



Universitatea
Transilvania
din Brașov
FACULTATEA DE MATEMATICĂ
ȘI INFORMATICĂ

Programul de studii:
Informatica aplicata

Lucrare de licență

Acustica spațiilor interioare folosind metoda Ray-Tracing

Autor:

Lixandru Andreea-Bianca

Mentori:

Nilă Costin

Gorobievschi Sebastian

(Siemens Industry Software)

Coordonator Științific:

Băicoianu Alexandra

Brașov, 2021

Cuprins

Scopul lucrării	5
Introducere	5
1 Noțiuni teoretice necesare	7
1.1 Sunetul	7
1.2 Reflexia sunetului	10
1.3 Difracția și interferența sunetului	11
1.4 Funcția de răspuns la impuls	13
1.5 Funcția de răspuns la frecvență	14
1.6 Absorbția sunetului	14
1.7 Simularea binaurală	15
1.8 Tehnica de trasare a razelor(Ray-Tracing)	15
1.9 Convoluția sunetului	18
1.10 Transformata Rapidă Fourier și Inversa Transofrmatei Rapide Fourier	18
2 Tehnologii utilizate	21
2.1 Limbajul de programare C#	22
2.2 Limbajul de programare Python și biblioteca matplotlib	22
2.3 Platforma Unity	23
2.4 Platforma Blender	23
2.5 Biblioteca NWaves, XCharts și StandaloneFileBrowser	24
3 Pașii elementari pentru implementarea aplicației	25
3.1 Prezentarea aplicației	26
3.2 Calcul geometric	27
3.2.1 Sfera lui Fibonacci	28
3.2.2 Propagarea razelor	29
3.2.3 Selecția razelor	30
3.3 Calcul fizic	31
3.3.1 Trecerea în domeniul frecvențelor	31
3.3.2 Calculul intensităților	31
3.3.3 Calculul presiunilor	32
3.3.4 Funcția de răspuns în frecvență și funcția de răspuns la impuls	32
3.3.5 Calculul distanțelor	33
3.3.6 Calculul timpilor	34
3.4 Post-procesarea	34
3.4.1 Convoluția sunetului	34
3.5 GUI	35

4 Testarea si analiza rezultatelor	37
4.1 Experimente realizate	37
4.2 Comparații	37
5 Concluzii	38
5.1 Concluzii generale	38
5.2 Concluzii personale	38
5.3 Dezvoltări ulterioare	38

ABSTRACT

urmeaza sa ma ocup de partea asta

Scopul lucrării

În ultimele decenii, au fost dezvoltate multiple modele ce calculează modul de propagare al undelor acustice în spațiul virtual. Din acest motiv am ales să dezvolt o aplicație care modelează propagarea sunetului.

Există multiple motive pentru dezvoltarea și îmbunătățirea modelării acustice în încăperi. Această lucrare va prezenta ce presupune implementarea unui model acustic, atât din punct de vedere fizic, cât și din punct de vedere geometric și vizual. Mai mult de atât, va fi prezentată și implementarea unui model de acest gen, dar și rezultatele obținute. Evident, nici un model care simulează propagarea sunetului într-un mediu virtual nu va imita 1 la 1 realitatea. Totuși, se încearcă găsirea unui algoritm cât mai aproape de realitate.

!aici nu este nici de cum finalizat, ma voi ocupa la final de aceasta bucata!

Introducere

aici mi-am propus putin sa prezint:

-*i* ce presupune realizarea unui model acustic, care ar fi etapele importante pe care urmeaza sa le dezvolt(geometric, fizic, post-procesare)

-*i* de unde a plecat idee, trimitere catre proiect Triton realizat de microsoft, acea lucrare de licenta din italia din care m-am inspirat, articol legat de acustica in moschee si o sa mai caut si altele

-*i* sa povestesc un pic si de psre ray-tracing ca este folosit pentru lumina de obicei dar in contextul acestei aplicatii este foarte util pentru ca in acustica, atunci cand avem frecvente mari undele au o amplitudine foarte mica si deci pot fi approximate folosind raze

Capitolul 1

Noțiuni teoretice necesare

1.1 Sunetul

Sunetul reprezintă vibrația undelor propagate într-un mediu, fie el gazos, lichid sau solid și prezintă, în multe aspecte, dar nu în toate, comportament similar cu alte mișcări de undă pe care le întâlnim în natură, adică undele de apă și undele de lumină, ale căror fenomene de propagare sunt ușor de observat.

În contextul propagării sunetului, aerul este mediul de interes despre care vom discuta atunci când vorbim despre acustica încăperilor, iar perturbarea reprezintă o alterare a presiunii atmosferice peste și sub valoarea sa medie, care produce o mișcare periodică a moleculelor de aer înapoi și înainte de-a lungul aceleiași direcții în care se propagă unda(unde longitudinale).

Propagarea sunetului prin aer este ilustrată în Figura 1.1. În partea superioară este ilustrată alterarea presiunii atmosferice, în timp ce în partea inferioară este ilustrată mișcarea moleculelor de aer asociate cu propagarea sunetului. Dacă intensitatea sunetului crește, gradientul presiunii crește și, ca urmare, mai multe molecule de aer se află în mișcare.

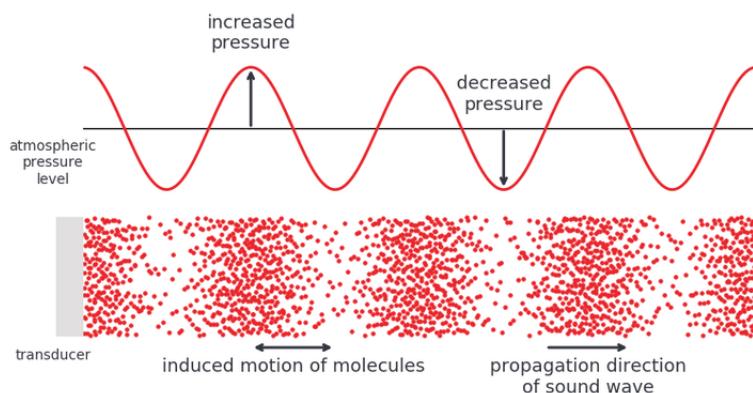


Figura 1.1: Propagarea sunetului prin aer[1]

Un principiu general, stabilit pentru prima dată de Fermat, afirmă că fiecare undă se propagă de la sursă către receptor prin calea cea mai rapidă. Dacă mediu este omogen, precum considerăm că este aerul, atunci viteza sunetului este uniformă prin acest mediu și astfel calea cea mai rapidă devine totodată și cea mai scurtă. Viteza sunetului depinde

doar de temperatură, nu și de alte proprietăți precum presiunea și densitatea. Astfel, putem observa în Tabelul 1.1 cum variază viteza și densitatea sunetului în funcție de temperatură.

Temperatură(°C)	Viteza sunetului($\frac{m}{s}$)	Densitatea aerului($\frac{kg}{m^3}$)
30	349.02	1.1644
25	346.13	1.1839
20	343.21	1.2041
15	340.27	1.2250
10	337.31	1.2466
5	334.32	1.2690
0	331.30	1.2922

Tabelul 1.1: Efectul temperaturii asupra proprietăților aerului[8]

Amplitudinea este valoarea absolută, maximă, a unei cantități care variază periodic. Amplitudinile sunt exprimate fie ca valori instantanee, fie mai ales ca valori de vârf. Aceasta reprezintă fluctuația sau deplasarea unei unde de la valoarea sa medie. În cazul undelor sonore, particulele de aer sunt deplasate, iar această amplitudine a sunetului este exprimată ca intensitatea sunetului. Amplitudinea nu este influențată de frecvență, de lungimea de undă, de perioada de timp sau de viteza sunetului și nici invers.

Lungimea de undă este reprezentată de distanța dintre punctele consecutive corespunzătoare ale aceleiași faze de undă, precum două creste adiacente. *Viteza de propagare*(c) a undei este dată de lungimea de undă, pe care o vom nota cu λ , fiind distanța parcursă de val între două instanțe de fază egală și de timp, T , timpul necesar acestei distanțe.

$$c = \lambda/T \quad (1.1)$$

Inversa perioadei este numită *frecvență*(f) și indică de câte ori particulele de aer s-au mișcat înapoi și înainte într-o secundă. Frecvența este măsurată în Hertz[Hz].

$$c = \lambda f \quad (1.2)$$

O caracteristică importantă a unei unde sonore este *faza*, care specifică locația unui punct în cadrul unui ciclu de undă al unei forme de undă repetitive. În majoritatea cazurilor, diferențele de fază dintre undele sonore sunt mai importante decât fazele în sine. Diferența de fază dintre două unde sonore cu aceeași frecvență care se deplasează dincolo de o locație fixă este dată de diferența de timp dintre aceleiasi poziții în cadrul ciclurilor de undă ale celor două sunete, exprimată ca o fracțiune dintr-un ciclu de undă.

Două unde sonore de aceeași frecvență care sunt perfect aliniate au o diferență de fază de 0 și se spune că sunt „în fază”. Două unde care sunt „în fază” se adaugă pentru a produce o undă sonoră cu o amplitudine egală cu suma amplitudinilor celor două unde.

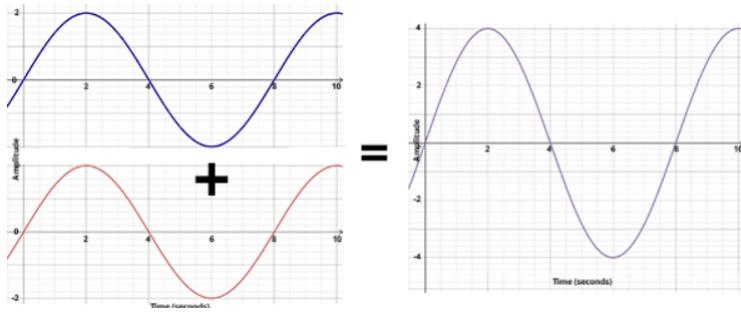


Figura 1.2: Două unde sonore „în fază”

Dacă una dintre cele două unde sonore cu aceeași frecvență este deplasată cu o jumătate de ciclu față de celalaltă, se spune că undele sonore sunt „defazate”. Două unde sunt „defazate” dacă se anulează reciproc exact când sunt adunate împreună.

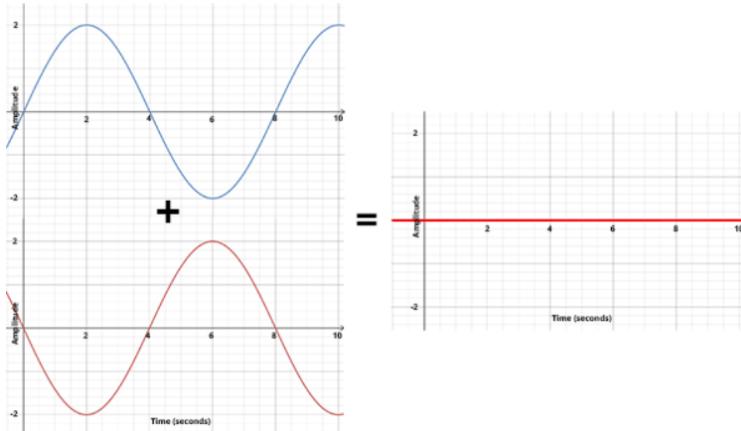


Figura 1.3: Două unde sonore „defazate”

Diferența de fază este exprimată ca un unghi, deoarece forma de undă a unui ton pur alcătuită dintr-o singură frecvență poate fi descrisă utilizând funcția sinus trigonometrică.

Cele mai multe sunete sunt mult mai complexe decât o singură frecvență, dar constau în schimb din multe unde sinusoidale diferite la frecvențe diferite. Când mai multe unde sinusoidale se combină pentru a crea un sunet, formele de undă ale tuturor undelor sinusoidale sunt adăugate la fiecare locație de-a lungul formei de undă.

Energia(W) poate fi auzită de ființele vii. Sunetul este o undă mecanică și ca atare constă fizic în compresie elastică oscilatorie și în deplasarea oscilatorie a unui fluid. Prin urmare, mediul acționează ca stocare atât pentru energia potențială, cât și pentru energia cinetică. În consecință, energia sonoră dintr-un volum de interes este definită ca suma densităților de energie potențială și cinetică:

$$W = W_{\text{potențială}} + W_{\text{cinetică}} \quad (1.3)$$

Presiunea acustică este abaterea presiunii locale față de presiunea atmosferică. În aer, presiunea poate fi măsurată cu ajutorul unui microfon, iar în apă cu ajutorul unui

hidrofon. Unitatea de măsură dată de către Sistemul internațional de unități de măsură, SI, pentru presiunea acustică este Pascal(Pa). Formula folosită pentru transformarea intensității în presiune este:

$$p = \sqrt{2I\rho c}, \quad (1.4)$$

unde ρ este densitatea aerului, iar c este viteza sunetului.

Puterea sunetului este rata la care energia sonoră este emisă, reflectată, transmisă sau recepționată, pe unitate de timp. Unitatea SI pentru puterea sunetului este Watt-ul(W). Pentru o sursă de sunet, spre deosebire de presiunea sonoră, puterea nu este dependentă nici de încăpere, nici de distanță. Presiunea sonoră este o proprietate a câmpului într-un punct din spațiu, în timp ce puterea sonoră este proprietatea unei surse sonore, egală cu puterea totală emisă de acea sursă în toate direcțiile.

Intensitatea(I) este definită ca putere pe unitate de suprafață purtată de o undă. *Puterea(P)* este rata la care energia este transferată de undă. Unitatea de măsură folosă pentru intensitate este $\frac{W}{m^2}$, iar formula acesteia este:

$$I = \frac{P}{A} \quad (1.5)$$

Nivelurile de intensitate ale sunetului sunt citate în decibeli (dB). Modul în care urechile noastre percep sunetul poate fi descris mai exact prin logaritmul intensității. Nivelul de intensitate(β) este definit astfel:

$$\beta(dB) = 10 \lg \left(\frac{I}{I_0} \right) \quad (1.6)$$

1.2 Reflexia sunetului

Principiile reflexiei pot fi aplicate undelor sonore, care constau din compresii și refracții. Dacă o undă sonoră se deplasează printr-un tub cilindric, în cele din urmă sunetul va ajunge la capătul tubului, acesta reprezentând granița dintre aerul din tub și aerul din afara tubului. La atingerea capătului tubului, unda sonoră va suferi o reflecție parțială(o parte din energia transportată va rămâne în tub și se va deplasa în direcția opusă) și o transmisie parțială(o parte din energia transportată va trece peste graniță, în afara tubului).

Reflexia pe suprafețe poate duce la unul dintre următoarele fenomene: ecou sau reverberație. *Ecou* este o reflexie a sunetului care ajunge la ascultător cu o întârziere față de sunetul direct. Această întârziere este direct proporțională cu distanța suprafeței reflectate de la sursă la ascultător. Undele acustice sunt reflectate de peretei sau de alte suprafețe dure, cum ar fi munții. Ecou poate fi auzit atunci când reflexia revine cu o amplitudine și o întârziere suficientă pentru a fi percepută distinct.

Reverberarea reprezintă o persistență a sunetului după ce acesta a fost produs. O reverberație este creată atunci când un sunet sau semnal provoacă multiple reflexii care

se acumulează și apoi se descompun pe măsură ce sunetul este absorbit de suprafețe. În Figura 1.4, observăm două diagrame cu axa x reprezentând timpul și cu axa y reprezentând SPL(Sound Pressure Level) ce ilustrează diferența dintre ecou și reverberație

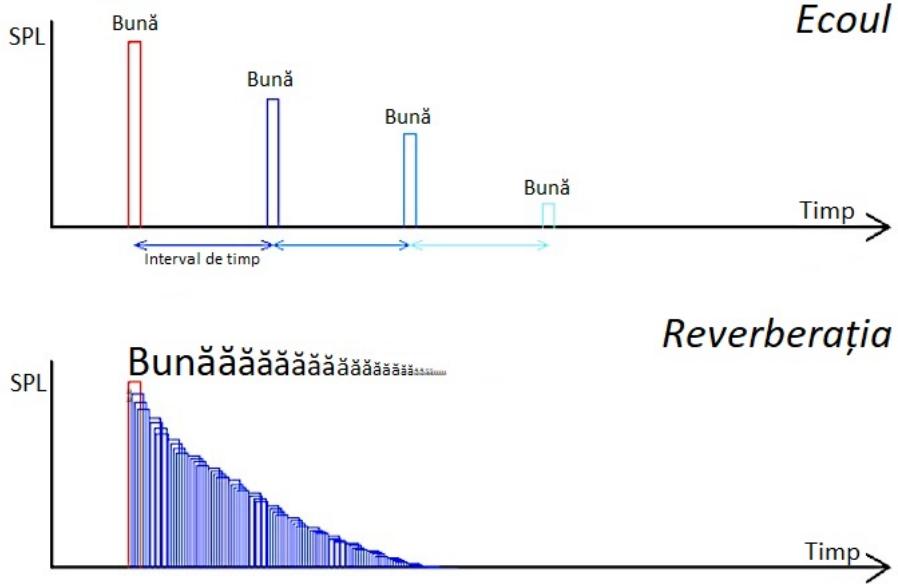


Figura 1.4: Diferența dintre ecou și reverberație

Trasarea razelor pentru calcularea unghiurilor de incidentă sau refracție este dată de legea lui Snell prin Ecuația (1.7)(vezi [9]):

$$\frac{\sin \theta_2}{\sin \theta_1} = \frac{v_2}{v_1} = \frac{n_1}{n_2} \quad (1.7)$$

unde θ_1, θ_2 sunt unghiurile de refracție, v_1, v_2 reprezintă viteza de propagare prin mediu, iar n_1, n_2 sunt indicii de refracție din mediu.

Reflexiile sunetului pe suprafețe nu urmează întotdeauna legile lui Snell. Astfel, nici o reflexie nu este perfect speculară, deci este parțial difuză, iar acest lucru se întâmplă ca o consecință pentru duritatea și dimensiunea suprafeței de coliziune. Când o rază întâlnește o suprafață difuză, se generează un număr aleatoriu în intervalul $[0, 1]$. Dacă numărul este mai mic decât un prag ales direcția razei este randomizată pentru a simula difuzia, altfel reflexia este speculară. Acest fenomen poate fi evidențiat prin Figura 1.5. În această lucrare se va folosi reflexia difuză prin randomizare normală.

1.3 Difracția și interferența sunetului

Difracția reprezintă schimbarea locală în direcția propagării undelor sonore trecând de marginea unui obstacol. Aceasta este unul dintre cele mai importante fenomene acustice cauzat de natura sunetului, fiind una dintre cele mai complexe probleme de rezolvat. Efectele difracției pot fi împărțite în trei grupe: barieră, margine și deschidere.

Fenomenul de difracție depinde semnificativ de raportul dintre lungimea de undă a sunetului și mărimea obstacolului. Cu cât lungimea de undă este mai mare, cu atât sunetul se intensifică. Când o undă sonoră întâlnește un obstacol, care este mic în raport

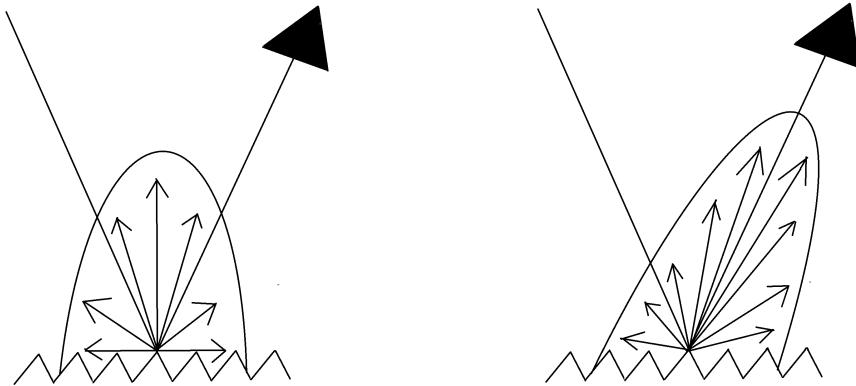


Figura 1.5: Reflexie difuză prin randomizare normală și reflexie difuză prin randomizare ponderată

cu lungimea de undă, valul trece în jurul ei ca și când nu ar exista, formând foarte puțină umbră. Dacă frecvența sunetului este suficient de mare, lungimea de undă este suficient de scurtă și ca urmare se formează o umbră vizibilă.

În Figura 1.6 se pot observa cele trei tipuri de difracții, unde situațiile din partea de sus sunt ideale pentru frecvențele joase, iar cazurile din partea de jos sunt de dorit pentru frecvențele înalte.

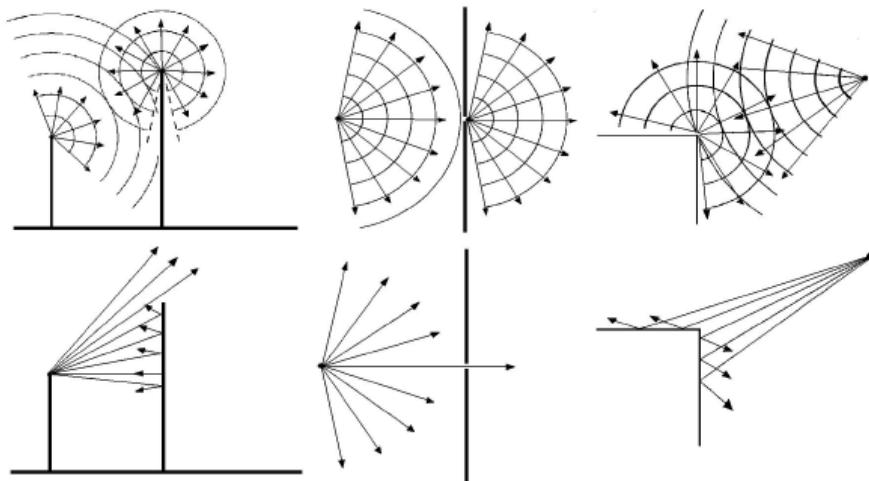


Figura 1.6: Difracție barieră, deschidere, margine[1]

Două unde care călătoresc în același mediu vor interfera una cu cealaltă. Dacă amplitudinile lor se adună, se spune că aceasta este o interferență constructivă. O interferență distructivă se realizează atunci când cele două unde sonore sunt defazate și scad.

În acest studiu, difracția și interferența nu vor fi abordate.

1.4 Funcția de răspuns la impuls

Propagarea sunetului de la o sursă audio la un receptor se caracterizează prin funcția de răspuns impuls și informațiile spațiale pe toate căile de propagare posibile. Prima detectare corespunde întotdeauna cu sunetul direct și, după acesta, primesc reflexii multiple. Întârzierile lor de timp în ceea ce privește sunetul direct sunt în funcție de lungimile căilor parcuse și intensitățile lor de presiune depind de absorbția sunetului prin aer, precum și de caracteristicile de absorbție a suprafețelor implicate în fiecare cale.

Răspunsul la impuls este funcția de ieșire a unui sistem dinamic atunci când la intrare se aplică o funcție unitară (funcția Dirac). Un impuls este un eveniment sonor foarte puternic și scurt, care este utilizat pentru testarea răspunsului la sunet într-o cameră sau pentru a testa eficiența unui sistem acustic. Un impuls conține toate frecvențele.

O funcție de răspuns la impuls se compune din: sunet direct, prima întârziere, reflecții timpurii și coada reverberantă. Figura 1.7 ilustrează componentele unei funcții de răspuns la impuls.

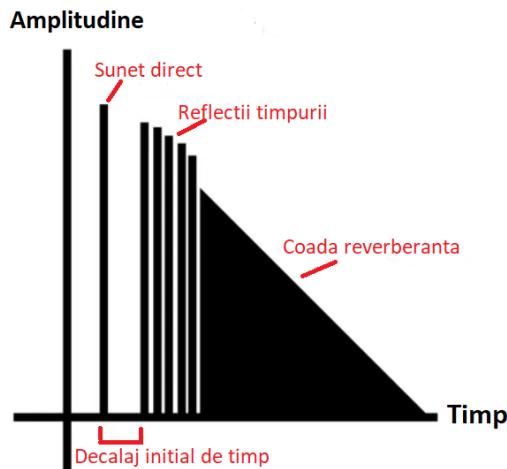


Figura 1.7: Funcția de răspuns la impuls

Sunetul direct(Direct Sound) are presiunea acustică ridicată, dar durată scurtă, reprezentând timpul necesar pentru ca sunetul să ajungă la cel primit (ex: ascultător sau microfon).

Decalaj inițial de timp(Initial Time Delay Gap) reprezintă timpul dintre sunetul direct și primele reflexii și ne spune cât de departe este sursa de sunet. Cu cât decalajul inițial de timp este mai lung, cu atât este mai apropiată sursa de sunet. Cu alte cuvinte, dacă sursa este departe, sunetul direct și primele reflexii vor fi auzite mai apropiate.

Reflecții timpurii(First Order Reflections sau Early Reflections) sunt primele pe care le auzim și se disting. Este posibil să fie doar câteva într-o cameră simplă dreptunghiulară, dar pot fi mai multe dacă camera este mai complexă. Reflecțiile timpurii indică cât de mare este o cameră.

Coada reverberantă constă în reflexii de ordin superior și nu se pot distinge între ele. Pe măsură ce numărul de reflexii crește, undele sonore pierd energie și, în cele din urmă, se descompun. Dezintegrarea reverberantă este adesea liniară atunci când este reprezentată grafic ca mai sus.

1.5 Funcția de răspuns la frecvență

Răspunsul în frecvență este măsura cantitativă a spectrului de ieșire al unui sistem sau dispozitiv ca răspuns la un stimул și este utilizat pentru a caracteriza dinamica sistemului. Este o măsură a magnitudinii și a fazei în funcție de frecvență, în comparație cu intrarea. În termeni simpli, dacă o undă sinusoidală este injectată într-un sistem la o frecvență dată, un sistem liniar va răspunde la aceeași frecvență cu o anumită magnitudine și un anumit unghi de fază relativ la intrare.

În contextul unui sistem audio, obiectivul poate fi reproducerea semnalului de intrare fără distorsiuni. Acest lucru necesită o amplitudine de răspuns uniform până la limitarea lățimii de bandă a sistemului, cu semnalul întârziat cu exact aceeași cantitate de timp la toate frecvențele.

Estimarea răspunsului în frecvență pentru un sistem fizic implică, în general, excitarea semnalului cu un semnal de intrare, măsurarea istoricelor de timp de intrare și de ieșire și compararea celor două printr-un proces precum Transformarea Fourier Rapidă(FFT).

Răspunsul în frecvență se caracterizează prin amploarea răspunsului sistemului, măsurată, de obicei, în decibeli(dB), și faza, măsurată în radiani sau grade.

1.6 Absorbția sunetului

În contextul propagării sunetului în spații închise vom considera două tipuri de absorbții: *absorbția aerului* și *absorbția suprafetelor*.

Coefficientul de absorbție în aer, depinde de temperatura atmosferică, de presiunea atmosferică și de frecvență. Absorbția sunetului în acustică este cauzată de caracteristicile de absorbție ale suprafetelor.

Astfel, absorbția este definită ca o disipare a energiei sonore la lovirea unei suprafete fizice. La fiecare reflexie, o parte α din energia sau puterea sa este absorbită. Acest factor α se numește coeficient de absorbție. Absorbția sunetului depinde de unghiul de incidentă.

Un alt mecanism de absorbție a sunetului este *absorbția suprafetelor*, fiind definit ca disiparea energiei sonore la lovirea unei suprafete fizice. La fiecare reflexie, o parte din energie sau putere este absorbită. Restul de energie care nu este absorbită este fie absorbită de material, fie transmisă mai departe ca sunet reflectat. Formula utilizată în această lucrare este evidențiată prin Ecuatia (1.8).

$$W = W(1 - \alpha) \quad (1.8)$$

unde W este energia.

1.7 Simularea binaurală

Simularea binaurală reprezintă o metodă de a realiza semnale binaurale la ambele urechi ale receptorului în spații inexistente prin intermediul unui model. În ultima perioadă, simularea acustică a devenit din ce în ce mai importantă pentru proiectarea spațiilor închise, precum sălile de concerte și de teatru.

Există o mulțime de tehnici de simulare binaurală în combinație cu metode de calcul, de cele mai multe ori tehnici geometrice pentru calcul acustic, precum: metoda sursei de imagine în oglindă, ray-tracing sau metode hibride. Cu toate acestea, genul acesta de tehnici vin la pachet cu creșterea timpului de calcul. Unul dintre cele mai dificile lucruri este directivitatea sursei.

Deși este imposibilă simularea cu exactitate a unui câmp sonor real folosind metode geometrice, lucrările realizate până în prezent pe aceste teme s-au bucurat de un mare succes, iar unele au fost chiar comercializate. O altă provocare în crearea unui astfel de algoritm este dată de coada reverberantă datorită efortului considerabil de timp. Simularea binaurală nu va fi tratată în acest studiu.

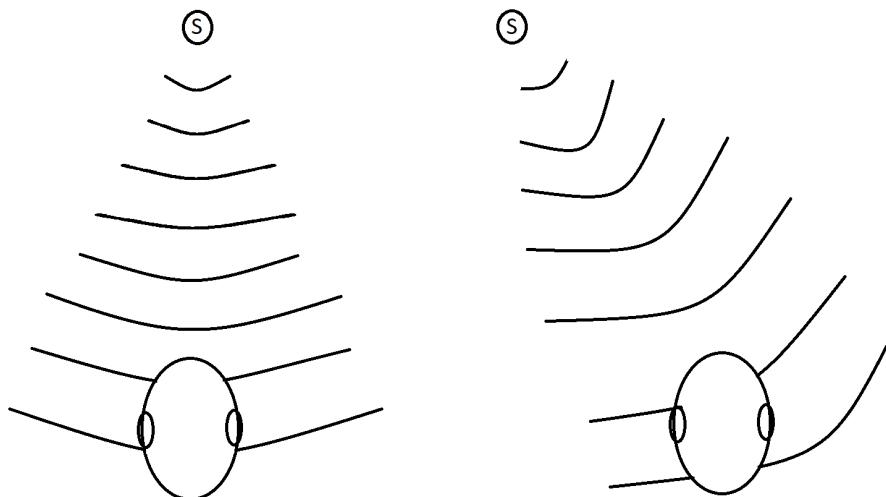


Figura 1.8: Exemplu de simulare binaurală

În Figura 1.8, se poate observa beneficiul unei simulări binaurale. În prima parte a figurii, sunetul ajunge de la sursă(S) la urechile omului în același moment și cu aceeași intensitate, pe când în cea de-a doua parte a imaginii sunetul ajunge mai devreme la urechea stângă decât la urechea dreaptă și la intensități diferite. Urechea stângă va auzi mai puternic sunetul decât urechea dreaptă.

1.8 Tehnica de trasare a razelor(Ray-Tracing)

Intensitatea emisă de o sursă este descrisă de un număr finit de raze, care vor fi considerate purtători de intensitate(sau de energie sau de putere). Aceste raze călătoresc prin spațiu la viteza sunetului și sunt reflectate după fiecare coliziune cu limitele camerei. În acest timp, intensitatea lor scade ca o consecință a absorbtiei aerului și a pereților pe care raza îi intersectează.

Dacă sursa sonoră are caracteristici omnidirectionale, direcțiile razelor sunt create prin distribuții aleatoare omogene. Calea posibilă pentru fiecare rază emisă de la sursă trebuie gasită. Raza este considerată un vector care își schimbă direcția la fiecare reflexie, fiind astfel posibilă determinarea următoarei suprafete de ciocnire.

De obicei, sursa este împărțită într-un număr mare de piese mici. Metoda deterministă este împărțirea sursei într-o manieră matematică sau geometrică. Piese din sursă ar trebui să fie identice sau cât de mult posibil identice. Punctele selectate aleatoriu pe suprafața unei sfere sursă pot fi reprezentate de trei parametrii: raza, azimutul și elevația[3]. Mai mult decât a propus Jeong Cheol-Ho în studiul său, în această lucrare împărțirea sursei în piese mai mici s-a realizat cu ajutorul sferei lui Fibonacci, algoritm ce urmează să fie descris în capitolele următoare.

Mai departe, aceste puncte reprezintă poziția de start pentru o rază. Fiecare rază poate avea nici una sau mai multe reflexii și, astfel, să fie considerată o rază directă sau indirectă. Fiecare coliziune este reținută și putem observa care a fost traseul pe care fiecare rază l-a parcurs în încăpere de la sursă până la ascultător.

Intensitatea pe care o rază o transmite receptorului este direct dependentă de distanța pe care raza a parcurs-o în încăpere. Intensitatea la un anumit moment se calculează astfel dependent de absorbția aerului, de lungimea traseului parcurs și de proprietățile de absorbție ale peretelui. Coliziunea cu suprafață poate fi considerată speculară, conform legii lui Snell. Tot acest proces trebuie repetat pentru fiecare frecvență.

În Figura 1.9 este ilustrat un mod de reprezentare pentru tehnica Ray-Tracing, unde se poate observa traiectul razelor. Linia continuă reprezintă raza directă, iar liniile punctate semnifică razele indirecte.

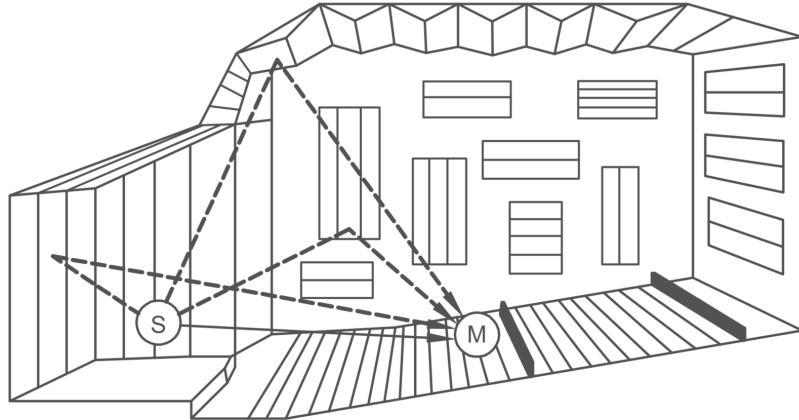


Figura 1.9: Exemplu de ray-tracing

S-a demonstrat că algoritmul de Ray-Tracing prezice nivelurile de zgomot cu o precizie foarte bună și este considerat drept unul dintre cele mai elegante metode de reprezentare a reflexiilor. Cu toate acestea, există limitări, excepții și probleme. Există câteva elemente care ar trebui luate în considerare atunci când implementăm sau evaluăm un astfel de algoritm, precum: *intervalul de frecvențe, factori dependenți de frecvență, geometria, numărul de raze*.

În mod normal, singurii factori dependenți de frecvență care pot fi inclusi sunt coeficienții de absorbție și estimarea statistică a proprietăților difuze. O ipoteză de bază în metodele care utilizează raze este că lungimea de undă corespunzătoare celei mai mici frecvențe este mai mică în comparație cu dimensiunile camerei și suprafetele acesteia.

Precum am menționat mai sus, coeficientul de absorbție α este dependent de unghi. Principalul motiv pentru evitarea măsurătorilor dependente de unghi este că acestea ar trebui să fie foarte precise. Acest lucru fiind destul de dificil, încrât presupune stocarea tuturor materialelor de construcție. Prin urmare, a fost acceptată estimarea caracteristicilor de absorbție a unei suprafete printr-o formă care este în medie peste toate unghurile de incidentă.

Erori numerice sunt introduse în rezultate prin utilizarea unui număr limitat de raze, deoarece unghiul dintre raze adiacente rămâne constant, iar această reprezentare devine treptat mai puțin exactă odată cu scăderea numărului de raze. Pentru a atinge criteriile de convergență, numărul de raze trebuie să fie cât mai mare, cu cât numărul de raze este mai mare, cu atât este mai mic unghiul solid pe care îl va acoperi o rază. Acest lucru poate afecta timpul de calcul, dar și un număr prea mic de raze poate genera probleme.

Pentru a putea înțelege mai bine traiectul unei raze vom folosi Figura 1.10. S și M reprezintă sursa și microfonul, iar linia trasată este drumul parcurs de rază până la microfon, drum care presupune 4 reflexii speculare. Modelul de propagare al sunetului ce urmează să fie prezentat în această lucrare va folosi doar reflexii speculare.

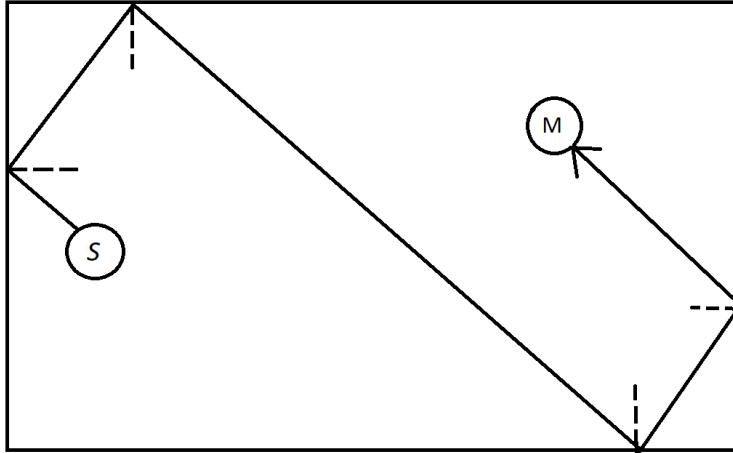


Figura 1.10: Traiectul unei raze de la sursă la microfon

Atunci când se dorește implementarea unui algoritm de acest gen, există câțiva factori foarte importanți când ne gândim la timpul de calcul, precum: numărul de raze, lungimea maximă pe care o rază o poate avea, numărul maxim de reflexii pe care îl poate avea o rază, chiar și procesorul pe care îl are mașina de pe care lucrăm poate influența semnificativ timpul de calcul.

Astfel, realizarea unui algoritm care modelează acustic spațiile închise presupune două mari etape: realizarea unui model geometric și implementarea unui model fizic. Prima etapă se ocupă de definirea dimensiunilor încăperii, așezarea sursei și a microfoanelor în spațiu, trasarea razelor și determinarea razelor intersectate cu microfoanele. A doua etapă

presupune calcularea măsurilor fizice: intensitatea, presiunea, faza, magnitudinea, timpul și funcția de răspuns la impuls.

1.9 Convoluția sunetului

Convoluția în domeniul timpului înseamnă că spectrele sunt multiplicate. Prin „multiplicarea” spectrelor înțelegem că orice frecvență care este puternică în ambele semnale va fi foarte puternică în semnalul convolut și invers orice frecvență care este slabă în ambele semnale de intrare va fi slabă în semnalul de ieșire. Convoluția implică două funcții matematice f și g care produc o a treia funcție h ce reprezintă modul în care forma uneia este modificată de către celalaltă.

Atunci când vorbim despre convoluție, sursa de sunet este numită semnal de intrare, iar fișierul de ieșire este răspunsul la impuls. Fișierul de răspuns la impuls are mereu o lungime fixă.

În practică, o aplicare relativ simplă a convoluției este locul în care avem „funcția de răspunsul la impuls” al unui spațiu. Acest lucru se obține înregistrând o scurtă explozie a unui semnal de bandă largă pe măsură ce este procesat de caracteristicile reverberante ale spațiului. Cu alte cuvinte, a fost procesat de funcția de răspuns în frecvență al spațiului similar cu modul în care ar funcționa acest proces în spațiul real. De fapt, convoluția din acest exemplu este pur și simplu o descriere matematică a ceea ce se întâmplă atunci când orice sunet este „colorat” de spațiul acustic în care apare, ceea ce este de fapt adevărat pentru toate sunetele din toate spațiile, cu excepția unei camere anecoice. Sunetul convolut va apărea, de asemenea, la aceeași distanță ca în înregistrarea originală a impulsului. Dacă convoluționăm un sunet de două ori cu același răspuns la impuls, distanța sa aparentă va fi de două ori mai mare.

Teorema de bază despre domeniul timpului și domeniul frecvenței este că multiplicarea într-un domeniu este echivalentă cu convoluția din celălalt domeniu.

În cele din urmă, există o diferență tehnică între *convoluție directă*, care este un proces foarte lent, dat fiind că fiecare eșantion din fiecare semnal trebuie să fie multiplicat cu fiecare eșantion din celălalt semnal. O variantă mai rapidă de a face acest lucru este folosirea Transformatei Rapide Fourier și folosirea Inverselor Transformatelor Rapide Fourier.

În practică, convoluția se realizează cel mai adesea prin calculul FFT sau analizele spectrale pentru fișierele de intrare și răspunsul la impuls, înmulțind spectrele lor împreună. Aceasta se numește *convoluție rapidă*.

1.10 Transformata Rapidă Fourier și Inversa Transformatelor Rapide Fourier

Transformata Fourier Rapidă(FFT) este un algoritm care calculează transformata directă(DFT) a unei secvențe sau inversa acesteia(IDFT). Analiza Fourier convertește un semnal din domeniul timpului sau al spațiului în domeniul frecvențelor și invers.

Transformata Fourier Discretă(DFT) transformă o secvență de N numere complexe $x_n = x_0, x_1, \dots, x_{N-1}$ într-o altă secvență de numere complexe $X_k = X_0, X_1, \dots, X_{N-1}$ și are următoarea formulă[2]:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn} \quad (1.9)$$

DFT este o operație foarte utilă, dar destul de costisitoare, din acest motiv a apărut FFT, care calculează rapid astfel de transformări factorizând matricea DFT într-un produs de factori rari(în mare parte zero). În acest mod, complexitatea algoritmului este redusă de la $O(N^2)$ la $O(N \lg N)$. Diferența de viteză poate fi enormă, în special pentru seturile de date lungi, unde N poate de ordinul miilor sau milioanelor. În prezența unei erori de rotunjire, mulți algoritmi FFT sunt mult mai exacti decât evaluarea definiției DFT direct sau indirect.

Reprezentarea în domeniul frecvență presupune descopunerea unui semnal acustic în semnale sinusoidale caracterizate prin frecvență, amplitudine și fază. FFT permite modificarea semnalului prin atenuare/eliminare de frecvențe - filtrare în domeniul frecvență.

Pentru a atinge performanțe există mai mulți algoritmi care calculează FFT, iar acești algoritmi, de obicei, presupun împărțirea polinomului inițial în două polinoame. Polinomul calculat poate fi caracterizat de rădăcinile complexe conjugate prin rădăcini de ordin n ale unității. Acest polinom trebuie evaluat la doar $\frac{n}{2}$ rădăcini ale unității. Una dintre condițiile necesare pentru a atinge aceste performanțe de viteze este ca n să fie o putere de-a lui 2.

În Figura 3.6, putem observa trei semnale x_1, x_2, x_3 care împreună formează semnalul x și sunt transformate cu ajutorul Transformantei Rapide Fourier pentru a compune semnalul transformat.

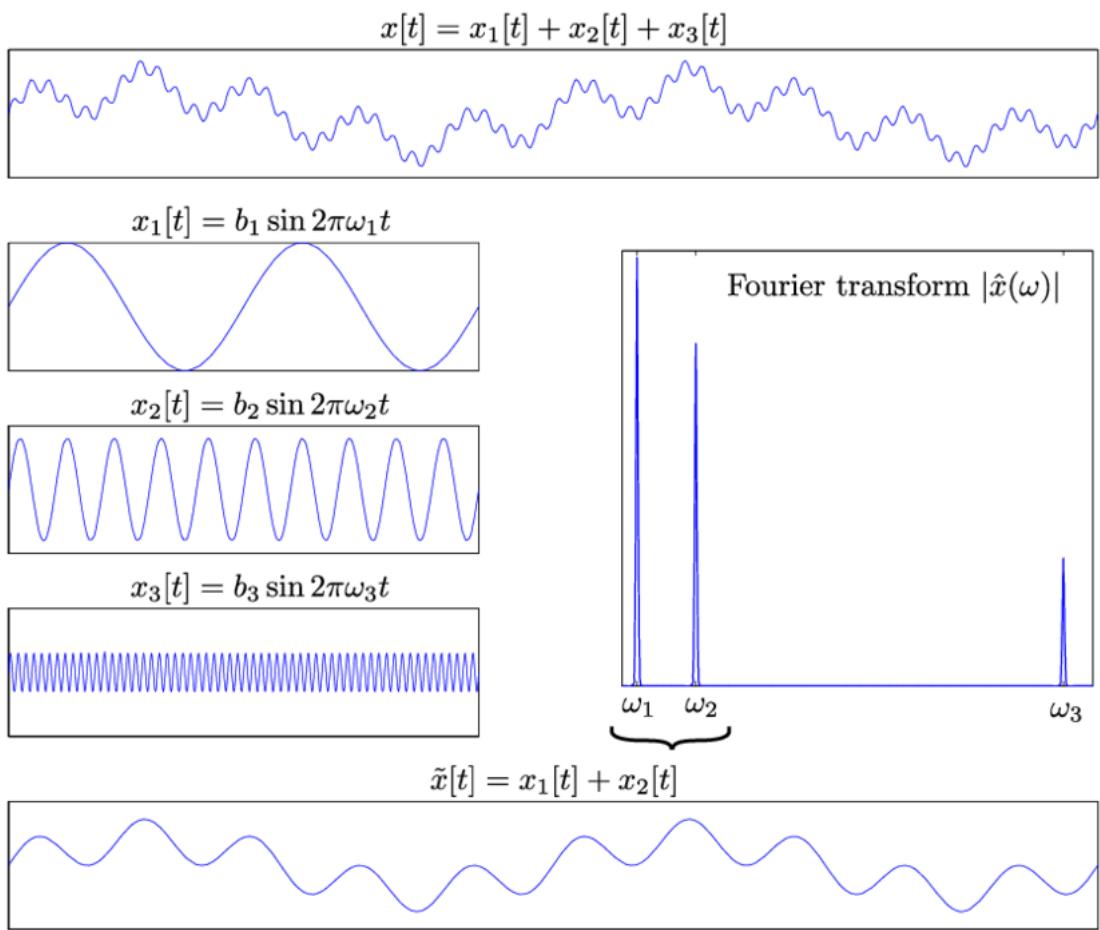


Figura 1.11: Exemplu de transformare al semnalului cu ajutorul Transformatei Rapide Fourier

Capitolul 2

Tehnologii utilizate

În zilele noastre calculatorul este folosit în multiple arii și are ca scop soluționarea sau optimizarea unor probleme. Acesta poate rezolva diferite sarcini, deoarece este programabil, adică a fost realizat pentru a putea soluționa orice cerere dată de un program.

Un program reprezintă un set de instrucțiuni pe care calculatorul le îndeplinește pentru a rezolva o problemă. Pe lângă procesul de programare în sine, se regăsesc și alte procese precum testarea, depanarea și mențenanța codului sursă care, asigură astfel o calitate superioară a codului sursă. Aplicația finală trebuie să îndeplinească o serie de proprietăți fundamentale, indiferent de limbajul de programare ales.

O parte din aceste proprietăți sunt:

- *Fiabilitatea*: proprietate care reprezintă corectitudinea programului, adică în ce măsură aplicația îndeplinește scopul pentru care a fost concepută, depinzând de factori externi precum corectitudinea algoritmilor și de cantumul de erori care pot apărea în timpul execuției programului. Mai exact, aceasta poate fi văzută ca o probabilitate.
- *Robustețea*: proprietate care definește în ce măsură programul software reacționează la evenimente mai puțin așteptate cum ar fi accesarea datelor indisponibile sau a unei zone de memorie neallocată, introducere de date eronate, etc.
- *Uzabilitate*: proprietate care vizează în mod direct utilizatorul și se referă la ușurința cu care acesta își poate rezolva problemele prin intermediul aplicației dezvoltate.
- *Portabilitate*: proprietate care definește multitudinea de platforme și sisteme de operare pe care poate rula aplicația dezvoltată. De asemenea, se referă și la ușurința cu care se poate muta codul sursă de pe o platformă pe alta.
- *Menținabilitate*: proprietate care evidențiază ușurința de a modifica aplicația, anume prin adăugare de noi funcționalități pentru a satisface noi cerințe, fixarea problemelor existente sau adaptarea codului la o versiune actuală aplicației.
- *Eficiență/Performanță*: proprietate care măsoară resursele de timp și spațiu de memorie folosite la execuția aplicației. Eficiența programului constă în minimizarea resurselor utilizate. Un alt aspect important este gestionarea corectă a memoriei și utilizarea unor algoritmi eficienți.

În continuare urmează o prezentare detaliată privind tehnologiile folosite în cadrul acestei lucrări.

2.1 Limbajul de programare C#

Un limbaj de programare este un limbaj formal care cuprinde un set de instrucțiuni care produc diferite tipuri de ieșire. Limbaje de programare sunt utilizate în programare pentru a implementa algoritmi.

Majoritatea limbajelor de programare constau în instrucțiuni pentru calculatoare. Există mașini programabile care folosesc un set de instrucțiuni specifice, mai degrabă decât limbaje de programare generale.

Limbajul de programare C# este un limbaj de programare imperativ, obiect-orientat, asemănător sintactic cu Java și C++. Acesta a fost creat de Microsoft, inițial în cadrul proiectului .NET, la sfârșitul anilor 90, fiind un concurrent al limbajului Java. Acestea sunt derivate ale limbajului C++. Limbajul C# a fost conceput ca să fie simplu, modern, să aibă un scop general și să fie orientat pe obiecte.

Acest limbaj conține două categorii de tipuri de date: *tipuri valoare* și *tipuri referință*. Prima categorie conține tipuri simple, precum: char, int, float, dar și tipurile enumerare și structură, fiind alocate pe stivă sau inline într-o structură. Cea de-a doua categorie conține tipurile interfață, delegat și tablou. Un aspect important al limbajului de programare C# este faptul că toate tipurile de date sunt derivate direct sau indirect din tipul de date System.Object[7].

Astfel, se pot crea aplicații web prin intermediul ASP.NET, aplicații desktop prin WPF(Windows Presentation Foundation) sau aplicații mobile pe Windows Phone. Common Language Runtime(CLR) gestionează execuția programelor .NET, fiind un Virtual Machine(VM) care rulează Intermediate Language(IL) și oferă multiple servicii, precum gestionarea memoriei, securitate, gestionare de excepții, garbage collector, dar și altele.

În cadrul acestui studiu, limbajul de programare a fost folosit pentru a realiza implementarea modelului acustic, iar rezultatele acestuia au fost ilustrate cu ajutorul platformei Unity, dar și cu ajutorul limbajului de programare Python.

2.2 Limbajul de programare Python și biblioteca matplotlib

Python este un limbaj de programare interpretat, obiect orientat, de nivel înalt, cu semantică dinamică. Structurile sale de date încorporează la nivel înalt, combinate cu tastarea dinamică și legarea dinamică, îl fac foarte atractiv pentru dezvoltarea rapidă a aplicațiilor, precum și pentru a fi utilizat ca limbaj de scriptare sau lipici pentru a conecta componente existente împreună. Sintaxa simplă, ușor de învățat accentuează lizibilitatea și, prin urmare, reduce costul întreținerii programului. Python acceptă module și pachete, ceea ce încurajează modularitatea programului și reutilizarea codului.

Acest studiu folosește biblioteca matplotlib pentru a reprezenta rezultate comparative pentru funcția de răspuns în timp sau sunetul după convoluție pe mai multe microfoane.

Aceasta este o bibliotecă utilă pentru crearea de vizualizări statice, animate și interactive în Python. Matplotlib poate fi utilizat în scripturi Python, shell-urile Python și IPython, servere de aplicații web și diverse seturi de instrumente grafice de interfață cu utilizatorul[13].

2.3 Platforma Unity

O platformă(IDE- Integrated Development Environment) este o aplicație software care oferă facilități programatorilor pentru dezvoltarea de soft. O platformă este alcătuită din cel puțin un editor de cod sursă, instrumente de automatizare a construcțiilor și un depanator.

Un singur program în care se realizează dezvoltarea unui soft reprezintă o platformă(IDE). Această platformă oferă multe caracteristici pentru autorizare, modificare, compilare, implementare și depanare a produsului soft. Unele platforme sunt specializate pe un limbaj de programare specific, oferind un set de caracteristici care se potrivesc cu paradigma de programare a aceluiași limbaj. Cu toate acestea, există multe IDE-uri care suportă mai multe limbi de programare.

Platforma Unity este folosită în general pentru a crea jocuri și poate rula pe mai multe platforme. A fost dezvoltată de Unity Technologies și lansată în 2005 la Apple Inc's Worldwide Developers Conference ca fiind un game engine exclusiv pentru macOS.

Unity este principala platformă mondială pentru crearea și operarea de conținut 3D interactiv, în timp real, oferind instrumente pentru a crea jocuri și pentru a le publica pe o gamă largă de dispozitive. Platforma de bază Unity permite echipelor creative să fie mai productive împreună.

De-a lungul anilor, această platformă s-a dezvoltat reușind astăzi să susțină peste 25 de platforme. Platforma poate fi utilizată pentru a crea jocuri 2D și 3D, realitate virtuală și realitate augmentată. Unity este utilizat nu numai pentru jocuri video, cât și în domeniul filmelor, arhitecturii, ingineriei și construcțiilor[6].

2.4 Platforma Blender

Blender este o aplicație software de grafică 3D folosită pentru crearea de informații, efecte vizuale, artă, modele de tipărire 3D, grafică de mișcare, aplicații 3D interactive și jocuri pe calculator. Aceasta este cross-platform și permite mapări UV, folosirea materialelor, a shaderelor, a mesh-urilor fără probleme.

În acest studiu, programul a fost folosit pentru a putea realiza încăperea sferică din aplicație, întrucât era nevoie de un soft ce trebuia să permită crearea unei fișiere de tipul „obj” care să conțină o sferă cu normalele inversate pentru a putea plasa alte elemente înăuntrul acesteia.

2.5 Biblioteca NWaves, XCharts și StandaloneFile-Browser

Biblioteca NWaves a fost inițial destinată cercetării, vizualizării și predării elementelor de bază ale programării DSP(Digital Signal Processing) și a sunetului. Toți algoritmii sunt implementați în limbajul de programare C#, cât mai simplu posibil și au fost proiectați în principal pentru procesarea offline(în prezent există și multe metode online)[14].

Această bibliotecă este una open source destinată procesării digitale a semnalelor audio, înglobând multiple funcționalități pentru acest domeniu. În cadrul studiului, biblioteca a fost folosită pentru citirea/creearea fișierelor cu extensia „*wav*”, dar și pentru convoluția sunetelor.

XCharts este o bibliotecă puternică, ușor de utilizat și de configurat pentru vizualizarea datelor în contextul platformei Unity. Acest proiect a fost dezvoltat sub Unity 2017 și .NET 3.5[16].

Diferite componente și date pot fi combinate în diferite tipuri de diagrame. O componentă XCharts este împărțită într-o componentă principală și în una sau mai multe componente secundare, unde cea principală conține toate componentele secundare.

Biblioteca a fost folosită în contextul acestei lucrări pentru a ilustra date statistice privind sunetul, precum: funcția de răspuns la impuls în frecvență și în timp, magnitudinea, faza, timpul, dar și altele.

Biblioteca StandaloneFileBrowser este una cross-platform ce permite lucrul cu fișiere pentru platforma Unity[15]. Aceasta a fost folosită în realizarea aplicației pentru a permite deschiderea și utilizarea fișierelor „*wav*” pentru a putea crea input-ul pentru modelul acustic.

Capitolul 3

Pașii elementari pentru implementarea aplicației

Acest capitol va descrie detaliat algoritmul de modelare acustică în spații interioare prin metoda Ray-Tracing și va conține pașii elementari pentru implementarea acestuia. Înținând cont de toate acestea, aplicația poate fi împărțită în patru etape:

- Calcul geometric
 - Sfera lui Fibonacci
 - Propagarea razelor
 - Selectia razelor
- Calcul fizic
 - Calculul intensităților
 - Calculul presiunilor
 - Funcția de răspuns la frecvență și funcția de răspuns la impuls
 - Calculul distanțelor
 - Calculul timpilor
- Post-procesare
 - Convoluția sunetului
- GUI

Astfel, pentru a realiza acest algoritm trebuie să încep prin a distribui uniform razele plecând de la sursă, după care trebuie să selectăm razele care intersectează microfoanele din încăpere. Pentru a obține performanță este necesar să excludem acele raze care nu ajung pe nici unul dintre microfoane sau acele raze care intersectează microfoanele, dar sunt foarte asemănătoare și să pastrăm doar una dintre acestea.

Pentru pasul următor calculăm intensitățile pentru fiecare rază pe care mai apoi le transformăm în presiuni. Următoarea etapă presupune calculul frecvențelor și calcularea fazelor. După acești pași mai este nevoie doar să calculăm distanțele și timpii pentru ca mai departe să obținem funcția de răspuns la impuls și funcția de răspuns la frecvență.

Acum că am considerat toate aceste etape mai râmâne doar să putem auzi rezultatul algoritmului nostru. Astfel, cu ajutorul convoluției ajungem să ascultăm soluția algoritmului, iar modelul nostru acustic este complet.

3.1 Prezentarea aplicației

Acustica face parte din domeniul fizicii și se ocupă cu studiul undelor mecanice în gaze, lichide, solide și este prezent în toate aspectele societății actuale, însă predominant în domeniul industriei pentru controlul zgomotului. Această ramură implică și studiul sunetelor, vibrațiilor, ultrasunetelor și infrasunetelor. Un model acustic va implica, în cel mai generic mod, simularea căilor pe care sunetul le parcurge de la sursă la destinație. Cel mai adesea, aceste modele presupun rezolvarea integralei Helmholtz-Kirchoff[5] folosind diverse abordări de calcul, precum: soluții numerice ale ecuației de undă, aproximări de înaltă frecvență ale ecuației de undă, dar și modele statistice. Modelul pe care îl propune acest studiu face parte din cea de-a doua categorie. Modelul acustic va conține etapa de calcul geometric, etapa de calcul fizic, convoluția sunetului și realizarea unei interfețe ce permite validarea acestor rezultate.

Tinând cont de toate aceste elemente este foarte important să reținem că aceste etape se află într-o legătură strânsă și se bazează unele pe celelalte. Astfel, ne putem da seama că calculul geometric este o etapă deosebit de importantă, întrucât modul în care distribuim razele în încăpere și selecția acestora dictează performanța modelului și optimalitatea acestuia. Algoritmii de post-procesare sunt folosiți, de obicei, pentru a suprima zgomotul sau orice artefact creat în cadrul primelor două etape, adică calculul geometric și cel fizic, concentrându-se pe eliminarea distorsiunii și a ecoului.



Figura 3.1: Meniul pentru setarea configurației

În Figura 3.1 putem observa meniul care ne ajută să setăm valorile pentru care dorim să vedem ce se întâmplă într-o încăpere cu sunetul. Prin intermediul acestei ferestre putem stabili care este numărul maxim de reflexii permis pentru o rază, numărul de raze ce se vor împrăștia în cameră, distanța maximă permisă pentru o rază, pasul de frecvență și sunetul ce va fi difuzat în încăpere. De asemenea, putem observa că avem la dispoziție încă 3 ferestre ce permit vizualizarea rezultatelor.



Figura 3.2: Meniul pentru vizualizarea razelor

Aplicația permite atât vizualizarea tuturor razelor din încăpere folosind butonul „Show all”, cât și vizualizarea unei raze individual prin intermediul butonului „Show”, aceste lucruri fiind ilustrate în Figura 3.2.

Pentru acest studiu au fost gândite trei încăperi: două dreptunghiulare și una sferică. Prima încăpere dreptunghiulară are dimensiunea de 3x5m, iar cea de-a doua are 30x30m. Încăperea sferică are diametrul 5m. Acestea pot fi vizualizate în Figura 3.3. Pentru încăperile dreptunghiulare sursa a fost poziționată în centru, iar pentru încăperea sferică sursa a fost poziționată pe o poziție astfel încât să aparțină unui cerc pe care vor fi plasate microfoanele.

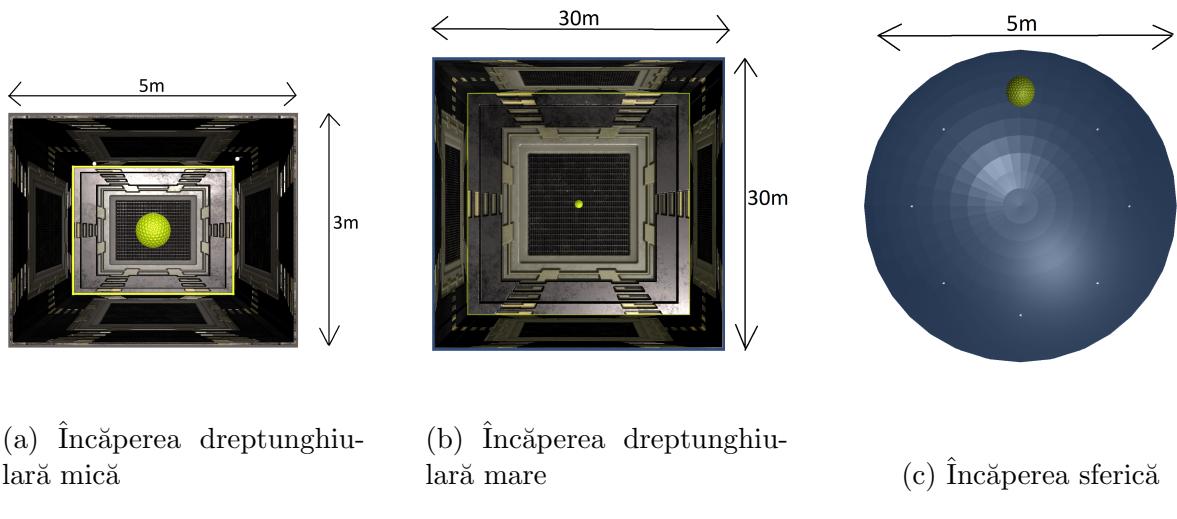


Figura 3.3: Încăperile propuse de către aplicație

3.2 Calcul geometric

Calculul geometric presupune simularea geometriei unei încăperi. Pentru acest lucru vom considera ca date de intrare suprafețele încăperii, sursa audio și microfoanele plasate în încăpere. Cu ajutorul acestora vom putea distribui uniform raze în încăpere folosind algoritmul sferei lui Fibonacci și vom păstra doar acele raze de care avem nevoie cu ajutorul unui algoritm pentru reducerea razelor dupicate. Aceste subetape urmează a fi prezentate în următoarele pagini.

3.2.1 Sfera lui Fibonacci

De-a lungul timpului, în literatura matematică, au existat multiple încercări pentru soluționarea problemei distribuirii uniforme a punctelor pe o sferă. Din păcate, cu excepția câtorva cazuri speciale, nu este posibil să se distribuie în mod egal punctele pe suprafața unei sfere.

Astfel, vom prezenta modul în care vor fi distribuite razele pe sursa audio în contextul algoritmului nostru. Sursa va fi reprezentată în modelul propus de o sferă pe care se vor alege puncte uniform distribuite din care vor porni razele.

Chiar dacă seturile de puncte Fibonacci nu reprezintă cea mai bună distribuție globală a punctelor pe o sferă, ele oferă proprietăți excelente de eșantionare și sunt extrem de simple de construit față de alte modele.

Una dintre cele mai mari provocări pentru acest algoritm este determinat de faptul că distribuția optimă depinde în mod critic de funcția obiectiv pe care o utilizăm, în cazul nostru este dependentă de N , numărul de puncte pe care vrem să îl distribuim pe sferă.

În Figura 3.4, se poate observa impactul pe care N îl are în cadrul algoritmului. Cu cât N este mai mare, cu atât punctele pe sferă sunt mai bine distribuite.

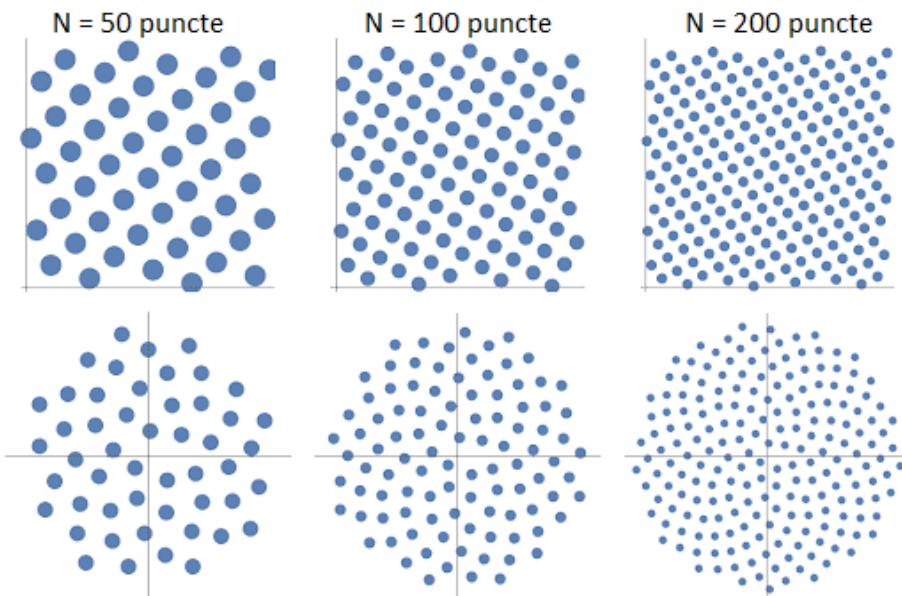


Figura 3.4: Rezultate pentru algoritmul sferei lui Fibonacci[10]

Acest algoritm se bazează pe o deplasare în spirală pe suprafața sferei incrementală cu unghiul de aur, care este legat de raportul de aur. Două cantități, a și b , se află în raportul de aur dacă: $\frac{a}{b} = \frac{a+b}{a} = \varphi$, unde $a > b$, iar acest raport este aproximativ egal cu $\varphi = \frac{1 + \sqrt{5}}{2}$. Unghiul de aur, ϑ , este definit în funcție de raportul de aur astfel: $\vartheta = 2\pi(2 - \varphi)$.

Conform Algoritmului 1, vom distribui razele uniform pe suprafața sferei, urmând ca mai departe să generăm geometria razelor.

Algorithm 1 Sfera lui Fibonacci

```
1: procedure generare(s)
2:    $\varphi \leftarrow \frac{1 + \sqrt{5}}{2}$ 
3:    $\vartheta \leftarrow 2\pi(2 - \varphi)$ 
4:   for  $i \leftarrow 0$  to  $N$  do
5:      $\theta \leftarrow \sin\left(-1 + \frac{2i}{N + 1}\right)$ 
6:      $\phi \leftarrow \vartheta i$ 
7:      $dir \leftarrow$  transformăm  $(1, \theta \text{ și } \phi)$  în coordonate carteziene
8:     Serie dir, direcția pe care se va deplasa viitoarea rază
9:   end for
10: end procedure
```

3.2.2 Propagarea razelor

După ce am stabilit la pasul anterior punctul de start pentru fiecare rază ne vom preocupa de modul în care vor fi generate acestea. Pentru acest lucru este nevoie să ținem cont de câteva aspecte precum: legea reflexiei și cea a refracției, care este distanța maximă pe care o poate acoperi o rază, care este numărul maxim de reflexii pe care îl dorim pentru modelul nostru.

Pentru a calcula lungimea unei raze vom folosi distanța Euclidiană:

$$d(p, q) = \sqrt{(p - q)^2} \quad (3.1)$$

În cazul în care nu am ține cont de lungimea razei am putea ajunge în situația în care am crea o rază care are mai puține reflexii decât numărul maxim de reflexii, dar care s-ar propaga la infinit pentru că nu ar mai întâlni o suprafață în care să se reflecte.

Ca să putem crea o rază trebuie să stabilim de la bun început și care este numărul maxim de reflexii pe care dorim să îl permitem. Acest parametru este ales dependent de problema pe care vrem să o rezolvăm, dacă dorim să ajungem la o anumită performanță sau dorim să obținem o soluție cât mai apropiată de realitate.

Pentru a rezolva problema propagării razelor am folosit următoarea metodă pusă la dispoziție de platforma Unity: *public static bool Raycast(Vector3 origin, Vector3 direction, float maxDistance = Mathf.Infinity, int layerMask = DefaultRaycastLayers, QueryTriggerInteraction queryTriggerInteraction = QueryTriggerInteraction.UseGlobal)*, unde *origin* este punctul din care pornește raza, *direction* este direcția pe care se deplasează raza, *maxDistance* este lungimea maximă pe care o rază o poate avea, *layerMask* este o mască care este utilizată pentru a ignora selectiv collider-urile atunci când este aruncată o rază, iar parametrul *queryTriggerInteraction* specifică dacă această interogare trebuie să atingă Triggerele, pentru a calcula direcția pe care urmează să se deplaseze raza[11].

Numărul maxim de reflexii influențează în mod direct atât complexitatea, cât și performanța algoritmului. Dacă vom alege un număr foarte mare de reflexii, acesta va duce la creșterea timpului de calcul. O rază va parcurge o suprafață mare a camerei atunci când considerăm un număr mare de reflexii. În schimb, dacă alegem un număr prea mic de raze, vom întâmpina probleme, pentru că putem fi puși în situația în care

nici una dintre raze nu a ajuns pe microfon, ceea ce va avea un impact negativ asupra rezultatului pe care îl obținem.

3.2.3 Selectia razelor

Unul dintre pașii esențiali pentru a obține performanță este reducerea numărului de raze, păstrându-le doar pe acele care ajung pe unul dintre microfoanele plasate în încăpere. Mai mult de atât, se pot face îmbunătățiri prin eliminarea duplicatelor. Duplicatele sunt acele raze care difeă printr-un prag definit, iar păstrarea acestora crește complexitatea algoritmului, fără a aduce valoare.

Pentru a selecta doar acele raze care intersectează microfonul am folosit ecuația sferei[12]:

$$(x - x_{centru})^2 + (y - y_{centru})^2 + (z - z_{centru})^2 = r^2 \quad (3.2)$$

și a liniei:

$$\begin{cases} x = x_1 + x_2 - x_1 \\ y = y_1 + y_2 - y_1 \\ z = z_1 + z_2 - z_1 \end{cases} \quad (3.3)$$

Substituind x, y, z din ecuația sferei obținem:

$$\begin{cases} a = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \\ b = -2[(x_2 - x_1)(x_{centru} - x_1) + (y_2 - y_1)(y_{centru} - y_1) + (z_2 - z_1)(z_{centru} - z_1)] \\ c = (x_{centru} - x_1)^2 + (y_{centru} - y_1)^2 + (z_{centru} - z_1)^2 - r^2 \end{cases} \quad (3.4)$$

iar pentru a verifica condiția de intersecție avem următoarele situații:

$$\begin{cases} \text{sunt tangente} & b^2 - 4ac = 0 \\ \text{se intersectează} & b^2 - 4ac > 0 \\ \text{nu se intersectează} & b^2 - 4ac < 0 \end{cases} \quad (3.5)$$

Două raze sunt duplicate dacă următoarele condiții sunt adevărate:

1. cele două raze trebuie să aibă același număr de puncte de coliziune;
2. diferența absolută dintre lungimile celor două raze nu trebuie să depășească un prag ales; pragul pe care l-am folosit pentru algoritmul propus a fost $\epsilon = 10^{-2}$;

În Algoritm 2 am prezentat modul în care a fost realizată reducerea de duplicate după ce am considerat doar acele raze care ajung pe microfon. Până la acest pas am stabilit cum vom distribui razele în orice încăpere, am stabilit care va fi geometria camerei și am păstrat doar acele informații de interes, adică acele raze care ajung pe microfoanele din încăpere, dar fără duplicate.

Astfel, descrierile următoare prezintă într-un mod detaliat modelul fizic și matematic pentru algoritmul nostru.

Algorithm 2 Reducerea duplițelor

```
procedure reducere_duplicate(raze)
2:   index = 0
    while index < no(raze) do
4:      if razeindex și razeindex+1 sunt raze directe then
            eliminăm razeindex din raze
6:      else if |distanță(razeindex) - distanță(razeindex+1)| < ε and
            no(razeindex.puncte_coliziune) = no(razeindex+1.puncte_coliziune) then
8:          if razeindex.puncte_coliziune = razeindex+1.puncte_coliziune then
              eliminăm razeindex din raze
10:         index ← index + 1
            end if
12:        else index ← index + 1
            end if
14:    end while
end procedure
```

3.3 Calcul fizic

Am ajuns astfel la etapa de mijloc a aplicației, ce propune calcularea câtorva elemente pentru a putea simula fizica unei încăperi. La pasul anterior, am realizat geometria unui spațiu interior, iar mai departe este nevoie să calculăm intensitățile, presiunile, distanțele, timpii și funcțiile de răspuns în frecvență. Aceste detalii urmează să fie prezentate mai jos.

3.3.1 Trecerea în domeniul frecvențelor

Trecerea valorilor din domeniul timpului în domeniul frecvențelor este un pas esențial în realizarea algoritmului pentru modelarea acustică a spațiilor închise, întrucât ajută la simplificarea calculelor, reducând astfel timpul computațional pe care îl necesită rularea algoritmului. Acest procedeu este realizat cu ajutorul Transformatei Rapide Fourier.

3.3.2 Calculul intensităților

Pentru a calcula intensitățile pentru fiecare rază am folosit Legea Pătratului Invers care afirmă ca o mărime fizică specificată este invers proporțională cu pătratul distanței de la sursa acelei mărimi fizice. Cauza fundamentală pentru aceasta poate fi înțeleasă ca diluție geometrică corespunzătoare radiației punct-sursă în spațiul 3D.

Pentru a preveni diluarea energiei în timpul propagării unui semnal, pot fi utilizate anumite metode, cum ar fi un ghid de undă, care acționează ca un canal pentru apă sau modul în care un butoi de pistol restricționează expansiunea gazului fierbinte la o dimensiune pentru a preveni pierderea transferului de energie către un glonț.

Legea pătratului invers se aplică în general atunci când o anumită forță, energie sau altă cantitate conservată este uniform radiată spre exterior dintr-o sursă punctuală în spațiul 3D. Deoarece suprafața unei sfere este proporțională cu pătratul razei, pe măsură ce radiația emisă se îndepărtează de sursă, aceasta se întinde pe o zonă care

crește proporțional cu pătratul distanței de la sursă. Prin urmare, intensitatea radiației care trece prin orice zonă unitară (direct orientată spre sursa punctuală) este invers proporțională cu pătratul distanței de la sursa punctuală.

Astfel, formula intensității este:

$$\frac{I_n}{I_{n+1}} = \frac{d_{n+1}}{d_n} \quad (3.6)$$

unde d_n reprezintă distanța de la sursă până la punctul al n -lea.

În cazul algoritmului nostru, atunci când o rază parcurge încăperea aceasta se lovește de diferite materiale ce au un factor de absorbție α care variază în funcție de material. Pentru a putea ține cont de fenomenul de absorbție am inclus acest factor în formula intensității:

$$\frac{I_n}{I_{n+1}} = \frac{d_{n+1}}{d_n} (1 - \alpha_k)^2 \quad (3.7)$$

unde d_n are aceeași semnificație ca la pasul anterior, iar α_k reprezintă coeficientul de absorbție pentru al k -lea material.

3.3.3 Calculul presiunilor

Pentru a putea calcula presiunile, ne vom folosi de intensitățile calculate la pasul anterior, ținând cont de densitatea aerului și de viteza sunetului prin aer. Vom considera că densitatea aerului, ρ_{aer} , are valoarea $1.2041 \frac{kg}{m^3}$, iar viteza sunetului prin aer, c_{aer} , este $343.21 \frac{m}{s}$, pentru o temperatură constantă de $20^\circ C$.

Prin urmare, formula de transformare a intensității(I) în presiune(p) este:

$$p = \sqrt{2 \cdot I \cdot \rho_{aer} \cdot c_{aer}} \quad (3.8)$$

Nivelul puterii sonore cuantifică energia sonoră total radiată de la un obiect. Spre deosebire de presiunea sonoră, intensitatea este independentă de distanță față de sursa de sunet, zona încadrătoare și alte influențe.

3.3.4 Funcția de răspuns în frecvență și funcția de răspuns la impuls

Pentru fiecare frecvență din interval vom calcula fază și magnitudinea:

$$\begin{cases} t = \frac{c_{aer}}{fr} \\ w = \frac{2\pi}{t} \\ \theta = \arctan \frac{-\sin w \cdot d}{\cos w \cdot d} \end{cases} \quad (3.9)$$

unde fr este frecvența dată, θ este faza pe care dorim să o calculăm, iar d este lungimea razei. Ca să obținem magnitudinea ne vom folosi de presiunea calculată anterior.

În urma acestui calcul obținem faza și magnitudinea în coordonate polare și dorim să le transformăm în coordonate carteziene. Astfel, vom obține pentru fiecare frecvență un număr complex de magnitudine m și fază θ . Lista de numere complexă obținută pentru fiecare microfon o vom numi *ecogramă*.

În Figura 3.5 putem observa care sunt etapele algoritmului. Pentru a putea ajunge la conoluția sunetului trebuie să trimitem ca input valori în domeniul timpului. Obținerea acestor valori se realizează prin aplicarea Inversei Transformatei Rapide Fourier peste funcția de răspuns în frecvență.

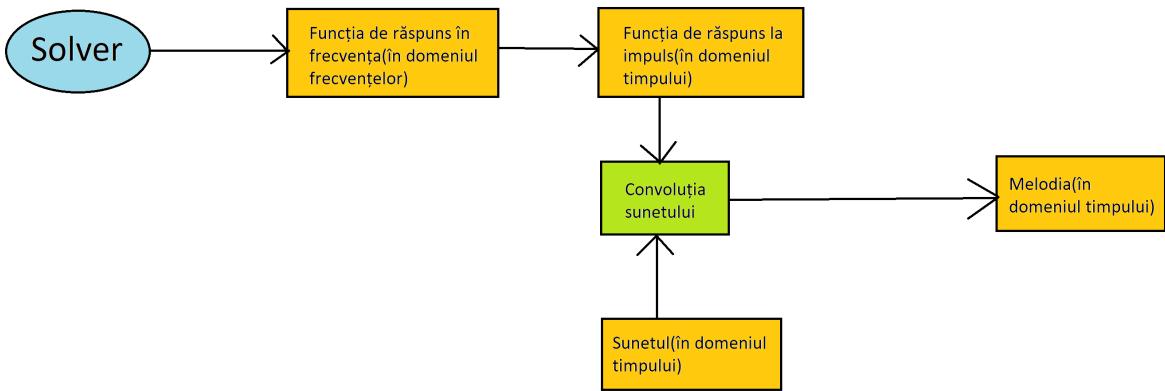


Figura 3.5: Etapele parcuse pentru obținerea sunetului pe microfon

Funcția de răspuns la impuls se calculează pe baza funcției de răspuns în frecvență cu ajutorul Inversei Transformatei Rapide Fourier care va permite trecerea de la domeniul frecvențelor la domeniul timpului pentru a pregăti semnalul de intrare ca mai departe să realizăm conoluția sunetului. Ca să putem trece mai departe și să calculăm funcția de răspuns la impuls trebuie să utilizăm Algoritm 3.

Acum că am obținut răspunsul în frecvență putem trece la a calcula răspunsul la impuls, iar acest lucru se poate face cu ajutorul IFFT, care va permite trecerea de la domeniul frecvențelor înapoi la domeniul timpului.

3.3.5 Calculul distanțelor

Pentru a calcula distanțele am folosit formula distanței lui Euler. Am considerat pentru fiecare rază, toate punctele de coliziune și am calculat lungimile pentru acestea iterativ.

$$d(p_1, p_2) = \sqrt{(p_{1x} - p_{2x})^2 + (p_{1y} - p_{2y})^2 + (p_{1z} - p_{2z})^2} \quad (3.10)$$

Algorithm 3 Crearea funcției de răspuns îm frecvență

```
procedure răspuns_frecvență(ecograme, microfoane, fr)
    răspunsFr ← dicționar gol
3:   for i ← 0 to microfoane do
        valori ← listă numere complexe goală
        for j ← 0 to fr do
            sum ←  $\sum$  ecograme[fr[j]][microfoane[i]]
            valori.adaugă(sum)
        end for
9:   răspunsFr[microfoane[i]] ← valori
    end for
end procedure
```

Prin urmare, lungimea unei raze este calculată astfel:

$$d_{razei} = \sum_1^{N-1} d(p_i, p_{i+1}) \quad (3.11)$$

unde p_1, p_2, \dots, p_N sunt punctele de coliziune, iar N reprezintă numărul de puncte de coliziune.

3.3.6 Calculul timpilor

Calculul timpilor implică calculul lungimilor pentru fiecare rază și viteza sunetului, folosind următoare formulă:

$$t = \frac{d}{c_{aer}} \quad (3.12)$$

unde t este timpul, d reprezintă distanța de la sursă până la ultimul punct de coliziune al razei, iar c_{aer} este viteza sunetului prin aer.

3.4 Post-procesarea

Am ajuns și la ultima etapă unde vom aplica operații de post-procesare pentru a putea asculta sunetul pe microfoanele plasate în încăpere. Astfel, la acest pas vom aplica un algoritm pentru convoluția sunetului ce va fi descris în paginile ce urmează.

3.4.1 Convoluția sunetului

Convoluția sunetului se face pe baza funcției de răspuns la impuls. Răspunsul este alcătuit dintr-o listă de valori reale care sunt transformate într-un semnal discret. Convoluția presupune un sistem ce primește ca intrare un semnal și pe care îl transformă pentru a obține un semnal de ieșire. În cazul acestui model, semnalul de intrare este funcția de răspuns la impuls și sunetul în domeniul timpului, iar ieșirea este reprezentată de valorile reale ale sunetului.

Teoria Fourier spune că orice semnal digital poate fi exprimat ca o combinație de sinusoide, din acest motiv preferăm să realizăm calculele în domeniul frecvențelor (este mult mai rapid).

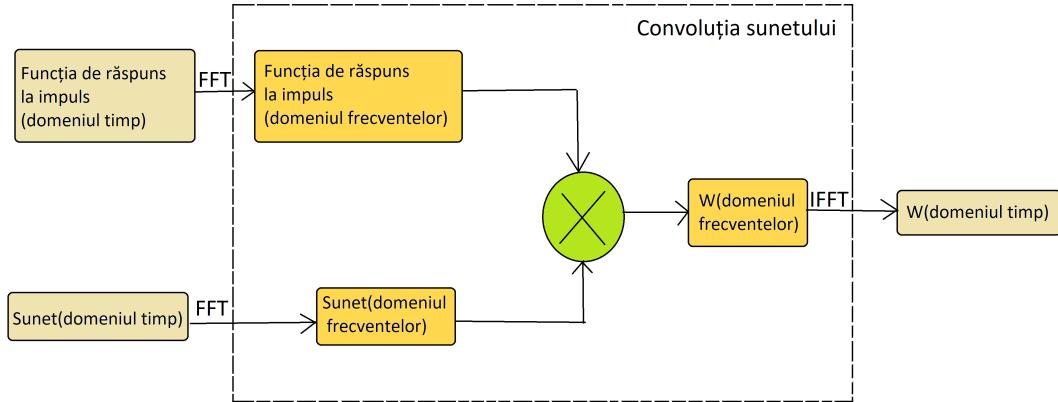


Figura 3.6: Etapele parcuse pentru convoluția sunetului

Precum este descris și în Figura 3.6, convoluția sunetului preia ca intrare funcția de răspuns la impuls și sunetul care a fost pornit pe sursa audio. Aceste valori sunt în domeniul timpului și pentru a putea să le aplicăm transformările trebuie să trecem în domeniul frecvențelor. Obținem astfel două polinoame ce trebuie înmulțite. Răspunsul obținut este în domeniul frecvențelor și pentru a calcula ieșirea trebuie să trecem înapoi în domeniul timpului cu ajutor Inversei Transformatei Rapide Fourier. Conoluția sunetului, transformările din domeniul timpului în domeniul frecvențelor și invers au fost realizate cu ajutorul metodelor din NWaves: *Convolve*, *RealFft* și *Inverse*.

3.5 GUI

Interfața aplicației a fost realizată cu ajutorul platformei Unity. Încăperile dreptunghiulare au fost construite folosind forme 3D puse la dispoziție de către această platformă, iar încăperea sferică a fost creată în Blender. Pentru cea din urmă cameră a fost folosit un obiect de tip sferă pentru care au fost inversate normalele. Fiecare spațiu folosește culori și materiale potrivite utilizând Unity Material, o clasă care permite utilizatorului să animeze un obiect prin intermediul proprietăților de care dispune această clasă.

Meniurile aplicației au fost realizate tot cu ajutorul platformei Unity folosind elemente de UI precum: Canvas, Panel, Button, Dropdown, Label, Input, Text, dar și altele. Canvas-ul este un element ce conține o arie în interiorul căreia ar trebui să se afle toate celelalte elemente de UI, iar un Panel este utilizat pentru a susține mai multe elemente precum Button, Label, Input, Dropdown, Text etc. De asemenea, pentru butoane au fost alese imagini intuitive care să ușureze folosirea acestora, dar și un font adecvat pentru text astfel încât să fie lizibil pentru cititor.

Mai mult de atât, pentru a realiza graficele ce definesc o parte din rezultatele solver-ului a fost folosită biblioteca XCharts. Aceasta a fost introdusă în aplicație pentru a crea un mod de a vizualiza funcția de răspuns la impuls, funcția de răspuns în frecvență, dar și ecograma de timp.

Biblioteca StandaloneFileBrowser a fost folosită în interfața aplicației pentru a permite utilizatorului să încarce un fișier cu extensia „wav” pentru a putea fi folosit mai departe drept parametru pentru modelul acustic într-un mod cât mai intuitiv.

Capitolul 4

Testarea și analizarea rezultatelor

4.1 Experimente realizate

aici as vrea sa prezint experimentele aceleia din incaperi:

-i cum arata camerele cand se trag un nr de raze

-i cum arata incaperea sferica atunci cand punem si un obstacol in fata unui microfon

-i cateva concluzii legat de aceste aspecte

4.2 Comparații

aici as vrea sa prezint comparatii legate de valorile obtinute de mine in incapere vs valorile obtinute in simcenter, dar nu sunt f sigura ca acest lucru va merge inca

as vrea sa spun macar ca aplicatia este testata ca sunt realizate mai multe unit teste si ce bucati ale aplicatie sunt testate

Capitolul 5

Concluzii

5.1 Concluzii generale

aici as vrea sa prezint cam care sunt concluziile legate de aplicatia mea, ce a mers bine, ce nu e asa bine(cu gandul ca acestea se pot imbunatatii)

sa pun putin accentul pe idee ca aplicatia nu este una finalizata si ca poate fi o idee mult imbunatita(ca nu se limiteaza doar la prezent, ca este o lucrare orientata spre viitor)

5.2 Concluzii personale

aici as vrea putin sa spun din prisma mea cum este aplicatia si cam ce am invatat eu din toate acestea

5.3 Dezvoltări ulterioare

aici as vrea sa prezint acele aspecte ce pot fi imbunatatite: ca se poate muta calculul pe gpu, se pot realiza alte moduri de trasare al razelor...

Bibliografie

- [1] David Oliva Elorza, *Room acoustics modeling using the raytracing method: implementation and evaluation*, Licentiate Thesis, University of Turku, 2005
- [2] Steve Haynal, Heidi Haynal, *Generating and Searching Families of FFT Algorithms*, Journal on Satisfiability, Boolean Modeling and Computation 7, 2011
- [3] Jeong Cheol-Ho, Ih Jeong-Guon, Rindel Jens Holger, *Consideration of Wall Reflection and Diffraction in the Room Acoustic Prediction Using the Phased Beam Tracing Method*, 9th Western Pacific Acoustics Conference, Seoul, Korea, 2007
- [4] Alexander Sengpiel, *Conversion: Sound pressure to Sound intensity and vice versa*, Tutorium, Berlin, 2013
- [5] Jonathan Andrew Hargreaves, Yiu Wai Lam, *An Energy Interpretation of the Kirchhoff-Helmholtz Boundary Integral Equation and its Application to Sound Field Synthesis*, Acta Acustica united with Acustica, 2014
- [6] Axon Samuel, *Unity at 10: For better- or worse- game development has never been easier*, Ars Technica, 2016(cit. on p. 10)
- [7] Lucian M. Sasu, *Medii vizuale de programare*, 2020(cit. on p. 23)
- [8] Effect of temperature on properties of air, 24 noiembrie 2020
https://en.wikipedia.org/wiki/Speed_of_sound
- [9] Legea lui Snell- Snell's law, 8 decembrie 2020
https://ro.qaz.wiki/wiki/Snell%27s_Law
- [10] How to evenly distribute points on a sphere more effectively than the canonical Fibonacci Lattice, 14 decembrie 2020
<http://extremelearning.com.au/how-to-evenly-distribute-points-on-a-sphere-more-effectively-than-the-canonical-fibonacci-lattice/>
- [11] Physics.Raycast din Unity, 15 decembrie 2020
<https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>
- [12] Intersecția unei sfere cu o linie, 22 decembrie 2020
http://www.ambrsoft.com/TrigoCalc/Sphere/SpherLineIntersection_.html
- [13] Biblioteca matplotlib, 20 ianuarie 2021
<https://pypi.org/project/matplotlib/>

- [14] Biblioteca NWaves, 3 decembrie 2020
<https://github.com/gkngkc/UnityStandaloneFileBrowser>
- [15] Biblioteca StandaloneFileBrowser, 10 decembrie 2020
<https://github.com/gkngkc/UnityStandaloneFileBrowser>
- [16] Biblioteca XCharts, 15 decembrie 2020
<https://github.com/monitor1394/unity-ugui-XCharts/blob/master/Assets/XCharts/README.md>