co-Processing (CP) board.

or its integer (2, 3, or 4) submultiples. For the signals gathered by the corresponding group of 32 probe elements, the FPGA also performs a DAS beamforming block, which can work in "parallel" mode for HFR imaging scan sequences [44]. Beamformed radiofrequency (RF) data are transferred to two 1.2 GHz, 8-core DSPs (TMS320C6678, Texas Instruments, Austin, TX, USA), which perform IQ demodulation, filtering, and downsampling. Low-pass filtering can be based either on cascaded integrator–comb filters, for minimum processing load, or finite impulse response filters, for better frequency response. The same DSPs can also perform coherent compounding when required for HFR imaging sequences based on the TX of plane or diverging waves [3,9].

A SerialRapidIO (SRIO) bus enables a fast (full-duplex transfer rate of 10 GB/s) communication between the DSPs on neighboring FE boards. Specifically, the DSPs of an FE board transfer, in a ring-communication topology, the beamformed baseband sample blocks to the DSPs of the next FE board, which completes the beamforming by coherently accumulating such samples with the "local" beamformed sample blocks [45].

Each FE board also includes 18 GB of Double Data Rate 3 Synchronous Dynamic Random-Access Memory (DDR3 SDRAM), which is split between the FPGA (2 GB) and the DSPs (8 GB each). The SDRAM works as a lookup table for beamforming delays and apodization values, implements a supporting buffer for real-time processing, and stores the acquired data.

The MC board includes one DSP, hereinafter referred to as MC DSP (the same model as those on the FE boards), which supervises the operations of the FE boards, i.e., controls, sequences, and interleaves the TX/RX phases and scan settings. In addition, the MC DSP can perform specific processing tasks, like those described in [3,29,47,49–51], on the fully beamformed baseband signals received from the nearest FE board.

Following preliminary indications that emerged from the work described in [52], the ULA-OP 256 has been upgraded with a co-processing expansion board, which scales up the processing power of the scanner by embedding a GPU SoM (Nvidia Jetson AGX Orin) [54]. High-speed communication between the MC DSP and the GPU is managed by a peripheral component interconnect express (PCIe) switch (PEX8796, Broadcom Limited, Irvine, CA, USA), capable of controlling up to 96 PCIe3.0 lanes. Specifically, the DSP on the MC board, which has one PCIe lane Gen 2, and the GPU module, which has ×8 PCIe lanes Gen 4, an Arm Cortex-A78AE 64-bit, i.e., a host central processing unit (CPU), 2048 CUDA cores, and 32 GB of LPDDR5 RAM, were connected to the PCIe switch, as shown in Fig. 2.1:.

In this configuration, the GPU acts as a co-processor for the MC DSP: post-beamforming, IQ data collected by the MC DSP are sent to the GPU module through the PCIe connection. The GPU module processes such data according to the selected method and sends the results to the MC DSP. They can then be forwarded to the personal computer (PC) for display through a universal serial bus (USB) 3.1 SuperSpeed.

2.2.2 Software and Firmware

The ULA-OP 256 is controlled by customizable software that runs on the host PC. It initializes the hardware when the system starts up, manages the processing function, and provides a user-friendly interface for real-time control and display of results. The software code is organized into processing and display modules, which can be invoked according to the specific application. Each software processing module communicates with a corresponding firmware module that runs on the MC DSP and is characterized by two main functions: Setup and Process. The Setup function is called only during the start-up of the system and initializes the

processing and communication parameters, while the Process function, shared with the firmware, is called in a loop to elaborate available data according to the specific algorithm.

The software and firmware of ULA-OP 256 were carefully modified to enable the use of the new hardware configuration based on the GPU SoM. The PCIe interface on the MC DSP was enabled by developing new control and communication functionalities in the DSP firmware, while the communication between the firmware processing module and the PC was still managed by the software. Accordingly, the upgraded software executes the processing module that initializes and establishes communication with the MC DSP, which, in turn, establishes communication with the GPU module.

Since the Nvidia Jetson Orin GPU module runs on a Linux operating system (OS), a driver to communicate with a PCIe device was custom developed. This driver allows allocating, reading, and writing the memory buffer to exchange data between the DSP and the GPU through the PCIe.

The GPU software was organized into modules as well. Specifically, the Process function splits the data into independent blocks that are processed in parallel on the GPU cores.

2.3 Proof of concept: High Frame Rate Color Flow Mapping

The new system was tested by implementing, on the embedded GPU, a parametric HFR CFM algorithm. A similar algorithm was previously implemented in the MC DSP and experimentally validated [29]. However, the poor amount of available fast-access memory imposed some constraints on parameters such as the size of the processable PS and ROI, and the achievable frame rate. The performance obtained by the two HFR CFM implementations is reported in the next section.

2.3.1 CFM Module

To overcome such limitations, a CFM processing module was developed for the GPU SoM. The real-time implementation of HFR CFM on the ULA-

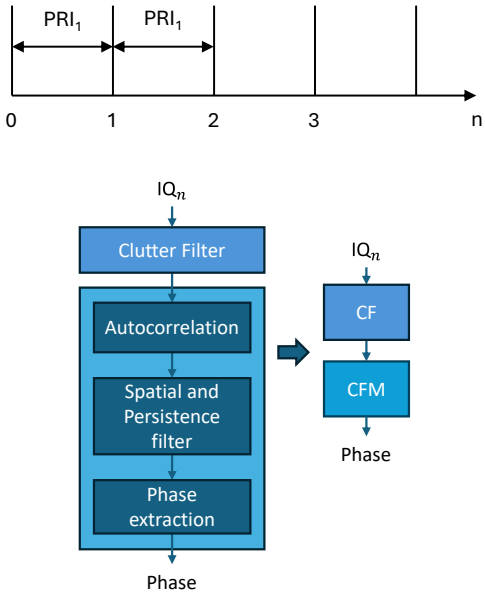


Fig. 2.2: Conventional PRF sequence and block diagram of the HFR CFM processing GPU pipelines implemented on the ULA-OP 256G.

OP 256G is sketched in Fig. 2.2. Starting from HFR IQ_n data frames, where *n* denotes the frame index within the sequence, each frame is processed on the GPU by a dedicated CUDA kernel. The detailed organization of the GPU implementation is illustrated in Fig. 2.3 and described below.

GPU Implementation

The data transfer is handled by the MC DSP and the host CPU of the GPU module, while the GPU cores take care of the processing. The IQ_n data are frame-by-frame pushed through the PCIe channel into a buffer memory of the GPU SoM, then, by a specifically designed Linux driver, IQ_n data are moved to a first-in-first-out (FIFO) memory allocated in the SoM global memory and shared with the GPU cores.

To efficiently handle varying processing rates, the code was split into two CUDA streams. Stream #1 is responsible for fetching the input complex samples from the FIFO-IN memory, calculating the signal power and simultaneously clutter-filtering the signal using a continuous-time, 8th-order, IIR high-pass filter (HPF). The resulting power and clutter-

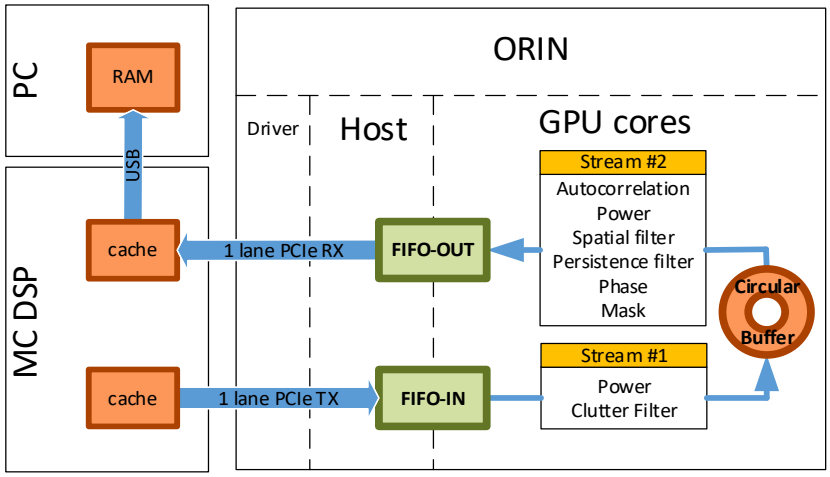


Fig. 2.3: Block diagram of the CFM implementation. The green sectors indicate FIFO memories, and the red sector indicates a circular buffer memory.

filtered frames are then transferred to the circular buffer. To improve the processing efficiency, the stream processes batches of 64 frames, which was empirically found to be a reasonable compromise between computational cost per frame and data group delay.

Stream #2 completes the CFM algorithm and is organized into two sequential kernels. The first kernel extracts a set of frames from the circular buffer: the number of frames per set corresponds to PS, while the lag between consecutive frames is defined as the percentual sliding factor (SF). The latter parameter determines the hop size (HS), i.e. the number of new samples in each autocorrelation calculation, as HS = PS(1-SF/100).

For each group of PS wall-filtered frames, this kernel computes the signal power and the autocorrelation with a programmable lag, *l*, as described in:

$$R_{x,y}[n,l] = \frac{1}{PS-l}\sum_{i=0}^{PS-l-1} s_{x,y}(n-i)\cdot s^{*}_{x,y}(n-l-i) \qquad (26)$$

where $s_{x,y}(n)$ is the complex sample value at the *x,y* position in the n-th frame.

The second kernel collects the autocorrelation complex output as well as the signal power before and after the wall-filter and smooths them spatially and temporally by 3×3 Gaussian low-pass and 1-st order IIR persistence filters. The computed powers and autocorrelation phase are compared with empirical thresholds to estimate their reliability [26,29] and thus the final CFM map is saved in the FIFO-OUT memory.

The chosen subdivision into kernels ensures that by the time the spatial filter is computed, the source frame is fully processed. Moreover, the kernel functions utilize different types of GPU memory: the first and last kernel functions read from and write to the FIFO-IN/OUT devices, allocated in the global memory and shared between the CPU and GPU, while the intermediate kernel functions use only registers and level 1 cache to store intermediate data, resulting in lower latency and faster processing compared to global memory.

Exploiting a full-duplex mode, simultaneously with the arrival of a new IQ_n data frame, the host ORIN CPU sends the last computed CFM frame to the MC DSP memory. From here, it is then routed to the PC for color palette transformation, selective overlay with the B-mode image, and display.

2.3.2 Experiments

The LA533 probe (Esaote SpA, Florence, Italy), a 192-element linear array with a pitch of 245 μm and a 100% bandwidth of around 8 MHz, was connected to the ULA-OP 256G. The scanner was configured to transmit plane waves from the central 128 elements of the probe, at a programmable PRF. The TX signal was a 6-cycle, Hamming-tapered, 6 MHz sinusoidal burst.

In RX, the system was set to acquire the RF signals from a depth range of about 25 mm (i.e., 4096 samples @78.125 MHz). The RF channel data were parallel beamformed, demodulated, downsampled, and finally streamed to the SoM.

The real-time system performance was evaluated, first, in terms of data transfer rate between the DSP and GPU, and then considering the parameters that most influence the processing load, i.e., the size of the ROI (ranging from 64 lines × 128 depths to 192×512), PS (in the range from 8

Table 2.1: Comparison between DSP and GPU real-time performance.

	NL×Ndepths	64 × 128					
	PS	8		64		128	
	HS	1	8	1	64	1	128
DSP	FR_{OUT} [Hz]	NF	570	NF	72	NF	NF
GPU	FR_{OUT} [Hz]	4300	538	4300	67	4300	34

	NL×Ndepths	128 × 256					
	PS	8		64		128	
	HS	1	8	1	64	1	128
DSP	FR_{OUT} [Hz]	NF	262	NF	33	NF	NF
GPU	FR_{OUT} [Hz]	1100	138	1100	17	1100	9

*Colors highlight a higher (green), a lower (yellow), or a not feasible (NF, in red) performance.

to 128), and HS (from 1 to 128). Specifically, for each parameter set-up, the tests aimed at comparing the performance of DSP and GPU implementations, in terms of achievable frame rate (FR_{OUT}). Nevertheless, for the sake of clarity and synthesis, only the most representative cases are shown in Table 2.1. Although PS values could be arbitrarily set, the following ones were reported: a typical value for classic focused CFM (PS=8), the maximum PS achievable for the DSP implementation (PS=64), and one value feasible only with the GPU, which is short enough to avoid filtering the flow accelerations of interest (PS=128). The HS values were chosen to highlight the maximum performance of the GPU (HS=1) and the only option feasible for the DSP implementation (HS=PS). It must be noted that in the GPU implementation, the performance varied almost linearly for intermediate values of PS and HS.

In vivo tests were performed on the carotid bulb and the femoral vessels of healthy volunteers. The *in vivo* protocol was approved by the institutional review board on ethics of the University of Florence (approval number 309/2024) and informed consent was obtained from each volunteer.

2.4 Results

2.4.1 Performance assessment

A preliminary test of the direct PCIe communication between the MC DSP and GPU showed a data transfer rate of ~425 MB/s, i.e., only a ~17% reduction compared to the nominal value (0.5 GB/s for PCIe ×1 Gen 2). Since this value is consistent with the protocol overhead, it can be stated that the system fully exploits the maximum available transfer bandwidth.

Table 2.1 shows that several parameter combinations are not feasible (NF) by the DSP mainly due to the limited amount of available cache memory. Consequently, the HS had to be always coincident with the PS, i.e., no sliding was possible, and any PS>64 could not work. Overall, the FR achieved with the GPU is at least 7.5 times higher for the 64×128 map and 4 times for the

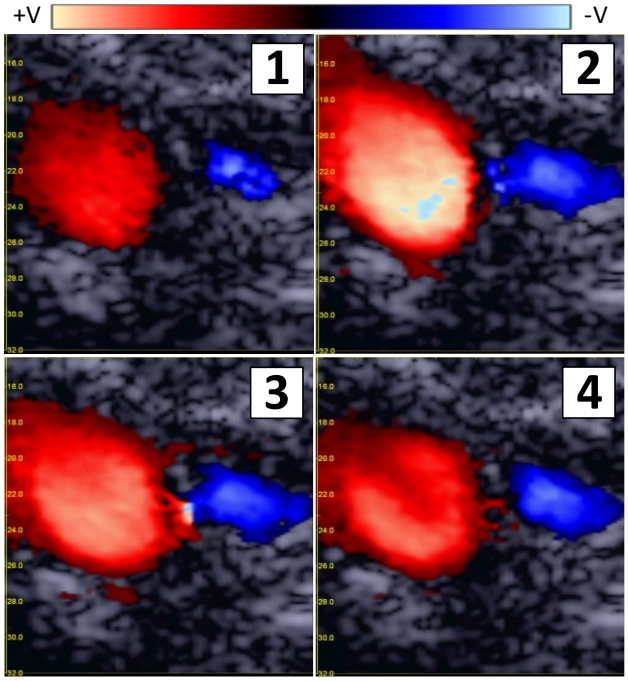


Fig. 2.4: Four frames extracted from an acquisition performed using HFR CFM on the carotid bulb (left) and jugular vein (right) of a healthy volunteer. The frames were obtained over a 64×128 ROI with PRF = 4.3 kHz, HS = 1, and EL = 64. The interval between these frames was 39.5 ms, corresponding to systolic acceleration (1), systolic peak (2), and systolic deceleration (3-4).

larger ROI. The GPU performance is, in most cases, limited by the PCIe transfer rate. Only when the ROI size is very large and HS=1 (which corresponds to producing a full CFM frame after each TX event, i.e., the peak output frame rate is equal to the PRF), the performance was limited by the current code implementation and the GPU computation power.

2.4.2 Real-time *in vivo* test

The carotid bulb of a volunteer was imaged at 4.3 kHz PRF over an ROI of 64×128 points. By setting PS = 64 and HS = 1, detailed frames at high temporal resolution (FR_{OUT} = 4.3 kHz) were obtained which was recorded at a 43 times slower speed. Fig. 2.4 shows four frames, extracted from the movie at intervals of 39.5 ms, corresponding to systolic acceleration (1), systolic peak (2), and systolic deceleration (3-4). Although the images were taken at the inlet of the carotid bulb, the classic C shape, so far predicted for slightly curved common carotid arteries [55–57], is quite evident here, especially, as expected, during the second part of the systolic deceleration (4).

The two frames in Fig. 2.5 were acquired while scanning the femoral vein bifurcation (A) and artery (B) of a healthy volunteer and was recorded at a 11 times slower speed. Here, a larger ROI (128×256 points) was scanned to provide a complete view of the flow dynamics in the femoral veins region. The PRF and frame rate were reduced to 1.1 kHz to better detect the lower velocities while still avoiding the aliasing effect. The movie highlights the complex flow behavior determined by the opening of the upper femoral vein's valve. In particular, the high temporal resolution allows for showing a turbulence region propagating through the vessel, from the right (closer to the valve) to the left. In Fig. 2.5, showing two frames following the valve's opening, the turbulence region appears as a blue spot (C) surrounded by the main (red) flow.

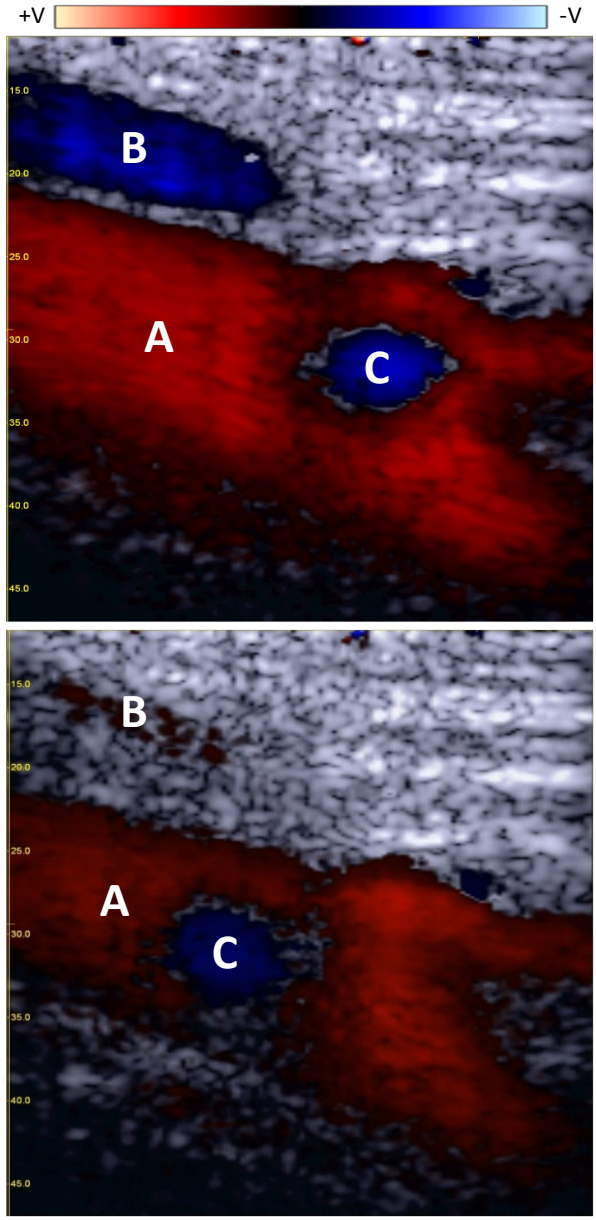


Fig. 2.5: Two frames extracted from an acquisition performed using HFR CFM of the femoral vein (A) bifurcation and artery (B) of a healthy volunteer. A 128×256 ROI was imaged with PRF = 1.1 kHz, HS = 1, and PS = 64. The frames, obtained at a time distance of 136 ms, highlight the slow movement, from right to left, of the turbulence region (C) following the valve aperture.

2.5 Discussion

In this work, the implementation of an open scanner exploiting heterogeneous processing devices, including FPGAs, DSPs, and a GPU, was presented. Such heterogeneity enables an optimal use of each device, thus providing high-performance and high-efficiency processing according to the needs of the application of interest. Additionally, the presence of a GPU SoM allows the user to achieve real-time data processing by programming with high-level languages.

The choice of embedding a GPU SoM into the ULA-OP 256 scanner, rather than using a GPU card on the PC, was encouraged by different considerations. First, the available USB communication is not fast enough to stream HFR IQ data in real-time to the PC. Furthermore, on the PC, transferring data from the CPU to the GPU memory is time-consuming, while on the SoM, any data transfer between independent memories is avoided (the SoM CPU and GPU share the same physical memory), thus favoring more efficient data processing.

Finally, the SoM is compact and well fits inside the rack of the scanner; this keeps the system portable and mitigates the requirements of the host PC, which does not need to embed a powerful GPU. Of course, the continuous advancement in commercially available boards will permit us to leverage the latest GPU modules, offering enhanced performance, increased memory capacity, and other advancements.

As a proof of concept, the HFR CFM modality was implemented and tested. Compared to the previous DSP-based implementation, different advantages were obtained. As shown in Table 2.1, the GPU enables configurations that are not feasible by the DSP. For example, any hop-size between 1 and PS can be set, thus significantly increasing the FR up to the PRF (for HS=1), while in the DSP-based implementation it was limited to PRF/PS. High-quality color maps (i.e., with long PS values, here extendable up to 128) are finally compatible with temporal resolutions of a few milliseconds, as shown in the accompanying movies that highlight the flow complexity in two different arterial regions.

The performance shown in Table 2.1was achieved by programming in CUDA, without ultimate optimization efforts. If further applications

require very long PS(≥128) and higher FRs, the performance could be improved by optimizing the code. Although the current PCIe switch is oversized for this application, it was chosen for its capability of connecting all FE boards' DSPs to the GPU module. Such communication architecture could increase the current transfer rate (~425 MB/s) up to a factor ×10. This could be relevant in other computationally demanding applications involving the transfer of huge amount of data to and from the GPU.

Of course, programming and testing of the system with multiple heterogenous processing units is laborious and requires skills in multiple languages, e.g., VHDL, C++, and CUDA. However, such a solution opens the perspective of implementing in real-time complex methods such as advanced beamforming methods [58–62], clutter filtering techniques [63–65], shear wave elastography [66,67], super-resolution [68,69], echocardiography [70–72], contrast enhanced ultrasound imaging [30,73].

Chapter 3. Real-time high-frame-rate vector Doppler ultrasound imaging by a hybrid open-platform

This chapter presents the implementation and validation of high-frame-rate vector Doppler imaging on the ULA-OP 256G platform. The heterogeneous architecture, combining FPGAs, DSPs, and an embedded GPU, enabled real-time processing by efficiently distributing the computational tasks. The method was experimentally validated using flow phantom acquisitions, demonstrating accurate two-dimensional velocity estimation with high temporal resolution.

The PhD candidate was primarily responsible for all stages of the work, including real-time software implementation, system integration, and experimental validation on the ULA-OP 256G platform.

This chapter is based on the following publication:

- ***G. Bonciani**, A. Ramalli, A. Bernard, F. Guidi, P. Tortoli, E. Boni, D. Garcia, and F. Varray, "Real-time, high-frame-rate, vector Doppler ultrasound imaging by a hybrid open platform," 2024 IEEE Ultrasonics, Ferroelectrics, and Frequency Control Joint Symposium (UFFC-JS), Taipei, Taiwan, 2024, pp. 1-3, doi: 10.1109/UFFC-JS60046.2024.10793839.*

3.1 Introduction

As introduced in Chapter 1, Doppler ultrasound has evolved from conventional single-direction measurements to advanced VDI techniques capable of providing 2-D and 3-D flow information. Unlike traditional Doppler, VDI does not rely on assumptions about flow direction and enables the accurate characterization of complex patterns such as vortices and turbulence. Among VDI methods, HFR approaches based on plane wave transmission have gained particular importance, as they combine high temporal resolution with the ability to reconstruct velocity vectors through multi-angle or multi-aperture reception strategies.

Although, HFR VDI techniques enable high spatio-temporal resolution imaging, they have some limitations. Processing the data from multiple angles or apertures and handling the corresponding large volume of generated information can be computationally demanding. This can result in long processing times and require powerful hardware for efficient data handling. The heavy computational loads may also constraint the number of measurement points that can be processed within a given timeframe. This can affect spatial resolution and the ability to capture detailed flow characteristics. Additionally, the increased processing time can impact the output frame rate of the imaging system. Lower frame rates can reduce the ability to capture rapid changes in blood flow, potentially affecting diagnostic accuracy and real-time monitoring capabilities.

This work aims to implement and validate a novel heterogeneous open platform with a real-time HFR VDI technique based on two oriented coordinate systems [74,75], using unsteered plane wave TX and two RX angles.

3.2 Real-time implementation of VDI

The ULA-OP 256G architecture, described in Chapter 2, is well suited for HFR VDI, which requires the processing of large volumes of data acquired from multiple RX angles. Consequently, the time needed to process each frame is significantly reduced by leveraging the GPU's parallel processing capabilities. This enables higher output frame rates and/or allows for processing more data points within each frame.

The method involves the transmission of unsteered plane waves using a linear array. For each TX event, the system FPGAs perform parallel beamforming of the echoes received by different dynamically selected RX apertures, according to the reception angle β. The width of the RX aperture is adaptively adjusted to maintain a constant f-number ($f_\#$) [21], while, as sketched in the top panel of Fig. 3.1, its center position, x_a, is determined for each generic point of interest $P_m(x_m,z_m)$ based on z_m and the reception angles ±β, as

$$x_a = z_m \tan\beta \tag{27}$$

As an example, points P_A and P_B correspond to different depths (z_A and z_B) along the same line x_A, while point P_C lies at the same depth as P_B (z_B) but along a different line (x_C). Their corresponding receiving apertures, A_A, A_B, and A_c, are shown above. It can be observed that the receiving aperture A_B is wider than A_A at greater depths, while A_C is laterally shifted with respect to A_B, maintaining the same geometric configuration. The bottom panel of Fig. 3.1 provides an example of the dynamic aperture evolution with depth during the reconstruction of the x_A line, where the same receiving apertures A_A and A_B are shown at depths z_A and z_B, respectively.

This dynamic selection allows the reconstruction of two raw images over the same region of interest but from differently oriented coordinate systems. The real-time GPU implementation of HFR VDI on the ULA-OP 256G is sketched in Fig. 3.2. The beamformed and IQ_n demodulated signals, where *n* denotes the frame index within the TX sequence, are

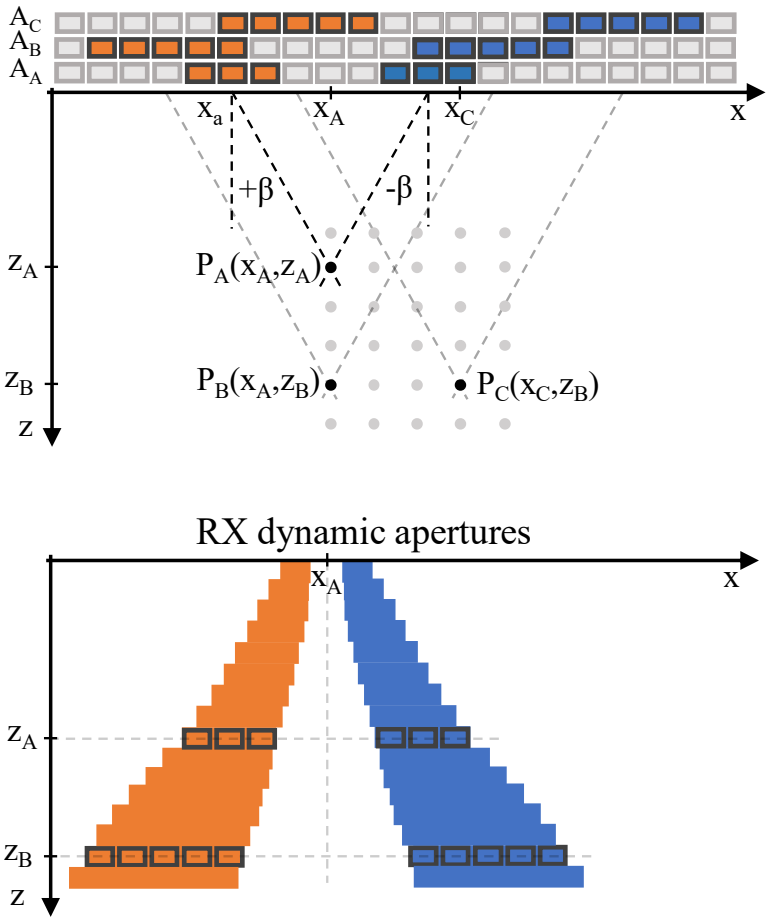


Fig. 3.1: Illustration of the dual-angle reception geometry (top) and dynamic aperture (bottom) used for real-time HFR VDI. The top panel shows the coordinate reference system and examples of the receiving apertures used to beamform the points of interest P_n with angles ±β (orange and blue). Points P_A and P_B lie along the same line at lateral position x_A but at different depths z_A and z_B; their corresponding receiving apertures (A_A and A_B) are indicated above. The same apertures are shown in the bottom panel, which illustrates the evolution of the dynamic reception aperture with depth for the reconstruction of the x_A line. Point P_C is located at the same depth as point B (z_B) but along a different line (x_C); therefore, its receiving apertures are laterally shifted with respect to the apertures used for point P_B, while the geometry is maintained. All points are assumed to be sonicated by the same plane wave.

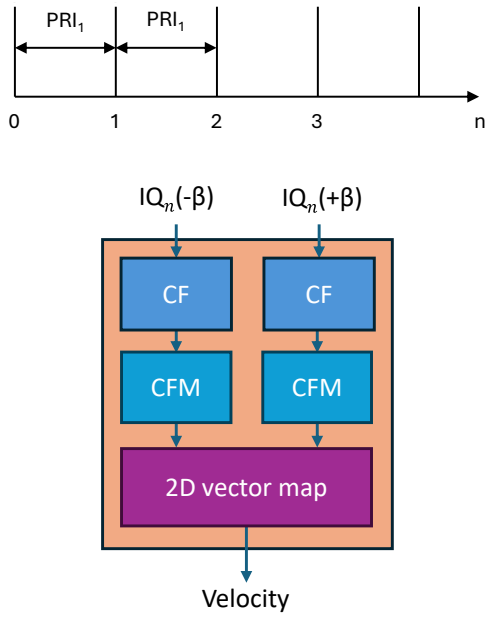


Fig. 3.2: Conventional PRF sequence and block diagram of the HFR VDI processing GPU pipelines implemented on the ULA-OP 256G.

streamed to the onboard GPU module. Here, the IQ_n frame corresponding to each reception angle (±β) are processed according to the HFR CFM method (see Chapter 2), which includes clutter removal through an IIR filter (CF block), followed by autocorrelation, phase extraction (Kasai's method [28]), and conversion of phase shift into axial velocity (CFM block). Furthermore, for each oriented image, the velocity maps are processed using a spatial 3×3 Gaussian low-pass filter, while an IIR persistence filter is applied to smooth the velocity measurements over time.

Finally, the 2-D vector field is reconstructed in the *2D vector map* block by combining the two sets of oriented velocity data using the following equations [74,75]:

$$V_x = \frac{(V(-\beta) - V(+\beta))}{\sin\beta} \quad V_z = -\frac{\big(V(-\beta) + V(+\beta)\big)}{(1+\cos\beta)} \qquad (28)$$

where, V_x and V_z represent the lateral and axial components of the 2-D velocity vector, while $V(-\beta)$ and $V(+\beta)$ correspond to the 1-D velocity

components of the flow, obtained from the acquisitions at the reception angles ±β.

3.3 Experimental setup

The HFR VDI method was validated through a flow phantom. A blood mimicking fluid (model 069-DTF, CIRS, USA) was pushed by a peristaltic pump (model 120U, Watson Marlow, UK) through a hydraulic circuit, including a pulsation dampener and a home-designed tissue mimicking phantom. The latter contains a 7-mm-diameter horizontal tube positioned at 20 mm depth. The revolution per minute (RPM) of the peristaltic pump was set to achieve flow rates of 51, 73, and 277 mL/min, and maximum steady flow speeds v_{max}^{Ref} of 4.4, 6.4, and 8.4 cm/s, respectively.

The GPU embedded in the ULA-OP 256G system was exploited to calculate the flow velocity components in real-time and then, were saved on binary files for post-processing and performance assessment. A global accuracy and precision were computed over the region-of-interest, for each depth z_p, as the mean relative bias ($\bar{B}_v$) and standard deviation ($\bar{\sigma}_v$) of the estimated velocity module, $v_{Module}(z_p)$, as follows:

$$\bar{B}_v = \frac{1}{v_{max}^{Ref}} \frac{1}{N_p} \sum_{p=1}^{N_p} \left(\bar{v}_{Module}(z_p) - v^{Ref}(z_p) \right) \tag{29}$$

$$\bar{\sigma}_v = \frac{1}{v_{max}^{Ref}} \sqrt{\frac{1}{N_p} \sum_{p=1}^{N_p} \sigma_v(z_p)^2} \tag{30}$$

with

$$\sigma_v(z_p) = \sqrt{\frac{\sum_{u=1}^{N_F} \left(v_{Module}(z_p, u) - \bar{v}_{Module}(z_p) \right)^2}{(N_F - 1)}} \tag{31}$$

where $\bar{v}_{Module}(z_p)$ and $v^{Ref}(z_p)$ are the temporal means of the estimated and reference velocity modules, respectively, averaged across the lines, N_p is the number of points of interest, and u is the frame index out of the total N_F frames.

3.4 Results

Fig. 3.3 shows an example of the estimated velocity profile obtained for v_{max}^{Ref} = 8.4 cm/s, PS of 128, and β = 15° compared to the reference profile. The latter was obtained using the analytical solution derived from the Poiseuille's equation for laminar flow in a pipe. Qualitatively, the estimated profile (in red) closely followed the theoretical reference profile (in cyan). The low standard deviation $\sigma_v(z_p)$, illustrated by the shaded yellow area around the mean profile, highlighted the slight variability in the real-time measurements, confirming the accuracy and precision of the proposed method.

Table 3.1 allowed quantitative comparison of the average relative bias ($\bar{B}_v$) and standard deviation ($\bar{\sigma}_v$) for the three tested flow conditions, each evaluated with three different β angles (7.5°, 10°, and 15°), three PS values (8, 64, and 128), and over a region spanning 90% of the radial size of the tube lumen. In all cases, the relative bias was always better than 20%, while

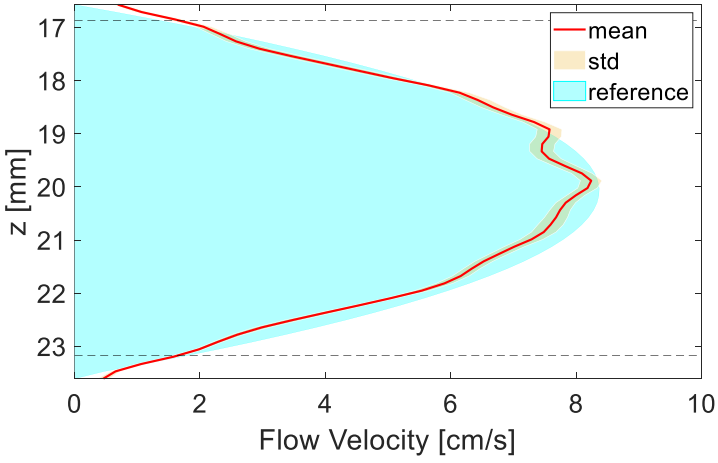


Fig. 3.3: Estimated and reference velocity profiles obtained for v_{max}^{Ref} *= 8.4 cm/s, PS = 128, and β = 15°.*

Table 3.1: Average relative bias and standard deviation.

		$\bar{B}_v \pm \bar{\sigma}_v$		
RPM	**β [°]**	**PS = 8**	**PS = 64**	**PS = 128**
100	**7.5**	-17.0 ± 43.2	-13.1 ± 55.3	-7.7 ± 13.4
	10	-3.9 ± 22.0	-0.4 ± 20.9	-0.6 ± 22.7
	15	-11.8 ± 19.3	-11.3 ± 18.2	-11.5 ± 18
150	**7.5**	-20.0 ± 9.9	-11.0 ±5.7	-10.0 ± 5.0
	10	-16.0 ± 8.2	-10.1 ± 4.3	-9.2 ± 4.0
	15	-16.8 ± 6.3	-16.0 ± 2.2	-15.6 ± 1.9
200	**7.5**	-2.9 ± 17.1	3.7 ± 8.8	4.5 ± 7.9
	10	-6.1 ± 7.8	-2.4 ± 4.2	-1.9 ± 3.5
	15	-4.2 ± 6.7	-4.0 ± 3.8	-4.0 ± 1.8

the standard deviation remained below 22%. The performance improved with higher PS values, larger beamforming angles, and higher RPMs. The best results were observed for v_{max}^{Ref} = 8.4 cm/s and PS = 128. Specifically, the lowest relative bias of -4.0 ± 1.8 % was achieved for β = 15°, while the lowest standard deviation of -1.9 ± 3.5 % was observed at β = 10°. This is expected because larger PS values enable more stable and averaged flow measurements, minimizing fluctuations. Similarly, the highest flow rates better exploit the Doppler spectrum and make the autocorrelation phase estimation less noisy, thus improving the flow estimates.

In vivo tests were performed on the carotid artery of a healthy volunteer. The *in vivo* protocol was approved by the institutional review board on ethics of the University of Florence (approval number 309/2024) and informed consent was obtained from the volunteer. Sample screenshots

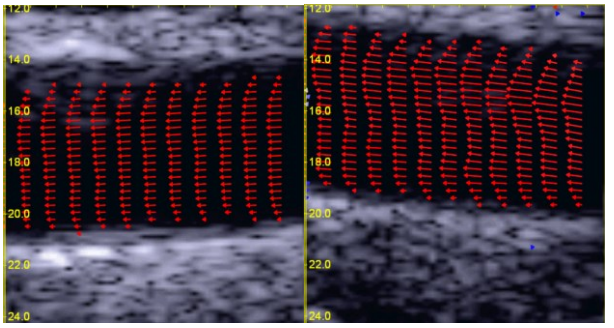


Fig. 3.4: In vivo VDI maps on the carotid artery of a healthy volunteer with 48×128 vector maps and 1.5 kHz PRF. The two images were obtained by slightly tilting the probe.

from the ULA-OP 256G real-time interface are shown in Fig. 3.4. They were captured for a 48×128-vector map with a 1.5 kHz PRF. The images show physiological profiles with velocity vectors parallel to the vessel walls, with asymmetric profiles, as expected in non-ideal non-straight vessels.

3.5 Discussion

The results obtained with the proposed real-time HFR VDI implementation demonstrate that the heterogeneous ULA-OP 256G platform is capable of supporting computationally demanding flow-imaging algorithms while maintaining high temporal resolution. The system successfully handled the dual-angle beamforming and velocity reconstruction pipeline entirely on-board, with frame-to-frame latencies compatible with real-time visualization.

The quantitative evaluation on the flow phantom confirmed the robustness of the implementation. The relative bias and standard deviation were consistently below 20–22% across all tested conditions, validating both the accuracy and the precision of the proposed method. Performance improved with increasing PS and reception angle, as expected from the enhanced signal-to-noise ratio and reduced estimation variance. The observed trend with higher flow rates further confirmed that the GPU-based processing preserved Doppler phase integrity without introducing additional noise or latency.

In vivo experiments on the carotid artery demonstrated the feasibility of applying the proposed technique in a clinical-like setting. The real-time interface provided stable velocity vector maps with physiological flow patterns, confirming that the GPU-embedded architecture can sustain interactive frame rates while processing dual-angle data streams.

Chapter 4. Real-Time High Frame Rate Color and Vector Doppler Imaging with Staggered Pulse Repetition Frequency

This chapter presents the implementation and validation of the staggered PRF (S-PRF) strategy to overcome aliasing limitations in high-frame-rate Doppler imaging. Both color flow mapping and vector Doppler imaging were realized in real time on the ULA-OP 256G platform, demonstrating robust alias suppression and accurate velocity estimation in phantom experiments.

The PhD candidate was primarily responsible for all stages of the work, including real-time software implementation, system integration, and experimental validation on the ULA-OP 256G platform.

This chapter is based on the following publication:

- ***G. Bonciani**, F. Guidi, C. Giangrossi, F. Varray, D. Garcia, E. Boni, and A. Ramalli, "Real-Time High Frame Rate Color Doppler Imaging with Staggered Pulse Repetition Frequency," 2025 IEEE International Ultrasonics Symposium (IUS), Utrecht, Netherlands, 2025.*

4.1 Introduction

In Chapter 1, the fundamentals of Doppler imaging and the evolution toward HFR modalities such as CFM and VDI were presented. Their real-time implementation on the heterogeneous ULA-OP 256G platform was discussed in Chapter 2 and Chapter 3, where the benefits of GPU acceleration for HFR CFM and VDI were demonstrated.

A fundamental limitation of all pulsed-wave Doppler imaging modalities, including HFR-CFM and VDI, is the possible occurrence of aliasing artifacts, which arise when the Doppler frequencies exceed the Nyquist limit. In particular, in HFR CFM/VDI, the maximum alias-free velocity is directly constrained by the achievable frame rate, which depends on the PRF and on possible compounding strategies. In addition, since real-time visualization and interactive feedback are core features of ultrasound imaging, the achievable frame rate is also limited by the computational capability of the scanner itself. Indeed, implementing HFR imaging demands high data transfer rates and fast parallel beamformers [45], while HFR CFM and VDI involve intensive processing pipelines. In particular, VDI is sometimes particularly demanding due to the processing of data from multiple reception steering angles or apertures [76].

Therefore, advanced techniques that effectively mitigate aliasing with low computational overhead represent a promising and practical solution for real-time implementation of HFR-CFM and VDI. Among them, staggered pulse repetition frequency (S-PRF) methods [77], originally developed in meteorological radar applications [78], extend the Nyquist velocity limit and suppress aliasing artifacts by interlacing multiple PRFs, each corresponding to a distinct Nyquist velocity. Initially, S-PRF strategies were adapted to classical pulsed wave Doppler to enhance the measurement of high flow velocities in narrow sample volumes [79]. Then, they were extended to imaging modalities, such as CFM and VDI, where velocity maps must be estimated over two-dimensional areas rather than single gates [77].

In this chapter, we investigate the real-time implementation of S-PRF on the heterogeneous ULA-OP 256G platform. Building upon the

developments described in Chapter 2 and Chapter 3, S-PRF was integrated into both HFR CFM and VDI pipelines, leveraging the combined processing power of FPGAs, DSPs, and GPU acceleration. Experimental validation on flow phantoms demonstrates that the proposed method effectively suppresses aliasing while maintaining high temporal and spatial resolution, thereby confirming its potential for quantitative Doppler imaging in challenging flow conditions.

4.2 Materials and methods

4.2.1 Staggered PRF

Several strategies have been proposed in the literature to disambiguate aliased Doppler velocity fields. Early methods relied on post-processing of color Doppler images using heuristic or region-growing algorithms [80,81], but these approaches are sensitive to noise and prone to failure in the presence of flow discontinuities. More advanced techniques [82,83], originally developed in InSAR (Interferometric Synthetic Aperture Radar) imaging [84], assume smooth spatial velocity variations and apply quality-guided or cost-optimization algorithms, but are too computationally demanding for real-time use. To overcome these limitations, transmission-

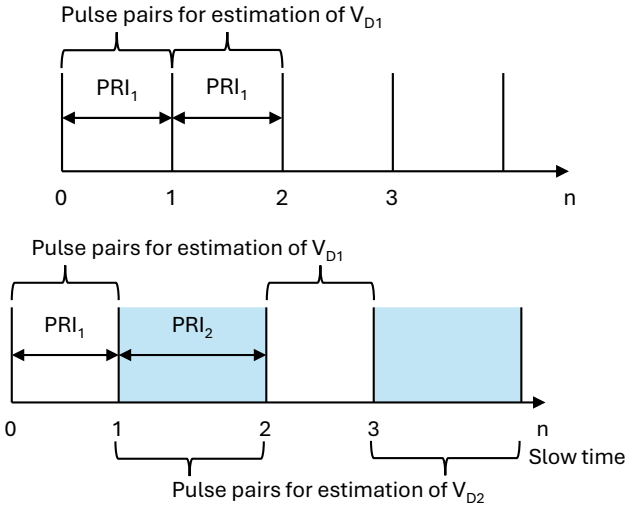


Fig. 4.1: Conventional and staggered Dual-PRF sequences.

based techniques, such as S-PRF, have been proposed. Rather than correcting aliasing through post-processing, these methods modify the timing of pulse emissions to enable alias-free Doppler imaging directly at acquisition.

S-PRF strategies have their origins in Doppler weather radar systems, where they were introduced to synthetically extend the measurable velocity range [78]. The concept was later adapted to medical ultrasound using dual-PRF sequences [79,85], to estimate high flow velocities [86]. The S-PRF method is a Doppler ultrasound technique designed to mitigate aliasing, which causes velocity wrap-around errors. S-PRF addresses this issue by transmitting pulses at varying repetition intervals (PRIs) rather than at a constant one, generating a staggered sequence, as illustrated in Fig. 4.1. This approach yields multiple differently aliased velocity estimates, from which the true velocity can be inferred.

The S-PRF technique for color Doppler proposed by Posada *et al.* [77], stands out for its efficient implementation in ultrafast ultrasound systems, enabling real-time alias-free Doppler imaging via a fast numerical method that estimates the so-called Nyquist numbers, which indicate how many times the true velocity exceeds the Nyquist interval. This method was validated *in vitro* and *in vivo* using dual and triple-PRF schemes, demonstrating alias-free imaging with Nyquist limits up to six times higher than conventional approaches.

Typically, two or more PRFs are used, where each supplementary PRF is expressed as a rational fraction of the primary PRF:

$$PRF_i = \frac{p_i}{q_i} PRF_1 \tag{32}$$

where p_i and q_i are relatively prime positive integers, with $p_i < q_i$ and $i > 1$. The use of relatively prime ratios ensures that the aliasing patterns are as distinct as possible. Under this formulation, the extended Nyquist velocity (V_{Ne}) becomes:

$$V_{Ne} = LCM(p_i) V_{N1} \equiv k V_{N1} \tag{33}$$

where LCM stands for least common multiple and V_{N1} is the Nyquist velocity associated with the PRF_1.

Several approaches allow reconstructing de-aliased velocities using staggered PRF sequences such as early applications of non-equally-spaced pulse transmission [79], systems based on nonuniform coherent sampling [85] or multi-PRF pulsed Doppler [86], and more recently the ultrafast color Doppler implementation proposed by Posada *et al.* [77].

4.2.2 The ULA-OP 256G

As explained in Chapter 2, the ULA-OP 256G is a high-performance, heterogeneous open scanner that integrates hardware and software, including FPGAs, DSPs, and an embedded GPU, to support advanced real-time applications.

In this work, ULA-OP 256G uses unsteered PW transmissions for HFR acquisition. Real-time parallel receive beamforming is performed on the FPGAs using a delay-and-sum algorithm [44], followed by in-phase quadrature (IQ) demodulation and digital filtering on the DSPs. The resulting complex data are then transferred via PCIe to the embedded GPU. This architecture has already proven effective for real-time implementations of HFR CFM and VDI, which were introduced in Chapter 2 and Chapter 3, respectively.

4.2.3 Real-time implementation of Staggered PRF

The ULA-OP 256G's main sequencer was modified to manage the interleaving of transmission events according to programmable S-PRF sequences.

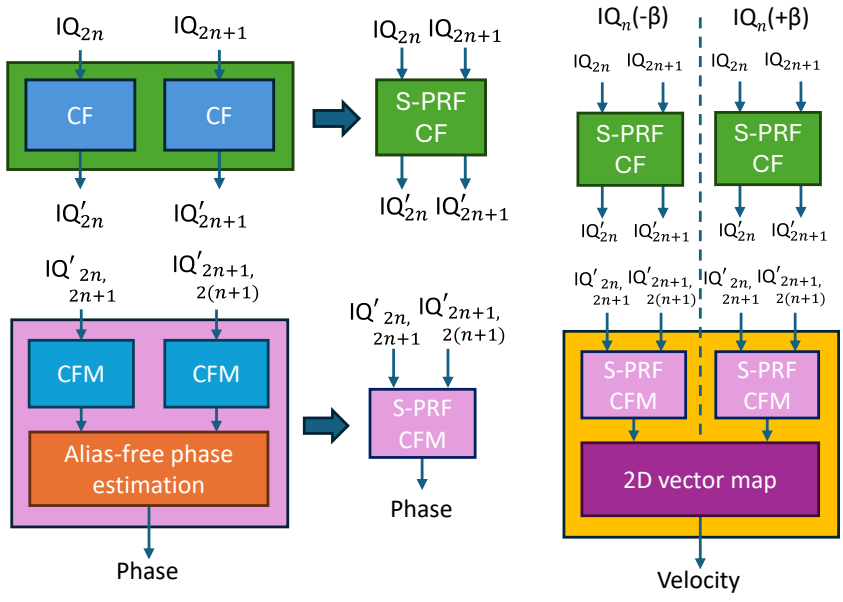
Specifically, the configuration files, interpreted by the real-time software, were extended to allow the user to define the duration of each PRI. Three options were made available: a classic constant PRI, a programmable duration expressed in nanoseconds, or a programmable duration expressed as a fraction of the base PRI.

To support this flexibility, the firmware of both the FPGA and DSP on the ULA-OP 256 main control board was upgraded. The FPGA generates synchronization pulses through a programmable counter with 12.8 ns

resolution, which triggers a pulse when it reaches a full-scale value stored in a memory register. The DSP updates this value for each interval based on the PRI duration defined in the configuration file and interpreted by the software. This modification enables the emission of staggered PRIs, allowing acquisition of multiple Doppler streams, each affected by different aliasing patterns. On the GPU, a dual S-PRF approach was integrated in both the CFM and VDI processing pipelines. The processing code of each method was modified to handle sequences acquired at different PRFs, each processed by a dedicated CUDA kernel, allowing parallel and independent computation of Doppler data.

Staggered CFM Module

The real-time implementation of HFR CFM on the ULA-OP 256G was presented in Chapter 2. In the S-PRF CFM configuration, as sketched in Fig. 4.2a, the IQ_n frames, where *n* denotes the frame index within the



a) b)

Fig. 4.2: Block diagrams of the Doppler processing GPU pipelines implemented on the ULA-OP 256G for staggered Dual-PRF configuration. a) Staggered-PRF CFM & b) Staggered-PRF VDI.

sequence, acquired with two staggered PRFs (PRF_1 and $PRF_2 = \frac{p_2}{q_2} PRF_1$) are divided into even (IQ_{2n}) and odd (IQ_{2n+1}) frame sequences, according to the indexes reported in Fig. 4.1. This separation enables maintaining a constant effective sampling frequency of $1/(PRI_1 + PRI_2)$ for both sequences, which is essential for the IIR clutter filtering process. Consequently, the even and odd frame sequences are independently processed by separate but identical instances of clutter filtering. The clutter-filtered frames (IQ'_n) are then processed through two instances of the CFM pipeline. One instance handles frame pairs separated by PRI_1 – specifically $[IQ'_{2n}; IQ'_{2n+1}]$ – while the second instance processes frame pairs separated by PRI_2 – namely $[IQ'_{2n+1}; IQ'_{2(n+1)}]$. Using the standard lag-one autocorrelation method, this processing ultimately yields two distinct phase maps, which are converted into axial velocities V_{D1} and V_{D2}. A subsequent velocity unwrapping stage combines the two maps to generate an alias-free velocity estimate. This is achieved using a precomputed lookup table, which is based on the theoretical relationships between each PRF and its corresponding Nyquist velocity limit. This table maps the measured velocity pairs (V_{D1}, V_{D2}) to integer Nyquist numbers (n_{N1}, n_{N2}), which indicate how many times the true velocity exceeds the Nyquist limit.

To obtain alias-free Doppler velocity estimates, for each pixel, the left-hand side of equation (34) is evaluated using the measured velocities V_{D1} and V_{D2}:

$$nint\left(q_2 \frac{V_{D2} - V_{D1}}{2V_{N1}}\right) = n_{N1} q_2 - n_{N2} p_2 \tag{34}$$

where *nint*() denotes the nearest-integer function, which rounds its argument to the closest integer value.

Next, for each pixel, the Nyquist number $n_{N1}$and n_{N2} are determined using the precomputed lookup table. Once the Nyquist numbers have been identified, the unaliased velocities are reconstructed and combined using a weighted average:

$$V_D^u = \frac{(V_{D1} + 2n_{N1}V_{N1}) + \frac{q_2}{p_2}(V_{D2} + 2n_{N2}V_{N2})}{1 + \frac{q_2}{p_2}} \quad (35)$$

Staggered VDI Module

The S-PRF VDI method, illustrated in Fig. 4.2b, extends the VDI pipeline (Chapter 3) to support dual-PRF acquisition. For each reception angle (±β), Doppler data are acquired using two distinct PRFs (PRF_1 and PRF_2). These two IQ_n data streams, $IQ_n(-\beta)$ and $IQ_n(+\beta)$, are independently processed through two parallel S-PRF CF and S-PRF CFM algorithms.

Specifically, each SPR-CF block separates the IQ_n frames into even (IQ_{2n}) and odd (IQ_{2n+1}) frame sequences, according to the indexes reported in Fig. 4.1, and processes them using identical instances of the clutter filtering algorithm. The clutter-filtered frames (IQ'_n) are then processed through two instances of the S-PRF CFM pipeline, which apply the standard lag-one autocorrelation method to compute two distinct phase maps. These phase maps are converted into axial velocity estimates V_{D1} and V_{D2}. Then, the unaliased velocities are reconstructed using equation (35), after determining the Nyquist numbers (n_{N1}, n_{N2}).

The resulting pair of unaliased axial velocity maps, one for each aperture $V_D^u(-\beta)$ and $V_D^u(+\beta)$, are then used to reconstruct the final 2-D vector velocity field (V_x and V_z) using the same triangulation approach employed in the standard VDI pipeline.

4.2.4 Experimental setup

The S-PRF algorithm implemented on the ULA-OP 256G platform was validated through phantom experiments using HFR CFM and VDI, with and without staggered sequences.

Experiments were performed using an LA533 linear array probe (Esaote SpA, Florence, Italy), incorporating 192 elements with a pitch of 245 μm, a center frequency of 8 MHz, and a 100% fractional bandwidth. The probe was connected to the ULA-OP 256G system, which was configured to transmit plane waves using all 192 elements. Six-cycle, Hamming-tapered, 6 MHz sinusoidal bursts were used in transmission. In

reception, echo data were processed from a depth range of approximately 25 mm, corresponding to 2538 samples at 78.125 MHz. RF data were parallel beamformed, demodulated, streamed to the embedded GPU, and processed according to the CFM or VDI algorithm, always in real-time.

Following real-time GPU processing, the acquired data are stored in dedicated files. This approach allows for comprehensive MATLAB post-processing and detailed analysis, facilitating a further evaluation of the imaging results and supporting quantitative assessments of the implemented algorithms.

HFR-CFM experiments were performed on a custom flow phantom designed to emulate physiological vascular conditions. A blood-mimicking fluid (Model 069-DTF, CIRS, USA) was circulated through a closed-loop hydraulic circuit using a peristaltic pump (Model 120U, Watson Marlow, UK), which generated a pulsatile flow. The flow passed through a tissue-mimicking phantom developed in-house [87], containing a 7-mm-diameter vessel embedded at a depth ranging between 20 mm and 40 mm, and tilted at an angle of 15° with respect to the probe surface.

HFR-VDI acquisitions were performed using a rotating disk phantom. The disk was immersed in water and rotated at controlled angular velocities ranging from 100 deg/s to 2500 deg/s, with increments of 50 deg/s. This configuration generated a radial velocity field with known tangential velocity profiles, providing a reference for quantitative validation. The probe was aligned to face the side of the rotating disk, perpendicular to its surface, and centered along its diameter. Vector Doppler data were acquired using receive angles β of $\pm 10°$, realized through dynamic receive beamforming. The PRF of the primary sequence was set to 2400 Hz, while the secondary PRF was configured as three-quarters of the primary PRF (p_2=3, q_2=4).

4.3 Results

Fig. 4.3 shows CFM images at the peak of the pulsatile flow obtained using conventional (left column) and staggered (right column) PRF sequences. Two acquisitions were performed: the first used a high PRF (PRF_{1A} = 1000 Hz) to avoid aliasing, and the second used a lower PRF

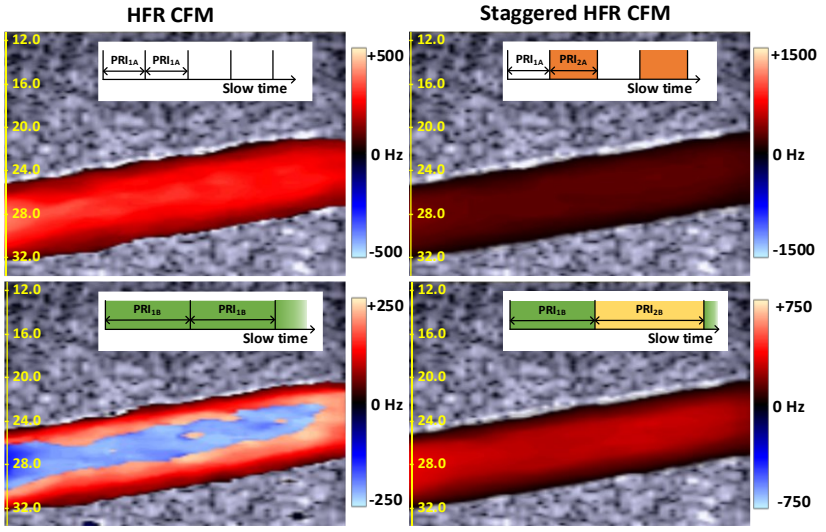


Fig. 4.3: Comparison between (left) conventional and (right) staggered HFR CFM at the peak of the pulsatile flow. At (top) high PRF, both methods yield alias-free results. At (bottom) low PRF, conventional CFM shows aliasing artifacts, while staggered PRF recovers a continuous and unaliased velocity field. Please note that the Doppler color bar differs in each subfigure, as the maximum velocity value is different.

(PRF_{1B} = 500 Hz) to induce aliasing in the conventional HFR CFM configuration. In staggered acquisitions, the secondary PRFs (PRF_{2A} and PRF_{2B}) were each set to three-quarters of the corresponding primary PRFs.

In the high-PRF condition (top row), both conventional and staggered HFR CFM generated alias-free results with similar phase distributions. However, in the low-PRF condition (bottom row), conventional HFR CFM displayed clear aliasing artifacts, visible as phase wrapping in the velocity map. In contrast, the staggered HFR CFM successfully compensated for aliasing and recovered a smooth, continuous phase map. The color scale of the images reflects different phase ranges: conventional CFM is limited to $\pm PRF_1/2$, while the staggered method, with $PRF_2=3/4PRF_1$, extends the range up to $\pm 3PRF_1/2$.

To further support the qualitative results, Fig. 4.4 reports quantitative data extracted from the GPU-processed acquisitions and analyzed offline in MATLAB. The top plot shows the temporal velocity profile at the vessel center for the three acquisitions (high PRF - PRF_{1A}, here used as reference,

low PRF (L) - PRF_{1B}, and staggered PRF (S) with low primary PRF – PRF_{1B} and PRF_{2B}). With the low-PRF setting, aliasing is clearly visible as abrupt sign inversions and discontinuities in the estimated velocity. In contrast, the staggered PRF trace closely follows the reference profile obtained at high PRF, accurately reproducing both the systolic peak and the deceleration phase.

The bottom plots present velocity profiles at the same point for 4 representative time instants (t_1–t_4), corresponding to early systolic acceleration (t_1), systolic peak (t_2), late systolic deceleration (t_3), and diastole (t_4). While the low-PRF acquisition is severely distorted by phase wrapping, producing inconsistent velocity distributions across the vessel segment, the staggered PRF reconstruction consistently overlaps with the

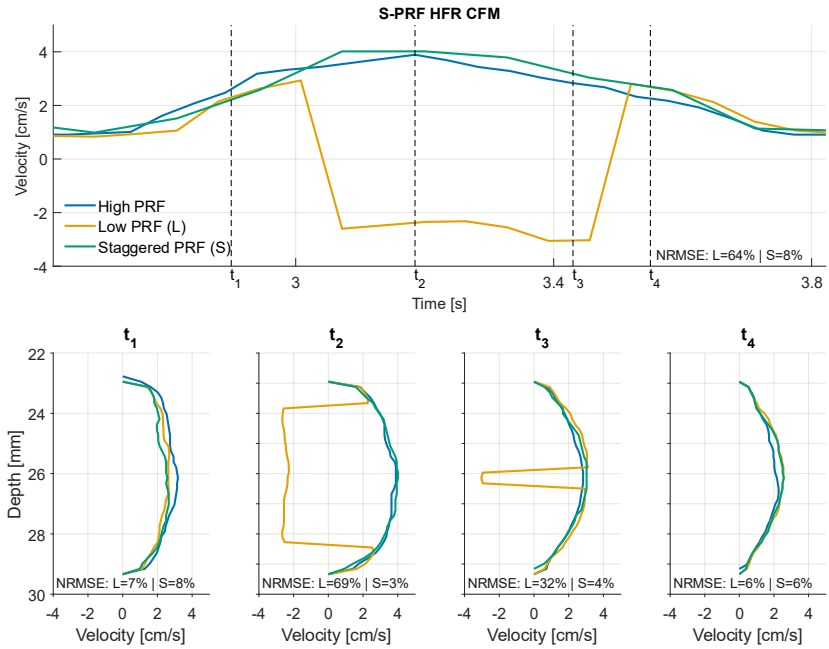


Fig. 4.4: Quantitative evaluation of HFR CFM acquisitions with high PRF (blue), low PRF (orange), and staggered PRF (green). Top: temporal velocity profile at a single point located at the vessel center. Bottom: spatial velocity profiles along a segment crossing the center of the vessel at four representative time instants (t_1–t_4). NRMSE values are reported for each case, showing large errors for the low-PRF acquisition (up to 69%), while the staggered PRF reconstruction consistently achieves low errors (<8%), closely matching the high-PRF reference.

high-PRF reference, confirming that the method restores a smooth and physiologically plausible velocity field. The normalized root mean square error (NRMSE) values reported in the figure confirm this observation: in the low-PRF case (L), errors increase at the instants affected by aliasing (69% at t_2 and 32% at t_3), whereas they remain below 8% for staggered PRF (S) across all time instants.

Staggered PRF vector Doppler imaging was validated through experimental tests based on the rotating disk. Representative vector velocity maps at 2000 deg/s are shown in Fig. 4.5 for conventional VDI (left) and dual-PRF VDI (right). In the conventional case, aliasing leads to discontinuities and incorrect flow direction in regions of higher velocity, particularly along the outer edge of the disk. In contrast, the dual-PRF implementation effectively resolves these artifacts, yielding smooth and continuous velocity vectors that correctly follow the rotational flow pattern.

Fig. 4.6 reports the global error analysis of dual-PRF velocity estimation as a function of the disk rotation speed, expressed in terms of tangential velocity. The root-mean-square error (RMSE, blue circles)

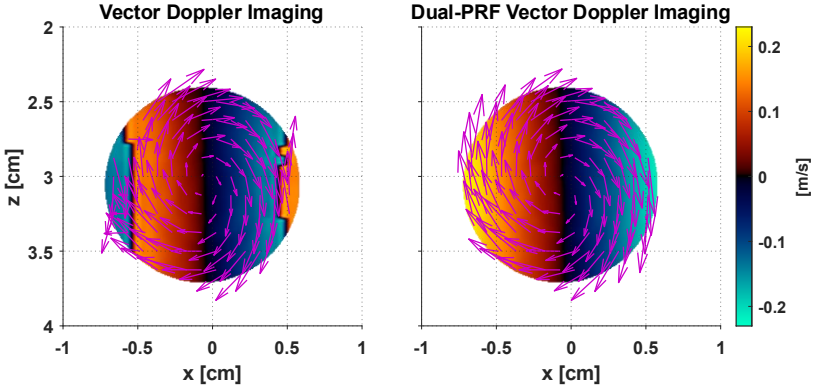


Fig. 4.5: Comparison between (left) conventional vector Doppler imaging and (right) dual-PRF vector Doppler imaging on the rotating disk phantom at 2000 deg/s. In the conventional case, aliasing artifacts produce discontinuities and incorrect flow directions, particularly in high-velocity regions. The dual-PRF implementation effectively resolves these artifacts, yielding smooth and continuous velocity vectors consistent with the expected rotational flow pattern.

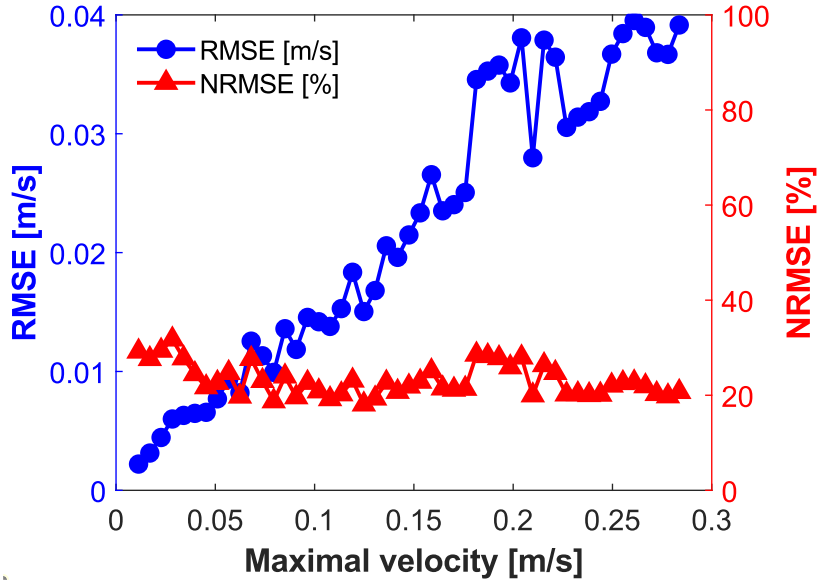


Fig. 4.6: Dual-PRF velocity estimation: global errors. Root-mean-square error (RMSE, blue circles) and normalized RMSE (NRMSE, red triangles) as a function of the maximal velocity.

increases with velocity amplitude but remains below 0.04 m/s across the tested range. The NRMSE (red triangles) remains around 20%, showing that while a residual error is present, the staggered approach effectively extends the unambiguous velocity range and ensures consistent velocity estimation at high rotational speeds.

4.4 Discussion

The results obtained with both CFM and VDI confirm the effectiveness of the S-PRF strategy in overcoming the PRF-related limitations of conventional HFR Doppler imaging. For CFM, the staggered approach quantitatively demonstrated robust alias suppression while preserving the temporal and spatial velocity profiles of the flow, thereby extending the usable PRF range without sacrificing accuracy. For VDI, the qualitative improvement in velocity vector continuity was supported by quantitative error analysis, which showed that the method maintains stable performance across a wide range of rotational speeds. Together, these findings validate

the robustness of S-PRF in extending the unambiguous velocity range for vector Doppler imaging.

A key strength of this work is the real-time implementation of the S-PRF method on a heterogeneous ultrasound open platform (ULA-OP 256G). By combining FPGAs for beamforming, DSPs for demodulation, and GPU acceleration for Doppler processing and velocity unwrapping, the system achieved high-throughput performance while maintaining both spatial and temporal resolution. This confirms the practical feasibility of applying staggered PRF Doppler imaging in real-time scenarios, even under reduced PRF constraints.

The proposed methodology is particularly advantageous when hardware resources, such as beamforming speed, data transfer bandwidth, or computational capacity, limit the achievable PRF. In such cases, S-PRF provides a computationally efficient and scalable solution to recover alias-free velocity information without requiring modifications to the base acquisition architecture.

Chapter 5. Computationally efficient SVD filtering for ultrasound flow imaging and real-time application to ultrafast Doppler

This chapter presents the implementation and validation of a computationally efficient SVD filter for ultrafast Doppler imaging. The method was integrated into the ULA-OP 256G platform through a GPU-based implementation, enabling real-time operation. Experimental results demonstrated the feasibility of clutter suppression for microvascular flow visualization, confirming the potential of SVD filtering in high-frame-rate applications.

The PhD candidate contributed to the real-time software implementation and system integration of the SVD-based processing pipeline within the ULA-OP 256G platform.

This chapter is based on the following publication:

- *B. Pialot, F. Guidi, **G. Bonciani**, F. Varray, T. Loupas, P. Tortoli, and A. Ramalli, "Computationally Efficient SVD Filtering for Ultrasound Flow Imaging and Real-Time Application to Ultrafast Doppler," in IEEE Transactions on Biomedical Engineering, vol. 72, no. 3, pp. 921-929, March 2025, doi: 10.1109/TBME.2024.3479414.*

5.1 Introduction

As introduced in Chapter 1, the suppression of tissue clutter is a crucial step in Doppler imaging, particularly in the visualization of the microvasculature where blood and tissue may exhibit similar velocity ranges [88,89]. Ultrafast imaging, based on plane or diverging waves, enables high spatiotemporal resolution and provides the large data volumes required by multidimensional filtering approaches [90,91]. In particular, the SVD, a data-driven matrix factorization algorithm, has become the reference clutter filtering method for ultrafast ultrasound [64,92]. In essence, SVD separates ultrasound data into spatiotemporal subspaces where the different components of the imaging medium are ranked according to their coherence in space and time. Since tissue presents more coherent displacements and is more homogeneous than blood, SVD isolates it from microvascular flow, even if their velocities are close. Thanks to SVD filtering, ultrafast ultrasound is now a very promising modality for microvasculature imaging, as its unique combination of high spatiotemporal resolution, penetration depth, and safety makes it an attractive alternative to magnetic resonance imaging, computed tomography, or optics [93,94].

The first ultrafast technique using SVD for microvascular imaging was ultrafast power Doppler (UPD) [64,95], which calculates the average power of microvascular blood flow within each packet after clutter filtering. As the power signal is proportional to blood volume, the strong clinical potential of UPD has been demonstrated in various applications involving microvascular hemodynamics, such as the detection of cognitive disorders in neonates [96], intraoperative temporal changes in coronary blood volume [97], or synovitis in rheumatoid arthritis [98]. Another essential use of ultrafast imaging is ultrasound localization microscopy (ULM) [68,99]. In this technique, microbubbles injected into the bloodstream are tracked over time with sub-wavelength precision to produce a super-resolved map of the microvasculature. SVD is currently the most widely used technique to isolate bubbles from tissue, although nonlinear approaches can be more suitable for specific ultrasound frequencies and blood velocity ranges [100]. Since ULM overcomes the

classic compromise between penetration depth and resolution, it allows the visualization of microvessels in deep tissue down to the microscopic scale. The potential clinical applications of the technique have been reviewed in [101] and [102], including tumor imaging, carotid plaque instability detection, and microvascular architecture assessment in liver and kidney chronic diseases.

However, a well-known limitation of SVD is its computational burden. Indeed, the computational complexity of SVD is much higher than that of a temporal filter, which translates into a detrimentally long processing time. Moreover, this complexity increases quadratically with the number of input images, thus posing a fundamental problem since SVD can require up to several hundred images to effectively isolate microvascular flow [64,103]. Therefore, the computational burden of SVD impacts UPD and ULM in different ways. SVD's long processing time is the main obstacle to a clinical translation of UPD. Indeed, real-time feedback is essential for UPD clinical imaging, as it is needed to interactively guide the clinician in probe positioning and image quality optimization, with obvious positive repercussions on the diagnostic capability. ULM, instead, does not require any real-time feedback, as it relies on long acquisition times (minutes) to accumulate enough information from bubbles. However, the accumulation also creates a large amount of data to be processed by computationally costly calculations, resulting in very long times to produce a single image [101,102]. Hence, the acceleration of SVD processing would be a valuable step towards a more computationally efficient implementation of ULM, facilitating its adoption in the clinic.

The first approach proposed to make SVD more computationally efficient in ultrafast imaging was to use a randomized algorithm (rSVD) [103]. Indeed, rSVD lightens the number of numerical operations of SVD as it only requires the tissue subspaces to perform the filtering instead of the complete factorization of data. The technique achieved real-time UPD imaging when implemented in a Verasonics research scanner using a 12-core CPU [104]. However, real-time was achieved through heavy spatial subsampling of the images, which inevitably leads to information loss. Another approach, proposed by some of this paper’s authors, is to use a simplified form of SVD (sSVD), which involves calculating its

components from the temporal covariance matrix of the input data [65]. From a numerical point of view, the counterpart of sSVD is a larger numerical error in estimating these components [105], but with no significant impact on the resulting UPD images. Although it has been demonstrated that sSVD-based UPD imaging has a significantly reduced computational load compared to a standard implementation of SVD, the processing time was still too high for real-time clinical applications. Furthermore, this implementation presents two fundamental drawbacks. Firstly, it takes advantage of suppressing the temporal components of SVD to speed up its calculation, since they are not necessary for UPD imaging [65]. However, it is not possible to omit these components in the case of ULM imaging. Second, the processing time is highly dependent on the number of subspaces occupied by blood and, consequently, on the characteristics of the ROI. Finally, for both rSVD and sSVD, the corresponding studies did not consider SVD subspace thresholding in their implementation. Yet, finding the thresholds that delimit the blood subspaces is the most crucial step in SVD filtering, as a wrong selection can significantly impact the quality of the resulting images [106]. As this selection should be automatized to guide the clinician, it is essential to also include adaptive thresholding methods without compromising the acceleration of SVD. In particular, one of the most widely used thresholding methods is the spatial similarity matrix (SSM) estimator, which is based on the correlation properties of the spatial components of SVD. Indeed, the robustness of this estimator to various tissue motion conditions has been demonstrated previously [22].

In this work, we show that by suitably implementing sSVD on a graphics processing unit (GPU sSVD), its processing time can be greatly reduced for ULM and made compatible with real-time UPD imaging. Indeed, while the classic SVD algorithms are not well suited to parallelization and, thus, suboptimal for GPUs implementation, the sSVD algorithm consists mainly of matrix products, which are highly parallelizable operations. Therefore, GPU sSVD can quickly filter large packets of frames without discarding temporal components, i.e., it is compatible with ULM. Furthermore, the efficiency of GPU sSVD is proved by the first implementation of UPD imaging with real-time SVD

filtering and the SSM estimator, on the ULA-OP 256G research scanner [107].

5.2 Materials and Methods

5.2.1 SVD filtering

Theory and numerical computation

Let's consider the matrix $I \in \mathbb{C}^{m,n_t}$ where beamformed and (possibly) compounded ultrafast frames have been stacked in the columns, with m being the number of pixels and n_t the number of images. The SVD of I is:

$$I = USV^H \tag{36}$$

where $U \in \mathbb{C}^{m,n_t}$ and $V \in \mathbb{C}^{n_t,n_t}$ are unitary matrices, $S \in \mathbb{C}^{n_t,n_t}$ is a diagonal matrix, and the superscript H represents the conjugate transpose. The matrices U and V contain the spatial and temporal singular vectors in their columns, respectively. Each pair of singular vectors $U(:,i) \in \mathbb{C}^m$ and $V(:,i) \in \mathbb{C}^{n_t}$ forms the i-th spatiotemporal subspace of I. The matrix S contains the n_t singular values $s_i \in \mathbb{R}^+$ of I in its diagonal that are ordered such that $s_1 > s_2 \ldots > s_{n_t}$. The singular value s_i weights the contribution of the i-th subspace to the frames. Note that equation (36) is the economy-size (or thin) expression of SVD, where only the n_t first subspaces have been kept since they are sufficient to obtain a full-rank factorization of I.

The SVD subspaces are linked to the three main components of ultrasound data: tissue, blood, and noise signals. The tissue signal, which is highly coherent both in space and time, occupies the low-rank subspaces with high singular values, the uncorrelated noise occupies the subspaces with low singular values, while blood components lie between the two subspaces.

In practice, the naive numerical approach to compute SVD is to use the covariance matrix $C = I^H I \in \mathbb{C}^{n_t,n_t}$. Indeed, the singular values are the square root of the eigenvalues of C and the singular vectors in V are its eigenvectors. However, this approach is usually not recommended, as the

multiplication $I^H I$ increases rounding errors and reduces numerical accuracy in estimating SVD components [105]. Consequently, SVD is generally performed using algorithms that access the eigencomponents of C but without actually computing the matrix. A review of the algorithms used in standard linear algebra libraries can be found in [105] and [108]. However, these algorithms consist of many serial operations that are not easily parallelizable and, thus, suboptimal for multicore devices such as multicore CPUs and GPUs.

Filtering

SVD filtering consists of finding the two thresholds i_t and i_n that separate the subspaces of blood from the ones of tissue and noise, respectively. Note that the threshold i_t is sufficient for isolating blood signals, while i_n is only used for denoising. Hereinafter, the indexes of the n_b blood subspaces will be denoted by α.

In practice, i_t and i_n are set either manually or using an adaptive algorithm. The SSM is probably the most common adaptive approach, and its high efficiency compared to numerous other algorithms has been demonstrated experimentally [106].

SSM thresholding

To find i_t and i_n, the SSM exploits the correlation properties of the spatial singular vectors in the columns of U. Indeed, the correlation within a subset of vectors belonging to the same component type, i.e., tissue, blood, or noise, is intrinsically higher than that of vectors belonging to different component types. This difference allows for the discrimination of tissue, blood, and noise subspaces. To this end, the SSM calculates the correlation between all the vectors as follows:

$$SSM = \frac{\widehat{U}^T \widehat{U}}{{n_t}^2} \tag{37}$$

where $\widehat{U}$ contains the singular vectors $U(:, i)$ in its columns, after they have been 1) converted in absolute value, 2) normalized to have zero mean, and

3) normalized to have a unit standard deviation. The superscript T represents the matrix transpose.

Because of the low correlation values between the tissue, blood, and noise subspaces, they generally appear as three distinct squares in the SSM. Each square shares the same diagonal as the SSM, with the boundary between the tissue and blood squares corresponding to i_t and that between the blood and noise squares corresponding to i_n. Consequently, both thresholds can be estimated by measuring the tissue/blood and blood/noise delineations. To this end, the most common approach is to fit two squares to the SSM, one for the tissue subspaces and the other for the blood subspaces. The fit then gives direct access to i_t and i_n [106].

Reconstruction of filtered images

Once α is identified, the filtered frames $I_\alpha \in \mathbb{C}^{m,n_t}$ can be reconstructed in space and time from SVD, as follows:

$$I_\alpha = U(:,\alpha)S(\alpha,\alpha)V(:,\alpha)^H \tag{38}$$

For example, this equation can be used to produce a ULM image. Otherwise, if the final goal is to have a UPD image $P \in \mathbb{R}^m$, it is sufficient to reconstruct the frames along space and use only the spatial vectors weighted by the singular values $U(:,\alpha)S(\alpha,\alpha) \in \mathbb{C}^{m,n_b}$ [65] :

$$P(m) = \sum_\alpha |U(m,\alpha)S(\alpha,\alpha)|^2 \tag{39}$$

where m is the pixel index.

5.2.2 sSVD filtering

Presentation and numerical computation

sSVD is based on the previously mentioned naive approach for calculating SVD. Indeed, as demonstrated in [65], the increase in numerical errors when computing C has no impact on UPD images, and it

will be shown, in this paper, that this statement is also true for ULM imaging. Compared with standard SVD algorithms, sSVD reduces the number of numerical operations and, thanks to its simplicity, enables filtering of tissue and noise subspaces to be included during the calculation of SVD components. Once the thresholds i_t and i_n have been manually fixed or estimated using SSM, the main steps of sSVD filtering are [65]:

1) MM1: Calculate the covariance matrix: $C = I^H I$
2) E1: Calculate the singular vectors $V(:, i)$ as the eigenvectors of C
3) MM2: Calculate $U(:, \alpha)S(\alpha, \alpha) = IV(:, \alpha)$
4) MM3: Obtain I_α using equation (38) (only for ULM)

where MM stands for 'matrix multiplication' and E 'eigendecomposition'.

The sSVD processing time is typically dominated by the matrix products MM1, MM2, and MM3. As the size of these products is fixed by α, the processing time of sSVD on the CPU is thus strongly dependent on the number of blood subspaces [65]. However, even if their execution is serial, each of these matrix products can be significantly accelerated using parallel routines like those widely available for GPUs. It will be shown that this acceleration minimizes the dependence of sSVD processing time on α.

GPU implementation

GPU sSVD was implemented in CUDA language using functions from optimized linear algebra libraries [109,110]. 32-bit floating-point complex data was used in all experiments. E1 was implemented with the cuSOLVER function 'cusolverDnCheevd', which calculates the eigen components of a matrix. As such, it is worth mentioning that this function also calculates singular values if needed. MM1, MM2 and MM3 were implemented with the cuBLAS matrix multiplication function 'cublasCgemm3m'. All steps were implemented sequentially, with each step waiting for the previous one to finish. For the validation (see next section), the data transfer to the GPU memory was accelerated using pinned memory.

SSM thresholding was implemented on GPU by adapting in CUDA kernels a recently published sum-and-table algorithm that strongly reduces its computational cost [111].

5.2.3 Performance assessment and validation

The proposed sSVD was implemented offline on a GeForce RTX 3060 GPU (NVIDIA, USA) for performance assessment and validation. Beamformed and IQ-demodulated datasets acquired under different conditions were analyzed to compare GPU sSVD with standard SVD algorithms in terms of processing time and UPD/ULM image quality. To this end, two implementations of SVD were used: the 'svd' function of MATLAB executed on an i7-12700 2.10 CPU processor (Intel, USA), and the 'cusolverDnCgesvd' GPU function of the cuSOLVER library. These functions will be referred to as MATLAB and cuSOLVER SVD, respectively. Both functions were set to run the economy-size SVD.

Processing time

The processing times of GPU sSVD, MATLAB SVD and cuSOLVER SVD were compared as a function of the PS, pixels, and blood subspaces α. To be the most generalizable, the processing time of GPU sSVD was measured using its complete algorithm, from MM1 to MM3, and equation (38) was included in the execution of MATLAB SVD and cuSOLVER SVD. The time needed to transfer data to the GPU was included in the processing time of GPU sSVD and cuSOLVER SVD. When varying PS, a beamforming grid of 128 depth samples and 96 lines was used (12288 pixels). The same beamforming grid was interpolated along depth to artificially increase the number of pixels. PS was fixed to 512. α was set at 80% of PS for both evaluations. When varying α, the beamforming grid was kept to its initial size, and 512 frames were used.

Image quality

The quality of the UPD images resulting from GPU sSVD and MATLAB SVD was compared in a controlled *in vitro* experiment. Unsteered plane wave transmissions were performed on a CIRS ATS 524

flow phantom (Sun Nuclear, USA) with the ULA-OP 256G research scanner [107] (MSDLab, Italy) when enabling 128 active elements of an 8 MHz linear probe (model LA533, Esaote, Italy). A steady flow was imposed in the 4 mm diameter tube of the phantom using a 120U peristaltic pump (Watson Marlow, UK).

The peak velocity was set at 10 cm/s, which is higher than the usual range for SVD filtering but sufficient to produce UPD images that were significantly better than with a temporal filter. Four acquisitions of 512 frames were made transmitting unsteered plane waves at a PRF of 2000 Hz. Acquisitions were repeated for different transmit voltages (from 20% to 100% of the maximum voltage allowed by the ULA-OP) to assess the robustness of GPU sSVD to different signal-to-noise ratios. Only one tissue/blood i_t threshold was used for all acquisitions and its value was the same for GPU sSVD and MATLAB SVD. UPD images from both filters were compared in terms of contrast-to-noise ratio (CNR):

$$CNR = 10log\left(\frac{|\mu_s - \mu_n|}{\sigma_n}\right) \tag{40}$$

where μ_n and μ_s, and σ_n are the mean and the standard deviation in the noise (subscript n) and flow regions (subscript s) selected on the UPD image, respectively. Also, the root-mean-square (RMS) difference between the GPU sSVD and MATLAB SVD images was measured in the flow ROI and normalized by the mean power of both images.

In addition, the UPD images were compared when automating the threshold selection with the SSM estimator. To this end, an acquisition was performed on the thyroid microvasculature of a healthy volunteer. Seven plane waves with angles from -5° to 5° were used, resulting in a post-compounding frame rate of 600 Hz. The thresholds i_t and i_n evaluated by the SSM and the resulting UPD images were compared with those obtained by MATLAB SVD and GPU sSVD as a function of PS, from 64 to 512. The *in vivo* tests were approved by the institutional review board on ethics of the University of Florence (approval number 309/2024) and informed consent was obtained from each volunteer.

Finally, the ULM images resulting from GPU sSVD and MATLAB SVD were compared using data and MATLAB codes from the open-source PALA toolbox [112]. The first 105 blocks of the InVivoRatBrainBolus dataset were used, each containing 800 frames acquired on a rat brain with circulating microbubbles. A threshold i_t=5 was used for both GPU sSVD and MATLAB SVD, followed by a [50, 250] Hz bandpass filter as recommended in the PALA toolbox. The localization of bubbles was performed using the radial algorithm provided by the toolbox. The saturation curves of the ULM images [113] were used to make a quantitative comparison between both filters. The saturation curve represents the % of pixels filled by at least one microbubble as a function of the number of blocks (or acquisition time).

5.2.4 Real-time UPD imaging

The computational effectiveness of sSVD was tested in the real-time UPD imaging case study. The setup consisted of a Jetson AGX Orin GPU SoM (NVIDIA, USA) with 32 GB RAM embedded in the ULA-OP 256G system. As introduced in Chapter 2, the SoM GPU is embedded in the scanner as a co-processor of IQ data, thus eliminating the data transfer cost to any external GPU memory. A PCIe connection handles high-speed communication between GPU and hardware. The GPU continuously receives 32-bit beamformed, compounded IQ data over a grid of 128 depth samples and 64 lines through the PCIe. The GPU applies a first CUDA stream to the incoming data, where sSVD filtering is performed continuously on a packet of frames with a pre-selected size, from 64 to 1024. The i_t and i_n thresholds can be set manually during the acquisition or automatically estimated using the SSM. The filtered packets of frames are placed in a circular buffer where a second CUDA stream is applied in parallel with the first one to calculate UPD images using the same PS as sSVD. The second stream averages the power Doppler signal in each pixel and applies a first-order IIR persistence filter to the images. Finally, the resulting UPD images are sent to a host PC through a USB 3.0 connection for real-time display.

To quantify the performance of GPU sSVD once embedded in the ULA-OP, its filtering rate was measured as $FiltRate = \frac{1}{t}$, where t is its processing time. The filtering rate must be higher than the display frame rate which is equal to $\frac{PRF}{N \times n_t}$, where PRF is the input (front-end) pulse repetition frequency, and N is the number of compounded plane waves, to make sure that the SVD stage does not have an impact on real-time imaging. Therefore, the maximum input PRF that can be used is $PRF_{max} = FiltRate \times N \times n_t$.

For validation, real-time UPD imaging was performed on the forearm and thyroid of two healthy volunteers using the LA533 probe and an unsteered plane wave. The transmission frequency was set at either 6 MHz or 8 MHz, which offers different trade-offs between resolution and attenuation. The imaging depth was set between 5 and 30 mm in all acquisitions.

5.3 Results

5.3.1 Offline validation

Processing time

Fig. 5.1 shows the comparison of processing times for MATLAB SVD, cuSOLVER SVD, and (A) GPU sSVD as a function of PS, (B) the number of depth samples, and (C) the number of blood subspaces. In all cases, the processing time of GPU sSVD increases much less with PS compared to MATLAB and cuSOLVER SVD. In (A), for 128 frames, the mean processing time of GPU sSVD is 3.0 ms, which represents a ×4 speedup over cuSOLVER SVD (12.6 ms), and a ×8 speedup over MATLAB SVD (22.8 ms). For 1024 frames, the mean processing time of GPU sSVD is 92.7 ms, which corresponds to more than a ×15 speedup compared to both MATLAB (1490.3 ms) and cuSOLVER (1595.4 ms) SVD. Note that the increase in processing time as a function of PS for MATLAB and cuSOLVER SVD is quadratic, as expected. The increase is also quadratic for GPU sSVD but with a smaller slope, making it appear flatter due to the scale of the figure. When varying the number of depth samples in (B), GPU

sSVD also presents a processing time that grows significantly lower than the other filters. However, its speedup over cuSOLVER SVD decreases when more samples are used. For instance, when increasing the number of depth samples from 128 to 512, the speedup of GPU sSVD increases from ×10 to ×15 when compared to MATLAB SVD, and decreases from ×18 to ×9 when compared to cuSOLVER SVD. Finally, we see in (C) that the flow characteristics of pixels inside the ROI have negligible influence on GPU sSVD. Indeed, its speedup over MATLAB SVD remains stable whatever the number of blood subspaces, and its speedup over cuSOLVER SVD slightly decreases, from ×14 to ×11 when 10% and 80% of subspaces are occupied by blood, respectively.

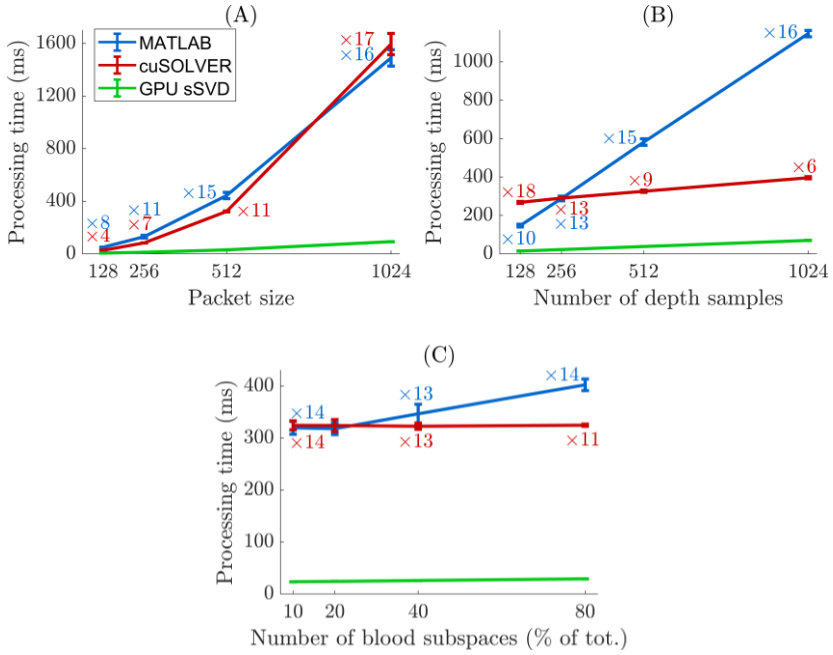


Fig. 5.1: GPU sSVD processing time compared to MATLAB SVD and cuSOLVER SVD as a function of (A) packet size, (B) number of depth samples, and (C) number of subspaces belonging to blood, expressed in %. For the three panels, the processing time is displayed on the vertical axis, while the speedup factor over MATLAB SVD and cuSOLVER SVD is highlighted directly on the trend lines for each tested configuration.

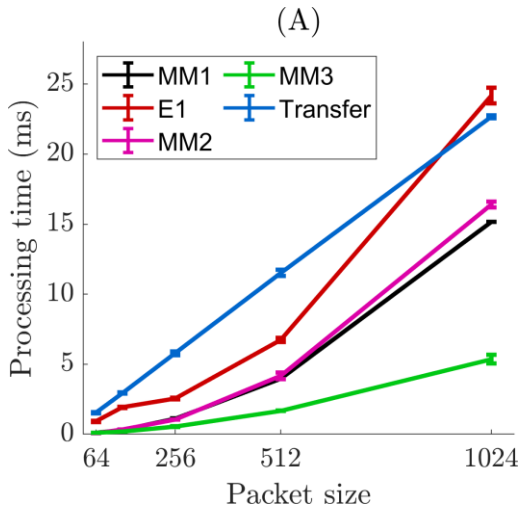


Fig. 5.2: Processing time of the GPU sSVD steps (MM1, E1, MM2, MM3) and data transfer to the GPU memory (Transfer), as a function of PS.

Fig. 5.2 shows the processing time of each GPU sSVD step as a function of PS, plus the time required to transfer the data to the GPU's memory (Transfer). The costliest operations are the memory transfers and eigendecomposition E1, where the latter is dominant only for 1024 frames.

UPD imaging

Fig. 5.3 shows a UPD image obtained during the phantom experiment and filtered using (A) MATLAB SVD and (B) GPU sSVD. Noise and flow ROIs are indicated by the green and black boxes, respectively. The CNRs calculated using these ROIs for both filters are shown in (C) as a function of transmit voltage. GPU SVD causes no degradation in image quality, as the UPD images resulting from GPU sSVD and MATLAB SVD have nearly the same CNR. As a result, the UPD images of (A) and (B) are virtually indistinguishable visually, with a RMS difference of -80 dB within the flow ROI.

Fig. 5.4 shows a sample UPD image with corresponding SSM calculated from 256 frames of the volunteer's thyroid, for (A) MATLAB SVD and (B) GPU sSVD. The green and black boxes indicate noise and flow ROIs, respectively. The SSM of GPU sSVD is partially corrupted by

vertical and horizontal lines, which causes fluctuations in the estimation of the i_t and i_n thresholds, as shown in graphs (C) and (D), which plot them for both filters as a function of PS. Nevertheless, these fluctuations have no impact on the quality of UPD images, as the CNR of MATLAB SVD and GPU sSVD in (E) is almost identical. Hence, the UPD images in (A) and (B) are visually indistinguishable, with a RMS difference within the flow ROIs of -67±2 dB.

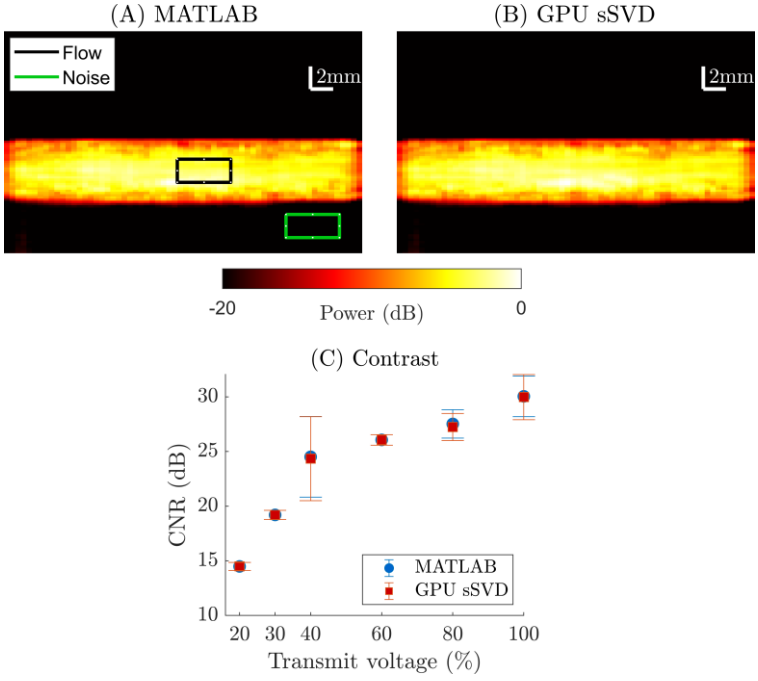


Fig. 5.3: (A) UPD image of the flow phantom obtained after clutter filtering with MATLAB SVD. The black/green box corresponds to the flow/noise ROI. (B) UPD image obtained with GPU sSVD. (C) CNR of UPD images measured as a function of the transmit voltage for MATLAB SVD and GPU sSVD.

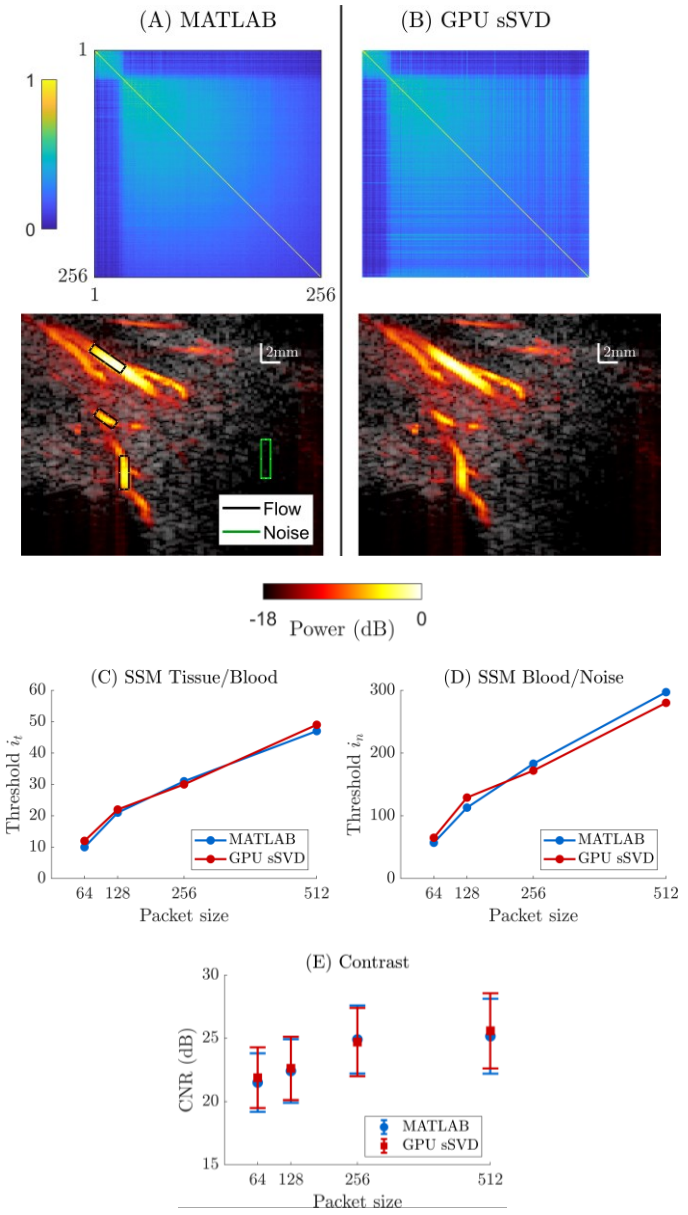


Fig. 5.4: Sample SSM maps (top panels) with the corresponding UPD images (lower panels) obtained in the thyroid with (A) MATLAB SVD and (B) GPU sSVD for 256 frames. The black/green box corresponds to the flow/noise ROI, (C) Tissue/ blood threshold i_t obtained with MATLAB SVD and GPU sSVD, (D) The same for the noise threshold i_n and E) CNR measured in the UPD images following MATLAB SVD and GPU sSVD as a function of the packet size.

ULM imaging

Fig. 5.5 shows the ULM images of the rat brain calculated after (A) MATLAB SVD and (B) GPU sSVD and the corresponding saturation curves in (C). Both filters produce indistinguishable ULM images and overlapping saturation curves. The 105 blocks of frames needed to calculate the ULM image were filtered in 3 minutes using MATLAB SVD and in 7 seconds only using GPU sSVD, resulting in a ×25 speedup.

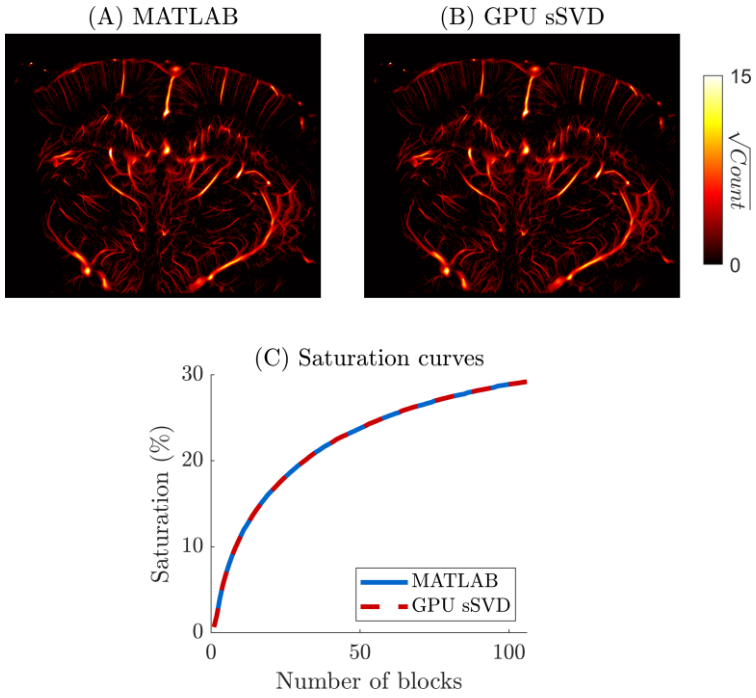


Fig. 5.5: ULM images obtained after clutter filtering with (A) MATLAB SVD and (B) GPU sSVD. (C) Saturation curves corresponding to both filters.

5.3.1 Real-time UPD imaging

Fig. 5.6 shows four UPD images acquired in real time with GPU sSVD filtering in the forearm (A, B, and C) and thyroid (D) of healthy volunteers. The image in (C) was acquired using the SSM to automate blood subspace selection. Transmit frequency, PS and i_t threshold (unless using SSM) are specified for each image. In (A) and (B), vascular networks can

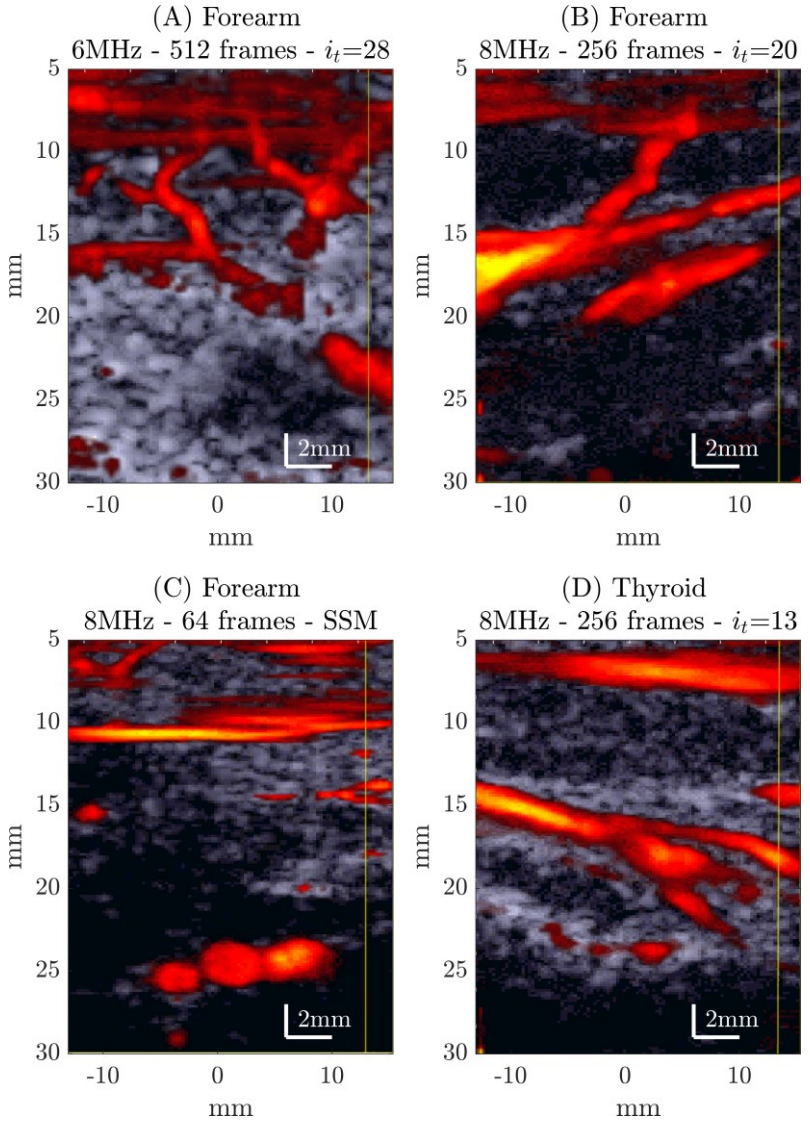


Fig. 5.6: UPD images acquired in real-time with GPU sSVD in the forearm (A, B, and C) and thyroid (D). The image in (C) has been acquired using the SSM estimator to automate the selection of blood subspaces. Transmit frequency, packet size, and i_t threshold (except when using SSM) are specified for each image. Videos corresponding to images (B) and (C) are included as supplementary materials.

be seen, with microvessels having a diameter of about 1 to 2 mm. In the lower part of the image (C), cross-sections of 3 adjacent microvessels are visible at 25 mm depth. Finally, (D) shows a microvessel dividing into several branches. For all four images, some artifacts visible in the near field (from 5 to around 10 mm in depth) were caused by correlated noise introduced by the acquisition system.

Fig. 5.7 shows the filtering rate of GPU sSVD once integrated into the ULA-OP 256. With or without SSM, the GPU sSVD has a very high throughput, being able for example to filter 31×512 = 15 872 input frames per second when a PS of 512 is used. Hence, GPU sSVD is fast enough to handle very high PRFs, enabling smooth real-time display even for large PS values. For example, for 1024 frames obtained with a single plane wave (no compounding), GPU sSVD can achieve a filtering rate of over 10 packets/s, which could support a maximum input PRF of over 10 kHz.

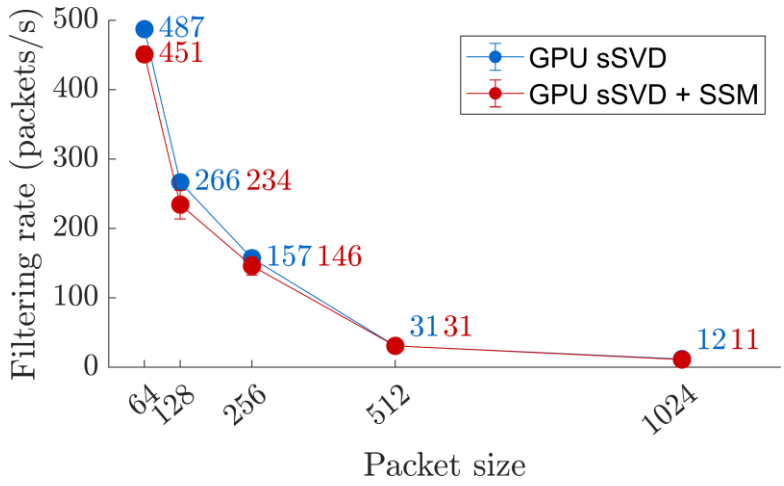


Fig. 5.7: Filtering rate of GPU sSVD and GPU sSVD combined with the SSM estimator.

5.4 Discussion

This work has presented the GPU implementation and real-time validation of a parallel formulation of the sSVD filter that was previously proposed in [65]. This approach significantly reduces the SVD processing

time compared with some of its standard implementations on both CPU and GPU. Despite the higher numerical error of GPU sSVD, this reduction has no impact on the resulting UPD and ULM images. In addition, the sSVD implementation proposed here is more versatile than the unparallel CPU implementation proposed in [65]. Indeed, the processing time of GPU sSVD depends mainly on the number of input frames and pixels, and, unlike on CPU, is marginally influenced by the number of blood subspaces and hence by the blood flow characteristics inside the flow ROI. Furthermore, GPU sSVD enables a sharp reduction in processing time without having to eliminate SVD temporal components, as in [65], making it suitable for ULM imaging. Finally, the effectiveness of GPU sSVD was here demonstrated by implementing it in the ULA-OP 256G research scanner and enabling, for the first time, ultrafast UPD imaging with real-time and adaptative SVD filtering.

The correlated noise in the SSM of GPU sSVD in Fig. 5.4 (B) is created by its higher numerical error than SVD. Indeed, like the covariance matrix calculated in MM1, the SSM is calculated by multiplying a matrix by its transpose, which further increases the rounding error to the point where it generates additional noise. However, as Fig. 5.4 shows, this noise preserves the pattern of squares in the SSM and has no significant impact on the automatic selection of thresholds. Nevertheless, if necessary, the noise can be eliminated by using a 64-bit floating-point data format when calculating the SSM on GPU with equation (37), as it increases numerical accuracy, but at the expense of an increased processing time.

Instead of SSM, GPU sSVD can be combined with any other thresholding methods, for example by exploiting the temporal information of the decomposition through the Doppler frequency [114] or central frequency [106] of the temporal singular vectors.

In all cases, we observed that the real-time capabilities of GPU sSVD once integrated in the ULA-OP 256G could favor the manual threshold selection. Indeed, the SSM estimator and probably most adaptive approaches are very sensitive to local ROI characteristics and probe positioning. Hence, using an adaptive method may generate greater sensitivity to hand movements, while the manual selection of thresholds can guarantee greater robustness, facilitated by the real-time display. It

should be noted that the sensitivity of SVD to local characteristics is a known problem, and one possible solution is to use a block-wise algorithm [114]. However, this approach may produce a grid-like pattern on the output images.

In addition to the applications presented here, the proposed GPU sSVD implementation is promising for all other applications that exacerbate the processing load requirements and, thus, significantly slow down the processing time. For example, an efficient implementation would be very useful in 3-D ULM, which inevitably increases the number of spatial samples to process, but allows for reducing operator dependency, accessing the out-of-plane component of blood flow [46], and limiting errors in bubble trajectory reconstruction [102].

The proposed method could be beneficial to speed-up other post-processing techniques for UPD and ULM imaging, such as coherence-based denoising [115] or adaptive beamforming [116], which require SVD filtering to be applied in the channel domain, i.e., after time-delaying the received signals, but before summing their contributions. However, here the number of spatial samples to be filtered overgrows, and, for a real-time operation, the performance of GPU sSVD will have to be further improved. An interesting solution could be to divide each of matrix product into several GPUs. Importantly, it must be noted that the eigen decomposition of size $n_t \times n_t$ in E1 will not be affected by the increase in the number of spatial samples.

Chapter 6. Summary and perspectives

This Chapter concludes the Ph.D. thesis by summarizing the work conducted and outlining the future research perspectives.

_____

The research work presented in this thesis was focused on the development and real-time implementation of innovative methodologies for high-frame-rate ultrasound imaging. This was made possible by the integration of an embedded GPU module into the ULA-OP 256 open scanner. Such an integration made it possible to overcome the limitations of previous implementations based on FPGAs and DSPs, enabling the real-time execution of advanced imaging techniques using high-level programming languages such as CUDA.

The main contributions of this thesis can be summarized as:

A. Increase of ULA-OP 256 processing power through GPU integration into the new system, named ULA-OP 256G;
B. Implementation of HFR CFM on the ULA-OP 256G;
C. Implementation of a new real-time vector Doppler imaging method on the ULA-OP 256G;
D. Development and validation on ULA-OP 256G of a staggered PRF strategy for extension of the Nyquist limit;
E. GPU-based implementation of SVD clutter filtering, with application to ultrafast Doppler imaging and to ULM.

The results achieved have shown the feasibility and effectiveness of approaches that had previously been based on DSPs (B), or confined to offline experimentation (C, D, E), opening new perspectives for their possible clinical translation.

The novel heterogeneous open platform has been demonstrated as a key enabler for innovation in ultrasound imaging, bridging the gap between methodological development and practical application. The direct integration of the GPU within the system, the direct access to raw data and the possibility of freely programming the processing chain make the ULA-OP 256G a versatile and powerful research environment, capable of accelerating the prototyping and validation of novel imaging techniques.

Several accelerated ultrasound research platforms have been proposed in the literature to support real-time or near real-time implementation of computationally demanding imaging techniques, including heterogeneous systems and GPU-based architectures. Early examples include platforms such as SARUS, as well as multi-GPU approaches designed to scale ultrasound processing across multiple discrete accelerators [5,6]. More

recent works have demonstrated the effectiveness of GPU acceleration for ultrafast Doppler imaging, vector Doppler methods, and SVD-based clutter filtering [82]. Compared to these approaches, the ULA-OP 256G platform adopts a system-level design that emphasizes tight integration between acquisition hardware and computational resources. While the proposed platform does not aim to maximize raw computational throughput, it provides a balanced and compact solution that effectively supports real-time processing, flexibility in algorithm development, and reproducibility of experimental results, making it particularly suitable for research-oriented and translational applications.

Although the proposed heterogeneous architecture enables the real-time execution of computationally demanding high-frame-rate ultrasound algorithms, some limitations related to the current implementation of the ULA-OP 256G system remain. The platform is presently optimized for two-dimensional imaging using linear array probes and plane-wave transmissions, which inherently limit spatial resolution and imaging depth in applications that would benefit from focused transmissions or volumetric acquisitions. Moreover, real-time performance is achieved under well-defined acquisition and processing configurations, where data throughput, memory transfers, and GPU workload are carefully balanced. More complex imaging scenarios—such as multi-angle compounding, larger fields of view, or three-dimensional imaging—would significantly increase data rates and computational demands, potentially approaching the limits of the current hardware configuration and requiring additional computational resources or further algorithmic optimizations.

Future developments of this work may include the extension of the proposed architecture to more advanced probe technologies, such as 2-D matrix array probes for real-time 3-D imaging. The integration of such probes would dramatically increase the amount of acquired data and the associated computational load, further highlighting the relevance of heterogeneous architectures with embedded GPUs. Additional future perspectives include the optimization of the implemented processing pipelines in terms of computational efficiency and robustness under more challenging imaging conditions, as well as extensive *in vivo* validation of the proposed methods, which represents a necessary step toward clinical

adoption. More broadly, the findings of this thesis suggest that heterogeneous processing architectures may inspire the design of a new generation of clinical ultrasound scanners, capable of integrating advanced computational resources to meet the growing demands of modern diagnostic imaging.

Bibliography

1. Powers J, Kremkau F. Medical ultrasound systems. Interface Focus. 2011 Aug 6;1(4):477–89. doi:10.1098/rsfs.2011.0027

2. Bercoff J, Montaldo G, Loupas T, Savery D, Mézière F, Fink M, et al. Ultrafast compound doppler imaging: providing full blood flow characterization. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2011 Jan;58(1):134–47. doi:10.1109/TUFFC.2011.1780

3. Tanter M, Fink M. Ultrafast imaging in biomedical ultrasound. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2014 Jan;61(1):102–19. doi:10.1109/TUFFC.2014.2882

4. Boni E, Yu ACH, Freear S, Jensen JA, Tortoli P. Ultrasound Open Platforms for Next-Generation Imaging Technique Development. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2018 Jul;65(7):1078–92. doi:10.1109/TUFFC.2018.2844560

5. Jensen JA, Holten-Lund H, Nilsson RT, Hansen M, Larsen UD, Domsten RP, et al. SARUS: A synthetic aperture real-time ultrasound system. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2013 Sep;60(9):1838–52. doi:10.1109/TUFFC.2013.2770

6. Cheung CCP, Yu ACH, Salimi N, Yiu BYS, Tsang IKH, Kerby B, et al. Multi-channel pre-beamformed data acquisition system for research on advanced ultrasound imaging methods. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2012 Feb;59(2):243–53. doi:10.1109/TUFFC.2012.2184

7. Jensen JA, Holm O, Jerisen LJ, Bendsen H, Nikolov SI, Tomov BG, et al. Ultrasound research scanner for real-time synthetic aperture data acquisition. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2005 May;52(5):881–91. doi:10.1109/TUFFC.2005.1503974

8. Smith PR, Cowell DMJ, Raiton B, Ky CV, Freear S. Ultrasound array transmitter architecture with high timing resolution using embedded phase-locked loops. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2012 Jan;59(1):40–9. doi:10.1109/TUFFC.2012.2154

9. Boni E, Bassi L, Dallai A, Meacci V, Ramalli A, Scaringella M, et al. Architecture of an Ultrasound System for Continuous Real-Time High Frame Rate Imaging. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2017 Sep;64(9):1276–84. doi:10.1109/TUFFC.2017.2727980

10. Risser C, Hewener H, Fournelle M, Fonfara H, Barry-Hummel S, Weber S, et al. Real-Time Volumetric Ultrasound Research Platform with 1024 Parallel Transmit and Receive Channels. Applied Sciences. 2021 Jan;11(13):13. doi:10.3390/app11135795

11. Lewandowski M, Walczak M, Witek B, Kulesza P, Sielewicz K. Modular & scalable ultrasound platform with GPU processing. In: 2012 IEEE International Ultrasonics Symposium [Internet]. Dresden, Germany: IEEE; 2012 [cited 2023 Dec 19]. p. 1–4. Available from: http://ieeexplore.ieee.org/document/6562104/ doi:10.1109/ULTSYM.2012.0518

12. So H, Chen J, Yiu B, Yu A. Medical Ultrasound Imaging: To GPU or Not to GPU? IEEE Micro. 2011 Sep;31(5):54–65. doi:10.1109/MM.2011.65

13. Vega M. Ultrasound Simulator Software. Verasonics [Internet]. 2019 Jan 30 [cited 2022 Feb 23]. Available from: https://verasonics.com/ultrasound-simulator/

14. Shamdasani V, Bae U, Sikdar S, Yoo YM, Karadayi K, Managuli R, et al. Research interface on a programmable ultrasound scanner. Ultrasonics. 2008 Jul;48(3):159–68. doi:10.1016/j.ultras.2007.11.009

15. Garland M, Le Grand S, Nickolls J, Anderson J, Hardwick J, Morton S, et al. Parallel Computing Experiences with CUDA. IEEE Micro. 2008 Jul;28(4):13–27. doi:10.1109/MM.2008.57

16. Lewandowski M, Jarosik P, Tasinkevych Y, Walczak M. Efficient GPU Implementation of 3D Spectral Domain Synthetic Aperture Imaging. In: 2020 IEEE International Ultrasonics Symposium (IUS) [Internet]. 2020 [cited 2024 Jun 11]. p. 1–3. Available from: https://ieeexplore.ieee.org/document/9251552 doi:10.1109/IUS46767.2020.9251552

17. Hyun D, Li YL, Steinberg I, Jakovljevic M, Klap T, Dahl JJ. An Open Source GPU-Based Beamformer for Real-Time Ultrasound Imaging and Applications. In: 2019 IEEE International Ultrasonics Symposium (IUS) [Internet]. 2019 [cited 2024 Jun 11]. p. 20–3. Available from: https://ieeexplore.ieee.org/document/8926193 doi:10.1109/ULTSYM.2019.8926193

18. Yiu BYS, Yu ACH. GPU-Based Minimum Variance Beamformer for Synthetic Aperture Imaging of the Eye. Ultrasound in Medicine & Biology. 2015 Mar;41(3):871–83. doi:10.1016/j.ultrasmedbio.2014.11.005

19. Yiu BYS, Walczak M, Lewandowski M, Yu ACH. Live Ultrasound Color-Encoded Speckle Imaging Platform for Real-Time Complex Flow

Visualization In Vivo. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2019 Apr;66(4):656–68. doi:10.1109/TUFFC.2019.2892731

20. Cacko D, Jarosik P, Lewandowski M. Real-time Shear Wave Elastography Implementation on a Portable Research Ultrasound System with GPU-accelerated Processing. In: 2023 IEEE International Ultrasonics Symposium (IUS) [Internet]. 2023 [cited 2024 Jun 11]. p. 1–4. Available from: https://ieeexplore.ieee.org/document/10307608 doi:10.1109/IUS51837.2023.10307608

21. Perrot V, Polichetti M, Varray F, Garcia D. So you think you can DAS? A viewpoint on delay-and-sum beamforming. Ultrasonics. 2021 Mar;111:106309. doi:10.1016/j.ultras.2020.106309

22. Evans DH, McDicken WN. Doppler Ultrasound: Physics, Instrumentation and Signal Processing. 2nd edition. Chichester ; New York: Wiley; 1999. 456 p.

23. Dunmire B, Beach KW, Labs KH, Plett M, Strandness DE. Cross-beam vector Doppler ultrasound for angle-independent velocity measurements. Ultrasound in Medicine & Biology. 2000 Oct 1;26(8):1213–35. doi:10.1016/S0301-5629(00)00287-8

24. Tortoli P, Manes G, Atzeni C. Velocity Profile Reconstruction Using Ultrafast Spectral Analysis of Doppler Ultrasound. IEEE Transactions on Sonics and Ultrasonics. 1985 Jul;32(4):555–61. doi:10.1109/T-SU.1985.31630

25. Bjaerum S, Torp H, Kristoffersen K. Clutter filter design for ultrasound color flow imaging. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2002 Feb;49(2):204–16. doi:10.1109/58.985705

26. Evans DH, Jensen JA, Nielsen MB. Ultrasonic colour Doppler imaging. Interface Focus. 2011 Aug 6;1(4):490–502. doi:10.1098/rsfs.2011.0017

27. Saxena A, Ng EYK, Lim ST. Imaging modalities to diagnose carotid artery stenosis: progress and prospect. BioMed Eng OnLine. 2019 Dec;18(1):66. doi:10.1186/s12938-019-0685-7

28. Kasai C, Namekawa K, Koyano A, Omoto R. Real-Time Two-Dimensional Blood Flow Imaging Using an Autocorrelation Technique. IEEE Transactions on Sonics and Ultrasonics. 1985 May;32(3):458–64. doi:10.1109/T-SU.1985.31615

29. Guidi F, Tortoli P. Real-Time High Frame Rate Color Flow Mapping System. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2021 Jun;68(6):2193–201. doi:10.1109/TUFFC.2021.3064612

30. Vos HJ, Voorneveld JD, Groot Jebbink E, Leow CH, Nie L, van den Bosch AE, et al. Contrast-Enhanced High-Frame-Rate Ultrasound Imaging of Flow Patterns in Cardiac Chambers and Deep Vessels. Ultrasound in Medicine & Biology. 2020 Nov 1;46(11):2875–90. doi:10.1016/j.ultrasmedbio.2020.07.022

31. Ramalli A, Rodriguez-Molares A, Avdal J, D'hooge J, Lovstakken L. High-Frame-Rate Color Doppler Echocardiography: A Quantitative Comparison of Different Approaches. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2020 May;67(5):923–33. doi:10.1109/TUFFC.2019.2958031

32. Rasouli R, Baranger J, Slorach C, Hui W, Venet M, Nguyen MB, et al. Local arterial stiffness measured by ultrafast ultrasound imaging in childhood cancer survivors treated with anthracyclines. Front Cardiovasc Med. 2023;10:1150214. doi:10.3389/fcvm.2023.1150214 PubMed PMID: 37346288; PubMed Central PMCID: PMC10279856.

33. Papadacci C, Finel V, Villemain O, Tanter M, Pernot M. 4D Ultrafast Ultrasound Imaging of Naturally Occurring Shear Waves in the Human Heart. IEEE Transactions on Medical Imaging. 2020 Dec;39(12):4436–44. doi:10.1109/TMI.2020.3020147

34. Jensen JA, Nikolov SI, Yu ACH, Garcia D. Ultrasound Vector Flow Imaging - Part II: Parallel Systems. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2016 Nov;63(11):1722–32. doi:10.1109/TUFFC.2016.2598180

35. Udesen J, Gran F, Hansen KL, Jensen JA, Thomsen C, Nielsen MB. High frame-rate blood vector velocity imaging using plane waves: Simulations and preliminary experiments. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2008 Aug;55(8):1729–43. doi:10.1109/TUFFC.2008.858

36. Lenge M, Ramalli A, Boni E, Liebgott H, Cachard C, Tortoli P. High-frame-rate 2-D vector blood flow imaging in the frequency domain. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2014 Sep;61(9):1504–14. doi:10.1109/TUFFC.2014.3064

37. Rossi S, Ramalli A, Fool F, Tortoli P. High-Frame-Rate 3-D Vector Flow Imaging in the Frequency Domain. Applied Sciences. 2020 Jan;10(15):15. doi:10.3390/app10155365

38. Yiu BYS, Yu ACH. Least-Squares Multi-Angle Doppler Estimators for Plane-Wave Vector Flow Imaging. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2016 Nov;63(11):1733–44. doi:10.1109/TUFFC.2016.2582514

39. Ricci S, Bassi L, Tortoli P. Real-time vector velocity assessment through multigate doppler and plane waves. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2014 Feb;61(2):314–24. doi:10.1109/TUFFC.2014.6722616

40. Hussain B, Yiu BYS, Yu ACH, Lacefield JC, Poepping TL. Investigation of Crossbeam Multi-receiver Configurations for Accurate 3-D Vector Doppler Velocity Estimation. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2016 Nov;63(11):1786–98. doi:10.1109/TUFFC.2016.2597135

41. Nowicki A, Tasinkiewicz J, Trots I. Flow imaging using differential Golay encoded ultrasound. Ultrasonics. 2022 Dec 1;126:106825. doi:10.1016/j.ultras.2022.106825

42. Jørgensen LT, Stuart MB, Jensen JA. Transverse oscillation tensor velocity imaging using a row–column addressed array: Experimental validation. Ultrasonics. 2023 Jul 1;132:106962. doi:10.1016/j.ultras.2023.106962

43. Liu P, de Hoop H, Schwab HM, Lopata RGP. High frame rate multi-perspective cardiac ultrasound imaging using phased array probes. Ultrasonics. 2022 Jul 1;123:106701. doi:10.1016/j.ultras.2022.106701

44. Meacci V, Boni E, Dallai A, Ramalli A, Scaringella M, Guidi F, et al. FPGA-based multi cycle parallel architecture for real-time processing in ultrasound applications. Lecture Notes in Electrical Engineering. 2019;550(9783030119720):295–301. Located at: Scopus. doi:10.1007/978-3-030-11973-7_34

45. Giangrossi C, Ramalli A, Dallai A, Mazierli D, Meacci V, Boni E, et al. Requirements and Hardware Limitations of High-Frame-Rate 3-D Ultrasound Imaging Systems. Applied Sciences. 2022 Jan;12(13):13. doi:10.3390/app12136562

46. Provost J, Papadacci C, Arango JE, Imbault M, Fink M, Gennisson JL, et al. 3D ultrafast ultrasound imaging *in vivo*. Phys Med Biol. 2014 Oct 7;59(19):L1–13. doi:10.1088/0031-9155/59/19/L1

47. Maffett R, Boni E, Chee AJY, Yiu BYS, Savoia AS, Ramalli A, et al. Unfocused Field Analysis of a Density-Tapered Spiral Array for High-Volume-Rate 3-D Ultrasound Imaging. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2022 Oct;69(10):2810–22. doi:10.1109/TUFFC.2022.3188245

48. Ramalli A, Harput S, Bézy S, Boni E, Eckersley RJ, Tortoli P, et al. High-Frame-Rate Tri-Plane Echocardiography With Spiral Arrays: From Simulation to Real-Time Implementation. IEEE Transactions on

Ultrasonics, Ferroelectrics, and Frequency Control. 2020 Jan;67(1):57–69. doi:10.1109/TUFFC.2019.2940289

49. Ramalli A, Dallai A, Guidi F, Bassi L, Boni E, Tong L, et al. Real-Time High-Frame-Rate Cardiac B-Mode and Tissue Doppler Imaging Based on Multiline Transmission and Multiline Acquisition. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2018 Nov;65(11):2030–41. doi:10.1109/TUFFC.2018.2869473

50. Mazierli D, Ramalli A, Boni E, Guidi F, Tortoli P. Architecture for an Ultrasound Advanced Open Platform With an Arbitrary Number of Independent Channels. IEEE Transactions on Biomedical Circuits and Systems. 2021 Jun;15(3):486–96. doi:10.1109/TBCAS.2021.3077664

51. Ricci S, Ramalli A, Bassi L, Boni E, Tortoli P. Real-Time Blood Velocity Vector Measurement Over a 2-D Region. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2018 Feb;65(2):201–9. doi:10.1109/TUFFC.2017.2781715

52. Rossi S, Boni E. Embedded GPU Implementation for High-Performance Ultrasound Imaging. Electronics. 2021 Jan;10(8):8. doi:10.3390/electronics10080884

53. Chang LW, Hsu KH, Li PC. Graphics processing unit-based high-frame-rate color doppler ultrasound processing. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2009 Sep;56(9):1856–60. doi:10.1109/TUFFC.2009.1261

54. Jetson Orin [Internet]. 2023. Available from: https://www.nvidia.com/it-it/autonomous-machines/embedded-systems/jetson-orin/

55. Tortoli P, Michelassi V, Bambi G, Guidi F, Righi D. Interaction between secondary velocities, flow pulsation and vessel morphology in the common carotid artery. Ultrasound in Medicine & Biology. 2003 Mar 1;29(3):407–15. doi:10.1016/S0301-5629(02)00705-6

56. Balbis S, Roatta S, Guiot C. Curvature affects Doppler investigation of vessels: Implications for clinical practice. Ultrasound in Medicine & Biology. 2005 Jan;31(1):65–77. doi:10.1016/j.ultrasmedbio.2004.09.007

57. Manbachi A, Hoi Y, Wasserman BA, Lakatta EG, Steinman DA. On the shape of the common carotid artery with implications for blood velocity profiles. Physiol Meas. 2011 Dec 1;32(12):1885–97. doi:10.1088/0967-3334/32/12/001

58. Camacho J, Parrilla M, Fritsch C. Phase Coherence Imaging. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2009 May;56(5):958–74. doi:10.1109/TUFFC.2009.1128

59. Matrone G, Savoia AS, Caliano G, Magenes G. The Delay Multiply and Sum Beamforming Algorithm in Ultrasound B-Mode Medical Imaging. IEEE Transactions on Medical Imaging. 2015 Apr;34(4):940–9. doi:10.1109/TMI.2014.2371235

60. Mehdizadeh S, Austeng A, Johansen TF, Holm S. Eigenspace Based Minimum Variance Beamforming Applied to Ultrasound Imaging of Acoustically Hard Tissues. IEEE Transactions on Medical Imaging. 2012 Oct;31(10):1912–21. doi:10.1109/TMI.2012.2208469

61. Wang W, He Q, Zhang Z, Feng Z. Adaptive beamforming based on minimum variance (ABF-MV) using deep neural network for ultrafast ultrasound imaging. Ultrasonics. 2022 Dec 1;126:106823. doi:10.1016/j.ultras.2022.106823

62. Paridar R, Asl BM. Plane wave ultrasound imaging using compressive sensing and minimum variance beamforming. Ultrasonics. 2023 Jan 1;127:106838. doi:10.1016/j.ultras.2022.106838

63. Huang Y, Chen X, Badescu E, Kuenen M, Bonnefous O, Mischi M. Adaptive higher-order singular value decomposition clutter filter for ultrafast Doppler imaging of coronary flow under non-negligible tissue motion. Ultrasonics. 2024 May 1;140:107307. doi:10.1016/j.ultras.2024.107307

64. Demené C, Deffieux T, Pernot M, Osmanski BF, Biran V, Gennisson JL, et al. Spatiotemporal Clutter Filtering of Ultrafast Ultrasound Data Highly Increases Doppler and fUltrasound Sensitivity. IEEE Transactions on Medical Imaging. 2015 Nov;34(11):2271–85. doi:10.1109/TMI.2015.2428634

65. Pialot B, Augeul L, Petrusca L, Varray F. A simplified and accelerated implementation of SVD for filtering ultrafast power Doppler images. Ultrasonics. 2023 Sep 1;134:107099. doi:10.1016/j.ultras.2023.107099

66. Sandrin L, Tanter M, Catheline S, Fink M. Shear modulus imaging with 2-D transient elastography. IEEE Trans Ultrason Ferroelectr Freq Control. 2002 Apr;49(4):426–35. PubMed PMID: 11989698.

67. Tanter M, Bercoff J, Sandrin L, Fink M. Ultrafast compound imaging for 2-D motion vector estimation: application to transient elastography. IEEE Trans Ultrason Ferroelectr Freq Control. 2002 Oct;49(10):1363–74. PubMed PMID: 12403138.

68. Christensen-Jeffries K, Couture O, Dayton PA, Eldar YC, Hynynen K, Kiessling F, et al. Super-resolution Ultrasound Imaging. Ultrasound in Medicine & Biology. 2020 Apr 1;46(4):865–91. doi:10.1016/j.ultrasmedbio.2019.11.013

69. Christensen-Jeffries K, Browning RJ, Tang MX, Dunsby C, Eckersley RJ. In Vivo Acoustic Super-Resolution and Super-Resolved Velocity Mapping Using Microbubbles. IEEE Transactions on Medical Imaging. 2015 Feb;34(2):433–40. doi:10.1109/TMI.2014.2359650

70. Caenen A, Bézy S, Pernot M, Nightingale KR, Vos HJ, Voigt JU, et al. Ultrasound Shear Wave Elastography in Cardiology. JACC: Cardiovascular Imaging. 2024 Mar 1;17(3):314–29. doi:10.1016/j.jcmg.2023.12.007

71. Joos P, Porée J, Liebgott H, Vray D, Baudet M, Faurie J, et al. High-Frame-Rate Speckle-Tracking Echocardiography. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2018 May;65(5):720–8. doi:10.1109/TUFFC.2018.2809553

72. Orlowska M, Ramalli A, Bézy S, Meacci V, Voigt JU, D'Hooge J. In Vivo Comparison of Multiline Transmission and Diverging Wave Imaging for High-Frame-Rate Speckle-Tracking Echocardiography. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2021 May;68(5):1511–20. doi:10.1109/TUFFC.2020.3037043

73. Engelhard S, Voorneveld J, Vos HJ, Westenberg JJM, Gijsen FJH, Taimr P, et al. High-Frame-Rate Contrast-enhanced US Particle Image Velocimetry in the Abdominal Aorta: First Human Results. Radiology. 2018 Oct;289(1):119–25. doi:10.1148/radiol.2018172979

74. Dort S, Muth S, Swillens A, Segers P, Cloutier G, Garcia D. Vector flow mapping using plane wave ultrasound imaging. In: 2012 IEEE International Ultrasonics Symposium [Internet]. Dresden, Germany: IEEE; 2012 [cited 2024 Aug 16]. p. 330–3. Available from: http://ieeexplore.ieee.org/document/6562473/ doi:10.1109/ULTSYM.2012.0081

75. Madiena C, Faurie J, Porée J, Garcia D. Color and Vector Flow Imaging in Parallel Ultrasound With Sub-Nyquist Sampling. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2018 May;65(5):795–802. doi:10.1109/TUFFC.2018.2817885

76. Udesen J, Jensen JA. Investigation of transverse oscillation method. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2006 May;53(5):959–71. doi:10.1109/TUFFC.2006.1632686

77. Posada D, Poree J, Pellissier A, Chayer B, Tournoux F, Cloutier G, et al. Staggered Multiple-PRF Ultrafast Color Doppler. IEEE Trans Med Imaging. 2016 Jun;35(6):1510–21. doi:10.1109/TMI.2016.2518638

78. Doviak RJ, Zrnic DS, Sirmans DS. Doppler weather radar. Proceedings of the IEEE. 1979 Nov;67(11):1522–53. doi:10.1109/PROC.1979.11511

79. Nishiyama H, Katakura K. Non-equally-spaced pulse transmission for non-aliasing ultrasonic pulsed Doppler measurement. Journal of the Acoustical Society of Japan (E). 1992;13(4):215–22. doi:10.1250/ast.13.215

80. Muth S, Dort S, Sebag IA, Blais MJ, Garcia D. Unsupervised dealiasing and denoising of color-Doppler data. Medical Image Analysis. 2011 Aug 1;Special section on IPMI 200915(4):577–88. doi:10.1016/j.media.2011.03.003

81. Yatchenko AM, Krylov AS, Sandrikov VA, Kulagina TYu. Regularizing method for phase antialiasing in color doppler flow mapping. Neurocomputing. 2014 Sep 2;139:77–83. doi:10.1016/j.neucom.2013.09.060

82. Nahas H, Yu ACH. SUP-Net: Slow-time Upsampling Network for Aliasing Removal in Doppler Ultrasound. IEEE Transactions on Medical Imaging. 2025;1–1. doi:10.1109/TMI.2025.3591820

83. Tortoli P. A tracking FFT processor for pulsed Doppler analysis beyond the Nyquist limit (medical ultrasound). IEEE Transactions on Biomedical Engineering. 1989 Feb;36(2):232–7. doi:10.1109/10.16470

84. Chiglia DC, Pritt MD. Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software. New York: Wiley; 1998.

85. Eduardo G, Nogueira C, Ferreira A, Vidal JT. A nonuniform sampled coherent pulsed Doppler ultrasonic velocimeter with increased velocity range. IEEE Trans Ultrason Ferroelectr Freq Control. 1999;46(2):452–6. doi:10.1109/58.753034 PubMed PMID: 18238442.

86. Nitzpon HJ, Rajaonah JC, Burckhardt CB, Dousse B, Meister JJ. A new pulsed wave Doppler ultrasound system to measure blood velocities beyond the Nyquist limit. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 1995 Mar;42(2):265–79. doi:10.1109/58.365240

87. Lagonigro F, Tortoli P, Ramalli A, Yiu BYS, Yu ACH. Accuracy of flow velocity estimations over regions including great depths: an experimental study. In: 2025 IEEE International Ultrasonics Symposium (IUS). 2025. p. 1–4.

88. Lal C, Leahy MJ. An Updated Review of Methods and Advancements in Microvascular Blood Flow Imaging. Microcirculation. 2016;23(5):345–63. doi:10.1111/micc.12284

89. Aziz MU, Eisenbrey JR, Deganello A, Zahid M, Sharbidre K, Sidhu P, et al. Microvascular Flow Imaging: A State-of-the-Art Review of Clinical

Use and Promise. Radiology. 2022 Nov;305(2):250–64. doi:10.1148/radiol.213303

90. Mauldin FW, Lin D, Hossack JA. The Singular Value Filter: A General Filter Design Strategy for PCA-Based Signal Separation in Medical Ultrasound Imaging. IEEE Trans Med Imaging. 2011 Nov;30(11):1951–64. doi:10.1109/TMI.2011.2160075 PubMed PMID: 21693416; PubMed Central PMCID: PMC3351208.

91. Tierney J, Baker J, Brown D, Wilkes D, Byram B. Independent Component-Based Spatiotemporal Clutter Filtering for Slow Flow Ultrasound. IEEE Trans Med Imaging. 2020 May;39(5):1472–82. doi:10.1109/TMI.2019.2951465 PubMed PMID: 31689187; PubMed Central PMCID: PMC7288756.

92. Ledoux LAF, Brands PJ, Hoeks APG. Reduction of the Clutter Component in Doppler Ultrasound Signals Based on Singular Value Decomposition: A Simulation Study. Ultrason Imaging. 1997 Jan 1;19(1):1–18. doi:10.1177/016173469701900101

93. Deffieux T, Demené C, Tanter M. Functional Ultrasound Imaging: A New Imaging Modality for Neuroscience. Neuroscience. 2021 Oct 15;Brain imaging474:110–21. doi:10.1016/j.neuroscience.2021.03.005

94. Villemain O, Baranger J, Friedberg MK, Papadacci C, Dizeux A, Messas E, et al. Ultrafast Ultrasound Imaging in Pediatric and Adult Cardiology. JACC: Cardiovascular Imaging. 2020 Aug;13(8):1771–91. doi:10.1016/j.jcmg.2019.09.019

95. Maresca D, Correia M, Villemain O, Biz é A, Sambin L, Tanter M, et al. Noninvasive Imaging of the Coronary Vasculature Using Ultrafast Ultrasound. JACC: Cardiovascular Imaging. 2018 Jun;11(6):798–808. doi:10.1016/j.jcmg.2017.05.021

96. Baranger J, Demene C, Frerot A, Faure F, Delanoë C, Serroune H, et al. Bedside functional monitoring of the dynamic brain connectivity in human neonates. Nature communications. 2021;12(1):1080.

97. Nguyen MB, Zhang N, Baranger J, Aguet J, Friedberg MK, Barron DJ, et al. Ultrafast Power Doppler for Detecting Intraoperative Myocardial Perfusion in Infants With Critical Congenital Heart Disease. JACC: Advances. 2023 May;2(3):100290. doi:10.1016/j.jacadv.2023.100290

98. Lai KL, Tsai MC, Li PC. Ultrafast Doppler Imaging for Early Detection of Synovitis in Rheumatoid Arthritis. Ultrasound in Medicine and Biology. 2024 Apr 1;50(4):484–93. doi:10.1016/j.ultrasmedbio.2023.12.007 PubMed PMID: 38242743.

99. Errico C, Pierre J, Pezet S, Desailly Y, Lenkei Z, Couture O, et al. Ultrafast ultrasound localization microscopy for deep super-resolution vascular imaging. Nature. 2015 Nov;527(7579):499–502. doi:10.1038/nature16066

100. Brown J, Christensen-Jeffries K, Harput S, Zhang G, Zhu J, Dunsby C, et al. Investigation of Microbubble Detection Methods for Super-Resolution Imaging of Microvasculature. IEEE Trans Ultrason Ferroelectr Freq Control. 2019 Apr;66(4):676–91. doi:10.1109/TUFFC.2019.2894755 PubMed PMID: 30676955.

101. Yi HM, Lowerison MR, Song PF, Zhang W. A Review of Clinical Applications for Super-resolution Ultrasound Localization Microscopy. Curr Med Sci. 2022 Feb;42(1):1–16. doi:10.1007/s11596-021-2459-2 PubMed PMID: 35167000.

102. Song P, Rubin JM, Lowerison MR. Super-resolution ultrasound microvascular imaging: Is it ready for clinical use? Zeitschrift für Medizinische Physik. 2023 Aug 1;Special Issue: Recent Advances in Ultrasound Imaging33(3):309–23. doi:10.1016/j.zemedi.2023.04.001

103. Song P, Trzasko JD, Manduca A, Qiang B, Kadirvel R, Kallmes DF, et al. Accelerated Singular Value-Based Ultrasound Blood Flow Clutter Filtering With Randomized Singular Value Decomposition and Randomized Spatial Downsampling. IEEE Trans Ultrason Ferroelectr Freq Control. 2017 Apr;64(4):706–16. doi:10.1109/TUFFC.2017.2665342 PubMed PMID: 28186887.

104. Lok UW, Song P, Trzasko JD, Daigle R, Borisch EA, Huang C, et al. Real time SVD-based clutter filtering using randomized singular value decomposition and spatial downsampling for micro-vessel imaging on a Verasonics ultrasound system. Ultrasonics. 2020 Sep 1;107:106163. doi:10.1016/j.ultras.2020.106163

105. Dongarra J, Gates M, Haidar A, Kurzak J, Luszczek P, Tomov S, et al. The Singular Value Decomposition: Anatomy of Optimizing an Algorithm for Extreme Scale. SIAM Rev. 2018 Jan;60(4):808–65. doi:10.1137/17M1117732

106. Baranger J, Arnal B, Perren F, Baud O, Tanter M, Demené C. Adaptive spatiotemporal SVD clutter filtering for ultrafast Doppler imaging using similarity of spatial singular vectors. IEEE transactions on medical imaging. 2018;37(7):1574–86.

107. Boni E, Bassi L, Dallai A, Guidi F, Meacci V, Ramalli A, et al. ULA-OP 256: A 256-Channel Open Scanner for Development and Real-Time Implementation of New Ultrasound Methods. IEEE Transactions on

Ultrasonics, Ferroelectrics, and Frequency Control. 2016 Oct;63(10):1488–95. doi:10.1109/TUFFC.2016.2566920

108. Cline A, Dhillon I. Computation of the Singular Value Decomposition. In. 2013. p. 1027–39. doi:10.1201/b16113-68

109. cuSOLVER [Internet]. [cited 2024 May 20]. Available from: https://docs.nvidia.com/cuda/cusolver/

110. cuBLAS [Internet]. [cited 2024 May 20]. Available from: https://docs.nvidia.com/cuda/cublas/index.html

111. Baranger J, Aguet J, Villemain O. Fast thresholding of SVD clutter filter using the spatial similarity matrix and a sum-table algorithm. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control [Internet]. 2023 [cited 2024 May 2]. Available from: https://ieeexplore.ieee.org/abstract/document/10163397/

112. Heiles B, Chavignon A, Hingot V, Lopez P, Teston E, Couture O. Performance benchmarking of microbubble-localization algorithms for ultrasound localization microscopy. Nat Biomed Eng. 2022 May;6(5):605–16. doi:10.1038/s41551-021-00824-8

113. Hingot V, Errico C, Heiles B, Rahal L, Tanter M, Couture O. Microvascular flow dictates the compromise between spatial resolution and acquisition time in Ultrasound Localization Microscopy. Sci Rep. 2019 Feb 21;9(1):2456. doi:10.1038/s41598-018-38349-x

114. Song P, Manduca A, Trzasko JD, Chen S. Ultrasound Small Vessel Imaging With Block-Wise Adaptive Local Clutter Filtering. IEEE Trans Med Imaging. 2017 Jan;36(1):251–62. doi:10.1109/TMI.2016.2605819 PubMed PMID: 27608455.

115. Pialot B, Lachambre C, Mur AL, Augeul L, Petrusca L, Basarab A, et al. Adaptive noise reduction for power Doppler imaging using SVD filtering in the channel domain and coherence weighting of pixels. Phys Med Biol. 2023 Jan;68(2):025001. doi:10.1088/1361-6560/acac5d

116. Yan J, Wang B, Riemer K, Hansen-Shearer J, Lerendegui M, Toulemonde M, et al. Fast 3D Super-Resolution Ultrasound With Adaptive Weight-Based Beamforming. IEEE Trans Biomed Eng. 2023 Sep;70(9):2752–61. doi:10.1109/TBME.2023.3263369 PubMed PMID: 37015124; PubMed Central PMCID: PMC7614997.