AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

DYSCYPLINA INŻYNIERIA BIOMEDYCZNA

ROZPRAWA DOKTORSKA

Symulacja w czasie rzeczywistym wybranych zjawisk falowych w oparciu o technikę śledzenia promieni

Autor: mgr inż. Kamil Szostek

Promotor rozprawy: dr hab. inż. Adam Piórkowski, prof. AGH

Praca wykonana: Akademia Górniczo-Hutnicza im. St. Staszica w Krakowie,
Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej
Katedra Biocybernetyki i Inżynierii Biomedycznej

Kraków, 2023 r.

AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

FIELD OF SCIENCE: ENGINEERING AND TECHNICAL SCIENCES

SCIENTIFIC DISCIPLINE: BIOMEDICAL ENGINEERING

DOCTORAL THESIS

Selected phenomena simulation using ray trace technique

Author: mgr inż. Kamil Szostek

Supervisor: dr hab. inż. Adam Piórkowski, prof. AGH

Completed in: AGH University of Science and Technology,
Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering
Department of Biocybernetics and Biomedical Engineering

Krakow, 2023

W miejscu tym pragnę złożyć podziękowania
za wsparcie, cierpliwość i wyrozumiałość
moim Rodzicom, Żonie oraz Dzieciom

Spis treści

1. Wykaz publikacji stanowiących podstawę rozprawy doktorskiej

Niniejsza praca składa się z cyklu następujących publikacji:

[A1] Szostek K., Piórkowski A.: Real-time simulation of ultrasound refraction phenomena using ray-trace based wavefront construction method. *Computer Methods and Programs in Biomedicine*, 2016, vol. 135, pp. 187–197. DOI: 10.1016/j.cmpb.2016.07.034. **IF $^{(2016)}$ = 2.503**

[A2] Szostek K., Lasek J., Piórkowski A.: Real-Time Simulation of Wave Phenomena in Lung Ultrasound Imaging. *Applied Sciences*, 2023, 13, 9805. DOI: 10.3390/app13179805. **IF $^{(2022)}$ = 2.7**

[A3] Szostek K.: Real-Time Deformation of Three-Dimensional Volumetric Models Using Mesh Models. Progress on Pattern Classification, Image Processing and Communications, CORES and IP&C 2023. Springer 2023, LNNS, vol. 766. DOI: 10.1007/978-3-031-41630-9_6

[A4] Szostek K., Leśniak, A.: Parallelization of the seismic ray trace algorithm. Parallel Processing and Applied Mathematics: 9th International Conference, PPAM 2011. Springer 2012, LNCS, vol. 7204, pp. 411-418. DOI: 10.1007/978-3-642-31500-8_42

Artykuł pomocniczy:

[B1] Szostek K., Piórkowski A., Kempny A., Banyś R., Gackowski A.: Using computed tomography images for a heart modeling. *Journal of Medical Informatics & Technologies,* 2012, vol. 19, pp. 75-84

2. Streszczenie w języku polskim

Symulatory medyczne stanowią współcześnie nieodłączny element kształcenia przyszłych kadr medycznych. Wielokrotnie potwierdzono, że trening skomplikowanych procedur medycznych z wykorzystaniem realistycznych symulatorów podnosi kwalifikacje uczących się osób. W pracy tej przedstawione zostały badania nad implementacją symulatora ultrasonografii, które pozwolą w czasie rzeczywistym na wizualizację wybranych zjawisk falowych przy użyciu techniki śledzenia promieni. Praca składa się z cyklu powiązanych tematycznie publikacji. Przedstawione w nich zostały metody reprezentacji modeli anatomicznych, trójkątowych oraz wolumetrycznych, pochodzących z tomografii komputerowej. Zaprezentowano sposób analizy i przygotowania, a następnie transformacji danych wolumetrycznych na potrzeby symulacji fal ultradźwiękowych. Następnie omówione zostały zagadnienia symulacji propagacji fali. Wykorzystując rozwiązania symulacji propagacji fali w geofizyce, opracowano algorytmy bazujące na technice śledzenia promieni, które upraszczają problem propagacji fali w tkankach z uwzględnieniem zjawisk takich jak osłabianie, ugięcie, załamanie i rewerberacja, z zachowaniem wysokiej wydajności obliczeń. W wyniku prac wspieranych przez program Doktorat Wdrożeniowy III powstały dwa symulatory, które zostały wdrożone w postaci produktów firmy Medical Simulation Technologies. Pierwszy produkt to symulator punkcji transseptalnej, w którym wykonano symulację obrazu USG serca oraz dynamiczną modyfikację tkanek w modelu wolumetrycznym. Drugi produkt opracowany został z grantu w ramach projektów szybkiego reagowania europejskiego centrum EIT Health Rapid Response w czasie pandemii SARS-COV-2. Jest nim symulator USG płuc LUS (Lung Ultrasound Simulator), ogólnodostępny w sieci Internet.

3. Streszczenie w języku angielskim

Nowadays, medical simulators are an integral part of educating future medical staff. It has been repeatedly confirmed that training in complex medical procedures using realistic simulators improves the qualifications of learners. This work presents research on the implementation of an ultrasound simulator that will allow real-time visualization of selected wave phenomena using the ray tracing technique. The work consists of a series of thematically related publications. They present methods for representing anatomical models: triangular and volumetric derived from computed tomography. A method of analyzing, preparing and then transforming volumetric data for the simulation of ultrasonic waves was presented. Then, the issues of wave propagation simulation were discussed. Using a wave simulation solutions in geophysics, algorithms have been developed based on ray tracing techniques, which simplify the problem of wave propagation in tissues, taking into account phenomena such as attenuation, refraction, diffraction and reverberation while maintaining high computational efficiency. As a result of the work supported by the Implementation Doctorate III program, two simulators were created and implemented as products from Medical Simulation Technologies. The first product is a transseptal puncture simulator in which a cardiac ultrasound image was simulated and tissues were dynamically modified in a volumetric model. The second product was developed from a grant as part of the European EIT Health Rapid Response projects, during the SARS-COV-2 pandemic. It is the LUS (Lung Ultrasound Simulator) lung ultrasound simulator, publicly available on the online platform.

4. Autoreferat

Niniejszy autoreferat dotyczy pracy naukowej nad symulacją w czasie rzeczywistym obrazowania ultrasonograficznego z uwzględnieniem wybranych zjawisk falowych. Omówiono cel takiej symulacji, dane wejściowe, metody realizacji i przedstawiono otrzymane wyniki oraz przeprowadzone wdrożenia.

4.1 Cel i założenia

Symulacja medyczna staje się ważnym elementem szkolenia przyszłych lekarzy, ale także elementem doskonalenia zawodowego czy pomocą w przygotowywaniu zabiegów. Znamiennym jest fakt, że nie wymaga ona uczestnictwa pacjenta czy też ochotnika do celów treningowych lekarzy, a zatem można ją przeprowadzać bezpiecznie i wielokrotnie. Mimo, że nie zastąpi ona prawdziwego kontaktu z człowiekiem, pozwala jednak na opanowanie manualne specyfiki badania, ale także kształtuje percepcję obrazów i zachowań, co odnosi się do np. wyrabiania orientacji przestrzennej lekarza, czy też zwracania uwagi na konkretne aspekty danej jednostki chorobowej. Obecne trendy w kierunku medycyny spersonalizowanej pozwalają przygotować się lekarzom do zabiegów w nietypowych przypadkach chorobowych czy anatomicznych. Aby jednak szkolenie było wartościowe dla trenującego, aby przebiegało w sposób maksymalnie zbliżony do badania na żywo, symulacja musi się odbywać w czasie rzeczywistym. Realizacja tego zagadnienia w przypadku rozchodzenia się fal mechanicznych w ośrodku trójwymiarowym jest bardzo czasochłonna. Stąd realny problem naukowy, który jest celem niniejszej pracy, czyli zbadanie możliwości i opracowanie implementacji symulacji obrazowania ultrasonograficznego, realizującego wybrane zjawiska falowe, takie jak ugięcie, załamanie czy wielokrotne odbicia tak, aby zachować płynność wizualizacji.

Przedstawiane badania mają charakter ściśle wdrożeniowy. Są one częścią działalności firmy Medical Simulation Technologies, sp. z o.o. w Krakowie (MST). Firma ta specjalizuje się w konstrukcji i dystrybucji symulatorów badań echokardiograficznych przezprzełykowych oraz symulatorów interwencji przezskórnych w chorobach serca. Dodatkowo, w czasie pandemii SARS-COV-2, Firma opracowała wirtualny symulator diagnostycznej ultrasonografii płuc, związanej właśnie z pandemią. Badania wdrożeniowe Firmy były realizowane w ramach dwóch grantów:

- „Elektroniczny symulator zabiegu punkcji transseptalnej z możliwością rozbudowy o symulację innych procedur z dostępem przez naczyniowym”, w ramach Regionalnego Programu Operacyjnego Województwa Małopolskiego na lata 2014-2020, nr RPMP.01.02.01-12-0027/19, realizowanego w okresie 10.2019 – 10.2021 (w trakcie rozliczenia).
- „Ultrasonographic Lung Simulator” – EIT Health, 20880 – europejski grant badawczo-rozwojowy, realizowany w okresie 01.05.2020 - 31.12.2020 (rozliczony).

Rozwój możliwości symulacji, związanej także ze zjawiskami występującymi poza obszarem serca, został wsparty w programie „Doktorat Wdrożeniowy III”, rozpoczętym 1 października 2019 r.. W ramach tego programu określona została współpraca między Firmą MST, a Akademią Górniczo-Hutniczą im St. Staszica w Krakowie, a której celem jest realizacja pracy doktorskiej mgr inż. Kamila Szostka pod nadzorem promotora z ramienia Akademii Górniczo-Hutniczej, dr hab. inż. Adama Piórkowskiego i opiekuna z ramienia Firmy MST, dr inż. Tomasza Dziwińskiego.

Nad poprawnością ze strony medycznej czuwał dr hab. n. med. Andrzej Gackowski, wsparciem sprzętowym służył dr hab. inż. Paweł Piątek, natomiast w kwestiach finansowych wspierał mgr Kamil Kipiel.

4.2 Dane i modele

Zrealizowane w pracy symulatory medyczne oparte zostały na jednym z dwóch sposobów reprezentacji struktur anatomicznych. Pierwszym z nich jest reprezentacja przy pomocy danych wolumetrycznych, np. poprzez wykorzystanie danych tomografii komputerowej (CT). Drugim sposobem jest wykorzystanie modeli trójkątowych, które reprezentują granice między strukturami.

Pierwszy sposób, opisany szerzej w [B1] i wykorzystany w [A1], pozwala na lepsze odwzorowanie rzeczywistych struktur, gdyż wykorzystywane dane pochodzą bezpośrednio z obrazowań pacjentów. Daje to zdecydowaną przewagę nad drugim sposobem, w którym uproszczenia mogą powodować utratę istotnych szczegółów anatomicznych. Jednakże, wykorzystanie metody trójkątowej pozwala na większą kontrolę zarówno nad samym modelem anatomicznym, jak również nad symulacją zjawisk fizycznych. Metoda ta wykorzystana została w [A2].

W przypadku pracy [A3] zastosowano połączenie dwóch wyżej wymienionych modeli reprezentacji danych, aby wykorzystać korzyści wynikające z obu tych metod.

4.3 Metody symulacji

W literaturze opisanych zostało wiele metod symulacji zjawisk falowych, które można zastosować przy konstrukcji symulatora obrazowania ultradźwiękowego. Kompletną i dokładną symulację zjawisk falowych można przeprowadzić za pomocą równań falowych. Inne podejście polega na obliczeniu impulsu przestrzennego dla każdego rozpraszacza w odniesieniu do modelowanych elementów przetwornika. Powszechnie znane modele ultradźwięków opierają się na równaniach Westervelta lub KZK. Niestety metody te wymagają bardzo intensywnych obliczeń. Rozwiązania takie nie są w stanie wykonywać obliczeń w czasie rzeczywistym.

Uproszczeniem problemu jest technika śledzenia promieni (*raytracing*), która koncentruje się przede wszystkim na śledzeniu pojedynczych promieni, co znacząco skraca czas przetwarzania danych. Technika ta pozwala na uwzględnienie zjawisk takich jak tłumienie, odbicie, załamanie i dyfrakcja. Ponadto możliwa jest implementacja tej techniki z wykorzystaniem procesorów kart graficznych, których główną charakterystyką jest wpieranie obliczeń równoległych.

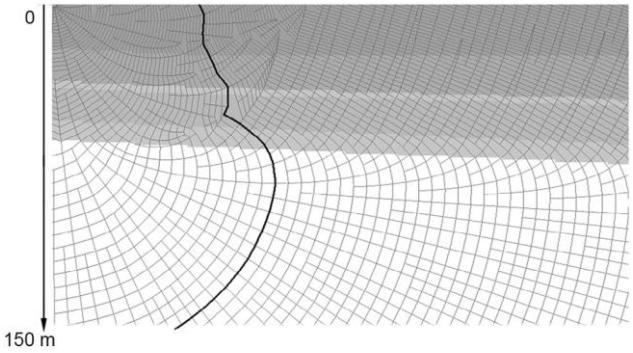
Symulację można również uprościć za pomocą techniki rzutowania promieni (*ray-casting*). Ta zakłada analizę promieni w linii prostej od nadajnika-odbiornika, co pozwala na symulację jedynie odbić i tłumienia. Uproszczenie to można stosować jedynie w szczególnych przypadkach, w których prędkość dźwięku i impedancja akustyczna tkanki są podobne, np. w sercu.

Na uwagę zasługuje również metoda, która najpierw przygotowuje rzut 2D modelu trójwymiarowego, a następnie przeprowadza symulację ultradźwiękową na tej projekcji. To podejście nie uwzględnia jednak w pełni zjawisk ugięcia i załamania fali.

4.4 Proponowane rozwiązania

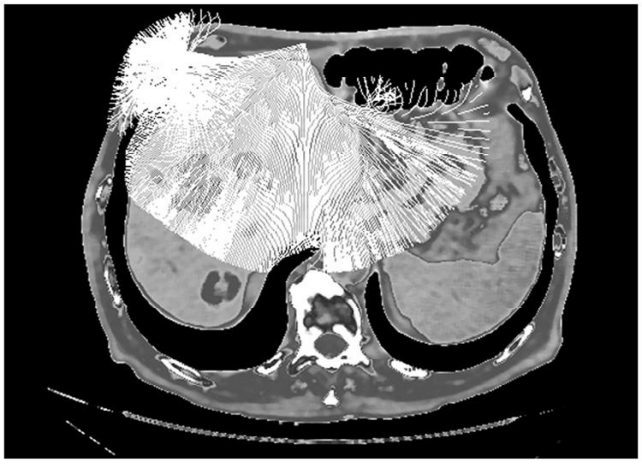
Rozwiązanie problemu symulacji fali mechanicznej, przedstawione w pracy [A4], jest oparte na konstrukcji frontów falowych. Metoda ta zakłada, że fala dźwiękowa rozchodzi się w medium w postaci frontów falowych, czyli powierzchni (albo krzywych w przypadku dwuwymiarowym) o równym czasie propagacji. Pierwszy front falowy tworzony jest w wybranym miejscu źródła fali jako sfera. Kolejne fronty falowe powstają iteracyjnie z wykorzystaniem techniki śledzenia promienia. W zagadnieniach geofizycznych, na podstawie trajektorii powstałych promieni możliwe jest

zarejestrowanie czasów propagacji fali sejsmicznej (Rysunek 1). Na podstawie otrzymanych czasów propagacji możliwe jest oszacowanie parametrów medium, przez które przechodziła fala mechaniczna.

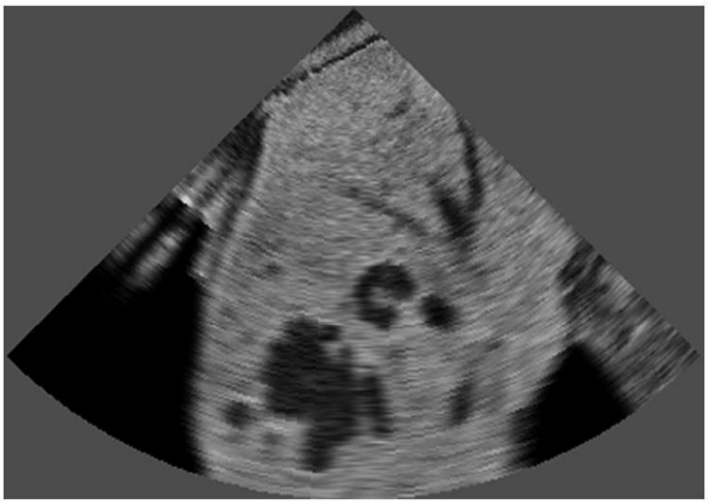


Rysunek 1. Przykład wyniku algorytmu konstrukcji frontów falowych w dwuwymiarowym modelu geologicznym. Zaznaczony został jeden z frontów. Usunięte zostały krzyżujące się promienie.

W pracy [A1] zaproponowane i opisane zostały modyfikacje powyższego algorytmu, które pozwoliły na wykorzystanie go do symulacji fali ultradźwiękowej w trójwymiarowym modelu wolumetrycznym, charakteryzującym struktury w ciele człowieka. Jako model danych wykorzystano pochodzące z tomografii komputerowej obrazy, które opisują model tkanek w jednostkach Hounsfielda. Zaprezentowano metodę konwersji tych danych na wartości prędkości dźwięku, co pozwoliło na symulację zjawisk załamania, ugięcia fali dźwiękowej oraz cieni akustycznych (Rysunek 2). Opisywany algorytm został zaimplementowany z wykorzystaniem metod zrównoleglania operacji na wielu procesorach, co pozwoliło na przyspieszenie obliczeń i uzyskiwanie wyników w czasie rzeczywistym.



Rysunek 2. Przykładowa wizualizacja propagacji promieni ultradźwiękowych na wybranym przekroju.



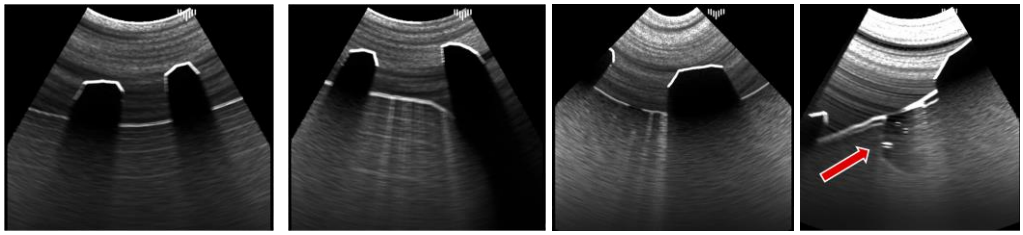
Rysunek 3. Przykładowy obraz symulacji USG w oparciu o algorytm frontów falowych.

Otrzymane wyniki symulacji na danych syntetycznych oraz rzeczywistych (Rysunek 3) zostały porównane z wynikiem symulacji przeprowadzonej przy pomocy rozwiązania „Field II" oraz z rzeczywistym obrazem ultrasonograficznym. Dokonano analizy opracowanego rozwiązania pod względem parametryzacji i wydajności, która dowiodła możliwości użycia go w symulacji w czasie rzeczywistym, czyli z prędkością przynajmniej 10 klatek na sekundę.

W kolejnej pracy [A2] zaprezentowano opracowany symulator USG płuc LUS (Lung Ultrasound Simulator), który znajduje się na ogólnodostępnym portalu internetowym (https://lus.mstech.eu). Wybór takiej formy symulatora wymusił zastosowanie konkretnych rozwiązań implementacyjnych, co stanowiło dodatkową trudność zarówno w kwestii konstrukcji algorytmów, jak również modelu danych wykorzystywanych w symulacji. W związku z tym symulator powstał w oparciu o modele trójkątowe, natomiast algorytmy symulacji zaimplementowano przy użyciu programów cieniujących (*shaders*) technologii WebGL. Programy te wykorzystują procesory kart graficznych (GPU), zrównoleglając obliczenia w celu osiągnięcia lepszej wydajności. Czas generowania pojedynczej klatki symulacji pozwalał na uzyskanie wysokiego odświeżania o wydajności około 270 klatek na sekundę.

W przypadku symulatora USG płuc, kluczowym było przygotowanie go w sposób, który ułatwi zapoznanie się z podstawowymi aspektami badania, tj. zaprezentuje w klarowny sposób artefakty obrazowania, stanowiące podstawę interpretacji obrazu USG płuc.

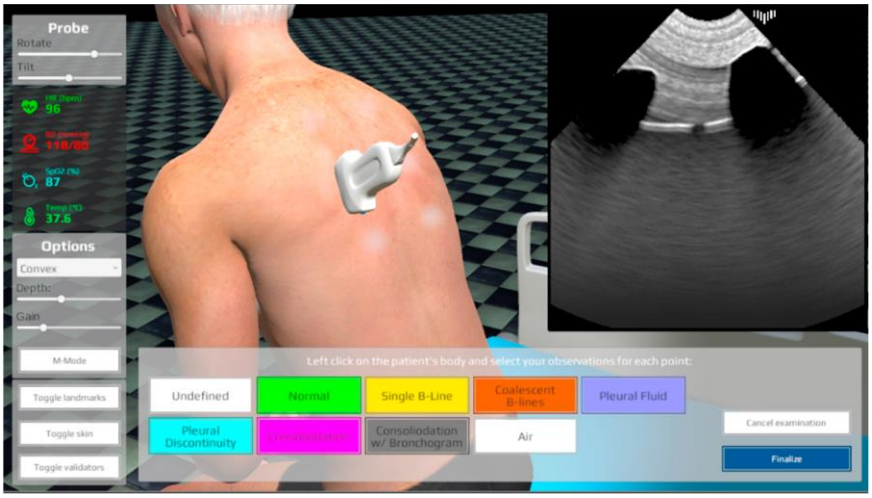
Przygotowane zostały modele trójwymiarowe pacjentów, wraz z animacją ruchu płuc. Opracowano algorytmy symulacyjne, których rezultatem był obraz USG badanych płuc z generowanymi najważniejszymi pod kontem diagnostyki artefaktami: cieniami akustycznymi, liniami A i B, przesuwaniem się oraz nieciągłością opłucnej, i wreszcie konsolidacjami (Rysunek 4).



Rysunek 4. Przykładowe artefakty zaimplementowane w symulatorze LUS.

Oprócz samej symulacji USG płuc, aplikacja pozawala użytkownikowi na wybranie do wirtualnego badania jednego z 5 pacjentów, zapoznanie się z krótkim opisem każdego z nich, a następnie wykonania badania. W trakcie tego badania użytkownik zaznacza zaobserwowane artefakty w ustandaryzowanych punktach przyłożenia głowicy USG, a następnie dokonuje wstępnej oceny jednostki chorobowej pacjenta. Całość możliwa jest do powtórzenia w celu sprawdzenia poprawności obserwacji (Rysunek 5).

Pomimo znaczącego uproszczenia niektórych elementów badania i symulacji, w tym konieczności operowania myszą w celu wykonania badania, opracowana aplikacja może być traktowana jako wstęp do nauki obrazowania USG płuc.



Rysunek 5. "LUS Simulator", przykładowy obraz w trakcie wirtualnego badania pacjenta.

4.5 Realizacje i wdrożenia

Doktorant brał udział w dwóch projektach badawczo-rozwojowych, które następnie zostały wdrożone w formie opracowanych produktów.

Pierwszy grant pt. „Elektroniczny symulator zabiegu punkcji transseptalnej z możliwością rozbudowy o symulację innych procedur z dostępem przez naczyniowym", w którym doktorant realizował badania w ramach dwóch zadań:

- Opracowanie technologii umożliwiającej dynamiczną modyfikację wybranych struktur trójwymiarowych serca (m.in. tenting). Rezultaty badań zostały częściowo przedstawione w pracy [A3]. Opisano w niej metodę dynamicznej modyfikacji modeli wolumetrycznych, w szczególności przegrody międzyprzedsionkowej, przy użyciu przygotowanej segmentacji trójkątowej oraz dostępnych rozwiązań symulacyjnych.
- Opracowanie metody obrazowania narządzi oraz przetwarzania danych CT na widok fluoroskopii. W ramach tego zadania doktorant opracował metodę symulacji obrazu fluoroskopii na podstawie danych z tomografii komputerowej, wraz trójwymiarowymi modelami narzędzi chirurgicznych.

Rezultatem prac w grancie jest opracowany symulator punkcji transseptalnej, z wielomodalnym obrazowaniem (USG oraz fluoroskopia). Stworzony symulator jest kolejnym produktem w ramach oferty firmy Medical Simulation Technolgies.

W drugim grancie pt. „Ultrasonographic Lung Simulator”, doktorant zrealizował badania nad symulacją zjawisk falowych w przypadku badania USG płuc. Następnie na tej podstawie zaimplementował i wdrożył symulator USG płuc z wykorzystaniem technologii WebGL oraz modeli trójkątowych. W ramach grantu powstała strona internetowa pod adresem https://lus.mstech.eu, na której zamieszczono stworzony symulator. Powstały symulator został opisany w [A2].

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Real-time simulation of ultrasound refraction phenomena using ray-trace based wavefront construction method

Kamil Szostek *, *Adam Piórkowski*

AGH University of Science and Technology, Department of Geoinformatics and Applied Computer Science, Cracow, Poland

ARTICLE INFO

Article history:
Received 20 November 2015
Received in revised form
6 July 2016
Accepted 29 July 2016

Keywords:
Biomedical image processing
Biomedical informatics
Echocardiography
Medical simulation
Ray tracing

ABSTRACT

Ultrasound (US) imaging is one of the most popular techniques used in clinical diagnosis, mainly due to lack of adverse effects on patients and the simplicity of US equipment. However, the characteristics of the medium cause US imaging to imprecisely reconstruct examined tissues. The artifacts are the results of wave phenomena, i.e. diffraction or refraction, and should be recognized during examination to avoid misinterpretation of an US image. Currently, US training is based on teaching materials and simulators and ultrasound simulation has become an active research area in medical computer science. Many US simulators are limited by the complexity of the wave phenomena, leading to intensive sophisticated computation that makes it difficult for systems to operate in real time. To achieve the required frame rate, the vast majority of simulators reduce the problem of wave diffraction and refraction.

The following paper proposes a solution for an ultrasound simulator based on methods known in geophysics. To improve simulation quality, a wavefront construction method was adapted which takes into account the refraction phenomena. This technique uses ray tracing and velocity averaging to construct wavefronts in the simulation. Instead of a geological medium, real CT scans are applied. This approach can produce more realistic projections of pathological findings and is also capable of providing real-time simulation.

1. Introduction

Simulation in medicine is a new educational method which accelerates the learning process for physicians in training and allows them to familiarize themselves with rare diseases. Computer based simulators are a popular research area and they are proven to have significant importance in enhancing competency at early training stages [1–10]. The main problem is that simulators need to produce realistic images in convenient time and should be designed in a way that enables users to practice. A simulation can be helpful for learning both diagnostic and surgical techniques. The aim of ultrasound (US) training is to master spatial orientation and the ability to manipulate a device while performing medical examinations or even surgery. Ultrasound examination, particularly when per-

* *Corresponding author.* AGH University of Science and Technology, Department of Geoinformatics and Applied Computer Science, Cracow, Poland.
E-mail address: kamil.szostek@agh.edu.pl (K. Szostek).

http://dx.doi.org/10.1016/j.cmpb.2016.07.034
0169-2607/

formed endoscopically, is a common issue that would benefit from simulation, especially in gastroenterology, urology or pulmonology. Moreover, the simulated images can be used for testing, validating and improving algorithms for real imaging registration [11]. There are some ultrasound simulators, but the problem is not fully covered by scientific research and not all the necessary solutions are implemented.

Ultrasound simulation is complicated due to the nature of the examination to be simulated. US images may be significantly distorted due to the fact that the lengths of waves used by this examination technique are relatively closer to the dimensions of the examined object. The impact of reflection, attenuation, refraction and diffraction must be mentioned as these fundamental wave phenomena have a significant impact on imaging, generating artifacts (mirroring, velocity errors, comet tail, ring-down, etc.) which can both facilitate and confuse diagnosis [12].

There are several ultrasound simulators. They differ in terms of model structure, input data, simulation method, and final effects. It is reasonable to present them in the context of ultrasound simulation methods.

A complete and accurate simulation of wave phenomena can be performed using a wave equation [13]. Another approach is to compute spatial impulse for every scatterer with respect to modeled transducer elements [14,15]. Widely known models for ultrasound are based on the Westervelt or KZK equations [16–18]. Unfortunately, these methods require highly intensive computation. Solutions such as those in references [13,14,19,20] are not able to perform the calculations in real time. A simplification of this problem is the technique of ray tracing, which focuses primarily on the tracking of individual rays. This procedure reduces the time required for processing simulation data. This technique allows such phenomena as attenuation, reflection, refraction and diffraction to be considered.

Simulation is best simplified with ray casting. This assumes a ray analysis in a straight line from the transmitter–receiver, which only allows the simulation of reflections and attenuation [21,22]. Simplifications can be applied only in special cases in which the speed of sound and the acoustic impedance in tissues are similar, for example, in the heart [23].

Classification of simulation methods was proposed in the work of Zhu and Salcudean [24]. Worthy of attention is the model which first prepares a 2D projection of the model and then performs an ultrasound simulation for the projection. This approach also does not take diffraction and refraction into account, but it is used in some real solutions. Shams et al. [25] created a simulation view using a combined reflection and scattering simulation for 2D projection. Kutter et al. [26] proposes a ray-based method that is based on transmission, reflection and scattering functions. Reichl et al. [27] take into account a combination of transmission, absorption, reflection and noise functions, but do not include diffraction and refraction. Also, Dillenseger et al. [28] presented a solution based on Bamber and Dickinson's model [29], but did not consider the aforementioned phenomena. Advanced simulation, based on convolution of the point spread function of the ultrasound probe with scatterer distribution, is presented in reference [30]. The authors implement such artifacts as shadows, attenuation, speckle and specular reflections, but not diffraction and refraction [31]. In reference [32] only reflection, transmission, attenuation, noise and blur are considered.

The presented solutions require high computational power to deliver real time results. Therefore, the authors performed their simulations using graphics processing units (GPU) [26,27,30,33]. Real-time performance of volumetric data processing and visualization on small mobile devices can be obtained with remote volume rendering approach [34].

The latest and most interesting research is the work presented by Burger et al. [35]. The authors propose a method that uses 3D polygonal meshes to render simulated images directly. The simulation engine uses the OptiX library [36]. This method allows the refraction phenomenon to be taken into account. The disadvantage of this method is the necessity of triangular mesh creation for input data. Described in this paper, our studies on the adaptation of wavefront construction algorithms (known in the field of geology) for ultrasound simulation, in particular modification, parallelization and implementation, provide a new solution that allows ultrasound to be simulated in real time, whilst still taking into account general wave phenomena such as refraction.

2. Method of ultrasound simulation

2.1. *Wavefront construction method*

The wavefront construction algorithm presented by Urdaneta in reference [37] and parallelized in reference [38] was developed to solve the problem of seismic wave propagation in geological mediums, in particular the travel times of the seismic waves from source to receivers. This algorithm assumes that a seismic P-wave propagates from the source in all directions as a wavefront, which is considered as a surface (or a closed curve in 2D) with equal propagation time. The initial wavefront is created at the selected shot point position as a sphere (or a circle in 2D) with a defined number of initial rays perpendicular to the wavefront. The following wavefront is constructed from the previous one using raytracing. These rays represent the shortest paths (which means, according to Fermat's principle, the path with the least traversal time) of the wave propagation in the medium. Every ray contains information about its starting point and direction and can collect data while passing through the medium. The directions of these rays change, depending on the medium velocity that they are passing through, which implies that the wavefront shape changes as well. The wavefront motion is then given by:

$$\vec{x}_{v+1} = \vec{x}_v + v(\vec{x}_v, T)\Delta t\hat{s} \tag{1}$$

where $\vec{x}_{v+1}$ is a new ray position, Δt defines time step and $v(\vec{x}_v, T)$ is velocity at point $\vec{x}$ and period T $\hat{s}$ is a unit vector that represents direction of propagation (Fig. 1).

The selected algorithm combines two approaches to solving the wave propagation problem. First is Lomax's waverays approximation method, described in reference [39]. This approximates the wave speed in a medium using a Gaussian-weighting curve whose shape depends on the frequency of the wave (Fig. 1). This approach increases the stability of the raytracing and makes it possible to handle inhomogeneities of

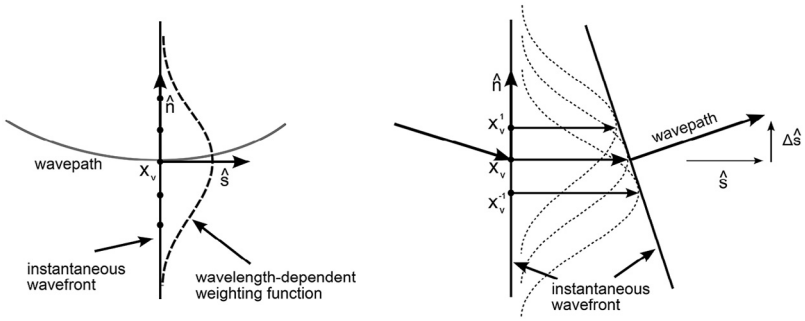


Fig. 1 – Lomax's velocity averaging at the point $\vec{x}_v$ using Gaussian curve (left) from reference [37] adapted from reference [39]. The ray from the point $\vec{x}_v$ is disturbed by the nearby rays and new wavefront is constructed (right). $\Delta\hat{s}$ is the change in direction of the waveray, calculated from the neighboring rays.

different sizes, hence simulating refractions and diffractions. Averaging in a two dimensional medium is given by:

$$\bar{v}(\vec{x}_v, T) = \frac{\int_{-\infty}^{\infty} \omega(\gamma) c(\vec{x}(\gamma, T)) d\gamma}{\int_{-\infty}^{\infty} \omega(\gamma) dy} \quad (2)$$

which averages velocity $\bar{v}$ at point $\vec{x}_v$ and period T, where γ is the arc length along the wavefront away from $\vec{x}_v$, in wavelengths. $c(\vec{x})$ is the value of speed at that point. The Gaussian weighting curve is chosen to average velocity:

$$\omega(\gamma) = e^{-4\ln 2(\gamma/\alpha)^2} \quad (3)$$

where α describes the half-width of the Gaussian bell in wavelengths. $\vec{x}(\gamma, T)$ is given by recursive relation and represents positions along the instantaneous straight wavefront:

$$\vec{x}(\gamma, T) = \vec{x}_v + \frac{T}{2\pi} \int_0^{\gamma} c(\vec{x}(\gamma', T)) \mathbf{n}(T) d\gamma' \quad (4)$$

where $\hat{\mathbf{n}}$ is the unit normal to the ray at point $\vec{x}_v$ [39]. For the discreet version of the above equations, please refer to reference [37].

The second technique implemented in Urdaneta's algorithm is the ray interpolation from Vinje et al. [40]. The idea of this method is as follows: whenever the distance between two neighboring wavefronts' rays is greater than an arbitrarily selected threshold value (maximum distance), new rays are added in-between by interpolating existing rays. This increases accuracy and guarantees constant density of the rays. What is more, Vinje et al. described a gridding process which recalculates values from the wavefronts into a grid of theoretical geophones or seismometers. However, for US simulation this process is redundant.

An example result from this algorithm for a geological medium is presented in Fig. 2. It shows the outcome of the algorithm for a selected profile of the Legnica–Glogow Copper Area (the vertical axis represents depth). The shot-point, which is the source of the wavefronts, was positioned at coordinate (0, 0), which is in the upper left corner of the model. The shades of gray vary according to velocity, from 800 m/s at the top to 6000 m/s at the bottom of the model. Strong refraction causes the rays to bend, as is clearly visible in the figure.

This algorithm was chosen for US simulation because of its parameterization features and computation speed. In reference [41] it was compared with two different methods: the Shortest Path Method and full-wave field modeling. The wavefront construction method can provide results similar to these techniques, but in a more convenient time. Also, this algorithm is based on velocity raster data, which makes it suitable for a CT-based US simulators. Finally, the algorithm takes into account the refraction, which can significantly increase the realism of the US simulation.

2.2. *The refraction problem*

When considering a wave (i.e. light or sound wave) passing the border of two mediums with different velocities, its direction changes according to Snell's law. This phenomenon is widely discussed in seismology, because it may lead to misinterpretation of the data [42].

Refractions in ultrasound may produce artifacts or change the observed objects size and position [43]. Fig. 3 presents a schematic ray refraction on a cyst, which leads to incorrect location of the nodule. A real example of the refraction problem in US is shown in Fig. 4 (left), where discontinuity of the diaphragm artifact is clearly visible [44]. As presented in Fig. 4 (right), rays from transducers meet the diaphragm at points P_1 and P_2, because the speed of sound in the liver is slightly

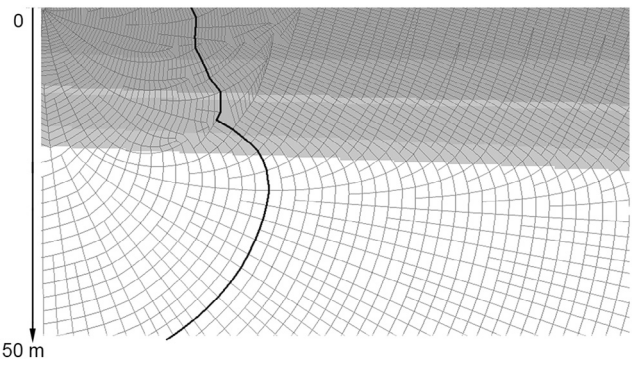


Fig. 2 – An example of wavefront construction method result for geological medium. All crossing rays are removed. One of the wavefronts has been marked.

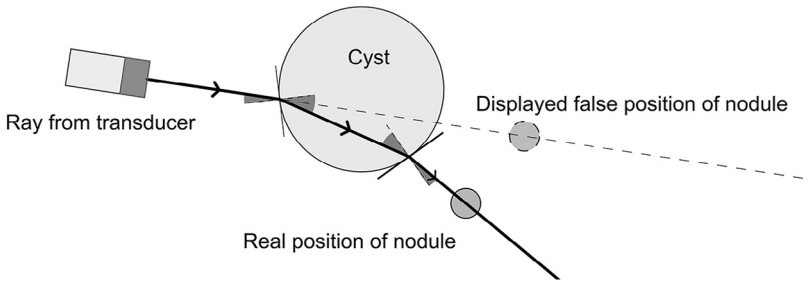


Fig. 3 – The refraction artifact example. The image of the nodule is displayed at a wrong position.

higher than in the surrounding ascites. These points are misplaced at P'_1 and P'_2 in the US image, therefore the diaphragm is wrongly displayed (marked with dashed line). This problem has also been discussed previously, i.e. in reference [45], in which the authors consider the US refraction problem for better image reconstruction and use the Fast Marching Method [46] and the High Accuracy Fast Marching Method from reference [47] to solve the Eikonal equation. The algorithm adopted in this paper simulates refraction. Also, other phenomena could be implemented to increase the quality of the images.

2.3. *Input data*

The basis for the US simulation is a 3D model of a patient body. Most solutions use a 3D CT scan [25,26,30,35,48]. This provides information on the shape of tissues, but the values describe the X-ray absorption and must be properly converted to be used in ultrasound simulation. In reference [49] a 4D US scan is used. The proposed method in reference [35] requires the construction of a triangular mesh based on CT volumes. Also, in reference [50] a mesh model is used to simulate deformations.

In this paper, synthetic data and CT images are used to simulate US images. This solution can provide results based on a real patients anatomy, including different pathological findings and movement of organs (if 4D data is applied).

The original CT data contains Hounsfield values (rescaled to 4096 shades of gray); therefore, it has to be preliminarily modified in two separate preprocessing stages. First, CT images are converted into US reflections, as is explained in reference [23]. The second transformation changes the same CT brightness values into a velocity model, which is required for the simulation algorithm. Fig. 5 shows the transfer function from CT brightness to the speed of sound, based on Table 1, which can be considered as the initial transfer function. Due to variations of tissues radiodensity of different patients, this conversion has to be adjusted manually for each data set. For example, the muscles and bones represent a wide range of CT brightness, as well as a high speed of sound changes. A solution for obtaining tissue parameters such as acoustic impedance or attenuation from CT images is presented in reference [31].

These transformations can be part of data preprocessing or can be performed during the simulation. The second solution is more flexible and less resource intensive, but is more time consuming, so the first option was chosen.

CT scans are 12-bit grayscale images, but 16 bit aligned, which means the maximum brightness of a CT pixel is $2^{12} - 1 = 4095$. This results in better image dynamics, therefore the transformation can be done more precisely, as shown in reference [53]. The Hounsfield Units $HU(x, y)$ for a given pixel $p(x, y)$ can be determined for DICOM files by formula (5):

$$HU(x, y) = (RescaleSlope * p(x, y)) + RescaleIntercept \quad (5)$$

where:

RescaleSlope—is of 1 for the all collected CT sets (for Siemens CT Definition),

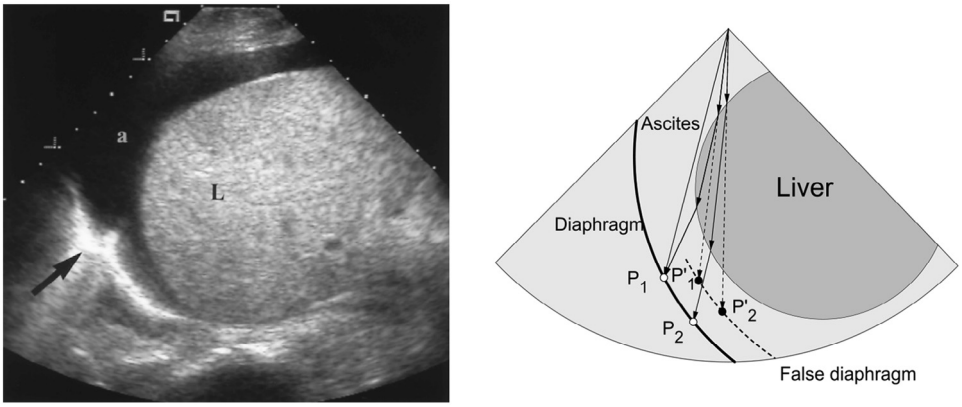


Fig. 4 – The refraction artifacts of the liver (L) image caused by the border with the ascites (a) [43] with corresponding diagram on the right.

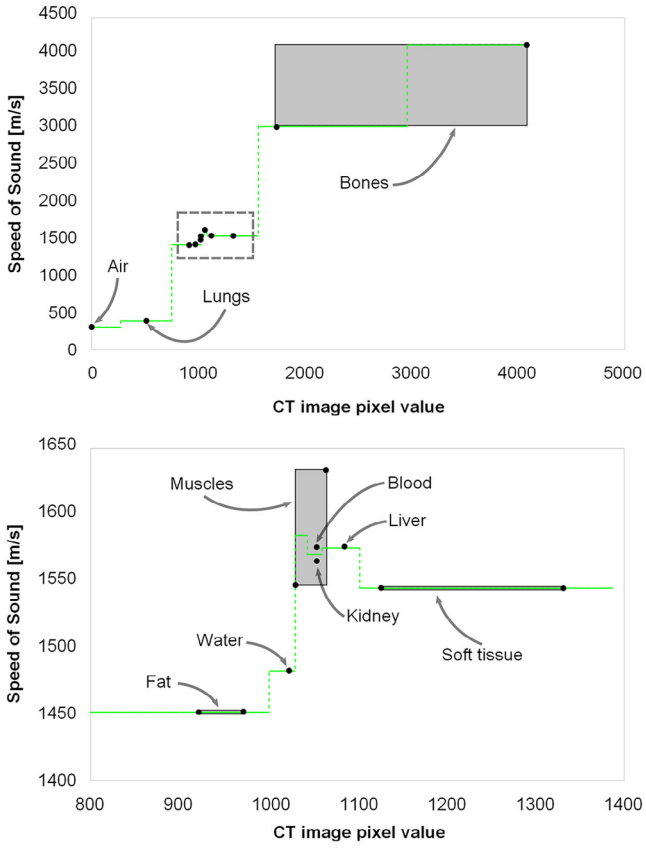


Fig. 5 – The graph shows the relation between the speed of sound for the selected tissues and CT scan pixel brightness. Green lines represent the speed of sound assigned to pixels in CT image during preprocessing. The detailed part marked with a dashed rectangle is shown below.

- *RescaleIntercept*—is of –1024 for the all collected CT sets (for Siemens CT Definition).

3. Algorithm for US simulator

Fig. 6 shows a diagram of the algorithm. At the beginning of simulation, a two-dimensional projection is extracted from the preprocessed CT dataset. This contains the reflections and velocities after application of the transfer function. Subsequently, the projection is used as a model for Urdaneta's algorithm, which, besides changes related to medium scale, is modified as explained in the following sections.

Table 1 – Hounsfield values and the speed of sound for different tissues [51,52].

Tissue/ medium	Hounsfield Units [HU]	Speed of sound [m/s]
Air	–1000	331
Lungs	–500	400
Fat	–100 to –50	1450
Water	0	1480
Kidney	30	1560
Blood	30 to 45	1570
Muscles	10 to 40	1542 to 1626
Liver	40 to 60	1570
Soft tissue	100 to 300	1540
Bones	700 to 3000	3000 to 4080

3.1. *Wavefront initialization*

The algorithm is initialized with the closed circular curve of the first wavefront (Fig. 6). The US simulation targets a 90° field of view, not 360°. Therefore, some of the rays are not used in the rendering. These extra rays can be assigned to a second group of rays and then attributed with a different interpolation threshold and flagged to be omitted in later steps. Thus, the first group simulates the transducer rays and its results are used in further processing. However, the second group is required for ray interpolations and velocity averaging. These two groups are simulated simultaneously (Fig. 7). Also, the different interpolation threshold of the second group increases algorithm speed.

3.2. *Simulation stop conditions*

As explained in reference [40], one wavefront may cross itself. This might happen when a wavefront meets inhomogeneity of velocity. In seismic simulation, *firstbreakes* (time of the first wavefront reaching a receiver) are usually sufficient for further calculations, so if a wavefront intersection appears, it is reduced by eliminating the later rays. Also, rays are removed when they reach the model boundaries. Therefore, in reference [37] wavefront propagation stops when the number of rays is below 5.

On the other hand, full paths are required in US simulation; therefore, removal occurs only for rays that leave the models boundaries. Unfortunately, when a wavefront becomes more complicated, the calculation time increases, so the process has to be limited either by a step count (which can correspond to US penetration depth) or wavefront length (calculated as the number of rays constructed over it, or simply list length). The first approach is chosen.

3.3. *Wavefront propagation*

The algorithm's diagram reveals the loop in which the wave iteratively propagates in the medium. Each iteration corresponds to the time step of the simulation. The propagation process which creates a new wavefront using ray tracing was described with equations earlier. As a result, the wavefront expands and changes its circular shape in response to the medium velocities. The new wavefront is then the subject of the following steps.

3.4. *Data acquisition*

The gridding process, as mentioned before, is redundant is this simulation. However, collecting data from the ray paths is still required. Therefore, during the ray tracing process, the reflection values from the model are gathered by the rays of group 1 and stored as array *G* for further reconstructions. The height of this image is given by the number of steps (US penetration depth). The width is calculated to ensure a sufficient amount

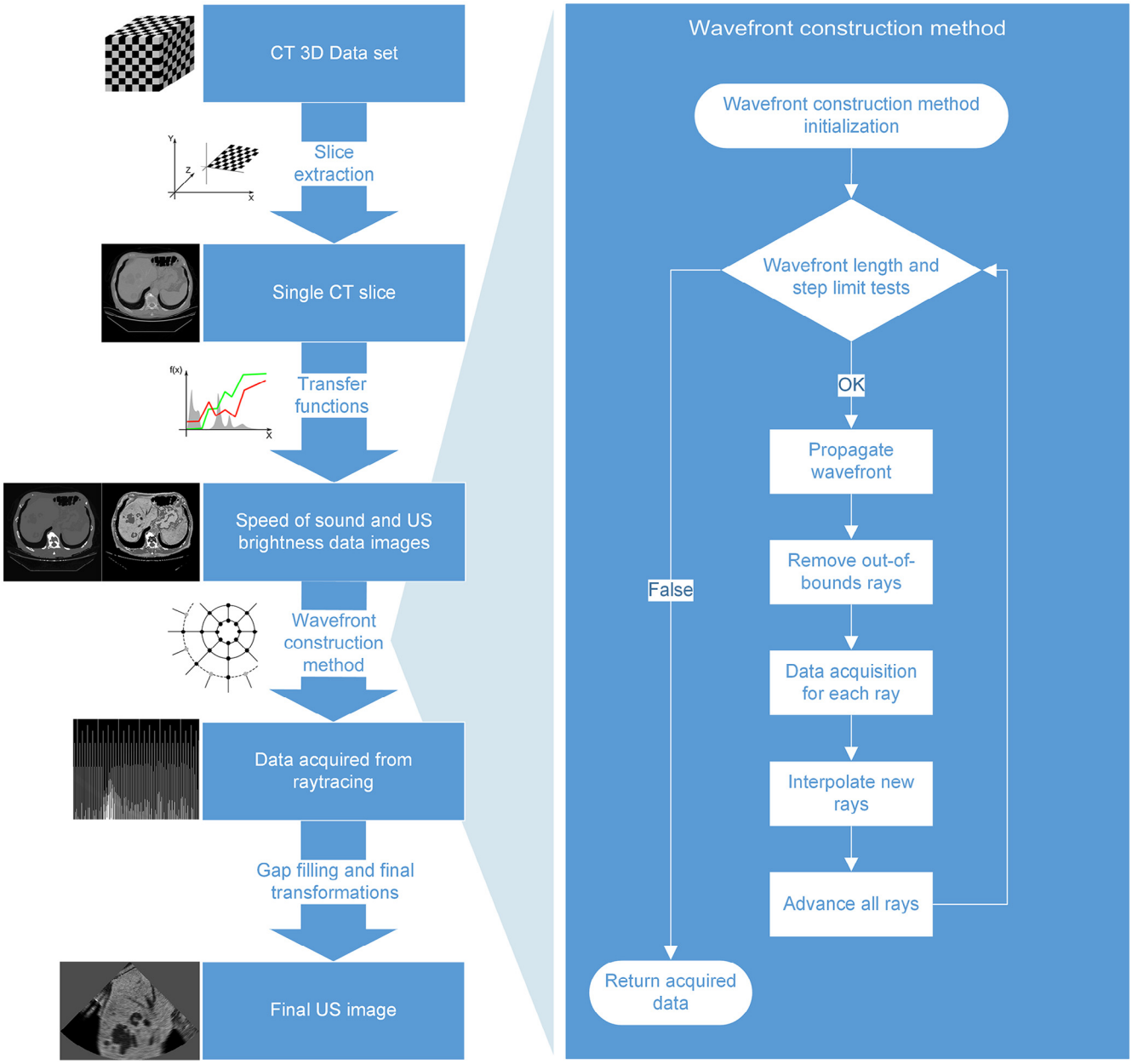


Fig. 6 – Diagram of the algorithm, explained in text. The images next to each steps illustrates intermediate results.

of rays at the last wavefront for the final transformation. For example, for the 90° field of view and 256 steps, the last wavefront should consist of 402 rays. If oversampling occurs (more rays than required), the collected values are averaged to fit the resulting image resolution.

Fig. 8 shows synthetic model and an example of images G for this model is presented in Fig. 9b. Each pixel $G(x, y)$ corresponds to the acquired data, where x (column) is a ray number and y (row) is a wavefront number or time step. Rays produce vertical traces. As new rays are interpolated at certain time steps, gaps might appear in the image where these rays have not been created yet. These gaps need to be interpolated. For every time step, (each row of image G) the linear interpolation is applied (Fig. 9c) and it is performed during this acquisition step.

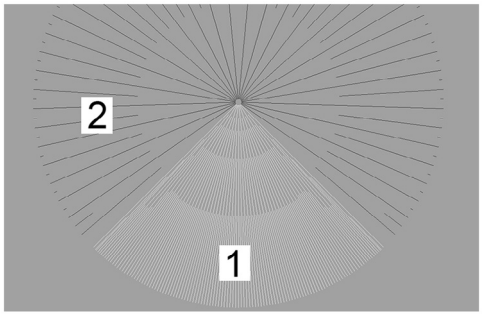


Fig. 7 – Two groups of rays used in US simulation. Lower interpolation threshold for group 2 decreases ray density and therefore increase algorithm speed.

Fig. 8 – Image of the synthetic model used in first simulation test.

Also, to simulate the ultrasound shadow artifact, the collected values are modified as explained in reference [23].

3.5. *Ray interpolation and ray advance*

These parts of the algorithm were described previously. Extensive explanations are presented in reference [40] as well as in reference [37]. In brief, new rays are interpolated (with the threshold depending on the group) over the wavefront for undersampled areas (when the distance between the rays is higher than selected threshold). All endpoints of rays construct a new wavefront which becomes the new source for the next step (Fig. 1).

3.6. *Final transformation*

The image *G* has to be transformed to the final ultrasound simulated image *U* – *a* 90° field of view of a sector type transducer (Fig. 9d). During this transformation, circular blurring (by averaging values in rows) and radial noise are applied.

4. Implementation highlights

The algorithm was implemented using the C/C++ language and compiled using the Microsoft Visual C++ compiler version 18.00.30501. To display the resulting projection, the OpenGL is used with help of the Freeglut library (version 2.6.0.1). The initial wavefront is constructed as a circular doubly linked list, which differs from the original approach where a simple array was used. This approach simplifies processes of adding and removing ray on the wavefront. Every element of the list contains the attributes necessary for ray simulation, such as ray vector and status flag. What is more, every ray holds its number corresponding to the column number of the image *G*. Initially, these numbers are integers, uniformly distributed in the range determined by the width of the image *G*. When the wavefront propagates and new rays are created, these numbers are interpolated. Then, the data acquisition process

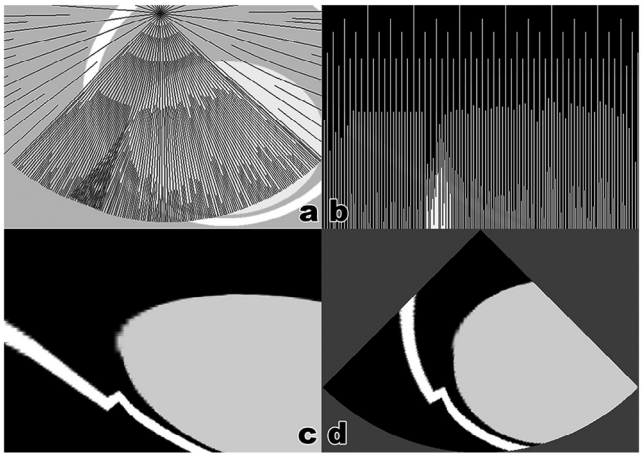


Fig. 9 – The algorithm main steps: wavefront with ray-tracing modeling (a), image *G* (b), interpolated gaps (c), final transformation (d). For clarity, neither blur nor noise is included.

averages values of the rays with decimal numbers into the column of image *G*.

5. Results

5.1. *Synthetic model*

For the first test a simple 2D synthetic model was used, which was 512 × 512 pixels in size. It is filled with constant velocity that corresponds to ascites fluids and two objects: a circular object representing the liver (1570 m/s from Table 1), and a second object representing the diaphragm (1584 m/s). The ascites fluids which fill the synthetic model have relatively lower speed than the liver. In the simulation, its speed was set to 1500 m/s. Fig. 8 shows this synthetic model and Fig. 9 presents the four main steps of the algorithm. Fig. 10 magnifies the image to illustrate how ray directions change while traveling through the tissues.

5.2. *Real model from CT data*

Selected input data from the CT scan of a human body is shown in Fig. 11 on the left panel. A single projection containing a

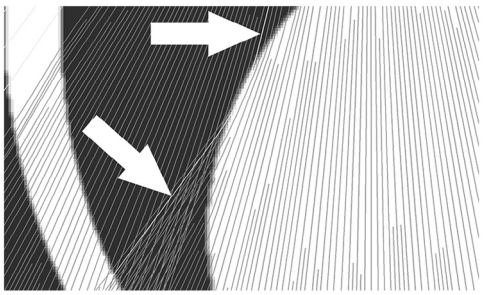


Fig. 10 – The close up of the ray passing the border of the synthetic liver. Ray bending is a result of difference in the speed of sound in tissues.

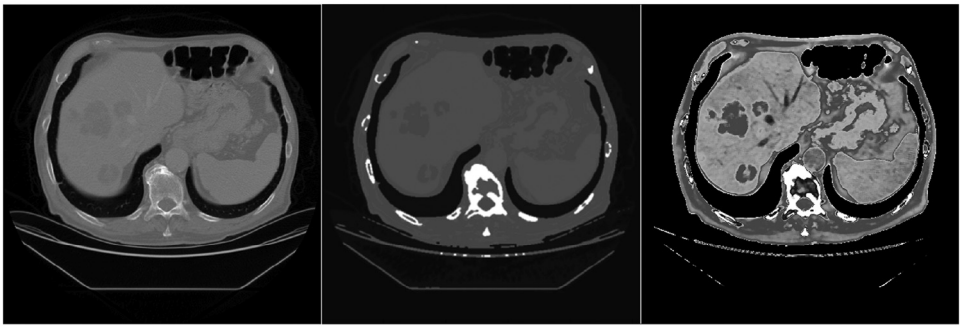


Fig. 11 – The CT scan selected for test (left), velocity model (middle) and reflection image (right) computed using transfer functions.

cross-section of a liver with cysts was extracted from 3D CT data. In spite of the scan size (512 × 512 pixels), the useful part of the data is considerably smaller: dark pixels (with values below 100, which correspond to surrounding air or lungs) cover more than 45% of the image. Nevertheless, it might still be used in the simulation. As mentioned previously, the proper transformation is required, and Fig. 11 (middle and right) shows the results of transfer function application in the preprocessing stage. Fig. 12 illustrates the second group of rays during the simulation, and the final image is shown in Fig. 13. To increase its quality in the context of US simulation, a circular blur and simple noise were introduced as previously mentioned.

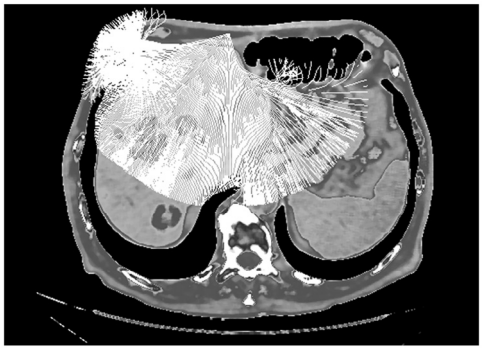


Fig. 12 – The second group of rays generated during simulation.

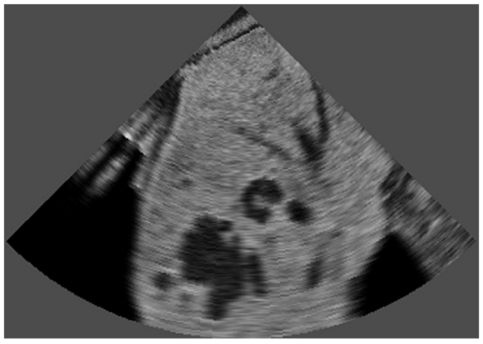


Fig. 13 – The result of the simulation with noise and radial blur.

What is more, Fig. 14 compares the results of the simulation with and without the refraction phenomenon. The performance increases without the refraction and the projection is then prepared about 7.6 times faster. Finally, one US projection was created using Field II software to compare the results. The generated image is presented in Fig. 15.

6. Program performance

To provide a real-time solution, the algorithm has to provide the requested projection in a satisfactory time, which should be greater than 10 frames per second. The presented algorithm requires about 70 ms to render one frame on Windows

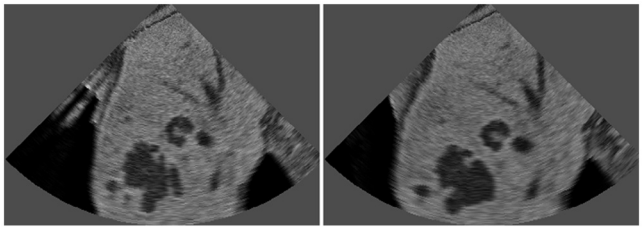


Fig. 14 – Comparison of the projection made with (left) and without refraction phenomenon (right).

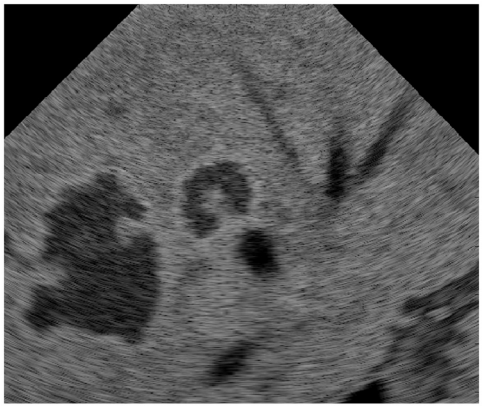


Fig. 15 – Field II results for the selected CT image.

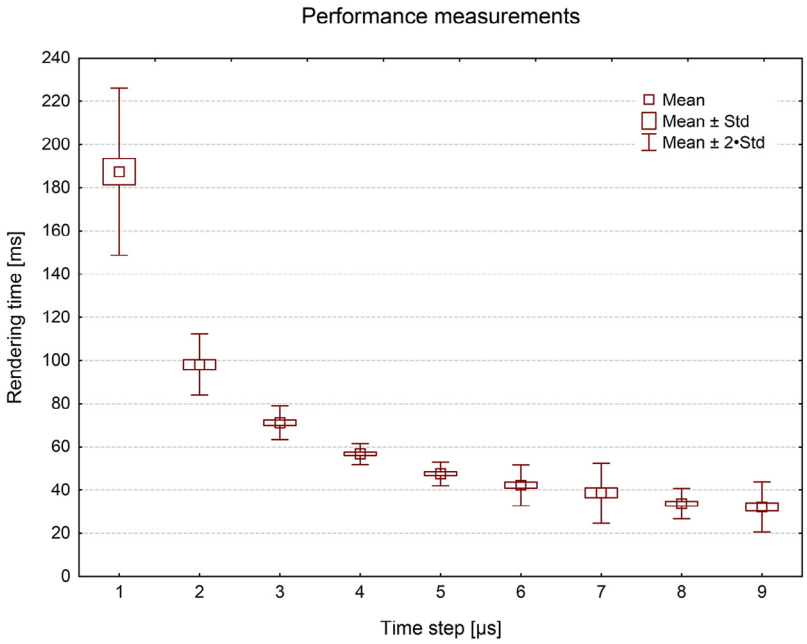


Fig. 16 – Performance measurements for 10 different projection and different time step values.

8 with an Intel(R) Core(TM) i7 3.5 GHz CPU. The average time is usually less than 100 ms, but the algorithm speed directly depends on its parameters and indirectly on the velocity model. The number of rays and the length of the wavefront, as well as the depth of propagation, can be controlled via interpolation threshold value, time step and stop condition (step limit particularly). Fig. 13 was achieved using 0.5 ms time step and 256 steps that reflect the projection radius. Interpolation threshold value was set to 2 pixels, which is about 1.56 mm. The final image was constructed in 72 ms as a 90° field of view sector projection. Increasing the interpolation threshold value increases the quality of the final image, as well as the computation time. Similar effects can be achieved with time step parameter adjustments.

To present the performance of the program, 10 different projections (different CT slice and starting point) were chosen and tested using different value of the time step parameter, which has the most significant influence on the program speed. To keep the same size of the projection, a change of the time step parameter must entail a change in the propagation depth. Fig. 16 shows mean times of the selected 10 projections rendering, for different values of time step parameter. The lower value of the time step should create larger projection with more details, including small inhomogeneities influences, while higher values creates smaller and less detailed images—in exchange for the performance of the program. Differences of the resulting projection are presented in Fig. 17. The qualitative measurements of differences will be investigated in the future work.

One of the most time consuming parts of the algorithm is wavefront propagation, which includes speed averaging using a Gaussian curve. This function might be replaced in the future with a faster alternative. The linear interpolation proposed for gap filling can also be replaced with a more accurate or faster method, i.e. the fast nearest neighbors method.

7. Discussion and conclusion

In spite of the 2D approach and simplicity of the synthetic model, the algorithm reveals the refraction artifacts on the resulting image. As expected, the diaphragm discontinuity artifact is clearly visible in Fig. 9d.

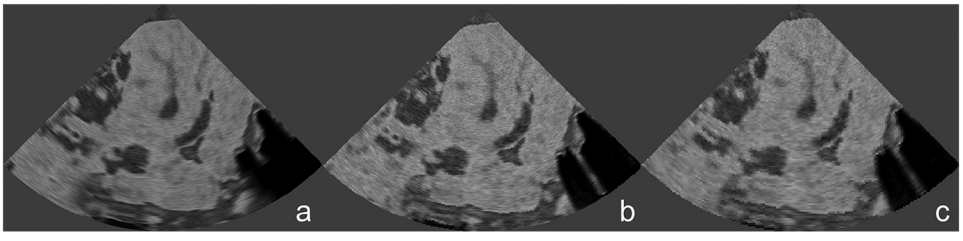


Fig. 17 – Three projections of different values of time steps parameter: 1 μs, 5 μs and 9 μs for image (a), (b) and (c), respectively.

More significant results are presented in Fig. 13, where real CT data was used for simulation. The projection depicts the cyst size change, which is a consequence of the velocity differences between liver and cyst. Also, an interesting comparison was presented in Fig. 15, where the influence of the refraction can be observed next to the undistorted projection. The larger cyst visible on both images has different shapes, which is the result of the discussed phenomenon. What is more, the area of the presented liver slice is also different.

The result of the algorithm was compared with the popular Field II software, however it is difficult to compare both algorithms, because Field II uses a different approach and focuses on different aspect of the simulation. This algorithm uses the concept of spatial impulse responses. The input data it uses is an MRI image, truncated into 8 bit gray image, which is transformed into the scatterer map of the tissue. What is more, the performance of the Field II is unsatisfactory in context of real time simulator—the result presented in Fig. 14 was created in about 1.5 hour. Despite its high quality results, it cannot be used as the simulator software. Finally, the algorithm does not simulate refraction—the sound speed is defined constant for the model.

The presented algorithm can be considered as a part of the even more complex US simulation process. It focuses mainly on the refraction phenomenon and successfully provides the required results, but also, by implementing shadowing, noise and circular blurring, it shows the potential for further algorithm development to simulate attenuation, reverberation, and other artifacts. Furthermore, in future the algorithm and final image may benefit from amplitude calculations, which are not discussed in this work but are explained in reference [37].

The algorithm uses 2D projections of 3D data sets, which might be considered as a compromise between speed and accuracy. 3D simulation, which will be undertaken in the future, will definitely be more precise, but also more time consuming.

The obtained times show that the modified wavefront construction method, adopted from geophysics, can successfully render images of a CT-based US simulation in real-time. Additionally, the presented algorithm is planned to be optimized and parallelized in future, using multiple CPU cores and graphic processing units (GPU). However, parallelization of the ray tracing part of the algorithm will significantly differ from usual approach. The speed averaging technique requires information from neighboring rays and therefore a sophisticated synchronization or completely different approach would be necessary.

Acknowledgments

This paper was supported by the AGH—University of Scienceand Technology, Faculty of Geology, Geophysics and Environmental Protection as a part of the statutory project. We thank Dr Robert Banyś, from Department of Diagnosis, Prevention and Telemedicine, John Paul II Hospital, Cracow, Poland, for his assistance in data acquisition.

REFERENCES

[1] R.R. Bose, R. Matyal, H.J. Warraich, J. Summers, B. Subramaniam, J. Mitchell, et al., Utility of a transesophageal echocardiographic simulator as a teaching tool, J. Cardiothorac. Vasc. Anesth. 25 (2) (2011) 212–215.
[2] A.U. Niazi, N. Haldipur, A.G. Prasad, V.W. Chan, Ultrasound-guided regional anesthesia performance in the early learning period: effect of simulation training, Reg. Anesth. Pain Med. 37 (1) (2012) 51–54.
[3] O. Shakil, F. Mahmood, R. Matyal, Simulation in echocardiography: an ever-expanding frontier, J. Cardiothorac. Vasc. Anesth. 26 (3) (2012) 476–485.
[4] J. Damp, R. Anthony, M.A. Davidson, L. Mendes, Effects of transesophageal echocardiography simulator training on learning and performance in cardiovascular medicine fellows, J. Am. Soc. Echocardiogr. 26 (12) (2013) 1450–1456.
[5] F. Clau-Terré, V. Sharma, B. Cholley, T. Gonzalez-Alujas, M. Galiñanes, A. Evangelista, et al., Can simulation help to answer the demand for echocardiography education?, Anesthesiology 120 (1) (2014) 32–41.
[6] T.M. Maus, Simulation: the importance of hands-on learning, J. Cardiothorac. Vasc. Anesth. 25 (2) (2011) 209–211.
[7] T. Blum, A. Rieger, N. Navab, H. Friess, M. Martignoni, A review of computer-based simulators for ultrasound training, Simul. Healthc. 8 (2) (2013) 98–108.
[8] J. McKinney, R.J. Scalese, R. Hatala, Simulation in non-invasive cardiology, in: The Comprehensive Textbook of Healthcare Simulation, Springer, 2013, pp. 289–298.
[9] R.E. Sedlack, J.C. Kolars, Computer simulator training enhances the competency of gastroenterology fellows at colonoscopy: results of a pilot study, Am. J. Gastroenterol. 99 (1) (2004) 33–37.
[10] R. Aggarwal, S.A. Black, J.R. Hance, A. Darzi, N.J.W. Cheshire, Virtual reality simulation training can improve inexperienced surgeons' endovascular skills, Eur. J. Vasc. Endovasc. Surg. 31 (6) (2005) 588–593.
[11] H. Rivaz, D.L. Collins, Simulation of Ultrasound Images for Validation of MR to Ultrasound Registration in Neurosurgery, Springer International Publishing, 2014, pp. 23–32.
[12] M.K. Feldman, S. Katyal, M.S. Blackwood, Us artifacts, Radiographics 29 (4) (2009) 1179–1189.
[13] J.A. Jensen, P.D. Fox, J.E. Wilhjelm, L.K. Taylor, Simulation of non-linear ultrasound fields, in: Ultrasonics Symposium, 2002. Proceedings, vol. 2, IEEE, 2002, pp. 1733–1736.
[14] A. Kowal, A. Piórkowski, T. Danek, A. Pieta, Analysis of selected component technologies efficiency for parallel and distributed seismic wave field modeling, in: Innovations and Advances in Computer Sciences and Engineering, Springer, 2010, pp. 359–362.
[15] H. Gao, H.F. Choi, P. Claus, S. Boonen, S. Jaecques, G.H. van Lenthe, et al., A fast convolution-based methodology to simulate 2-dd/3-d cardiac ultrasound images, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 56 (2) (2009) 404–409.
[16] P.J. Westervelt, Parametric acoustic array, J. Acoust. Soc. Am. 35 (4) (1963) 535–537 http://dx.doi.org/10.1121/1.1918525.
[17] E. Zabolotskaya, R. Khokhlov, Quasi-plane waves in the nonlinear acoustics of confined beams, Sov. Phys. Acoust. 15 (1) (1969) 35–40.
[18] V. Kuznetsov, Equations of nonlinear acoustics, Sov. Phys. Acoust. 16 (4) (1971) 467–470.
[19] Y. Li, J.A. Zagzebski, Computer model for harmonic ultrasound imaging, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 47 (4) (2000) 1000–1013.
[20] G.F. Pinton, J. Dahl, S. Rosenzweig, G.E. Trahey, A heterogeneous nonlinear attenuating full-wave model of

ultrasound, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 56 (3) (2009) 474–488.
[21] A. Piorkowski, Parallel processing in web-based interactive echocardiography simulators, Comput. Inf. 33 (3) (2014) 707–720.
[22] M. Borzęcki, A. Skurski, M. Kamiński, A. Napieralski, J. Kasprzak, P. Lipiec, Applications of ray-casting in medical imaging, in: Information Technologies in Biomedicine, vol. 3, Springer, 2014, pp. 3–14.
[23] A. Piórkowski, A. Kempny, The transesophageal echocardiography simulator based on computed tomography images, IEEE Trans. Biomed. Eng. 60 (2) (2013) 292–299.
[24] M. Zhu, S.E. Salcudean, Real-time image-based b-mode ultrasound image simulation of needles using tensor-product interpolation, IEEE Trans. Med. Imaging 30 (7) (2011) 1391–1400.
[25] R. Shams, R. Hartley, N. Navab, Real-time simulation of medical ultrasound from ct images, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008, Springer, 2008, pp. 734–741.
[26] O. Kutter, R. Shams, N. Navab, Visualization and gpu-accelerated simulation of medical ultrasound from ct images, Comput. Methods Programs Biomed. 94 (3) (2009) 250–266.
[27] T. Reichl, J. Passenger, O. Acosta, O. Salvado, Ultrasound goes gpu: real-time simulation using cuda, in: SPIE Medical Imaging, International Society for Optics and Photonics, 2009, p. 726116.
[28] J.-L. Dillenseger, S. Laguitton, É. Delabrousse, Fast simulation of ultrasound images from a ct volume, Comput. Biol. Med. 39 (2) (2009) 180–186.
[29] J. Bamber, R. Dickinson, Ultrasonic b-scanning: a computer simulation, Phys. Med. Biol. 25 (3) (1980) 463.
[30] S.U. Gjerald, R. Brekken, T. Hergum, J. D'hooge, Real-time ultrasound simulation using the gpu, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 59 (5) (2012) 885–892.
[31] S.U. Gjerald, R. Brekken, L.E. Bø, T. Hergum, T.A. Nagelhus Hernes, Interactive development of a ct-based tissue model for ultrasound simulation, Comput. Biol. Med. 42 (5) (2012) 607–613.
[32] Y. Law, S. Ullrich, T. Knott, T. Kuhlen, S. Weg, Ultrasound image simulation with gpu-based ray tracing, in: Virtuelle und Erweiterte Realität, 2011, pp. 183–194.
[33] M.R. Ogiela, T. Hachaj, Brain and neck visualization techniques, in: Natural User Interfaces in Medical Image Analysis, Springer, 2015, pp. 153–204.
[34] T. Hachaj, Real time exploration and management of large medical volumetric datasets on small mobile devicesevaluation of remote volume rendering approach, Int. J. Inf. Manag. 34 (3) (2014) 336–343.
[35] B. Burger, S. Bettinghausen, M. Radle, J. Hesser, Real-time gpu-based ultrasound simulation using deformable mesh models, IEEE Trans. Med. Imaging 32 (3) (2013) 609–618.
[36] S.G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, et al., Optix: a general purpose ray tracing engine, in: ACM Transactions on Graphics (TOG), vol. 29, ACM, 2010, p. 66.
[37] H. Urdaneta, Wavefront construction using waverays, Stanford Explor. Proj. Rep. 80 (2001) 85–100.
[38] K. Szostek, A. Leśniak, Parallelization of the seismic ray trace algorithm, in: Parallel Processing and Applied Mathematics, vol. 7204, LNCS, Springer, 2012, pp. 411–418.
[39] A. Lomax, The wavelength-smoothing method for approximating broad-band wave propagation through complicated velocity structures, Geophys. J. Int. 117 (2) (1994) 313–334.
[40] V. Vinje, E. Iversen, H. Gjøystdal, Traveltime and amplitude estimation using wavefront construction, Geophysics 58 (8) (1993) 1157–1166.
[41] M. Dwornik, A. Pieta, K. Szostek, Comparision of selected seismic forward problem method, in: The Computer–Aided Scientific Research, No. 217 in The Works of Wrocaw Scientific Society, 2011, pp. 283–290.
[42] H. Tewari, The effect of thin high velocity layers on seismic refraction data: an example from Mahanadi Basin, India, Pure Appl. Geophys. 151 (1) (1998) 63–79.
[43] W.E. Brant, The Core Curriculum, Ultrasound, Lippincott Williams & Wilkins, 2001.
[44] W. Middleton, G. Melson, Diaphragmatic discontinuity associated with perihepatic ascites: a sonographic refractive artifact, AJR Am. J. Roentgenol. 151 (4) (1988) 709–711.
[45] S. Li, K. Mueller, M. Jackowski, D.P. Dione, L.H. Staib, Fast marching method to correct for refraction in ultrasound computed tomography, in: Biomedical Imaging: Nano to Macro, 2006, 3rd IEEE International Symposium on, IEEE, 2006, pp. 896–899.
[46] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Natl. Acad. Sci. U.S.A. 93 (4) (1996) 1591–1595.
[47] J.A. Bærentzen, On the implementation of fast marching methods for 3d lattices, Tech. rep., 2001.
[48] A. Kempny, A. Piórkowski, CT2TEE—a novel, internet-based simulator of transoesophageal echocardiography in congenital heart disease, Kardiol. Pol. 68 (3) (2010) 374–379.
[49] M. Weidenbach, F. Wild, K. Scheer, G. Muth, S. Kreutter, G. Grunst, et al., Computer-based training in two-dimensional echocardiography using an echocardiography simulator, J. Am. Soc. Echocardiogr. 18 (4) (2005) 362–366.
[50] O. Goksel, S.E. Salcudean, B-mode ultrasound image simulation in deformable 3-d medium, IEEE Trans. Med. Imaging 28 (11) (2009) 1657–1669.
[51] P. Tsaklis, Presentation of acoustic waves propagation and their effects through human body tissues, Hum. Mov. 11 (1) (2010) 58–65.
[52] N. Joseph Jr., T. Rose, Quality assurance and the helical (spiral) scanner.
[53] M. Ciecholewski, Automatic liver segmentation from 2d ct images using an approximate contour model, J. Signal Process. Syst. 74 (2) (2014) 151–174.

applied sciences

MDPI

Article

Real-Time Simulation of Wave Phenomena in Lung Ultrasound Imaging

Kamil Szostek [1,2], Julia Lasek [3] and Adam Piórkowski [1,*]

[1] Department of Biocybernetics and Biomedical Engineering, AGH University of Science and Technology, 30-059 Krakow, Poland; szostek@agh.edu.pl
[2] Medical Simulation Technologies, 30-055 Krakow, Poland
[3] Faculty of Geology, Geophysics and Environmental Protection, AGH University of Science and Technology, 30-059 Krakow, Poland
* Correspondence: pioro@agh.edu.pl

Abstract: Medical simulations have proven to be highly valuable in the education of healthcare professionals. This significance was particularly evident during the COVID-19 pandemic, where simulators provided a safe and effective means of training healthcare practitioners in the principles of lung ultrasonography without exposing them to the risk of infection. This further emphasizes another important advantage of medical simulation in the field of medical education. This paper presents the principles of ultrasound simulation in the context of inflammatory lung conditions. The propagation of sound waves in this environment is discussed, with a specific focus on key diagnostic artifacts in lung imaging. The simulated medium was modeled by assigning appropriate acoustic characteristics to the tissue components present in the simulated study. A simulation engine was developed, taking into consideration the requirements of easy accessibility through a web browser and high-performance simulation through GPU-based computing. The obtained images were compared with real-world examples. An analysis of simulation parameter selection was conducted to achieve real-time simulations while maintaining excellent visual quality. The research findings demonstrate the feasibility of real-time, high-quality visualization in ultrasound simulation, providing valuable insights for the development of educational tools and diagnostic training in the field of medical imaging.

Keywords: ultrasound simulation; lung ultrasound imaging; wave phenomena; reverberations

Citation: Szostek, K.; Lasek, J.; Piórkowski, A. Real-Time Simulation of Wave Phenomena in Lung Ultrasound Imaging. *Appl. Sci.* **2023**, *13*, 9805. https://doi.org/10.3390/app13179805

Academic Editor: Athanasios Nikolaidis

Received: 7 July 2023
Revised: 23 August 2023
Accepted: 29 August 2023
Published: 30 August 2023

1. Introduction

Lung ultrasonography (LUS), also known as lung ultrasound or chest ultrasound, is a non-invasive imaging technique that allows real-time assessment of pulmonary structures. Research evidence indicates the importance of this technique, which may be a significant tool for a wide range of lung diseases, including both diagnosis and treatment considerations. There is a steady increase in the prevalence of LUS, and it has been shown to be effective in conditions including pneumonia [1], pleural effusion, lung consolidation [2], pneumothorax [3] and much more common etiologies of respiratory failure. Lung ultrasonography is recommended as a "point of care" examination in patients with dyspnea, chest pain, and any symptoms in the chest [4]. Moreover, LUS played a significant role during the COVID-19 pandemic [5,6]. In [7], authors present machine learning-based diagnostic tools for COVID-19 detection that are among other factors based on LUS examination. Despite the fact that ultrasonographic methods offer many inestimable advantages, such as portability (bedside usage), no ionizing radiation exposure, efficiency, and low-cost [8], its main limitation—operator-dependency, should also be mentioned. The appropriate interpretation of lung ultrasound images requires adequate training, experience and expertise to ensure the accuracy and reliability of the results [9].

Lung ultrasound has been shown to be an easy-to-learn technique by personnel without previous experience in a relatively short period of time [10–12]. Chiem et al. found that after a brief 30-min training, clinicians achieved a 95% success rate in obtaining adequate images and demonstrated comparable B-line interpretation sensitivity and specificity to expert sonographers [11]. In order to achieve effective training, several approaches are used. There are traditional ways of education based on lectures and seminars [13], in addition, e-learning courses with online lectures [14–16] and hands-on sessions involving healthy live models [16,17] or phantoms [17,18]. Each method has certain advantages; for example, learning from online lectures is distinguished by high accessibility, flexibility, and affordability, while hands-on learning on phantoms simulates the process of a real examination without exposing the patient to risks [19]. A promising solution that combines a number of advantages is a web-based LUS simulator, which visualizes US images on the screen, allows users to manipulate the probe, and presents different disease scenarios.

The development of an LUS simulator requires a comprehensive understanding of the fundamentals of ultrasound physics and the phenomena used in lung ultrasound diagnosis. LUS is achieved by interpreting ultrasound reverberation artifacts, generally referred to as A and B lines.

A-lines are horizontal, hyperechoic (bright) lines that are evenly spaced and parallel to the pleural line. These artifacts are caused by subpleural air, resulting in large differences in the acoustic impedance between the chest wall and alveoli of the lung. The sound waves travel through the lung and reflect between the pleural line and the transducer, producing a sequence of horizontal lines that are typical findings in air-filled tissues. The B-lines (known as ultrasound lung comets) are vertical, hyperechoic lines that originate from the pleural lines and extend without fading to the bottom of the screen. They replace normal A-lines and are synchronized with the lung sliding. B-lines are the consequence of interlobular septal thickening covered by air-filled alveoli, creating a significant acoustic impedance gradient that produces the reverberation artifacts. B-lines are associated with various lung diseases, including pulmonary edema, interstitial lung disease and pneumonia [20–23].

In this work, we presented a web-based LUS simulator that was created to provide a comprehensive learning tool that visualizes ultrasound images, allows probe manipulation, and presents various disease scenarios. The simulator includes artifacts that are commonly seen during lung exams, such as shadows, A-lines, B-lines, pleural breaks, pleural sliding, and consolidation. The simulator has been tested for compatibility and performance. Its usability and effectiveness in providing practical experience in lung ultrasound interpretation have been demonstrated.

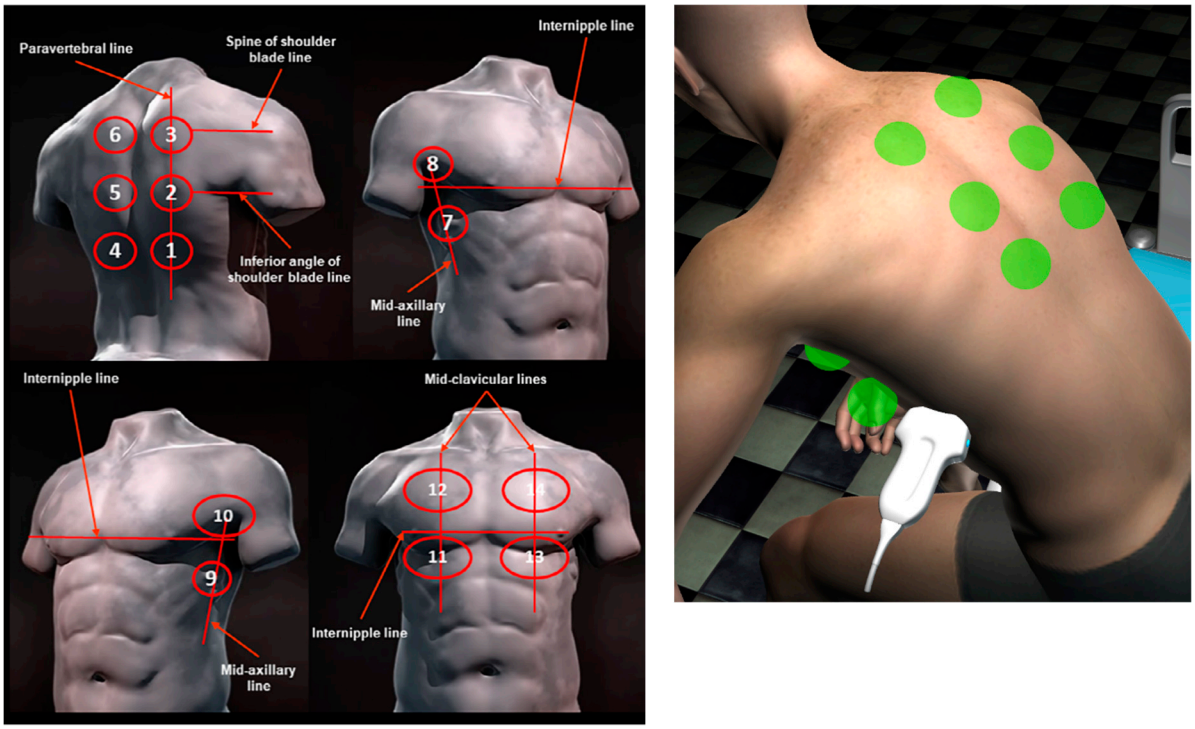
2. Simulation Method

The Lung Ultrasound Simulator was intended to be a web-based application, available for free using only a web browser. This requirement had a significant impact on the selected technology and method of algorithm implementation. Particularly, instead of relying on common 3D libraries, the Lung Ultrasound Simulator had to utilize WebGL technology based on OpenGL ES. WebGL is specifically designed for web-based applications and comes with certain limitations compared to the full OpenGL standard [24]. For example, it may lack support for certain advanced features, such as compute shaders for parallel computations, and it may have restrictions on certain texture types or dimensions.

The first stage of the work includes collecting anonymized real images of the actual ultrasound examinations, with detailed descriptions, and then analyzing and defining the requirements for the project lung simulator. A database of real US examinations was created and used as a source of reference images. The images were prepared for educational purposes and supplied by the Department of Coronary Disease and Heart Failure at John Paul II Hospital, Kraków, Poland.

The next stage was a definition of the learning tool in the simulator. The chosen approach assumes the existence of 14 predefined standard application points for the ultrasound probe on each patient, following established guidelines presented in Figure 1 [25].

Each of these points corresponds to a specific observation that the user should identify on the ultrasound image. After noting all the observed phenomena and considering additional patient information, the user is asked to select the most probable diagnosis. In the subsequent step, the simulator can provide feedback to ascertain the accuracy of the user's observations.



(**a**) (**b**)

Figure 1. Schematic representation of the acquisition landmarks on chest anatomic lines. (**a**) From guidelines [22]; (**b**) represented on the virtual patient in the simulator.

The next stage was to prepare scenarios: different pathologies and diseases were selected, and virtual patients were created to reflect defined conditions. For each of the 14 landmarks, the corresponding pathology was selected, along with the appropriate artifacts that would be visible when the ultrasound probe was positioned over that point.

2.1. Simulation Model

There are typically two main models used in ultrasound simulations: vector-based and volumetric. The vector-based model is created by 3D graphic artists and often relies on CT data as a foundation. This model involves the generation of geometric shapes to simulate ultrasound wave propagation.

On the other hand, the volumetric model is derived directly from CT data, allowing for a more accurate representation of the scanned anatomy. By utilizing the actual CT data, the volumetric model offers a higher level of realism, anatomical details and precision in simulating ultrasound images [26–34].

Both approaches have their strengths and are commonly employed in ultrasound simulation technology. Vector-based (or triangle-based) models are easier to edit, animate and apply interactions to. The Ultrasound simulation can then be performed using a cross-section of that 3D model or using a raytracing technique [35], where rays are sent from the transducer position and travel through the model, simulating a wavefront propagation.

On the other hand, the volumetric model may have lower temporal resolution, may be corrupted by different artifacts, is more challenging to process, and occupies more memory space. A volume model can be sampled using a ray-marching technique, which can implement most of the required phenomena to produce a realistic final ultrasound image.

2.2. Ultrasound Simulation

The first step was a 3D vector model creation of a patient and internal structures. The authors decided to use the Daz 3D [36] software and an asset of male and female with internal organs available in the Daz 3D Assets Store [37]. To meet the simulation requirements, models were adjusted and animated; in particular, the motion of the lungs was added. The animation was manually implemented to reflect the lung's movement as visible on the acquired ultrasound videos. Ribs movement was not implemented because the complexity of modeling exceeds the benefits and impact of animation on the final outcome. The speed and the range of the lung animation vary depending on the selected patient, which reflects the breathing speed and the depth of inhalation.

In the following step, the Unity framework was used to create a simulation platform for the Lung Ultrasound Simulator. The Unity framework is widely utilized for developing 2D and 3D games, and it has also found extensive application in the medical field [38–41]. Moreover, it serves as a solid foundation for creating web-based 3D applications, offering a versatile platform for delivering immersive experiences over the Internet.

The previously prepared models were imported into the Unity framework, and then a virtual examination room and virtual US probe were created. Users can manipulate the probe and position it on the patient's body using a computer mouse.

Next, a disease scenario was implemented for each patient by configuring the landmark states to the appropriate values based on the expected observations.

Then, the simulation algorithm is used to render the simulated US image:

1. According to the virtual probe position and orientation, the 3D model of the patient and its internal structures are rendered using a virtual camera and custom shader program. Only the cross-section is visible and rendered as an image into the framebuffer. This is an approximation of an ultrasound image in which individual tissues are rendered using a custom shader (called Material) that, among others, encodes the required physical parameters (like absorption coefficient and acoustic impedance) as a color. Additionally, the cross-section is completed with a simulation of the tissue structure. This is achieved using a position-dependent pseudorandom values function [42].
2. The same virtual camera renders a UV map of the moving tissues. UV value is a vertex attribute required to correctly map a texture on an object. However, in this case, this step is used to simulate pulmonary pleurae movement and related artifacts. The values rendered on the UV map correspond to the local position of the lung surface. The lung's movement corresponds to the changes in these values.
3. Another virtual camera is employed to render the ultrasound (US) sector. This camera utilizes a shader program that incorporates the modified ray-marching method to simulate wave propagation. This step was challenging because instead of a compute shader, better suited for this problem, a regular fragment shader had to be used. Compute shaders were not supported in the web browser environment at the time of writing. In the resulting image, each column corresponds to a ray, and each row is a ray marching step.
4. A shader program that extends the results of previous steps with additional wave phenomena, including A-lines and noise.
5. A shader program that transforms the results into a selected sector view.

2.3. Artifacts Simulation

In order to create an effective learning tool, it is crucial to simulate various ultrasound artifacts that are commonly encountered during lung examinations. These artifacts play a significant role in the interpretation and analysis of ultrasound images. By accurately replicating these artifacts in the Lung Ultrasound Simulator, users can develop the necessary skills to identify and interpret them in real-life scenarios.

The artifacts can be rendered according to the classical wave equation [43,44]:

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \tag{1}$$

where:

p—acoustic pressure,
c—speed of sound in the medium,
t—time,
and with a relaxation term:

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = -\tau\frac{\partial p}{\partial \mathrm{t}} \tag{2}$$

where:

τ—is a relaxation coefficient.
In cases where there is negligible or no scattering, it leads to (1-D):

$$I(x) = I_0 e^{-2\alpha x} \tag{3}$$

where:

$I(x)$—remaining intensity at the distance x to the wave source,
I_0—initial intensity,
α—absorption coefficient—a product of the medium attenuation coefficient and the frequency of the ultrasound wave.

To solve Equation (2), ultrasound waves can be simplified as a ray that is generated at the transducer and marched over a 2D cross-section plane. During that process, a ray interacts with the medium (tissue characteristics set up with a custom material) and renders the intensity image. The attenuation is implemented using the Equation (3). No reflection or refraction rays are implemented. However, the reflected intensity is calculated as follows:

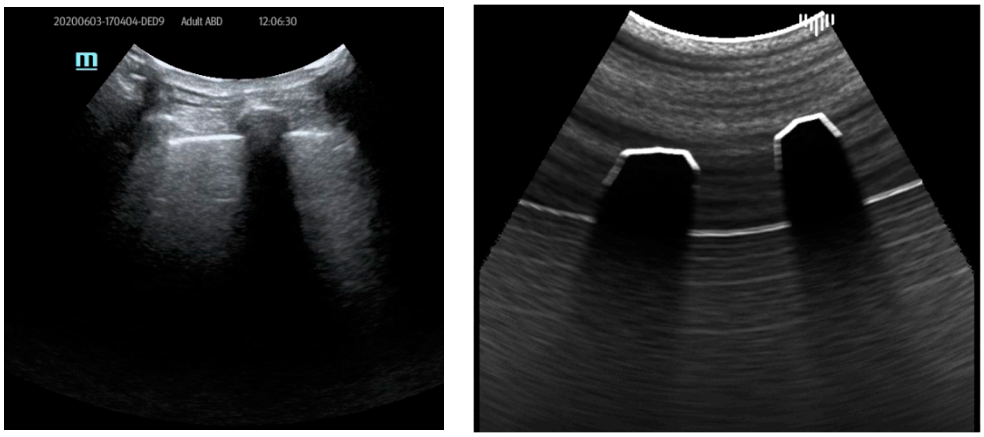
$$I_r = I_i\left(\frac{Z_2 - Z_1}{Z_2 + Z_1}\right)^2 \tag{4}$$

where:

I_r—is the reflected intensity,
I_i—is the incoming intensity,
Z_1, Z_2—are the acoustic impedances, defined as a product of tissue density and speed of sound.

Key artifacts that should be simulated include shadows, A-lines, B-lines, pleural breaks, and pleural sliding. The strength of each artifact can be adjusted as needed and defined for each standard probe application point.

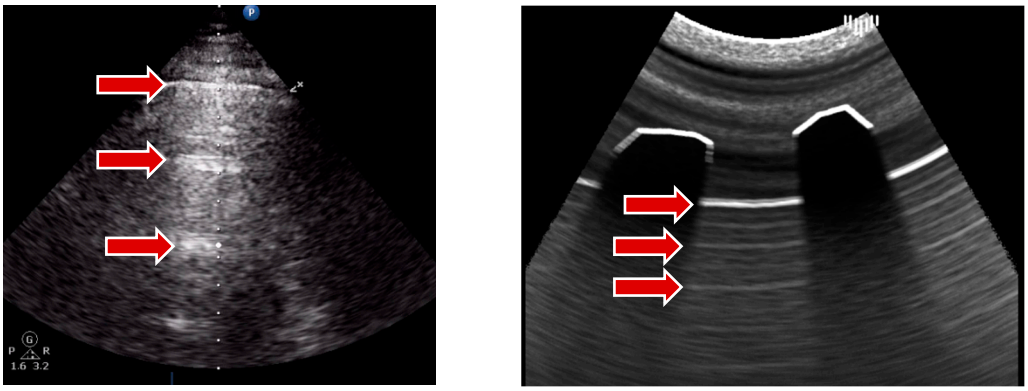
1. Shadows are dark areas that occur when ultrasound waves are blocked or absorbed by dense structures, such as ribs or air-filled spaces (Figure 2). To simulate shadows, first, the cross-section image is rendered with Material attributes defined for each tissue. Then, a ray marching technique utilizes these values to compute shadows for each ray.



(**a**) (**b**)

Figure 2. Ultrasound shadows example: (**a**) real examination; (**b**) the LUS simulation.

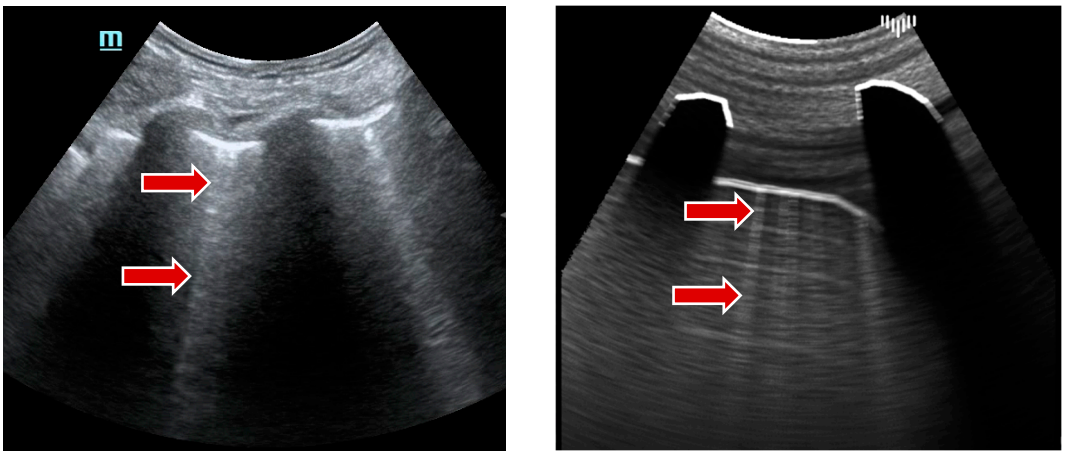
2. A-lines are horizontal, parallel lines seen in normal lung tissue and are created by the multiple reflections of ultrasound waves between the pleural line and the transducer. This is achieved during the ray marching step without a time-consuming ray generation: the input frame buffer, which stores the image of the first render pass, is repeatedly sampled relative to the distance of a pleural line to the transducer. The results can be seen in Figure 3.



(**a**) (**b**)

Figure 3. A-lines example: (**a**) real examination; (**b**) the LUS simulation.

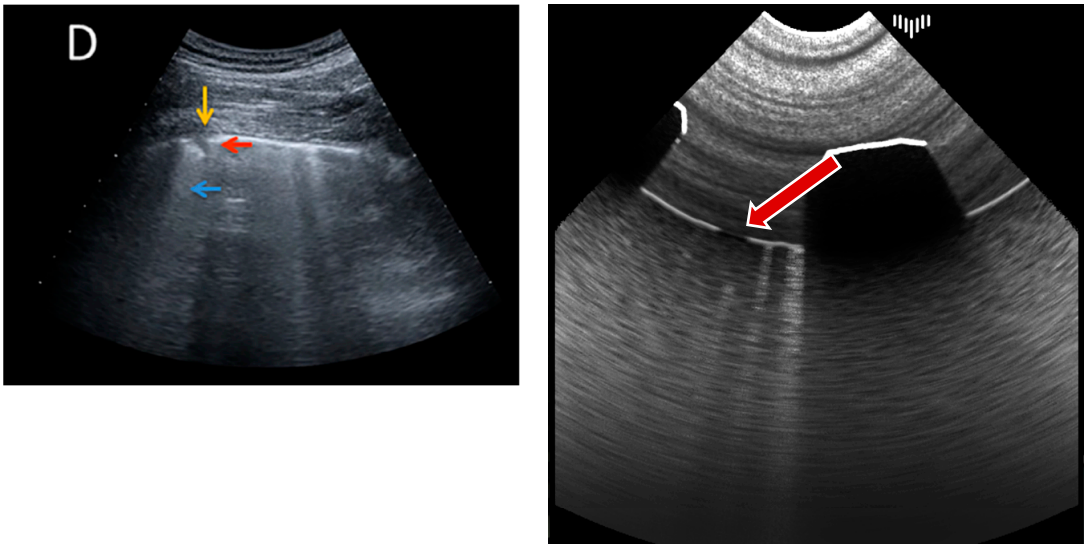
3. B-lines, on the other hand, are vertical lines that represent thickened or edematous lung interstitial spaces and are commonly associated with pulmonary edema or interstitial lung diseases. To simulate this artifact, the intensity of the response is randomly increased at the pleural line, leaving a bright trail interpreted as a B-line. A pseudo-random value is calculated from the UV map (rendered in the previous step) for each ray of the ray marching algorithm. Then, a strength parameter associated with the patient's landmark determines the visibility of the B-lines. This parameter ranges from 0 (no B-lines) to 1 (multiple coalescent B-lines). Radial blur, added in the final step, increases visual similarity to real artifacts. An example of that artifact results is visible in Figure 4.



(**a**) (**b**)

Figure 4. B-lines example: (**a**) real examination; (**b**) the LUS simulation.

4. Pleural breaks, also known as pleural irregularities or pleural abnormalities, appear as disruptions in the smooth pleural line. It was achieved by changing the pleural reflectance parameter in the function of distance to the axis of the probe standard application point (landmark). Figure 5 shows this simulated artifact.



(**a**) (**b**)

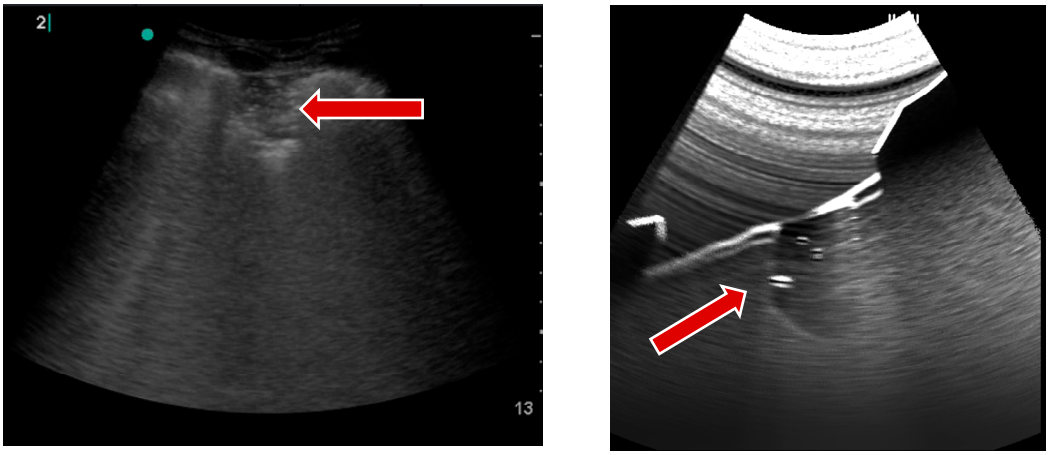
Figure 5. Pleural break example: (**a**) real examination [18]; (**b**) the LUS simulation.

5. Lastly, pleural sliding refers to the dynamic movement of the visceral and parietal pleura during respiration. As mentioned before, it was created using the UV maps of the animated lungs model and calculating pseudorandom values on that basis. The sliding is then achieved by modifying the intensity of the outer part of the pleural line according to the random values.

The attenuation phenomenon is also applied during ray marching, according to the traversed tissue attenuation values. Furthermore, pleural fluid was implemented using a custom 3D object and appropriate material parameters.

In addition to the mentioned artifacts that are highly important during lung ultrasound examinations, it was crucial for educational purposes to simulate the presence of consolidation. Consolidations are regions of lung tissues filled with liquid instead of air. It ensues from the accumulation of inflammatory cellular exudate within the alveoli and

contiguous ducts. The simulation of the image of consolidation was achieved using a predefined mesh object created in 3D software (3D Studio Max, Autodesk) and positioned in a virtual patient interior. Material attributes are added to the consolidation object to increase attenuation and induce more reflectance and then rendered in the final image (Figure 6).



(**a**) (**b**)

Figure 6. Consolidation example: (**a**) real examination; (**b**) the LUS simulation.

Finally, image noise and radial blur are applied to the image.

By simulating these artifacts within the Lung Ultrasound Simulator, users can develop a comprehensive understanding of their appearance, characteristics, and clinical significance. This simulation enables medical professionals and students to gain practical experience in identifying and interpreting these artifacts, enhancing their diagnostic capabilities and preparing them for real-world lung ultrasound examinations.

Due to their lesser importance and the complexity involved in their implementation, other artifacts have not been incorporated into the system at this stage.

3. Results

The LUS simulator is available for free at https://lus.mstech.eu/ (accessed on 20 August 2023). Using only a web browser and a computer mouse, the user can examine a selected virtual patient (5 cases available) and afterward choose the most probable diagnosis (Figures 7 and 8). The simulator shows the correct answers and allows one to revisit all the points to check the correct observations.

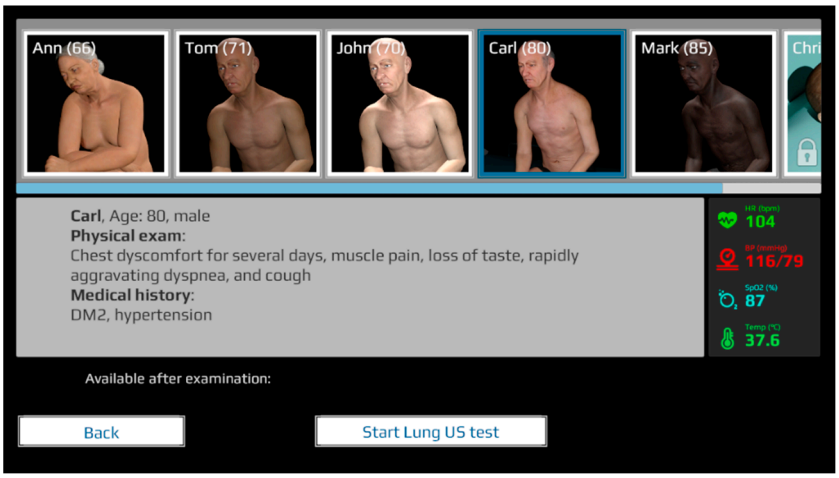


Figure 7. The LUS simulator, patient selection screen.

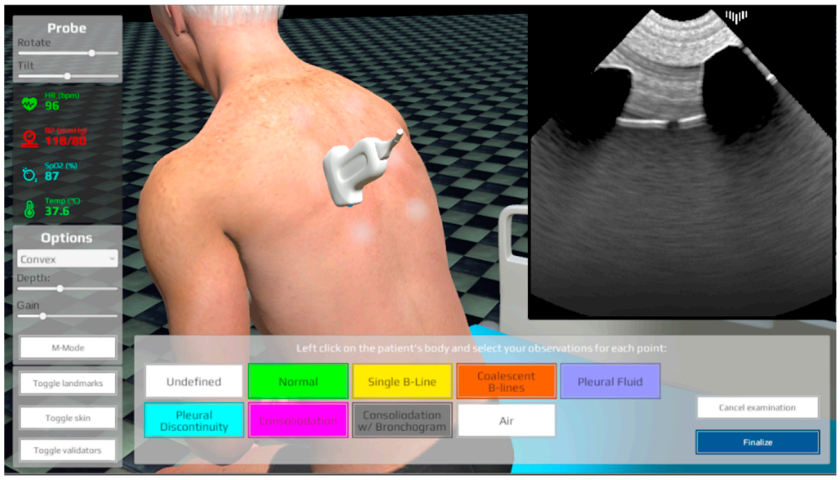


Figure 8. The LUS simulator, patient examination screen.

The simulator underwent thorough testing on various machines, operating systems, and web browsers, demonstrating its compatibility without encountering significant issues. Performance testing was conducted by measuring the simulator's performance with different frame buffer sizes, and the findings are summarized in Table 1. The results indicate that the size of the framebuffers has an impact on performance, with the largest framebuffer making the simulator unusable. Furthermore, a resolution of 512 by 512 pixels was selected as it provides the optimal balance between image quality and rendering efficiency, ensuring that the resulting images maintain sufficient visual clarity.

Table 1. Frames per second (FPS) for different sizes of framebuffers.

Framebuffers Size	Max FPS	Min FPS
4096 × 4096	2.6	1.4
2048 × 2048	78.3	74.3
1024 × 1024	215	181
512 × 512	261	239
256 × 256	271	241
128 × 128	274	247

4. Discussion

Lung ultrasonography (LUS) is a non-invasive imaging technique that allows real-time assessment of pulmonary structures. It has proven to be effective in diagnosing and treating various lung diseases, including pneumonia, pleural effusion, lung consolidation, and pneumothorax. LUS is recommended as a point-of-care examination for patients with symptoms in the chest. We presented a web-based LUS simulator developed to provide a comprehensive learning tool that visualizes ultrasound images, allows probe manipulation, and presents different disease scenarios. The simulator utilizes WebGL technology and a combination of vector-based and ray-marching techniques to simulate ultrasound waves and generate realistic ultrasound images. It incorporates various artifacts commonly encountered during lung examinations, such as shadows, A-lines, B-lines, pleural breaks, pleural sliding, and consolidation. By simulating these artifacts, the simulator helps users develop skills in identifying and interpreting ultrasound findings. The simulator has been tested for compatibility and performance, demonstrating its usability and effectiveness in providing practical experience in lung ultrasound interpretation.

The LUS simulator developed as part of the COVID-19 Rapid Response Projects and made available free of charge constituted a significant stride in responding to the COVID-19 pandemic while simultaneously aiming to encourage the utilization of ultrasonography in diagnosing COVID-19 patients. Faced with the challenges imposed by the pandemic, this innovative tool facilitated the learning and practice of ultrasonography from the comfort of

one's own home, offering immense value to aspiring medical professionals and physicians. The unrestricted access to the simulator not only democratized the acquisition of this crucial skill but also facilitated the commencement of ultrasonography education, which gained paramount importance in the context of COVID-19.

5. Conclusions

In conclusion, the LUS simulator stands as a remarkable innovation in the field of medical education, particularly within the context of the COVID-19 pandemic. While its quality may not rival that of top-tier commercial counterparts, its distinct advantage lies in its accessibility, simplicity, and cost-free availability. The online platform offers an easy-to-use interface that incorporates essential learning features, making it an invaluable resource for aspiring medical practitioners.

In the future, research on the effectiveness of learning using the developed simulator is planned. Additionally, there is an intention to enhance its capabilities through improved implementations of both simulation algorithms and 3D models. Furthermore, the simulator will be supplemented with additional scenarios.

Author Contributions: Conceptualization, K.S.; methodology, K.S.; software, K.S.; validation, K.S.; formal analysis, K.S.; investigation, K.S.; resources, K.S., A.P. and J.L.; data curation, K.S. and J.L.; writing—original draft preparation, K.S. and J.L.; writing—review and editing, K.S.; visualization, K.S.; supervision, A.P.; project administration, K.S.; funding acquisition, A.P. and K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by EIT Health, grant number 20880, "Ultrasonographic Lung Simulator (20880)", "COVID-19 Rapid Response Projects". The project is also partially funded by the Doctoral School of the University of Science and Technology, Cracow, Poland, as a part of the Industrial Doctoral Program "Selected phenomena simulation using ray trace technique".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The simulator is available at https://lus.mstech.eu/ (accessed on 20 August 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chavez, M.A.; Shams, N.; Ellington, L.E.; Naithani, N.; Gilman, R.H.; Steinhoff, M.C.; Santosham, M.; Black, R.E.; Price, C.; Gross, M.; et al. Lung Ultrasound for the Diagnosis of Pneumonia in Adults: A Systematic Review and Meta-Analysis. *Respir. Res.* **2014**, *15*, 50. [CrossRef]
2. Hansell, L.; Milross, M.; Delaney, A.; Tian, D.H.; Ntoumenopoulos, G. Lung Ultrasound Has Greater Accuracy than Conventional Respiratory Assessment Tools for the Diagnosis of Pleural Effusion, Lung Consolidation and Collapse: A Systematic Review. *J. Physiother.* **2021**, *67*, 41–48. [CrossRef] [PubMed]
3. Alrajab, S.; Youssef, A.M.; Akkus, N.I.; Caldito, G. Pleural Ultrasonography versus Chest Radiography for the Diagnosis of Pneumothorax: Review of the Literature and Meta-Analysis. *Crit. Care* **2013**, *17*, R208. [CrossRef] [PubMed]
4. Demi, L.; Wolfram, F.; Klersy, C.; De Silvestri, A.; Ferretti, V.V.; Muller, M.; Miller, D.; Feletti, F.; Wełnicki, M.; Buda, N.; et al. New International Guidelines and Consensus on the Use of Lung Ultrasound. *J. Ultrasound Med.* **2023**, *42*, 309–344. [CrossRef] [PubMed]
5. European Society of Radiology (ESR); Clevert, D.-A.; Sidhu, P.S.; Lim, A.; Ewertsen, C.; Mitkov, V.; Piskunowicz, M.; Ricci, P.; Bargallo, N.; Brady, A.P. The Role of Lung Ultrasound in COVID-19 Disease. *Insights Imaging* **2021**, *12*, 81. [CrossRef]
6. Lê, M.P.; Jozwiak, M.; Laghlam, D. Current Advances in Lung Ultrasound in COVID-19 Critically Ill Patients: A Narrative Review. *J. Clin. Med.* **2022**, *11*, 5001. [CrossRef]
7. La Salvia, M.; Torti, E.; Secco, G.; Bellazzi, C.; Salinaro, F.; Lago, P.; Danese, G.; Perlini, S.; Leporati, F. Machine-Learning-Based COVID-19 and Dyspnoea Prediction Systems for the Emergency Department. *Appl. Sci.* **2022**, *12*, 10869. [CrossRef]
8. Raheja, R.; Brahmavar, M.; Joshi, D.; Raman, D. Application of Lung Ultrasound in Critical Care Setting: A Review. *Cureus* **2019**, *11*, e5233. [CrossRef]
9. Tsou, P.; Chen, K.P.; Wang, Y.; Fishe, J.; Gillon, J.; Lee, C.; Deanehan, J.K.; Kuo, P.; Yu, D.T.Y. Diagnostic Accuracy of Lung Ultrasound Performed by Novice Versus Advanced Sonographers for Pneumonia in Children: A Systematic Review and Meta-analysis. *Acad. Emerg. Med.* **2019**, *26*, 1074–1088. [CrossRef]

10. Russell, F.M.; Ferre, R.; Ehrman, R.R.; Noble, V.; Gargani, L.; Collins, S.P.; Levy, P.D.; Fabre, K.L.; Eckert, G.J.; Pang, P.S. What Are the Minimum Requirements to Establish Proficiency in Lung Ultrasound Training for Quantifying B-lines? *ESC Heart Fail.* **2020**, *7*, 2941–2947. [CrossRef]
11. Chiem, A.T.; Chan, C.H.; Ander, D.S.; Kobylivker, A.N.; Manson, W.C. Comparison of Expert and Novice Sonographers' Performance in Focused Lung Ultrasonography in Dyspnea (FLUID) to Diagnose Patients with Acute Heart Failure Syndrome. *Acad. Emerg. Med.* **2015**, *22*, 564–573. [CrossRef] [PubMed]
12. International Liaison Committee on Lung Ultrasound (ILC-LUS) for the International Consensus Conference on Lung Ultrasound (ICC-LUS); Volpicelli, G.; Elbarbary, M.; Blaivas, M.; Lichtenstein, D.A.; Mathis, G.; Kirkpatrick, A.W.; Melniker, L.; Gargani, L.; Noble, V.E.; et al. International Evidence-Based Recommendations for Point-of-Care Lung Ultrasound. *Intensive Care Med.* **2012**, *38*, 577–591. [CrossRef] [PubMed]
13. Noble, V.E.; Lamhaut, L.; Capp, R.; Bosson, N.; Liteplo, A.; Marx, J.-S.; Carli, P. Evaluation of a Thoracic Ultrasound Training Module for the Detection of Pneumothorax and Pulmonary Edema by Prehospital Physician Care Providers. *BMC Med. Educ.* **2009**, *9*, 3. [CrossRef] [PubMed]
14. Cuca, C.; Scheiermann, P.; Hempel, D.; Via, G.; Seibel, A.; Barth, M.; Hirche, T.O.; Walcher, F.; Breitkreutz, R. Assessment of a New E-Learning System on Thorax, Trachea, and Lung Ultrasound. *Emerg. Med. Int.* **2013**, *2013*, 1–10. [CrossRef] [PubMed]
15. Edrich, T.; Stopfkuchen-Evans, M.; Scheiermann, P.; Heim, M.; Chan, W.; Stone, M.B.; Dankl, D.; Aichner, J.; Hinzmann, D.; Song, P.; et al. A Comparison of Web-Based with Traditional Classroom-Based Training of Lung Ultrasound for the Exclusion of Pneumothorax. *Anesth. Analg.* **2016**, *123*, 123–128. [CrossRef]
16. Heiberg, J.; Hansen, L.; Wemmelund, K.; Sørensen, A.; Ilkjaer, C.; Cloete, E.; Nolte, D.; Roodt, F.; Dyer, R.; Swanevelder, J.; et al. Point-of-Care Clinical Ultrasound for Medical Students. *Ultrasound Int. Open* **2015**, *1*, E58–E66. [CrossRef]
17. Breitkreutz, R.; Dutiné, M.; Scheiermann, P.; Hempel, D.; Kujumdshiev, S.; Ackermann, H.; Seeger, F.H.; Seibel, A.; Walcher, F.; Hirche, T.O. Thorax, Trachea, and Lung Ultrasonography in Emergency and Critical Care Medicine: Assessment of an Objective Structured Training Concept. *Emerg. Med. Int.* **2013**, *2013*, 1–9. [CrossRef]
18. Rojas-García, A.; Moreno-Blanco, D.; Otero-Arteseros, M.; Rubio-Bolívar, F.J.; Peinado, H.; Elorza-Fernández, D.; Gómez, E.J.; Quintana-Díaz, M.; Sánchez-Gonzalez, P. SIMUNEO: Control and Monitoring System for Lung Ultrasound Examination and Treatment of Neonatal Pneumothorax and Thoracic Effusion. *Sensors* **2023**, *23*, 5966. [CrossRef]
19. Pietersen, P.I.; Madsen, K.R.; Graumann, O.; Konge, L.; Nielsen, B.U.; Laursen, C.B. Lung Ultrasound Training: A Systematic Review of Published Literature in Clinical Lung Ultrasound Training. *Crit. Ultrasound J.* **2018**, *10*, 23. [CrossRef]
20. Marini, T.J.; Rubens, D.J.; Zhao, Y.T.; Weis, J.; O'Connor, T.P.; Novak, W.H.; Kaproth-Joslin, K.A. Lung Ultrasound: The Essentials. *Radiol. Cardiothorac. Imaging* **2021**, *3*, e200564. [CrossRef]
21. Lichtenstein, D.A.; Mezière, G.A.; Lagoueyte, J.-F.; Biderman, P.; Goldstein, I.; Gepner, A. A-Lines and B-Lines. *Chest* **2009**, *136*, 1014–1020. [CrossRef] [PubMed]
22. Bhoil, R.; Ahluwalia, A.; Chopra, R.; Surya, M.; Bhoil, S. Signs and Lines in Lung Ultrasound. *J. Ultrason.* **2021**, *21*, e225–e233. [CrossRef] [PubMed]
23. Demi, M.; Soldati, G.; Ramalli, A. Lung Ultrasound Artifacts Interpreted as Pathology Footprints. *Diagnostics* **2023**, *13*, 1139. [CrossRef] [PubMed]
24. WebGL Overview—The Khronos Group Inc. Available online: https://www.khronos.org/webgl/ (accessed on 28 June 2023).
25. Soldati, E.; Roseren, F.; Guenoun, D.; Mancini, L.; Catelli, E.; Prati, S.; Sciutto, G.; Vicente, J.; Iotti, S.; Bendahan, D.; et al. Multiscale Femoral Neck Imaging and Multimodal Trabeculae Quality Characterization in an Osteoporotic Bone Sample. *Mater. Basel Switz.* **2022**, *15*, 8048. [CrossRef]
26. Borzęcki, M.; Skurski, A.; Kamiński, M.; Napieralski, A.; Kasprzak, J.; Lipiec, P. Applications of Ray-Casting in Medical Imaging. In *Information Technologies in Biomedicine, Volume 3*; Piętka, E., Kawa, J., Wieclawek, W., Eds.; Advances in Intelligent Systems and Computing; Springer International Publishing: Cham, Switzerland, 2014; Volume 283, pp. 3–14. ISBN 978-3-319-06592-2.
27. Piórkowski, A.; Kempny, A. The Transesophageal Echocardiography Simulator Based on Computed Tomography Images. *IEEE Trans. Biomed. Eng.* **2013**, *60*, 292–299. [CrossRef]
28. Zhu, M.; Salcudean, S.E. Real-Time Image-Based B-Mode Ultrasound Image Simulation of Needles Using Tensor-Product Interpolation. *IEEE Trans. Med. Imaging* **2011**, *30*, 1391–1400. [CrossRef]
29. Shams, R.; Hartley, R.; Navab, N. Real-Time Simulation of Medical Ultrasound from CT Images. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2008*; Lecture Notes in Computer Science; Metaxas, D., Axel, L., Fichtinger, G., Székely, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5242, pp. 734–741. ISBN 978-3-540-85989-5.
30. Kutter, O.; Shams, R.; Navab, N. Visualization and GPU-Accelerated Simulation of Medical Ultrasound from CT Images. *Comput. Methods Programs Biomed.* **2009**, *94*, 250–266. [CrossRef] [PubMed]
31. Reichl, T.; Passenger, J.; Acosta, O.; Salvado, O. *Ultrasound Goes GPU: Real-Time Simulation Using CUDA*; Miga, M.I., Wong, K.H., Eds.; Society of Photo-Optical Instrumentation Engineers (SPIE): Lake Buena Vista, FL, USA, 2009; p. 726116.
32. Dillenseger, J.-L.; Laguitton, S.; Delabrousse, É. Fast Simulation of Ultrasound Images from a CT Volume. *Comput. Biol. Med.* **2009**, *39*, 180–186. [CrossRef]
33. Bamber, J.C.; Dickinson, R.J. Ultrasonic B-Scanning: A Computer Simulation. *Phys. Med. Biol.* **1980**, *25*, 463–479. [CrossRef]
34. Gjerald, S.U.; Brekken, R.; Hergum, T.; D'hooge, J. Real-Time Ultrasound Simulation Using the GPU. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **2012**, *59*, 885–892. [CrossRef]

35. Burger, B.; Bettinghausen, S.; Radle, M.; Hesser, J. Real-Time GPU-Based Ultrasound Simulation Using Deformable Mesh Models. *IEEE Trans. Med. Imaging* **2013**, *32*, 609–618. [CrossRef] [PubMed]
36. Daz 3D—3D Models and 3D Software. Available online: https://www.daz3d.com/ (accessed on 28 June 2023).
37. Anatomy 4 Pro Bundle. Available online: https://www.daz3d.com/anatomy-4-pro-bundle (accessed on 28 June 2023).
38. Unity Real-Time Development Platform. Available online: https://www.unity.com (accessed on 28 June 2023).
39. Von Haxthausen, F.; Rüger, C.; Sieren, M.M.; Kloeckner, R.; Ernst, F. Augmenting Image-Guided Procedures through In Situ Visualization of 3D Ultrasound via a Head-Mounted Display. *Sensors* **2023**, *23*, 2168. [CrossRef] [PubMed]
40. Korzeniowski, P.; Plotka, S.; Brawura-Biskupski-Samaha, R.; Sitek, A. Virtual Reality Simulator for Fetoscopic Spina Bifida Repair Surgery. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–7 October 2022; IEEE: Kyoto, Japan, 2022; pp. 401–406.
41. Korzeniowski, P.; Chacon, C.S.; Russell, V.R.; Clarke, S.A.; Bello, F. Virtual Reality Simulator for Pediatric Laparoscopic Inguinal Hernia Repair. *J. Laparoendosc. Adv. Surg. Tech. A* **2021**, *31*, 1322–1330. [CrossRef] [PubMed]
42. Gustavson, S. *Simplex Noise Demystified*; Linköping University: Linköping, Sweden, 2005.
43. Huijssen, J.; Verweij, M.D. An Iterative Method for the Computation of Nonlinear, Wide-Angle, Pulsed Acoustic Fields of Medical Diagnostic Transducers. *J. Acoust. Soc. Am.* **2010**, *127*, 33–44. [CrossRef] [PubMed]
44. Soulioti, D.E. Ultrasound in Reverberating and Aberrating Environments: Applications to Human Transcranial, Transabdominal, and Super-Resolution Imaging. Doctoral Dissertation, Department of Biomedical Engineering of the University of North Carolina at Chapel Hill and North Carolina State University, Chapel Hill, NC, USA, 2022.

Real-Time Deformation of Three-Dimensional Volumetric Models Using Mesh Models

Kamil Szostek

Abstract Medical simulations based directly on CT data are appreciated for their realism in reflecting anatomy. On the other hand, such an approach introduces a number of complications when it comes to the possibilities of constructing interventional simulators. In this paper an approach to CT-based volumetric data deformation is presented using the Obi simulation system and Unity framework. This method allows for easier and faster development, while yielding realistic outcomes built upon established and reliable solution. The method was implemented within the MrTEEmothy simulation system, where deformations of the heart tissues are visible in the simulated ultrasound images, during the transeptal puncture procedure.

1 Introduction

In medical simulations, employing CT-based volumetric data is a means to create a realistic simulation and visualizations. This approach ensures not only anatomical accuracy but also enables training on specific patients using their CT scans. Unfortunately, CT data introduces many limitations and increases processing difficulties. One significant challenge is the implementation of deformation and distortion of tissues. This issue has been discussed in the literature and various methods have been proposed for the implementation of interactive deformations. Torres *et al.* proposed a volume data resampling and constructing a tetrahedral mesh and use it to calculate volume deformations [2, 10]. They have also introduced an interactive deformation of medical 3D scans using a touchscreen interface [11]. In [8] authors use a regular free form deformation structures as a wrapper for a volume deformation, similar to the solution presented in this paper.

In Ultrasound Simulation system, like MrTEEmothy from Medical Simulation Technologies, the CT scans of hearts are used directly to create realistic ultrasound

K. Szostek (✉)
Department of Biocybernetics and Biomedical Engineering, AGH University of Science and Technology, Mickiewicza 30 Av., 30–059 Krakow, Poland
e-mail: szostek@agh.edu.pl

Medical Simulation Technologies, Miechowska St. 5b apt. 4, 30–055 Krakow, Poland

R. Burduk et al. (eds.), *Progress on Pattern Classification, Image Processing and Communications*, Lecture Notes in Networks and Systems 766,
https://doi.org/10.1007/978-3-031-41630-9_6

images of transesophageal echocardiography examinations [3, 7, 9]. This approach allows to experience the real anatomy and pathologies without any interference from a graphic designer. The visualization is based on a 4D data sets, with minimum 10 heart phases and about 512^3 voxels per frame. This data is processed in real time to simulate ultrasound images from transesophagus heart examination.

2 Motivations

Using CT data for multi-modal medical imaging simulations, including ultrasound or fluoroscopy, provides a realistic representation of a patient's tissues, facilitating teaching and skills development. However, when volumetric data is used for simulating surgical procedures, the interactions of the surgical devices and other tools with the volumetric model becomes challenging, as well as its deformation and haptic response.

In this paper an approach to volumetric data deformation is presented, based on an existing solutions available in the Unity Framework [12]. This method offers advantages of easier and faster development, allowing for extensions of existing simulation systems, and delivers realistic results based on well-tested and stable solutions. The volume deformation technique is then used in simulation where the heart tissue reaction is visible on the simulated ultrasound image. Specifically, a deformation of a *septum*, a structure between the atriums of the heart, is presented. This structure is susceptible to deformation during structural heart interventions, i.e. transeptal puncture.

3 Deformation Simulation

In this section the approach to CT data deformation is presented. By leveraging the Obi simulation system available in Unity and authors scripts, the ray marching-based ultrasound simulation was enhanced with the capability to visualize tissue deformations.

3.1 Obi Based Simulation

Obi is a powerful physics simulation engine designed for Unity, a popular framework used for developing 2D and 3D games, simulations, and other interactive applications [6]. With Obi, developers can create realistic and dynamic soft-bodies, cloths, and fluids simulations, which can interact with the environment and other objects in the scene. The engine offers an intuitive and flexible interface, empowering users to adjust physical properties and behaviors to achieve precise control over the performance

and appearance of the simulation. Obi features multi-threaded capabilities, however it does not utilize GPU for computation purposes.

The Obi approach is to simulate all physics using particles that can interact with one another and be affected by other objects. This methodology implements the solution proposed in [5], where authors shows a technique for resampling a vertex model into particles with defined constraints. The physics calculations are performed on these particles, and the results are then mapped to the original vertex mesh. In this work, the Obi's particles are mapped onto the CT-based volumetric model.

The physical simulation in Obi system is controlled by a wide range of adjustable parameters to cater to specific requirements. These parameters can be fine-tuned at different levels, including individual particle properties such as mass, plasticity, and size, as well as global settings for the entire soft-body object and the solver. The solver is a core processing module, where simulation details can be defined, including environment parameters like gravity, damping, collisions and the number of iterations for constraint calculations. The correct set up of the solver and the particle-based object's details is essential in achieving accurate simulation results for the selected soft body, i.e. heart tissues. Using the Obi's editor, the user can precisely set the physical parameters of objects.

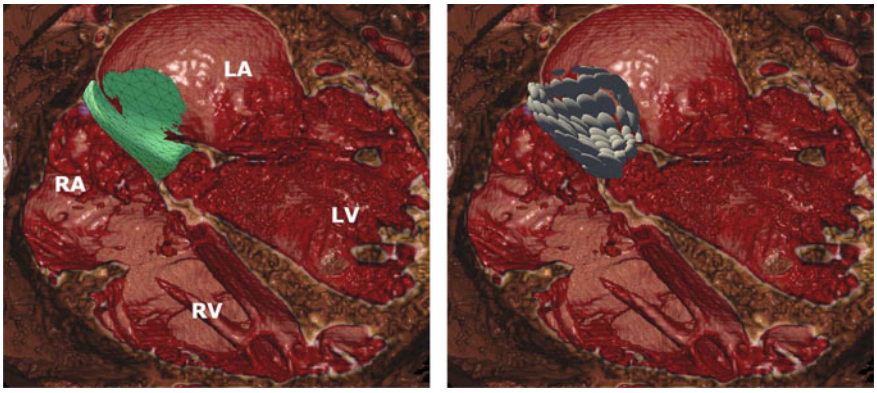
3.2 3D Volume Deformation

The first step of volumetric model deformation was to create a triangle based 3D models from the selected tissue of the volume model. This segmentation process was performed manually using the 3D Slicer software [1], utilizing the methods proposed in [4]. The outcome of the partial segmentation (only septum) is presented on Fig. 1a.

The segmented model was imported into Unity framework, simplified and transformed into a soft-body "particle blueprint". This blueprint contains particles information required to conduct a simulation. The Obi plugin was used to automatically populate particles over the mesh, using a predefined density (Fig. 1b). This step is one of the crucial part in ensuring both the quality and performance of the final simulation.

With the particle model in place, the next step requires a solver configuration. Depending on the requirements and complexity of the simulation system, some of the constraints iterators can have a reduced number of steps or be disabled altogether. The parameters like damping, gravity, sleep threshold and number of collision's iterations must be adjusted.

After setting up the scene with volume models and Obi particles, the simulation can be performed. To reflect the deformations of the particle structure, it is necessary to map them to the voxel data in rendering algorithm. The following approach was implemented: the positions of the particles and their translated location, are copied to the GPU buffers and utilized in a compute shader. This shader creates a 3D texture,



(a) The *septum* tissue segmentation, represented as a green mesh (L - left, R - Right, A - atrium, V - ventricle).

(b) Particles used for calculations.

Fig. 1 Rendering of the CT scan data, heart model, with *septum* vertex model and particle representation. Volume rendering is performed using author's custom method, implemented in Unity Framework.

where each voxel V at position x, y, z contains a vector of its deformation. The vectors are calculated as a weighted sum of all particle transitions (Eq. 1):

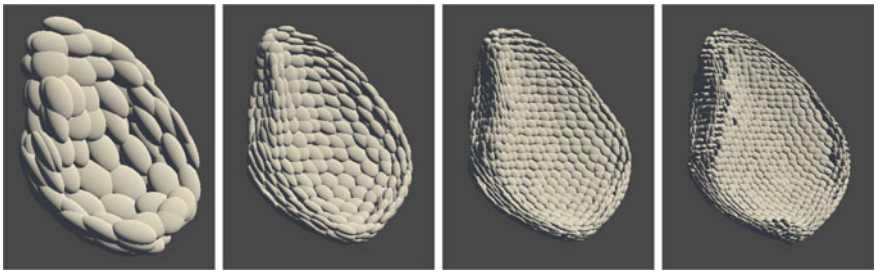
$$V'_{x,y,z} = \sum_{i=1}^{N} -(\boldsymbol{T_i} - \boldsymbol{S_i}) e^{-H \| V_{x,y,z} - S_i \|^2} \tag{1}$$

$V_{x,y,z}$ - original sampling source at x, y, z
N - number of particles,
$\boldsymbol{S}$ - position of source particle,
$\boldsymbol{T}$ - position of current particle,
H - a distance scaling factor.

In this approach the performance is dependent on the number of particles and the size of the generated proxy texture. However, it can be optimized by adding an influence distance limit.

The resulting 3D texture contains a deformation map that is applied during the rendering of the simulated ultrasound image. It can be also used for 3D volume visualization of the heart model.

The number of particles generated for the *septum* model was 115, 417, 965 and 1546 (Fig. 2). The deformation 3D texture size was 64^3, 128^3, 196^3 and 256^3 voxels. The heart model was a CT data set of $512 \times 512 \times 640$ voxels. The resulting ultrasound image was a 512×512 pixels. The visualization was done using a Intel(R) Core(TM) i7-8700 CPU, 3.20 GHz and NVIDIA GeForce RTX 2060 GPU. The average framerate was 30 fps (including US simulation and deformations). The performance results are presented on the Fig. 3 and Table 1 (Fig. 4).



(a) 115 (b) 417 (c) 965 (d) 1546

Fig. 2 Different numbers of particles generated for the *septum* mesh.

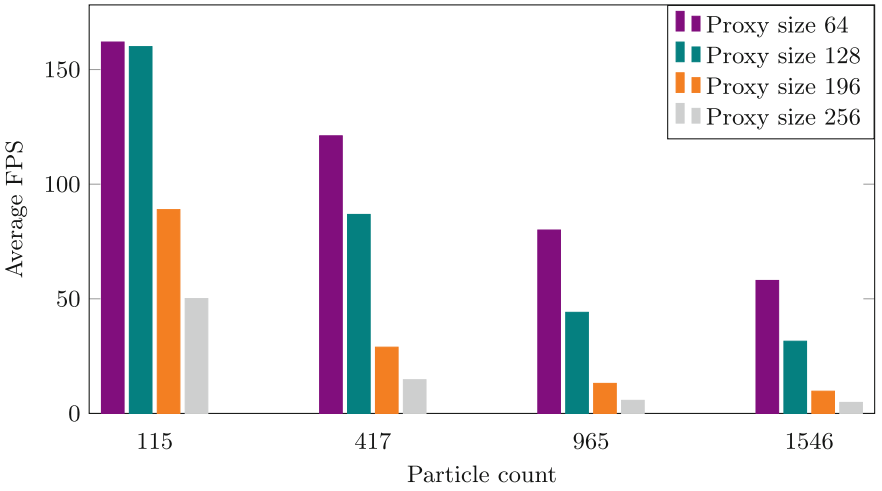


Fig. 3 Average number of frames per seconds (FPS) for different configurations.

Table 1 Average number of frames per seconds for different proxy sizes and particles counts.

	Proxy Size			
Particle count	**64**	**128**	**196**	**256**
115	162	160	88.9	50.1
417	121.1	86.8	28.9	14.7
965	80	44.1	13.1	5.7
1546	58	31.5	9.7	4.8

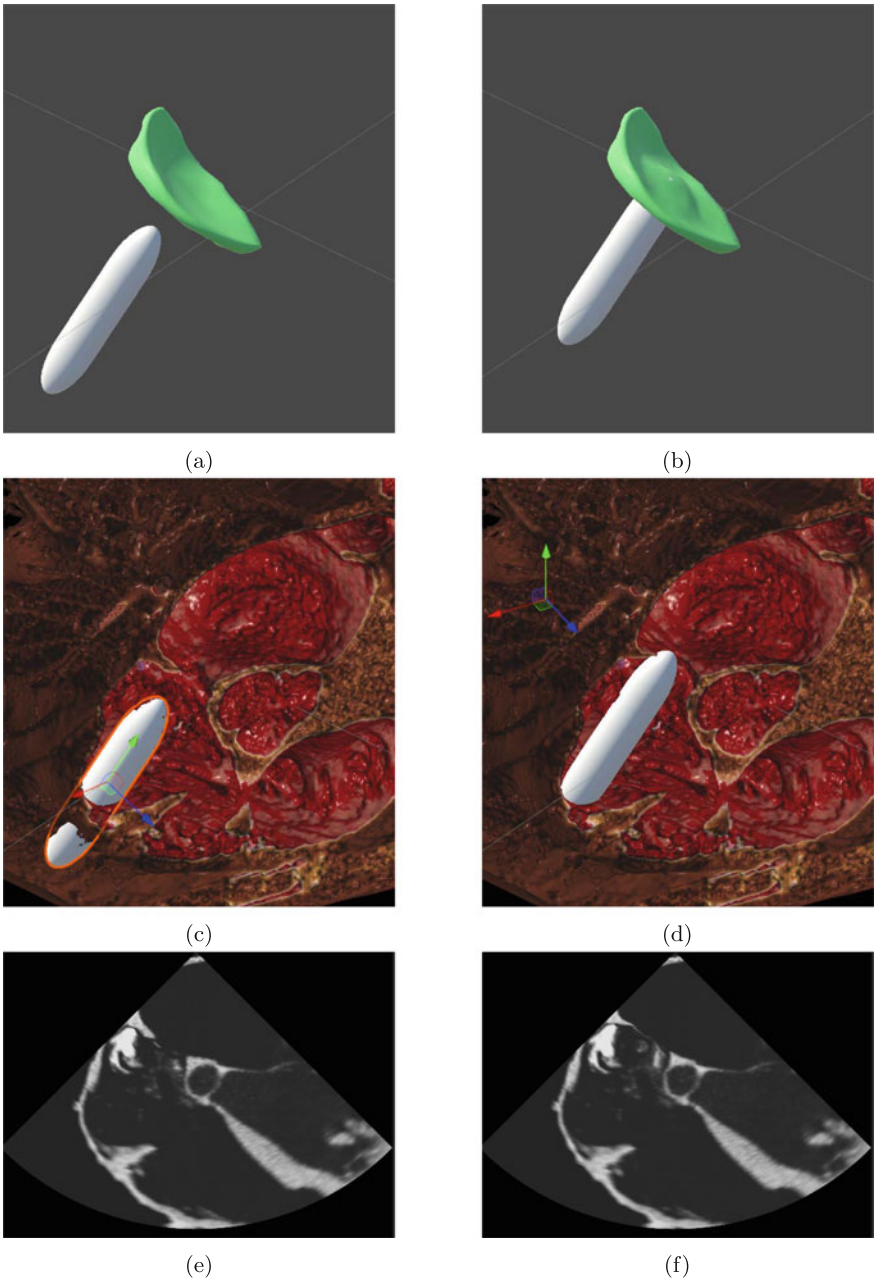


Fig. 4 The rendering of the *sptum* mesh (a, b), CT data (c, d) and simulated ultrasound images (e, f). The right side has no deformation, left side demonstrates the influence of a custom object on the tissue.

4 Results and Discussion

In this work, a solution for a volume data deformation is presented. The Unity framework with Obi extension was used to implement complex soft body physics, i.e. elastic and plastic deformation, and map it onto CT data. This method provides an easy way to extend existing ultrasound simulation platform with device interactions, including not only rigid bodies, but also other types of physical object supported by Obi extensions, such as ropes, rods, clothes or fluids. Furthermore, this solutions offers a straightforward approach to implementing haptic feedback if needed, as the Obi extensions provides force data for each particle or body used in simulation.

The performance and quality of the simulation are influenced by the size of the proxy volumes and the number of particles. Table 1 shows that the best performance is achieved with small proxy volumes and a limited number of particles. However, this approach decreased the level of detail that can be visualized.

One of the limitations of presented method is the manual segmentation process required to create the 3D model of the heart. Although segmentation methods are well-known, they are often time-consuming and require a high level of expertise. Therefore, automating the segmentation process would be desirable to reduce the time and effort required.

Another limitation is the lack of details in the resampled models. The sampled volume points are limited by the resolution of the mesh model, which can result in a loss of detail in the 3D volume model. This may impact the accuracy of the simulation, particularly for small or intricate structures. Increasing the size of the resampled models have significant impact on the performance. However, if the physical simulation is not required for the whole model all the time, it could be divided into smaller parts or with different sampling rate, increasing the quality and the simulation performance.

Finally, the lack of GPU computing in the Obi plugin can also be considered as a limitation of presented method. Although presented method is capable to create real-time simulation, it still requires significant computational resources. Using GPU computing could further accelerate the simulation process, allowing for even faster and more efficient simulations.

Acknowledgment The financial support was provided by the Regional Operational Programme for the Małopolska Region 2014–2020, grant number RPMP.01.02.01-12-0027/19-00, title "Digital Transseptal Puncture Simulator".

References

1. Fedorov, A., et al.: 3D slicer as an image computing platform for the quantitative imaging network. Magn. Reson. Imaging **30**(9), 1323–1241 (2012). https://doi.org/10.1016/j.mri.2012.05.001. https://www.slicer.org/

2. Gascon, J., Espadero, J.M., Perez, A.G., Torres, R., Otaduy, M.A.: Fast deformation of volume data using tetrahedral mesh rasterization. In: Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2013, pp. 181-185. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2485895.2485917
3. Kempny, A., Piórkowski, A.: CT2TEE - a novel, internet-based simulator of transoesophageal echocardiography in congenital heart disease. Kardiologia Polska **68**(3), 374–379 (2010)
4. Lasek, J.: The impact of data preprocessing on the accuracy of CNN-based heart segmentation. In: Chorań, M., et al. (eds.) CORES/IP&C/ACS -2021. LNNS, vol. 255, pp. 173–180. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-81523-3_17
5. Macklin, M., Müller, M., Chentanez, N., Kim, T.Y.: Unified particle physics for real-time applications. ACM Trans. Graph. **33**(4) (2014). https://doi.org/10.1145/2601097.2601152
6. Mendez, J.M.: Obi: Unified particle physics for unity (2023). http://obi.virtualmethodstudio.com
7. Piórkowski, A., Kempny, A.: The transesophageal echocardiography simulator based on computed tomography images. IEEE Trans. Biomed. Eng. **60**(2), 292–299 (2013)
8. Prados, F., León Salas, A., Torres, J.: Haptic interaction with elastic volumetric structures. Int. J. Creat. Interfaces Comput. Graph. (IJCICG) **3**, 63–73 (2012). https://doi.org/10.4018/jcicg.2012010105
9. Szostek, K., Piórkowski, A., Kempny, A., Banyś, R., Gackowski, A.: Using computed tomography images for a heart modeling. J. Med. Inf. Technol. **19**, 75–84 (2012)
10. Torres, R., Rodríguez, A., Espadero, J.M., Otaduy, M.A.: High-resolution interaction with corotational coarsening models. ACM Trans. Graph. **35**(6) (2016). https://doi.org/10.1145/2980179.2982414
11. Torres, R., Rodríguez, A., Otaduy, M.: Hands-on deformation of volumetric anatomical images on a touchscreen. Appl. Sci. **11**(20) (2021). https://doi.org/10.3390/app11209502
12. Unity Technologies: Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine (2023). https://unity.com/. Accessed 24 Apr 2023

Parallelization of the Seismic Ray Trace Algorithm

Kamil Szostek and Andrzej Leśniak

AGH University of Science and Technology,
Faculty of Geology, Geophysics and Environment Protection,
Department of Geoinformatics and Applied Computer Science,
al. A. Mickiewicza 30, 30-059 Kraków, Poland
{szostek,lesniak}@agh.edu.pl

Abstract. This article presents the parallelization of seismic ray trace algorithm. The chosen Urdaneta's algorithm is shortly described. It provides wavelength dependent smoothing and frequency dependant scattering thanks to the implementation of Lomax's method for approximating broad-band wave propagation. It also includes Vinje *et al.* wavefront propagation technique that provides fairly constant density of rays. Then the parallelized algorithm is preliminarily tested on synthetic data and the results are presented and discussed.

Keywords: raytrace, 2D seismic modeling, parallel algorithm.

1 Introduction

Ray trace algorithms are the techniques widely used in computer calculation and simulation, whenever tracing rays needs to be performed. In computer graphics, one of the most popular ray trace problems, these algorithms are implemented to achieve high quality images of virtual scenes, including reflections, refractions, shadows and more. Also in medicine ray trace algorithms are used to achieve extremely high quality images in simulation or reconstruction [1]. In seismology, the accuracy of ray trace algorithms is essential in successful seismic events localization and inversion of seismic source parameters [2]. Furthermore, in seismic modeling different ray trace algorithms have evolved during last years to satisfy computation time and accuracy needs. Eikonal equation solving technique has been developed in parallel systems and presented in [3]. Different approach showed in [4] adapt staggered-grid, finite-difference scheme. Another promising raytracing algorithm was introduced by Asakawa in [5] with later improvement by W. Hao-quan [6].

Algorithms parallelization is one of the most important techniques to improve computational speed. The development of multi-cored processors, highly efficient graphic cards and cluster technologies offers programmers and computer scientists significant acceleration of their algorithms with minimum costs [7]. Urdaneta presented sophisticated join of two techniques and his solution will be briefly presented in this paper and analyzed in terms of calculation speed and parallelization potential.

R. Wyrzykowski et al. (Eds.): PPAM 2011, Part II, LNCS 7204, pp. 411–418, 2012.

2 Urdaneta's Algorithm

2.1 Wavefront and Waverays

Lomax presented a new method for approximating broad-band wave propagation in complex velocity models.[8] This approximation is done using Gaussian weight curve (Fig. 1a). Then the Huygen's principle provides wavepath bending $\Delta\hat{s}$ from points $\boldsymbol{x}_v^{-1}$ and $\boldsymbol{x}_v^1$ (Fig. 1b). Advantages of this technique are: wavelength dependent smoothing which increases wavepaths stability, frequency dependant scattering and capability of handling large to small inhomogeneity sizes (Fig. 2).

Urdaneta in [9] has joined together Lomax's waverays approximation method with Vinje *et al.* [10] wavefront method. Wavefront is defined as a curve with equal traveltime. Its construction presented by Vinje *et al.* implements different

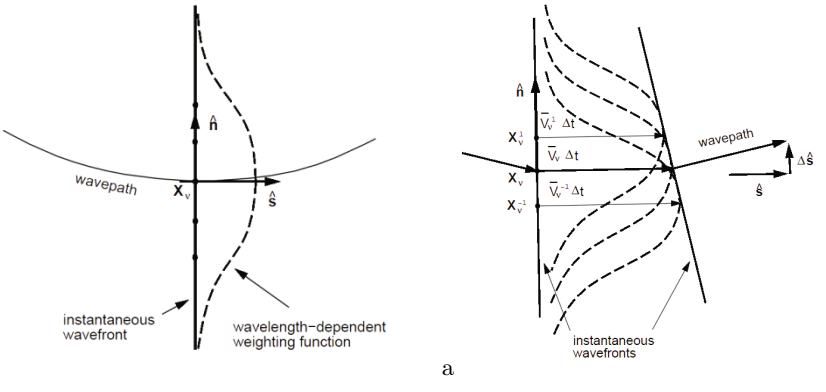


Fig. 1. Lomax's wavelength-dependent weighting function is evaluated by averaging velocities with Gaussian curve centered at point $\boldsymbol{x}_v$[9]

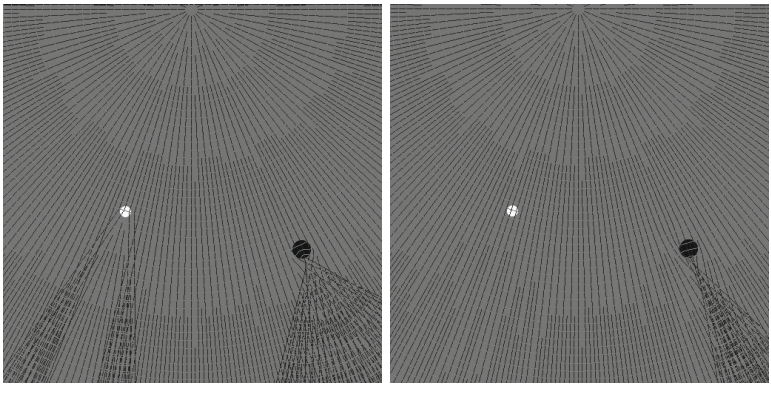


Fig. 2. Lomax's raytracing is wavelength and frequency dependant: on the left frequency was set to 500Hz, on the right to 10Hz. Rays are influenced by small inhomogeneities. Here they are also interpolated using Vinje *et al.* method.

approach to parameters computation: instead of calculating each ray separately, traveltimes and amplitudes are obtained from constructed wavefront. Propagation in medium is achieved by performing raytrace on preceding wavefront (Fig. 3a). If the distance between two successive rays is greater than predefined *DSmax*, a new ray is interpolated between two old ones. This interpolation is done using third order polynomial. Grid values are calculated from surrounding waverays and wavefront (Fig. 3b). New amplitudes are calculated in a similar way.

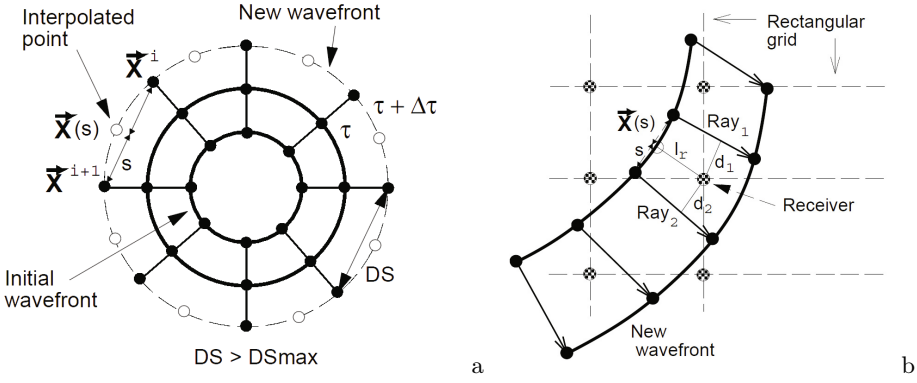


Fig. 3. Wavefront construction presented by Vinje *et al.* Points at time $\tau + \Delta\tau$ are calculated from previous wavefront. If distance between new points is greater than predefined *DSmax* a new point $\boldsymbol{x}_s$ is interpolated (a). Traveltimes and amplitudes are interpolated onto the grid as presented in figure b [9].

2.2 Urdaneta's Algorithm

Urdaneta's final algorithm begins with the definition of structure of ray in the wavefront. This structure, called *cube*, is as followed:

- points *x0* and *x1* – positions of the beginning and the end of the ray, respectively,
- *angle* – arriving angle of the ray at point *x1*,
- *ampl* – amplitude at the ray end,
- *cf* – flag that describes the ray status.

The maximum size of the wavefront is predefined. In case of exceeding this value, an error message is sent and the algorithm stops. In the initialization function, the first wavefront is defined – the 'shot point' – with starting points *x0* and proper angles. The propagation of wavefront is performed using Lomax's algorithm for waverays. The wave frequency and timestep are arbitrarly chosen. For each ray in the wavefront array, a new position (point *x1*) and arriving angle are calculated. Then, if there is self crossing wavefront or any ray is out of borders, status flag is properly set. In next step, these rays are removed from the array. Amplitudes, *gridding* and new rays interpolation are done using Vinje

et al. methods. Finally, the wavefront proceeds to next timestep. It is complete by replacing point *x0* with *x1*. These calculations repeat until wavefront array size is more than 4 (Fig. 4).

In order to calculate all arrivals times, *selfcrossing check* subroutine should be deactivated, but, for the sake of simplicity and to reduce the computational time, crossed rays are removed.

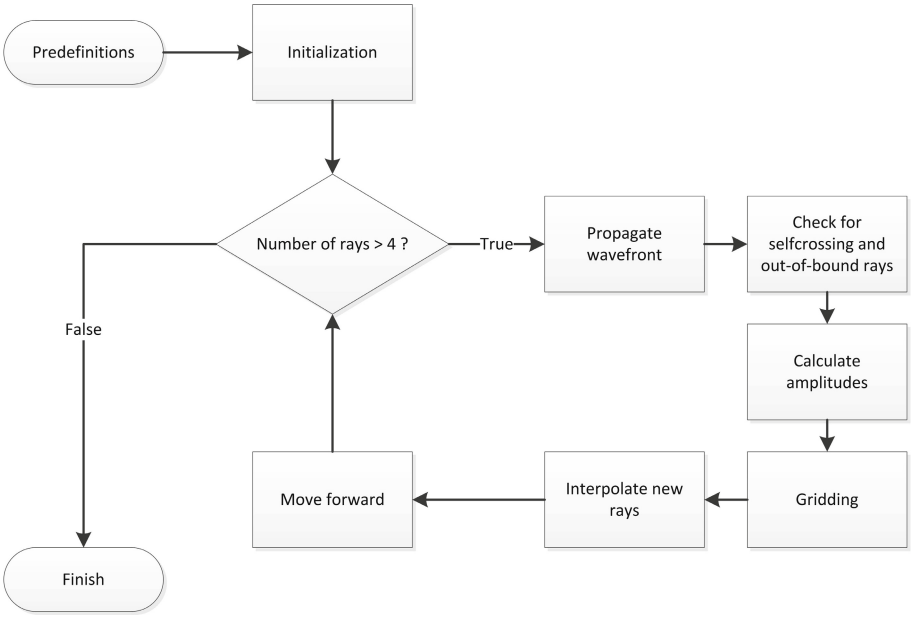


Fig. 4. Diagram of Urdaneta's algorithm [9]

3 Algorithm Parallelization

3.1 CPU Utilization

Urdaneta's algorithm has been implemented in C# (.NET 4.0) and tested using synthetic velocity models. Before any parallelization, CPU workload should be measured to get information, which procedures have the largest influence to the computation speed. The ANTS Performance Profiler was used to obtain CPU most time consuming function. Table 1 shows the results of this analysis. Insignificant methods are omitted. The profiling and later tests was performed using 512m x 512m synthetic isotropic velocity model, with 1600 grid nodes and timestep equal 3ms. Shot-point was set to the x=256m and z=256m.

The most time consuming procedure, according to profiler results, is gridding. It takes about 80% of whole computation time for chosen set of parameters. This procedure loops over every grid point on every wavefront step, so the size of the grid has a major influence on raytrace computation time. Enlarging the number of nodes will result in increasing gridding computation time.

Table 1. Results of Urdaneta's algorithm time analysis with ANTS Performance Profiler

Procedure Name	CPU Time [%]	CPU Time With Children[%]	Hit Count
RaytraceTest	0.029	97.851	1
Gridding	76.964	79.811	56
PropagateWavefront	0.009	5.204	56
CalculatePropagation	0.079	5.173	5581
GetAvarageVelocity	2.345	5.044	16743
CheckForSelfCrossing	1.734	3.480	55
GetDistance	2.862	2.862	3648859
CheckIntersection	1.715	1.715	371403
CalculateAmplitudes	0.240	0.240	56

3.2 Parallelization Procedure

The major idea is to change the most time consuming process, *gridding*, into its parallelized version by domain decomposition of the nodes grid. The *gridding* procedure consists of loop over all grid nodes and this loop can be parallelized, which means that grid nodes can be calculated simultaneously, e.g. in separated threads. As Table 1 shows, *gridding* procedure was called 56 times during profiling – each call for each of wavefronts. Therefore *gridding* procedure is moved outside the main loop. Tests have showed that this can decrease the computation time by about 10%. *Gridding* function is then performed with another loop over all calculated wavefronts, which are stored in array during wavefront propagation (Fig. 5).

The Parallelization is made by creating a thread pool for each processor found and performing *gridding* partially in each thread. Each processor works on part of the grid and calculates nodes values using Vinje *et al.* interpolation method (Fig. 3b). This solution is possible, because there is no interaction between already calculated wavefronts as well as between grid nodes.

3.3 Results of Preliminary Test

Parallelized algorithm was preliminarily tested on dual core processor. Synthetic tests were made for different grid size, starting with 10x10 grid nodes up to 500x500. Results of tests are presented in Table 2.

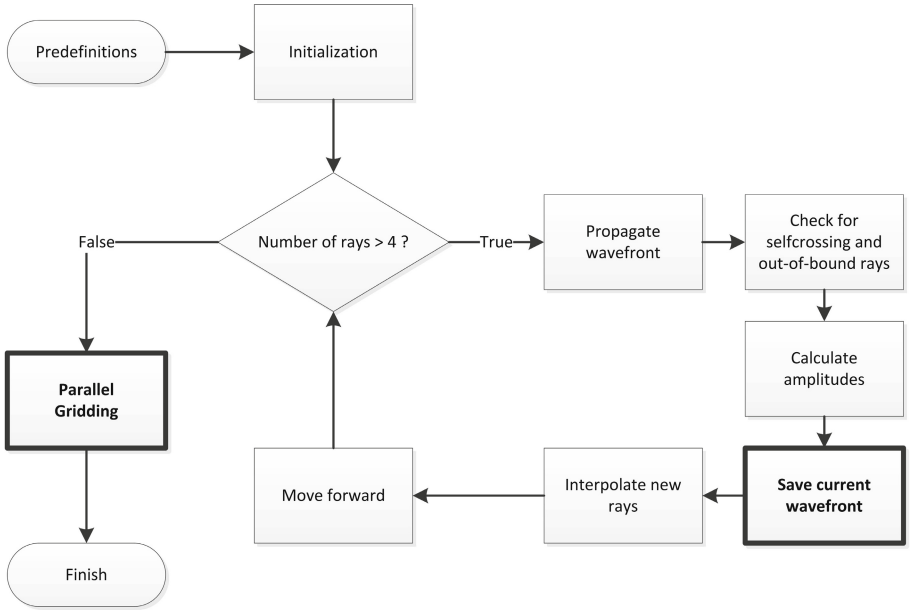


Fig. 5. Diagram of modified Urdaneta's algorithm with parallelization

Table 2. Preliminary test results

Grid size	*gridding* part [%]	Computation time [ms]		Acceleration
		1 CPU	2 CPUs	
10x10	44.09	156.00	140.40	1.111
20x20	76.51	358.80	249.60	1.438
30x30	87.80	702.00	436.80	1.607
40x40	92.63	1170.00	686.40	1.705
50x50	94.46	1778.40	998.40	1.781
60x60	96.44	2527.20	1388.40	1.820
70x70	96.88	3416.41	1872.00	1.825
80x80	97.58	4399.21	2418.00	1.819
90x90	98.25	5506.81	3010.81	1.829
100x100	98.79	6817.21	3697.21	1.844
150x150	99.47	15319.23	8174.41	1.874
200x200	99.63	26988.05	14414.43	1.872
250x250	99.82	42510.07	22604.44	1.881
300x300	99.85	60886.91	32370.06	1.881
350x350	99.87	83116.95	43960.88	1.891
400x400	99.90	108154.99	57486.10	1.881
450x450	99.94	136859.04	72680.53	1.883
500x500	99.93	169260.30	89996.56	1.881

The tests proved that parallelization was done correctly. For each grid size the acceleration is more than 1.85, except the small ones: small grid sizes are calculated faster and parallelizable part takes less time, as presented in Table 2. What is more, usually small grid sizes are considered as less relevant.

4 Summary and Future Work

This paper is an introduction to the subject of ray trace algorithms parallelization, presenting the parallelization of one of the seismic ray trace algorithm. It computes first arrival travel-times and amplitudes estimation in 2D medium using Lomax's ray trace method and Vinje *et al* wavefront propagation technique, both joined by Urdaneta. Parallelization has been done on the most time consuming procedure, *gridding*, which has the most influence on the algorithm speed. Preliminary tests on dual core CPU proved that acceleration is considerably high for relatively dense grids of nodes. Future work will be done in several areas of interest. Firstly, the presented parallel algorithm is going to be tested on machines with more than two CPUs to prove its reliability. Secondly, presented technique will be compared to other popular travel time and amplitude estimation techniques and their parallel versions in the matter of computation time and errors. What is more, as the GPU calculations become more popular and efficient, presented ray trace algorithm will be implemented and tested on graphic hardware, also in context of server side calculations [11].

Acknowledgements. The study was financed from the statutory research project No 11.11.140.032 of the Department of Geoinformatics and Applied Computer Science, AGH University of Science and Technology.

References

1. Hadwiger, M., Ljung, P., Rezk Salama, C., Ropinski, T.: Advanced illumination techniques for GPU-based volume raycasting. In: ACM SIGGRAPH 2009 Courses (SIGGRAPH 2009), pp. 1–166. ACM, New York (2009)
2. Scales, J.: Theory of seismic imaging. Samizdat Press (1994), `http://samizdat.mines.edu/imaging/`
3. Jeong, W.-K., Whitaker, R.T.: A fast eikonal equation solver for parallel systems. Imaging 84112, 1–4 (2007)
4. Shunhua, C., Greenhalgh, S.: Finite-difference solution of the eikonal equation using an efficient, first-arrival, wavefront tracking scheme. Geophysics 59(4), 632–643 (1994)
5. Asakawa, E., Kawanaka, T.: Seismic raytracing using linear traveltime interpolation. Geophysical Prospecting 41(1), 99–111 (1993)
6. Hao-quan, W.: An Improved Method of Linear Travel-Time Interpolation Ray Tracing Algorithm. Acta Physica Polonica A 118(4), 521–526 (2010)
7. Danek, T.: Parallel computing and PC clusters in seismic wave field modeling. Geoinformatica Polonica 7, 25–34 (2005)

8. Lomax, A.: The wavelength-smoothing method for approximating broad-band wave propagation through complicated velocity structures. Geophys. J. Int. 117(2), 313–334 (1994)
9. Urdaneta, H.: Wavefront construction using waverays. Stanford Exploration Project, Report 80, 85–100 (2001)
10. Vinje, V., Iverson, E., Gjøystdal, H.: Traveltime and amplitude estimation using wavefront construction. Geophysics 58(8), 1157–1166 (1993)
11. Szostek, K., Piórkowski, A.: OpenGL in Multi-User Web-Based Applications. In: Innovations in Computing Sciences and Software Engineering, SCSS 2009, pp. 379–383. Springer, Heidelberg (2010)

echocardiography
image processing
medical simulation

Kamil SZOSTEK[1], Adam PIÓRKOWSKI[1], Aleksander KEMPNY[2],
Robert BANYŚ[3], Andrzej GACKOWSKI[4]

USING COMPUTED TOMOGRAPHY IMAGES FOR A HEART MODELING

In this paper the quality and analysis of the computed tomography scan sets are presented in the context of creating a 3D/4D model of a heart for the ultrasonography simulator. Data was collected during regular patients examination, using various equipment and technique, therefore not every set has required quality. CT data can be fast characterized with histogram that can show if the brightness ranges of objects (heart structures) are selective. This makes CT data usable for simulation by applying a transform function on the CT images to produce ultrasonography-like images. The aim is to use a PACS system of Hospital, which is the source of data. Therefore a proper technique and system for analysis is needed.

1. DATA FOR CT2TEE SIMULATOR

CT2TEE simulator is a software that renders echocardiograms (both transthoractic and transesophageal) based on patient's real heart model [1]. The data is obtained from the computed tomography (CT) called "cardio-CT", which means its intensity is synchronized with ECG and includes the contrast agent usage to increase the visibility of blood vessels. This technique was described in [2] and [3].

CT data is collected in DICOM files, where each file contains an uncompressed 2D CT image of the body scan in vertical and horizontal axis as presented on the Figure 1. Spatial dimension of that image is either 512x512 or 256x256 and dynamic dimension is 12 bits per pixel, but aligned to 16 bits.

Set of files from one examination creates a three-dimensional block of data, which is evaluated by CT2TEE simulator. Blocks may represent different heart stage of the cycle, hence set of these blocks can be used to simulate animation of the heart activity. Otherwise one of these blocks can be selected for a static simulation. Figure 2 shows an example of the blocks set. The percentage values indicate the heart motion phase.

One CT examination produces large amount of data (from 2 up to 4GB) that should be evaluated in the context of the use in the CT2TEE simulator. This process should be automatic, as the CT data is kept only few days after the examination.

[1] AGH University of Science and Technology, Department of Geoinfomatics and Applied Computer Science, Cracow, Poland, email: pioro@agh.edu.pl
[2] Adult Congenital and Valvular Heart Disease Center, University of Muenster, Muenster, Germany
[3] Department of Diagnosis, Prevention and Telemedicine, John Paul II Hospital, Kraków, Poland
[4] Jagiellonian University, Medical College, Dept. of Coronary Disease, John Paul II Hospital, Kraków, Poland

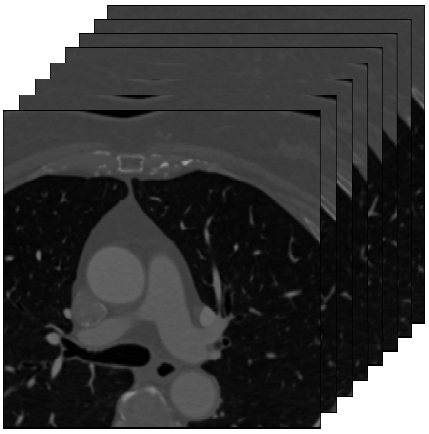


Fig. 1. An example of CT data block. 12bit values are rescaled to 8bit.

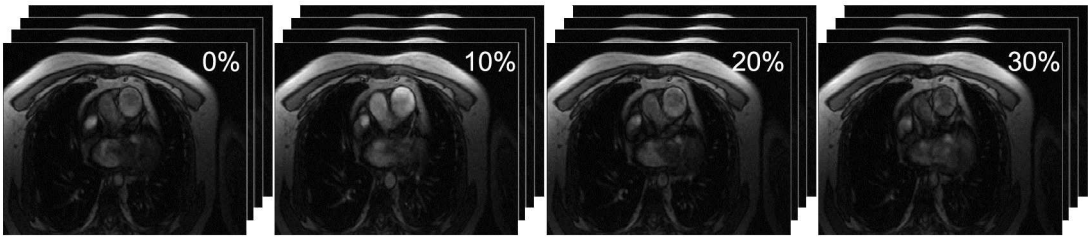


Fig. 2. Example of DICOM CT data set for different heart phase.

2. CT DATA TRANSFORMATION FOR ULTRASONOGRAPHY SIMULATION

CT data must be transformed in order to use for ultrasonography simulation (Figure 3). The reason is that computed tomography shows different than ultrasound-based imaging characteristics of the human body tissues, as this examination methods are based on different physical phenomenon. Human tissue response in the ultrasonography imaging has been already widely discussed [4-8].

What is more, each CT data set, or even a block, has its own features, as a result of different factors, for example: CT scanner features, amount and type of the contrast agent applied, X-Ray absorption characteristics of the tissue. This necessitates individual approach to each CT data set, which means that the transform function [9] should be selected so to ensure the best possible object separation (i.e. heart muscle from blood with contrast agent). Quality of the CT scans is very important in this process.

2.1. OBJECT IDENTIFICATION PROBLEM

To use CT data in the ultrasonography simulations it must be possible to differentiate objects by their brightness on the image. Detection of the heart chambers in images from computed tomography was presented in [10], where various problems of automatic processing where also discussed. Another example of heart structure recognition and selection using image brightness was presented in [11].

Figure 4 presents an example of CT image and its histogram. It shows group of peaks and how they correlates with certain object, for example two peaks on the left represents lungs. In the simulation this part of data is removed (truncated to 0) as the ultrasound waves cannot propagate in this medium, because of the high difference of the acoustic impedance on the tissue-air border [12].

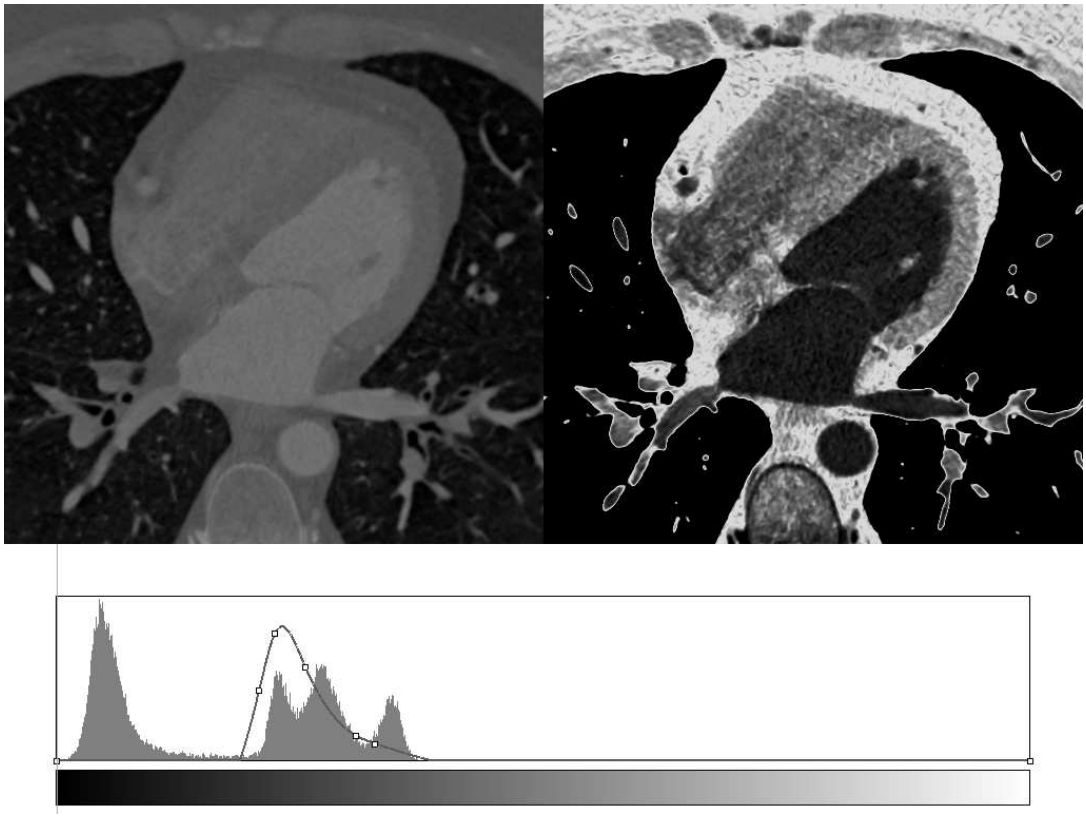


Fig. 3. An example of a transformation function of spline type, applied on a CT image (left) and the result (right).

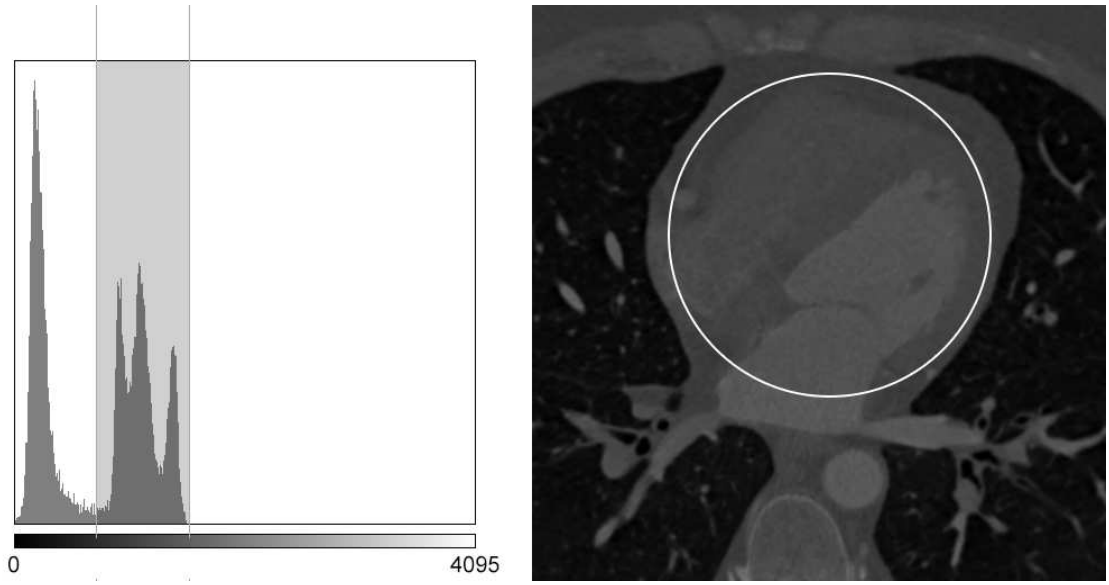


Fig. 4 The most important part of the CT image with the corresponding range on the histogram. Histogram is normalized and represents values from 0 to 4095 of 12bit DICOM image.

The most important part of the CT image is the range of tree peaks, usually between 800 and 1300 level of brightness, as presented on the Fig. 3. The first peak from the left, usually in the range of 800-1000, represents a heart muscle, the next one is correlated with contrasted blood. The last one is associated with saturated blood and bones (sternum and ribs). Quality and utility of the CT data might be defined as the selectivity of these histogram groups.

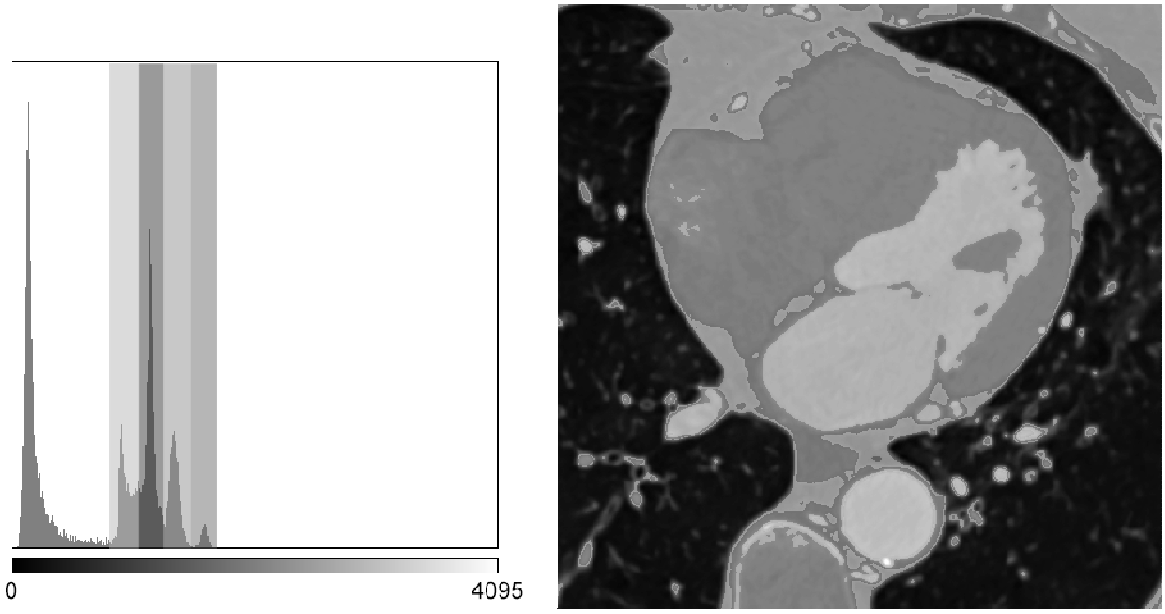


Fig. 5. Example of the correlation of objects on CT image with its histogram.

3. CT DATA ANALYSIS

In the next paragraphs the results of the CT data analysis is presented. The CT data was received from The John Paul II Hospital in Cracow as a part of cooperation with the AGH University of Science and Technology. The CT data is collected in sets of varying number of blocks and slices. Table 1 shows general characteristics of these data.

These data have not been gather with the intent of use in the ultrasonography simulation - it was a result of standard medical examinations. Selective sets were performed for assessment of coronary arteries. These sets may give the best results, because the intensity was not adjusted during examination and the spatial resolutions and contrast are high.

All the data have been anonymized before further processing.

Table 1. General characteristics of the CT data sets.

Name	Dimension [pixels]	No. blocks	No. slices	Data size [MB]	Equipment vendor
SET 2	512x512	1	324	161	Siemens
SET 3	512x512	1	351	176	Siemens
SET 11	512x512	8	262	1044	Siemens
SET 12a	256x256	10	261	326	Siemens
SET 12b	256x256	10	233	291	Siemens
SET 12c	256x256	10	218	272	Siemens
SET 15	512x512	20	210	2201	Siemens

3.1. QUALITY OF THE CT DATA SETS

As mentioned before, there are numerous and various factors that has significant influence on the CT image quality. One of them is the machine characteristic. First analysis of the CT data derived from the hospital showed that machine has implemented method of decreasing the X-ray intensity when a heart is in motion (some phases) to reduce patient radiation dose (CARE Dose technology). This solution creates streaks of noise on the CT images as can be seen on Figure 6. Noise is clearly visible at images of heart in motion phase as well on the related magnifications and their histograms, which show the destructive influence of noise. Therefore, only noise-free data blocks can be used for simulations without de-noising. Otherwise additional processing has to be made on the data.

One of the methods of de-noising was presented in [13]. For the ultrasonography simulation the median filter followed by the average blurring is satisfying and fast enough to be use in automatic CT data processing (Fig. 7).

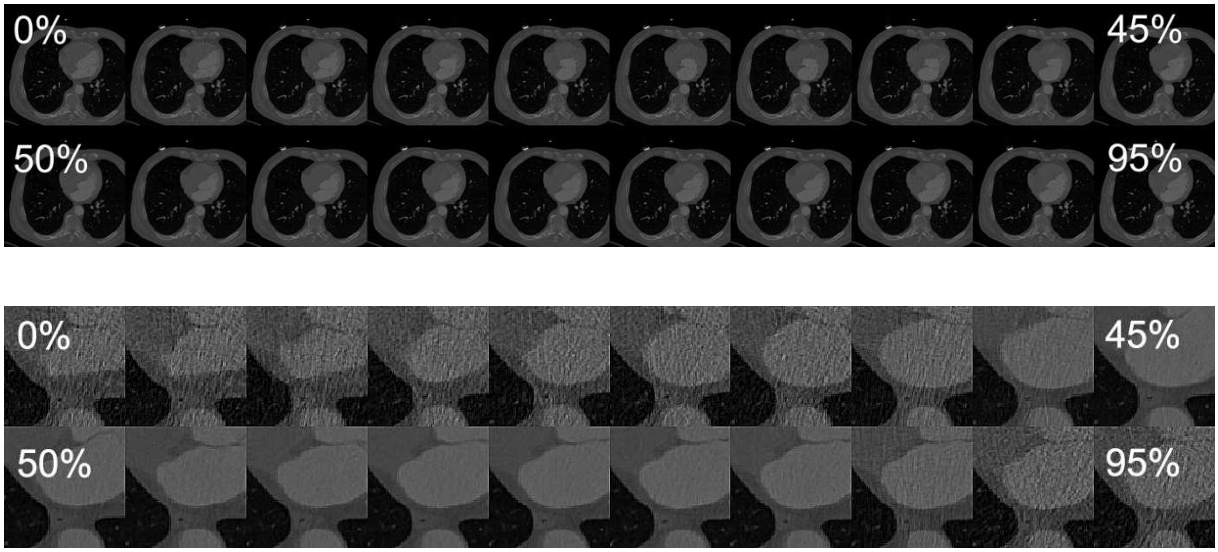


Fig. 6. CT data frames from Set 15 that represent one slice of the heart activity animation (above) with corresponding noise level magnifications (middle) and histograms.

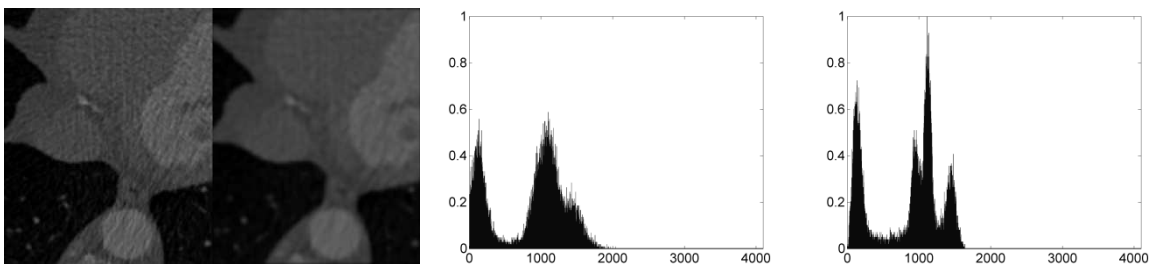


Fig. 7 Example of de-noising filter application on low-dose CT image (0% heart stage) with corresponding histograms.

Another important feature of the CT images is their dimension. Data sets of 512x512 size (in pixels) might provide more details, but on the other hand, large data set may be difficult to use in ultrasonography simulation, because of the limited memory resources.

3.2. DATA BLOCK ANALYSIS

Data block where analyzed in the context of usability in the simulator. Analysis was focused on the histogram characteristics.

Block from Set 3 can be described as a good data set for the simulator (Figure 8). The histogram of a single image yields three main peaks correlated to the heart muscle, same the block histogram. The variance of each slice (in the range from about 1.7e5 up to about 2.7e5) gives information about the domain resolution of the set, which can be considered as good. What is more, the images are 512x512 pixel size and the majority of their space is covered with the heart muscle.

Block from data sets 12a, 12b and 13c also present desirable characteristics (Figure 9). The single image histogram is good and three peaks can be found on the block histogram as well. The variance graph also proves that the blocks quality is more than satisfying. However, this data set has lower resolution (256x256), which has negative impact on the image quality, but might also be considered as advantage, because its smaller size makes this data set more portable.

The last block described in this paper is from set 15 (Figure 10). It has very poor usability in ultrasonography simulation, as only one peak related to the heart muscle can be found on histograms. As described before, streak noise has to be removed with de-noising processing to make this set useful. Another disqualifying feature of this set is the black area where no tissue can be found.

Other data sets, not described in this paper, shows similar characteristics, and will be used to create automatic decision algorithm.

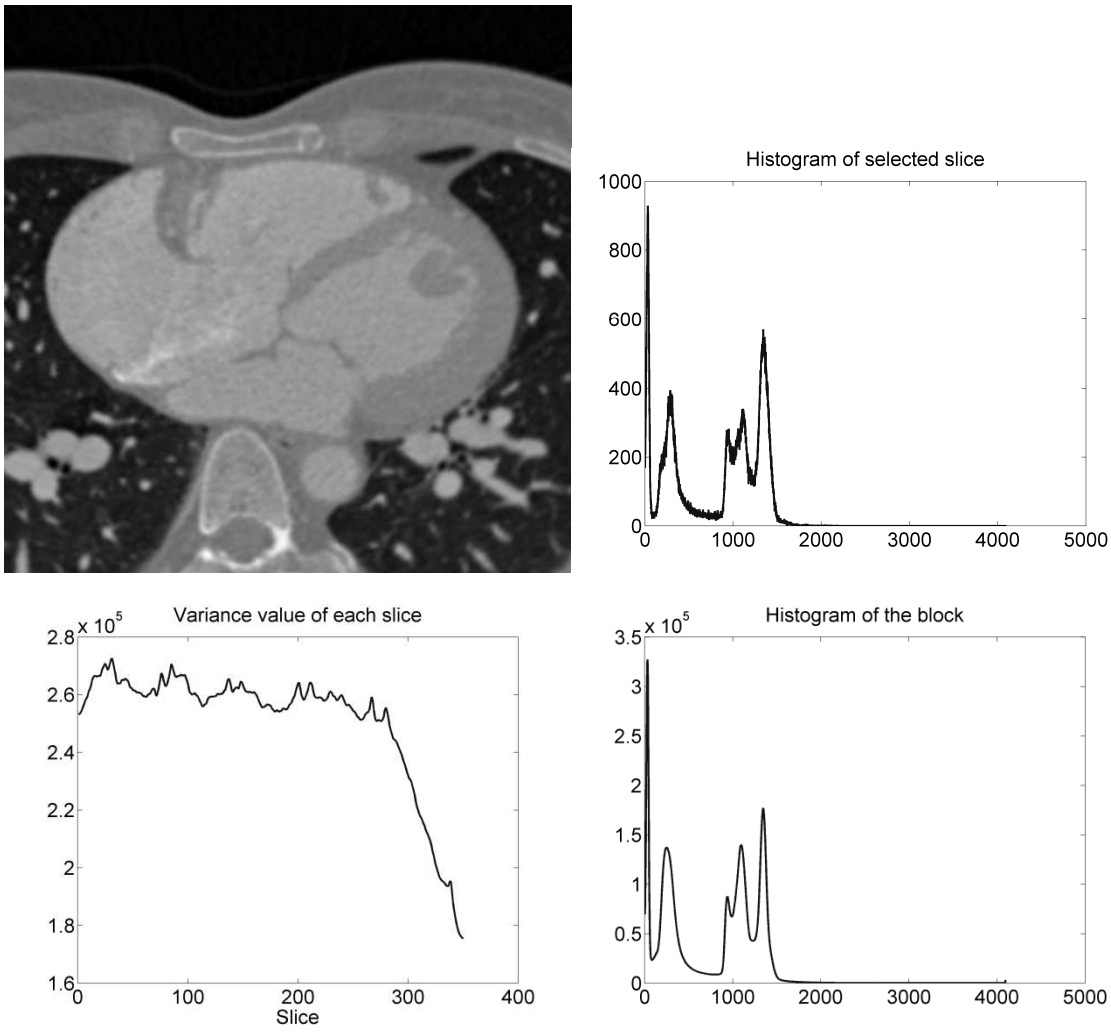


Fig. 8 Set 3 characteristics.

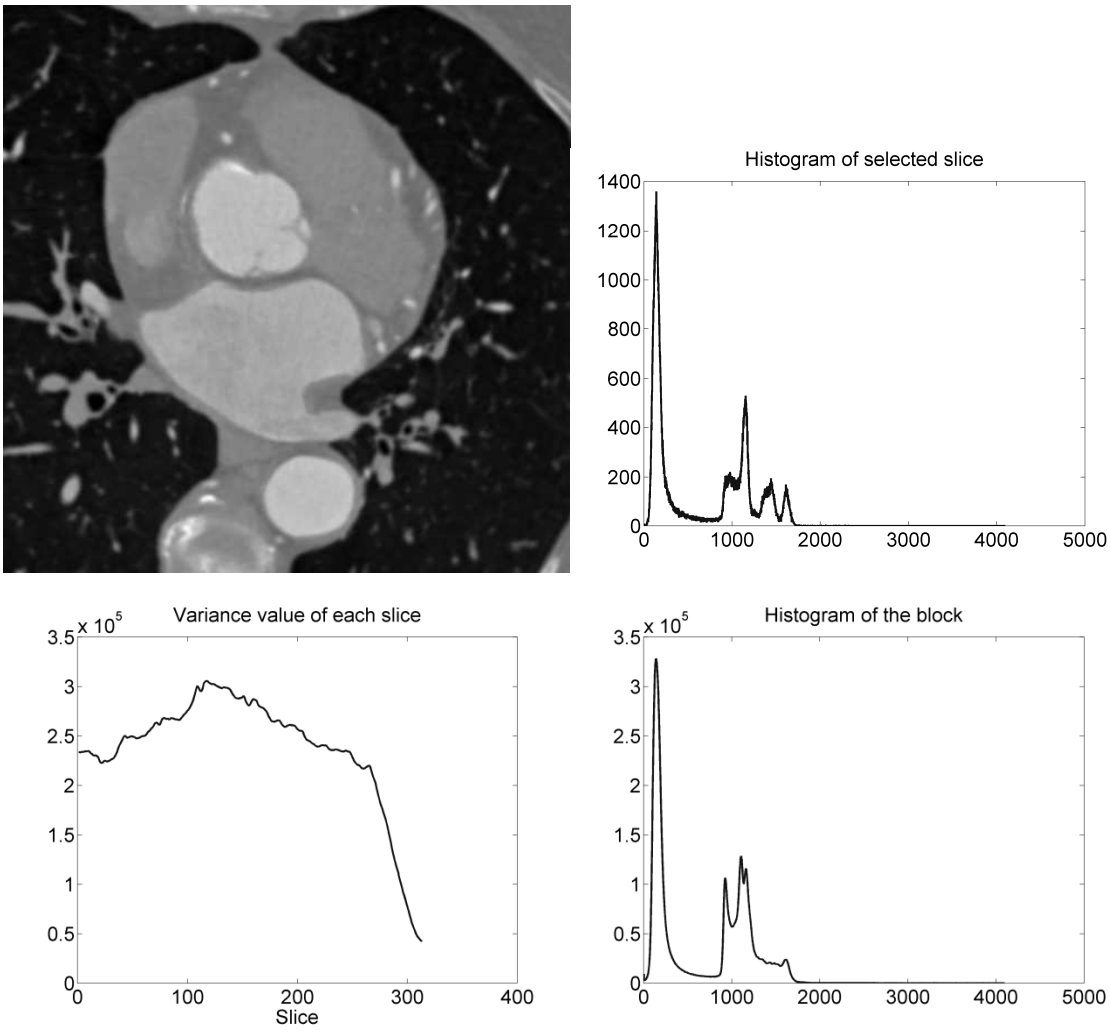


Fig. 9. Set 12 characteristics.

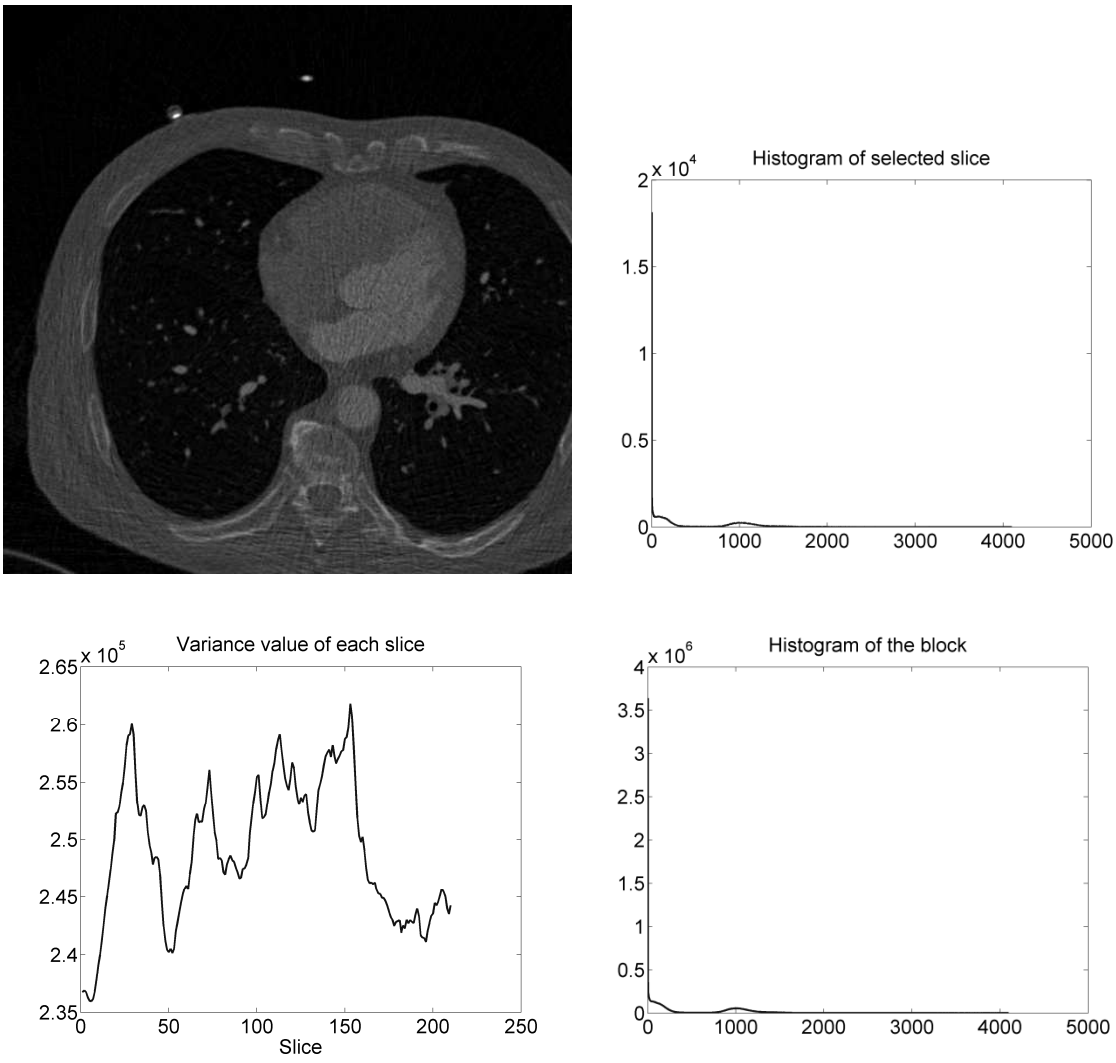


Fig. 10. Set 15 characteristics.

4. SUMMARY

In this paper the quality and analysis of the real CT data sets were presented. The analysis focuses on the selectivity of the heart structures by using the histogram of the individual images as well as the whole sets. The best results was achieved for the CT scans that came from the CT coronary angiography, because it does not include the intensity change during the examination and produces high spatial resolution images.

The histograms analysis is essential during the decision process of CT scan sets usability, as described in this paper. What is more, automatic decision process, part of the future work, will also rely on the histogram analysis. Furthermore, future work involves creating a complex Internet-based solution for eTraining using simulators [14].

ACKNOWLEDGEMENT

This work was financed by the grant No. 15.11.140.203 from the dean of Faculty of Geology, Geophysics and Environment Protection at AGH University of Science and Technology.

BIBLIOGRAPHY

[1] KEMPNY A., PIÓRKOWSKI A.: CT2TEE - a Novel, Internet-Based Simulator of Transoesophageal Echocardiography in Congenital Heart Disease. Kardiol Pol 2010; 68: , pp. 374–379.

[2] PORWIK P., SOSNOWSKI M., WRÓBEL K., WESOŁOWSKI T.: The attempt of the Blood Vessel contractibility estimation on the basis of the Computed Tomography imaging. Journal of Medical Informatics &Technologies, 2011, Vol. 17.

[3] PORWIK P., SOSNOWSKI M., WESOŁOWSKI T., WRÓBEL K.: A Computational Assessment of a Blood Vessel's Compliance: A Procedure Based on Computed Tomography Coronary Angiography. HAIS 2011. Lecture Notes in Artificial Intelligence. Part I, 2011, Springer-Verlag.

[4] DROIN P., BERGER G., LAUGIER P., Velocity Dispersion of Acoustic Waves in Cancellous Bone. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 1998, Vol. 45, No. 3, pp. 581-591.

[5] VAN VENROOIJ, G. E. P. M. , Measurement of ultrasound velocity in human tissue, Ultrasonics 1971, 9, pp. 240-242.

[6] TOPCHYAN A., TATARINOV A., SARVAZYAN N., SARVAZYAN A., Ultrasound velocity in human muscle in vivo: Perspective for edema studies, Ultrasonics 2006, 44(3), pp. 259-323.

[7] HOSKINS P. R., Physical properties of tissues relevant to arterial ultrasound imaging and blood velocity measurement, Ultrasound in Med. & Biol., 2007, Vol. 33, No. 10, pp. 1527–1539.

[8] PEOLSSON A., BRODIN L., PEOLSSON M., A tissue velocity ultrasound imaging investigation of the dorsal neck muscles during resisted isometric extension, Manual Therapy, 2010, 15, pp. 567-573.

[9] PIÓRKOWSKI A., KEMPNY A.: The Transesophageal Echocardiography Simulator Based on Computed Tomography Images. IEEE Transactions on Biomedical Engineering (in print).

[10] TARATORIN, A.M.; SIDEMAN, S., Constrained detection of left ventricular boundaries from cine CT images of human hearts. Medical Imaging, IEEE Transactions on , 1993, vol.12, no.3, pp. 521-533.

[11] BERNADY G., GACKOWSKI A., KEMPNY A., PIÓRKOWSKI A.: Pattern Matching Algorithms In Preparation Of A 3d Heart Models. Journal of Medical Informatics & Technologies, 2011, Vol. 17.

[12] SOMMER F.G., FILLY R.A., MINTON M.J., Acoustic shadowing due to refractive and reflective effects. AJR American Journal of Roentgenol, 1979, 132(6), pp. 973-982.

[13] WEIMIN YU; YANG CHEN; LIMIN LUO;, De-noising of low-dose CT images using space-time nonlocal means over large-scale neighborhoods. Complex Medical Engineering (CME), 2011 IEEE/ICME International Conference on , 2011, vol., no., pp. 455-459.

[14] PIÓRKOWSKI A., WEREWKA J.: A Concept of eTraining Platform for Cardiology Learning based on SOA Paradigm. Proceedings of ICEIS 2012 - 14th International Conference on Enterprise Information Systems, pp. 261-264.