

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА  
ШЕВЧЕНКА**

ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ

Кафедра медичної радіофізики

«На правах рукопису»

Робота допущена до захисту в ЕК  
рішенням кафедри медичної радіофізики

від \_\_\_\_ \_\_\_\_ 2024 року, протокол № \_\_\_\_

Завідувач кафедри канд. фіз.-мат. наук, доцент  
\_\_\_\_\_ Сергій РАДЧЕНКО

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

«МОДЕЛЮВАННЯ РОЗПОДІЛУ РОЗСІЯНОГО ВИПРОМІНЮВАННЯ ПІД ЧАС  
РЕНТГЕНОГРАФІЧНИХ ДОСЛІДЖЕНЬ»

**Виконав:**

студент 4-го курсу  
денної форми навчання  
спеціальності 105 «Прикладна фізика та наноматеріали»  
ОПП «Електроніка та інформаційні технології в медицині»  
Завадський Кирило Юрійович \_\_\_\_\_

**Науковий керівник:**

канд. фіз.-мат. наук, доцент  
Радченко Сергій Петрович \_\_\_\_\_

**Рецензент:**

канд. фіз.-мат. наук, доцент  
Оберемок Євген Анатолійович \_\_\_\_\_

Засвідчую, що у цій бакалаврській роботі немає запозичень з праць інших авторів без відповідних посилань

Київ - 2024

## РЕФЕРАТ

Дипломна робота бакалавра: с.36, рис.10, таб.2, додатки 1, джерела 7.

Було розроблено методику відтворення рентгенівського зображення, і її ефективність було підтверджено на основі результатів моделювання інформаційних складових і шумових складових під час реєстрації первинного сигналу. У результаті було з'ясовано поліпшення якості придушення шумової складової, створюваної частинками X-доз, розсіяними по товщині досліджуваного об'єкта. Кількісна оцінка ефективності розробленого методу отримується як стандартне відхилення реконструйованого зображення в порівнянні зі змодельованим вихідним реєстраційним зображенням. Показано поліпшення пікового відношення сигнал/шум, а також зниження рівня шуму на відновленому зображенні.

РЕНТГЕНІВСЬКЕ ВИПРОМІНЮВАННЯ, РЕКОНСТРУКЦІЯ ЗОБРАЖЕННЯ,  
ЗАДАЧА, РОЗСІЯННЯ, МОДЕЛЮВАННЯ, PYTHON

## ЗМІСТ

РЕФЕРАТ.....	2
ТАБЛИЦЯ СКОРОЧЕНЬ.....	4
ВСТУП.....	6
1. ОСНОВНІ ПРИНЦИПИ ОТРИМАННЯ РЕНТГЕНІВСЬКИХ ЗОБРАЖЕНЬ ТА МЕТОДИ ЇХ ПОРІВНЯННЯ.....	7
1.1 Рентгенівське випромінювання.....	7
1.2 Проста модель отримання рентгенографічного зображення.....	8
1.3 Розсіювання поперечних перерізів.....	10
1.4 Методи схожості між двома зображеннями.....	12
1.4.1 Метод середньоквадратичного відхилення (RMSE).....	12
1.4.2 Пікове співвідношення сигналу до шуму PSNR.....	13
1.4.3 Індекс структурною збіжності SSIM.....	13
2. ФОРМУВАННЯ ПРОЕКЦІЙНОГО ЗОБРАЖЕННЯ.....	15
2.1 Математична модель.....	15
2.1.1 Поглинання та розсіяння.....	15
2.1.2 Реєстрація у площині $xOy$ .....	17
2.2 Опис алгоритму отримання та аналізу зображень.....	20
2.2.1 Загальні положення.....	20
2.2.2 Алгоритм отримання змодельованого рентгенівського зображення.....	22
3. РЕКОНСТРУКЦІЯ ПРОЕКЦІЙНИХ ДАНИХ.....	24
Висновки.....	27
Перелік посилань.....	28
Додаток А.....	29

## ТАБЛИЦЯ СКОРОЧЕНЬ

Таблиця 1.1

RMSE (Root Mean Square Error)	корінь середньоквадратичної помилки, один з найпоширеніших методів оцінки якості моделей. RMSE вимірює середню величину помилки між прогнозованими значеннями та фактичними значеннями
PSNR (Peak Signal-to-Noise Ratio)	пікове відношення сигналу до шуму, що використовується для оцінки якості реконструйованих або стиснутих зображень і відео, вимірює якість зображення шляхом порівняння максимальної можливої потужності сигналу з потужністю шуму, що впливає на якість відновленого зображення
SSIM (Structural Similarity Index)	індекс структурної схожості, який використовується для вимірювання подібності між двома зображеннями, оцінює якість зображень на основі того, як сприймає людське око, на відміну від традиційних показників, таких як MSE або PSNR
SPR (Surface Plasmon Resonance)	методика, яка використовується для дослідження взаємодії між молекулами в режимі реального часу без необхідності мічення, застосовується в біохімії та

	біофізиці для вивчення кінетики та афінності взаємодій між біомолекулами
DSSIM (Difference Structural Similarity Index)	показник, який використовується для вимірювання різниці між двома зображеннями на основі індексу структурної схожості (SSIM). DSSIM є похідним від SSIM і часто використовується як доповнення до нього для кращої інтерпретації результатів

## ВСТУП

Рентгенівська ендоскопія - один із найбільш часто використовуваних методів обстеження людьми без хірургічного втручання. Він дає змогу правильно подивитися на форму кісток і локалізувати в них ґрунт перелому. На жаль, через високу енергію рентгенівського потоку та фізичну природу взаємодії отримане зображення не може бути зменшене фізичними методами. Медико-наукова спільнота поступово переходить на використання цифрових рентгенівських променів, що дає змогу обробляти отримане зображення за допомогою комп'ютера.

Оскільки однією з основних причин шуму було первинне розсіювання частинок X-доз(випромінювання), метою дослідження було вивчити можливість фільтрації такого шуму на основі розв'язання оберненої задачі з використанням статистичної реконструкції проекційних даних.

Це досягнення сприяє розробці технології придушення шуму для рентгенівських зображень, заснованої на обробці первинних даних.

# 1. ОСНОВНІ ПРИНЦИПИ ОТРИМАННЯ РЕНТГЕНІВСЬКИХ ЗОБРАЖЕНЬ ТА МЕТОДИ ЇХ ПОРІВНЯННЯ

## *1.1 Рентгенівське випромінювання*

Короткохвильове електромагнітне випромінювання з довжиною хвилі від 10 нм до 0,01 нм. В електромагнітному спектрі діапазон частот рентгенівських променів знаходиться між ультрафіолетовими і гамма-променями [1].

Рентгенівське випромінювання виникає в результаті різкого уповільнення руху швидких електронів в речовині під час енергетичного переходу внутрішніх електронів атома. Він використовується в науці, техніці та медицині. Рентгенівські промені використовуються в флюороскопії, рентгенофлуоресцентному аналізі та кристалографії для визначення атомної структури кристала. Метод дослідження речовини за допомогою рентгенівських променів об'єднується з терміном рентгенівська спектроскопія [1].

При взаємодії рентгенівського випромінювання з речовиною відбувається один із трьох процесів:

- Поглинання
- Розсіяння
- Без змін

## 1.2 Проста модель отримання рентгенографічного зображення

Розглянемо монохроматичне рентгенівське джерело, що випромінювало кванти з енергією  $E$ , розташоване на достатній відстані від пацієнта, так що потік квантів можна вважати паралельним осі  $z$  (рис. 1.2). Зображення при цьому реєструється в площині  $xy$ . Припустимо, що кожен взаємодіючий з приймачем квант локально поглинається, а характеристика приймача є лінійною; при цьому рентгенівське зображення можна розглядати як розподіл поглиненої енергії квантів. Якщо  $N$  - кількість квантів, що падають на одиницю площі тіла пацієнта, а  $I(x, y)dx dy$  - енергія, що поглинається елементом  $dx dy$  поверхні детектора, то справедливе співвідношення:  $dx dy$  поверхні детектора, то справедливе співвідношення:

$$I(x, y) = N\varepsilon(E, 0)E \exp\left(-\int \mu(x, y, z)dz\right) + \int \varepsilon(E_s\theta) E_s S(x, y, E_s, \Omega) d\Omega dE_s =$$

де інтегрування ведеться вздовж ліній прольоту первинних фотонів, що приходять у точку з координатами  $(x, y)$ , а  $\mu(x, y, z)$  - лінійний коефіцієнт ослаблення випромінювання у тканинах [2].

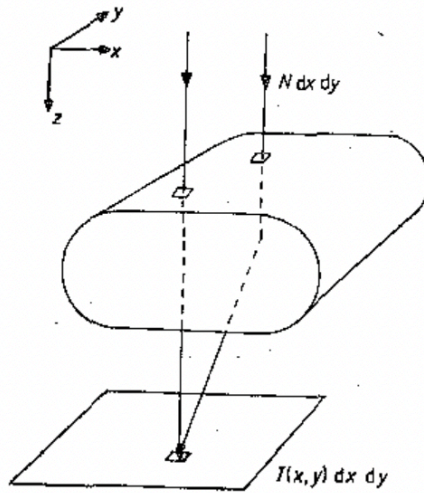


Рис 1.2. Проста модель отримання рентгенографічного зображення

Функція  $S$  розподілу розсіяних квантів, визначена таким чином, що величина  $S(x, y, E, \Omega) dE d\Omega dx dy =$  кількості квантів з енергією в діапазоні від  $E$  до  $E + dE$ , розсіяних в області тілесних кутів від  $\Omega$  до  $\Omega + d\Omega$ , що проходять через майданчик  $dx dy$  на поверхні детектор.

### *1.3 Розсіювання поперечних перерізів*

Переріз розсіювання  $\sigma$  - це значення, пропорційне швидкості, з якою відбувається дана взаємодія випромінювання-мішень. Точніше, якщо врахувати, що падаюче випромінювання складається з квантів або "частинок" (наприклад, фотонів і нейтронів), площа поперечного перерізу - це швидкість розсіювання на одиницю падаючого променя випромінювання (кількість подій розсіювання за одиницю часу), по-друге, кількість падаючих частинок. цільова поверхня за одиницю часу на одиницю кадру. Якщо випромінювання розглядається як безперервна класична хвиля, як у випадку довгохвильового електромагнітного випромінювання, площа поперечного перерізу розсіювання визначається як потужність хвилі, що розсіюється, поділена на інтенсивність падаючої хвилі. З точки зору розміру, поперечний переріз-це простір, корпус якого-сарай, що представляє площу. Розсіяний поперечний переріз слід інтерпретувати не як справжню геометричну площу поперечного перерізу, а як ефективну площу, пропорційну ймовірності взаємодії випромінювання і цілі.

Диференціальні перерізи розсіювання вимірюють у лабораторних умовах. Це визначається розміщенням детектора в певному кутовому положенні на значній відстані від мішені, що розсіюється. На рисунку 1.3 показано типову форму розсіювання. Промінь, що падає, рухається в позитивному  $z$  напрямку, беручи до уваги випромінювання, спрямоване на диференціальний тілесний кут  $d\Omega$  під полярним  $\theta$  і азимутальним  $\varphi$  кутами.

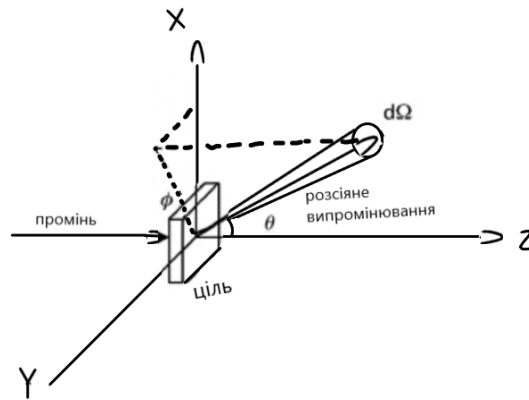


Рис. 1.3. Стандартна геометрія розсіювання для вимірювання диференціального поперечного перерізу розсіювання.

Потім можна виміряти кутовий диференціальний поперечний переріз розсіювання, заданий формулою  $\frac{d\sigma}{d\Omega} = \frac{dR/d\Omega}{\Phi_{in}}$  де  $dR$  – швидкість розсіювання на тілесний кут  $d\Omega$ , а  $\Phi_{in}$  – потік що падає. Найосновнішим типом поперечного перерізу є подвійний диференціальний переріз розсіювання,  $d^2\sigma/d\Omega dE$ . Величина  $[d^2\sigma/d\Omega dE] d\Omega dE$  — це кількість частинок, кожна з енергією падіння  $E$ , розсіяних (за одиницю часу) на тілесний кут  $d\Omega$  з енергією між  $E'$  і  $E + dE$ , поділена на потік падаючого променя. Після того як подвійний диференціальний поперечний переріз отримано або виміряно,  $d\sigma/d\Omega$  і  $\sigma$  можна обчислити шляхом інтегрування за енергією розсіяного випромінювання та тілесного кута [2].

## 1.4 Методи схожості між двома зображеннями

### 1.4.1 Метод середньоквадратичного відхилення (RMSE)

Перейдемо до схожостей між зображеннями. Розглянуто 3 методи порівняння монохромних зображень один з одним. Як от від методу середньоквадратичного відхилення (простих), так до більш комплексних.

У статистиці, стандартне відхилення використовується для вимірювання розмаїтості або розкиду значень у множині даних. Низьке стандартне відхилення вказує на те, що більшість значень знаходяться близько до середнього значення множини, тобто вони мало відхиляються від середнього. У той же час, високе стандартне відхилення свідчить про значний розкид значень.

Це вимірювання відображає квадратний корінь дисперсії випадкової змінної, вибірки або розподілу ймовірності. Важливо розрізнити стандартне відхилення загальної сукупності і вибірки від стандартної похибки статистики, наприклад, середнього значення вибірки. Стандартна похибка середнього представляє собою стандартне відхилення множини зразків, взятих з загальної сукупності, поділене на квадратний корінь від розміру вибірки. Наприклад, стандартна похибка опитування відображає стандартне відхилення оцінюваних середніх значень від багатьох повторюваних опитувань. Таким чином, ці концепції дозволяють оцінити, наскільки точно оцінки залежать від конкретних вибірок з популяції [3].

Для двох монохромних зображень  $I, K$  розміром  $m \times n$  середньоквадратичне відхилення вираховується за формулою:

$$RMSE = \sqrt{\frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |I(i, j) - K(i, j)|^2}$$

### **1.4.2 Пікове співвідношення сигналу до шуму PSNR**

PSNR (Peak Signal-to-Noise Ratio) є технічним терміном, який позначає відношення між максимально можливим значенням сигналу і інтенсивністю шуму, яка спотворює ці значення сигналу. Оскільки багато сигналів мають широкий динамічний діапазон, PSNR зазвичай вимірюється у логарифмічних децибелах. Вираховується наступним чином:

$$PSNR = 10 \log_{10} \left( \frac{MAX_I^2}{RMSE^2} \right) = 20 \log_{10} \left( \frac{MAX_I}{RMSE} \right) \quad \text{де } MAX_I -$$

максимальне значення пікселя

### **1.4.3 Індекс структурною збіжності SSIM**

SSIM (Structural Similarity Index) є одним із методів вимірювання схожості між двома зображеннями. Це метод повного порівняння, який оцінює якість на основі оригінального зображення без стиснення або спотворення. SSIM є вдосконаленням традиційних методів, таких як PSNR (Peak Signal-to-Noise Ratio) і MSE (Mean Square Error), які виявилися несумісними з фізіологією людського сприйняття.

На відміну від MSE і PSNR, метод SSIM враховує "сприйняття помилки" через структурні зміни інформації. Ідея полягає в тому, що пікселі мають сильні взаємозв'язки, особливо коли вони просторово близькі. Ці взаємозв'язки надають важливу інформацію про структуру об'єктів та сцени загалом.[4]

Метрика SSIM розрахована на різні розміри вікон. Різниця між двома вікнами  $x$  та  $y$  мають однаковий розмір  $N \times N$ :

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

де

$\mu_x$  - середнє  $x$

$\mu_y$  - середнє  $y$

$\sigma_x^2$  - дисперсія  $x$

$\sigma_y^2$  - дисперсія  $y$

$\sigma_{xy}$  - коваріація  $x$  та  $y$

$c_1 = (k_1 L)^2, c_2 = (k_2 L)^2$  - дві змінні

$L$  - динамічний діапазон пікселів

$k_1 = 0.01, k_2 = 0.03$  - константи

## 2. ФОРМУВАННЯ ПРОЕКЦІЙНОГО ЗОБРАЖЕННЯ

### 2.1 Математична модель

#### 2.1.1 Поглинання та розсіяння

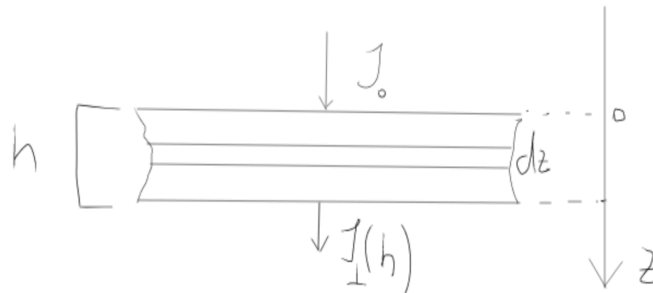


Рис 2.1.1. Схема моделювання поглинання рентгенівського випромінювання

$$dI \sin -I dz$$

$$dI = -\mu I dz$$

$$\int_{I_0}^{I_z} \frac{dI}{I} = - \int_0^z \mu dz$$

$\mu$  може залежати від  $z$  (не однорідний матеріал)

$$I_a \rightarrow I(z) = I_0 \exp\left(- \int_0^z \mu(z) dz\right)$$

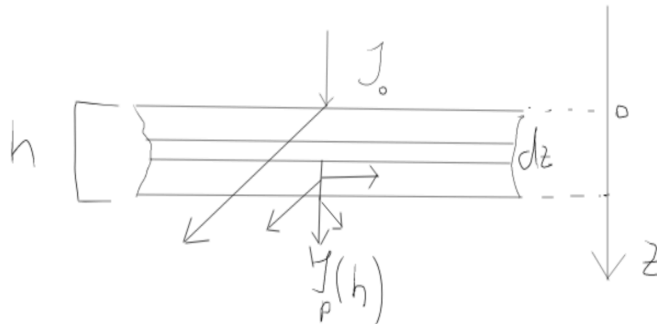


Рис 2.1.2 Схема моделювання розсіювання рентгенівського випромінювання

$$I_s \rightarrow I(z) = I_0 \exp\left(- \int_0^z k(z) dz\right)$$

Корисна складова сигналу складається із суми випромінювання, яке не було поглинуте, та випромінювання, яке не було розсіяне.

$$I_{\perp} = I_a + I_b$$

$$I_{\perp}(x, y) = I_0(x, y) \exp \left[ - \int_{z_2}^{z_1} (\mu + k) dz \right]$$

де  $k$  та  $\mu$  в загальному є функціями від координат

$$k = f(x, y, z)\mu = f(x, y; z) \Rightarrow \text{отже } I_z \sim I_0$$

Позначимо

$$\alpha(x, y) = \alpha = \exp \left( - \int_{z_2}^{z_1} (\mu + k) dz \right)$$

$\alpha$  - ослаблення випромінювання

$$I_{\perp} = \alpha(x, y)I_0(x, y)$$

### 2.1.2 Реєстрація у площині $xOy$

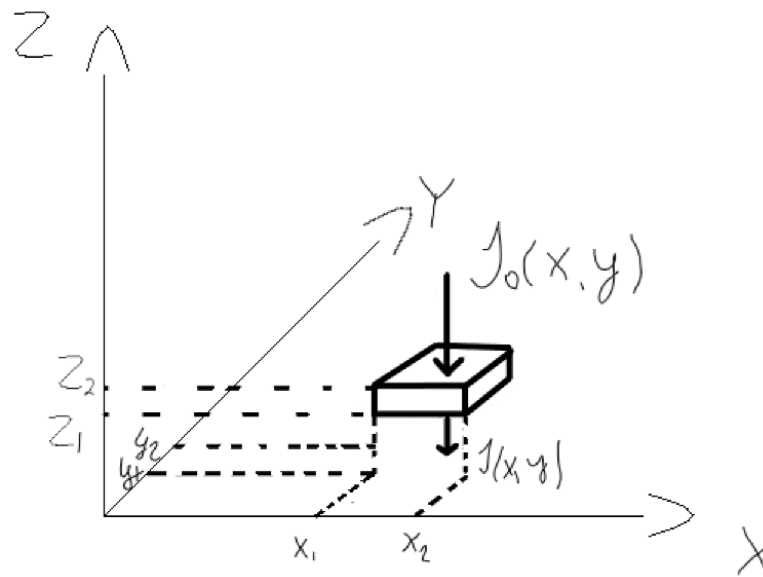


Рис 2.1.2. Схема проходження променів при реєстрації у площині  $xOy$

$$I(x, y) = I_{\perp}(x, y) + I_p(x, y), \text{ де}$$

$I(x, y)$  - отримане зображення;

$I_{\perp}(x, y)$  - корисна складова (наша ціль, те що хочемо реконструювати),

$I_p(x, y)$  - шуми (розсіяння складова)

Зареєстроване зображення це - розподіл  $I(x, y, z)$  в площині  $xOy \Leftrightarrow I(x, y, 0) = I(x, y)|_{z=0} = I(x, y)$

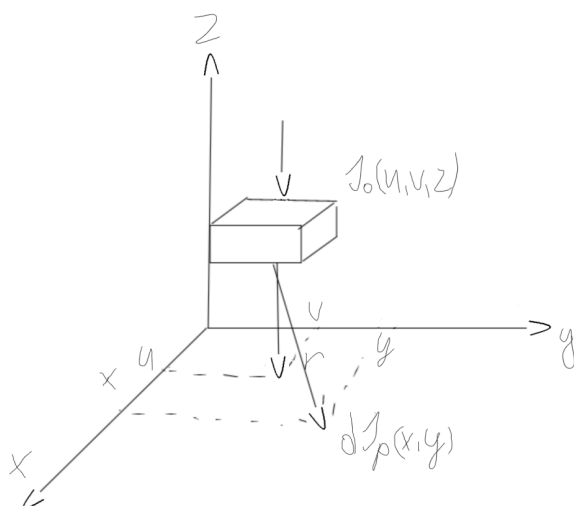


Рис 2.1.3. Схема моделювання проходження рентгенівських променів та ілюстрація фізичних величин які використовувалися у роботі  $xOy$

$$I(x, y) = I_{\perp}(x, y) + I_p(x, y)$$

$$dI_p(x, y) \sim \frac{I(u, v, z)dV}{r^2} \quad [1]$$

$$dV = du * dv * dz$$

За теоремою Піфагора:

$$z = f(x; y, z, u, v) = \sqrt{(x - u)^2 + (y - v)^2 + z^2}$$

$$dI_p(x, y) = \frac{\gamma(u, v, z)I_{\perp}(u, v, z)dV}{(x-u)^2+(y-v)^2+z^2}$$

$\gamma(u, v, z)$  - коефіцієнт пропорційності

$$I_{\perp}(u, v, z) = I_0(u, v) \exp \left[ - \int_{d+h}^z (\mu(u, v, \omega) + k(u, v, \omega)) d\omega \right] =$$

$= I_0(u, v) \cdot \alpha(u, v, z)$ , де  $\alpha(u, v, z)$  - ослаблення при проникненні до  $z$ .

$$I_p(x, y) = \int_0^a \int_0^b \int_d^{h+d} \frac{I_0(u, v) \gamma(u, v, z) \alpha(u, v, z) du dv dz}{(x-u)^2+(y-v)^2+z^2}$$

### 2.1.3 Спрощення моделі

- відстань від об'єкта до джерела набагато більше за розміри об'єкта  
 $\Leftrightarrow l \gg a; b$

- джерело ізотропне g

$$I_0(x, y, z)|_{z=z_2} = \text{const} \Rightarrow I_0(u, v) = I_0 = \text{const} \neq f(u, v)$$

Перепишемо  $I_p$ :

$$I_p(x, y) = I_0 * \int_0^a \int_0^b \int_d^{h+d} \frac{\gamma(u, v, z) \alpha(u, v, z) du dv dz}{(x-u)^2 + (y-v)^2 + z^2}$$

Позначимо  $\beta(x, y) = \int_0^a \int_0^b \int_d^{h+d} \frac{\gamma(u, v, z) \alpha(u, v, z) du dv dz}{(x-u)^2 + (y-v)^2 + z^2}$  тоді

$$I_p(x, y) = I_0 \beta(x, y) \Rightarrow I_p(x, y) \sim I_0$$

$$\beta(x, y) = \frac{I_p(x, y)}{I_0} \quad \text{- частка розсіяного випромінювання відносно}$$

падаючого(первинного), SPR(scatter-to-primary ratio)

Отже

$$\begin{aligned} I(x, y) &= I_0 * \alpha(x, y) + I_0 * \beta(x, y) = \\ &= I_0 (\alpha(x, y) + \beta(x, y)) = I_0 \left( 1 + \frac{\beta}{\alpha} \right) = \\ &= I_{\perp} \frac{\alpha + \beta}{\alpha} \Rightarrow I_{\perp} = \frac{I(x, y) \alpha}{\alpha + \beta} \end{aligned}$$

## ***2.2 Опис алгоритму отримання та аналізу зображень***

### ***2.2.1 Загальні положення***

- Усі розрахунки та безпосереднє моделювання виконується за допомогою мови програмування Python [6]
- Кожен квант розраховувався окремо
- В один момент часу квант може:
  - Розсіятися
  - Поглинутись
  - Поникати (не взаємодіяти з речовиною)
- В задачі розглядається лише первинне розсіяння так як воно найбільш вірогідне
- Об'єкт - однорідний та має дві зони різної товщини
- Розмір детектуючої площини у 3 рази більший за розмір об'єкта [5]

Для представлення об'єкту дослідження у програмі використовувався двовимірний масив [7] де координати кожного елемента відповідає координатам об'єкта, а значення елемента - відповідає товщині об'єкта у даній точці.

Також для графічного представлення об'єкту а потім і результатів використовувалася нормована на 255 значень кольорова схема представлення результатів де білий колір відповідає відсутності значень у даній точці, та чорний колір - показує максимальне значення

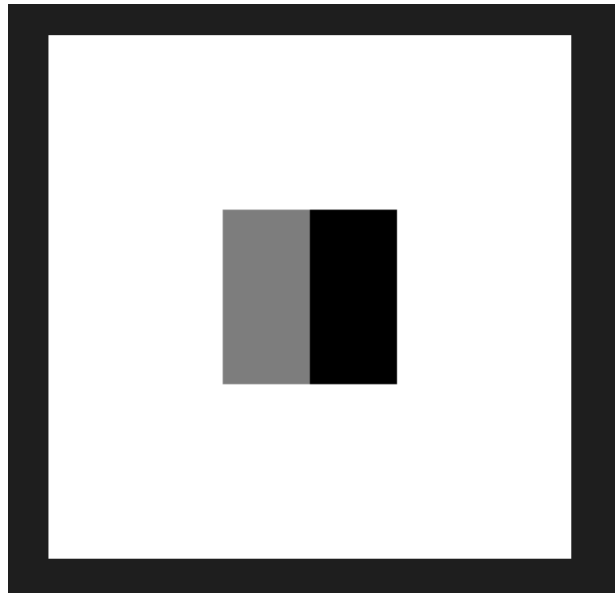


Рис 2.2.1. Представлення дворівневого об'єкта за допомогою градацій сірого кольору

### 2.2.2 Алгоритм отримання змодельованого рентгенівського зображення

Найголовнішим принципом для отримання змодельованого рентгенівського зображення є детермінованість можливих подій для одиничного кванту. Таким чином кванти в моделі підпорядковувалися наступним правилам: поза об'єктом квант не розсіюється і не поглинається; в об'єкті він може пролетіти відстань  $d_b$  до поглинання, де відстань прольоту вибирається випадковим чином за експоненціальним розподілом  $f(x) = \lambda e^{-\lambda x}$ , де  $\lambda$  це коефіцієнт поглинання, і якщо ця відстань більша за довжину об'єкта - переходимо до наступного пункту, якщо ж ні - цей квант вважається поглинутим і переходимо до симуляції наступного кванту. Квант не поглинувся, але може бути розсіяним через певну відстань  $d_p$ , яка так само вибирається за експоненціальним розподілом, але  $\lambda$  в даному випадку це коефіцієнт розсіяння. При  $d_p$  більше за довжину об'єкта - квант потрапляє у ту саму координату з якої вилітає, саме з таких квантів складається корисна частина нашого фінального зображення. У випадку коли  $d_p$  менше за товщину об'єкта - квант ще має шанс потрапити у детектор але вже з іншими координатами, такі кванти створюють шумову компоненту сигналу.

При розсіянні кванту всередині об'єкта постає задача знаходження нових координат де буде зареєстрований квант. Для цього випадковим чином вибираються кути  $\theta$  та  $\phi$ , де  $\theta$  - це кут відносно  $z$  та новим напрямком польоту кванта, кут  $\phi$  - відповідає новому напрямку в площині  $xOy$ , кут  $\phi$  обирається з рівномірного розподілу ( $f(x) = \frac{1}{2\pi}$ ) в межах  $[0, 2\pi)$ . Розподіл кута  $\theta$  більш складний, і його точний розподіл буде одним із пунктів у наступних роботах, для даної задачі було обрано розподіл  $f(x) = \frac{\pi}{2} \cos^2 \frac{x}{2}$  в межах  $[0, \pi]$  виходячи із фізичних міркувань, а саме те що квант з більшою вірогідністю відхилиться на незначний кут і майже з 0ю вірогідністю після зіткнення полетить у протилежному напрямі.

### **2.2.3 Алгоритм реконструкції рентгенівського зображення**

Нехай  $A$  - площина реконструйованого зображення,  $A_{\text{real}}$  - площина зображення що була отримана в результаті моделювання,  $\alpha$  і  $\beta$  - площини коефіцієнтів, тоді за нашою гіпотезою

$$A_{ij} = \frac{A_{\text{real}_{ij}} * \alpha_{ij}}{\alpha_{ij} + \beta_{ij}}$$

### 3. РЕКОНСТРУКЦІЯ ПРОЕКЦІЙНИХ ДАНИХ

Однією з ключових переваг моделювання є можливість отримання даних, недоступних у реальних експериментах. Так, в результаті моделювання, крім змодельованого рентгенівського зображення (рис. 3.1), було отримано ідеальне зображення, тобто зображення без шумів. Таким чином, головним критерієм порівняння стає визначення рівня відхилень зображень від ідеального стандарту.

Для перевірки можливості використання даного методу в майбутньому перш за все потрібно довести що рентгенівське зображення отримане даним методом краще за оригінальне зображення. Для цього було підбрано значення  $\alpha$  і  $\beta$  наступним чином: за вкладками вказаними у розділі 2.1.2  $\alpha(x, y) = \frac{I_{\perp}(x, y)}{I_0(x, y)}$  де  $I_{\perp}$  - це розподіл інтенсивності рентгенівського зображення за відсутності шумів, а  $I_0$  - інтенсивність випромінювання рентгенівської трубки; в свою чергу  $\beta$  (розділ 2.1.3)

$\beta(x, y) = \frac{I_p(x, y)}{I_0}$  - частка розсіяного випромінювання відносно падаючого(первинного), SPR (scatter-to-primary ratio), де  $I_p$  - розподіл шумів

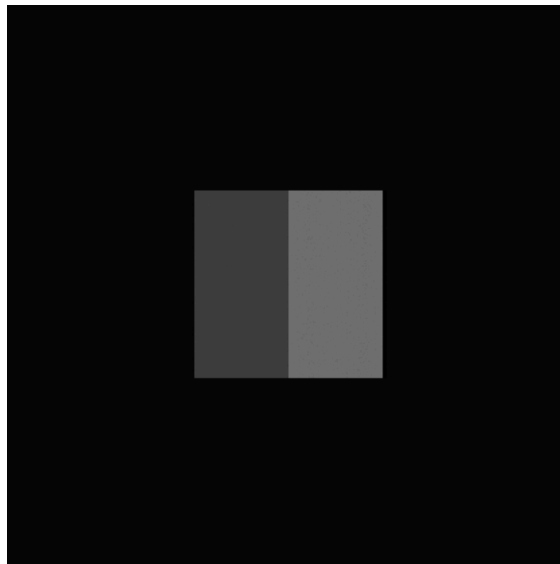


Рис 3.1. Монохромне представлення рентгенівського зображення. Чорний колір відповідає максимуму отриманого опромінення.

Рис 3.2 та 3.3 наочно демонструють ефективність застосування методики реконструкції для отримання більш якісного рентгенівського зображення

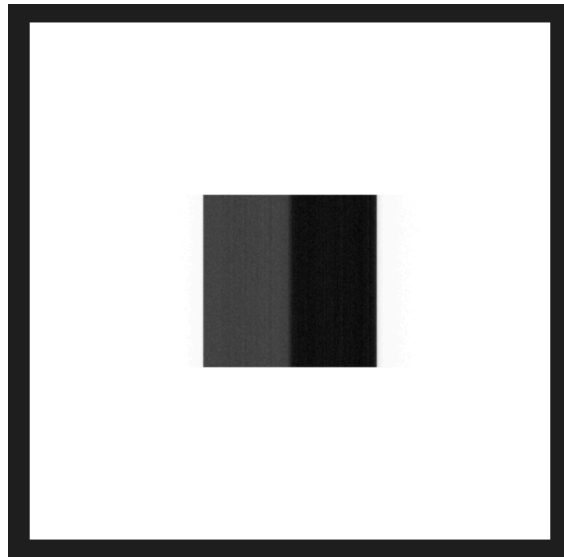


Рис 3.2. Різницевий сигнал між ідеальним та відфільтрованим зображенням

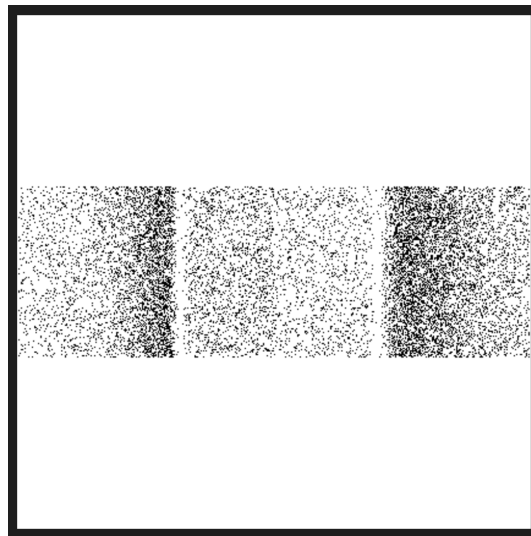


Рис 3.3. Різницевий сигнал між ідеальним та відфільтрованим зображенням

Чисельне порівняння (таблиця 1.2) підтверджує ефективність методики реконструкції.

Таблиця 1.2. "Числова оцінка порівнянь зображень з ідеальним випадком. Чим більше значення параметрів RMSE та DSSIM, тим більше відрізняються два зображення одне від одного. Для PSNR: чим вище значення, тим менше шуму присутньо у зображенні (виражено в децибелах), к-ть розрядів і похибка.

	RMSE	PSNR	DSSIM
Початкове зображення	778.6	28.396	0.1250
Реконструйоване зображення	0.255	97.896	0.1090

## *Висновки*

1. Модель, запропонована в дослідженні, дозволяє створювати розподіл значень SPR (відношення розсіяння до первинного випромінювання) у площині реєстрації рентгенівського проєкційного зображення. Цей розподіл відображається на основі обробки вихідних даних і застосування математичних моделей, що враховують фізичні властивості рентгенівського випромінювання та його взаємодію з матеріалом об'єкта.

2. Застосування математичного моделювання вказує на можливість ефективного приглушення впливу розсіяного випромінювання під час формування проєкційного рентгенівського зображення за допомогою запропонованого алгоритму статистичної реконструкції. Цей результат досягається за певних умов освітлення та параметрів експерименту, таких як інтенсивність первинного випромінювання та геометрія системи реєстрації.

3. За допомогою кількісних оцінок, таких як середньоквадратичне відхилення, досліджено можливість використання розробленої методики статистичної реконструкції проєкційних даних рентгенівського випромінювання для вирішення оберненої задачі. Оцінки показали, що методика забезпечує високу точність реконструкції, зокрема, зменшення похибок у відновлених зображеннях.

4. Результати математичного моделювання, отримані шляхом симуляції формування рентгенівського зображення та його обробки з використанням мови програмування Python, підтверджують ефективність запропонованої методики реконструкції. Конкретні числові показники, такі як покращення пікового відношення сигнал/шум на 2000 разів та зниження рівня шуму на 70 дБ, свідчать про високу ефективність методу

## Перелік посилань

1. Krane, Kenneth S. 1983. *Modern Physics*. N.p.: Internet Archive.  
<https://archive.org/details/modernphysics00kran>
2. International Atomic Energy Agency (IAEA). *Radiation Oncology Physics: A Handbook for Teachers and Students*. N.p.: IAEA.  
[https://www-pub.iaea.org/MTCD/Publications/PDF/Pub1196\\_web.pdf](https://www-pub.iaea.org/MTCD/Publications/PDF/Pub1196_web.pdf)
3. Thornton, Stephen T. 2004. *Modern Physics for Scientists and Engineers*. N.p.: Internet Archive.  
[https://archive.org/details/modernphysicsfor0000thor\\_g5v7](https://archive.org/details/modernphysicsfor0000thor_g5v7)
4. Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4): 600-612, 2004.  
Режим доступу :  
<http://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf>
5. Даник, А. Ю. Фізичні основи мінімізації дозового навантаження на пацієнта при корекції впливу розсіяного випромінювання у рентгенівській діагностиці. Дис. доктора філософії, Київ, 2021, 122 с.
6. Welcome to Python.org. Accessed June 08, 2024.  
<https://www.python.org/>.
7. numpy.array — NumPy v1.22 Manual. June 08, 2024.  
<https://numpy.org/doc/stable/reference/generated/numpy.array.html>.

## Додаток А

```
import random # Для генерації випадкових чисел
from dataclasses import dataclass # Для створення простих класів для
зберігання даних
from math import atan, cos, log10, pi, sqrt, tan # Для математичних операцій
from multiprocessing import Pool # Для паралельного обчислення
from typing import List, Tuple, Union # Для типізації

import cv2 # Для роботи із зображеннями
import numpy as np # Для роботи з масивами
import scipy.stats as st # Для статистичних функцій
from skimage.metrics import structural_similarity # Для обчислення індексу
структурної подібності
from tqdm import tqdm # Для відображення прогресу

# Константи
MAX_QUANTUM = 20000
MU = 0.2
K = 0.2

def dist(p: Tuple[float, float], q: Tuple[float, float]) -> float:
    """Обчислення евклідової відстані між двома точками."""
    return sqrt(sum((px - qx) ** 2.0 for px, qx in zip(p, q)))

def norm_array(arr: np.ndarray, normalize: bool = True) -> np.ndarray:
    """Нормалізація масиву."""
    arr_max = np.max(arr)
    if normalize:
        return (-1 * (arr - arr_max)) / arr_max * 255
    else:
        return -1 * (arr - arr_max)
```

```

@dataclass
class Coordinate:
    """Клас для зберігання координат."""
    x: float
    y: float

    def get(self) -> Tuple[float, float]:
        return (self.x, self.y)

    def move(self, dx: float, dy: float) -> 'Coordinate':
        return Coordinate(self.x + dx, self.y + dy)

class CosinePDF(st.rv_continuous):
    """Клас для ймовірнісного розподілу на основі косинуса."""
    def _pdf(self, theta):
        return 2 / pi * cos(theta / 2) ** 2

class XRaySimulation:
    def __init__(self, dimension: int, height_list: List[int]) -> None:
        """Ініціалізація рентгенівської симуляції."""
        level_count = len(height_list)
        width_obj = dimension * level_count
        object_row = [height for height in height_list for _ in range(dimension)]
        object = np.array([object_row] * width_obj)
        zeros = np.zeros((width_obj, width_obj), dtype=np.int64)
        whole_obj = np.block([[zeros] * 3, [zeros, object, zeros], [zeros] * 3])

        self.width = dimension * level_count * 3
        self.obj = whole_obj
        self.ref = np.zeros_like(whole_obj, dtype=np.int64)
        self.pred = np.zeros_like(whole_obj, dtype=np.int64)
        self.theta_dist = CosinePDF(a=0, b=pi, name='CosinePDF')

```

```

np.savetxt('obj.csv', self.obj)
cv2.imwrite('obj.png', norm_array(self.obj))

def get_cross_section(self, coord: Coordinate, phi: float) -> np.ndarray:
    """Отримання поперечного перерізу об'єкта."""
    y, x = coord.get()
    return self.obj[y, :x+1][::-1] if phi > 0 else self.obj[y, x:]

def trace_calculation(self, height: float, theta: float, cross_section: List[int]) ->
Union[int, None]:
    """Обчислення траєкторії кванта через об'єкт."""
    if atan(len(cross_section) / height) < theta:
        return None

    x0 = 0
    while height > 0:
        if height > cross_section[x0]:
            x1 = int(x0 + (height - cross_section[x0]) * tan(theta))
            height = cross_section[x0]
            x0 = x1
        else:
            x1 = x0 + 1
            while cross_section[x0] == cross_section[x1]:
                x1 += 1
            dx = x1 - x0
            new_height = height - dx / tan(theta)
            if new_height < 0:
                dx = int(height * tan(theta))
                x1 = x0 + dx
                new_height = 0
            length = dist((x0, height), (x1, new_height))
            if length > random.expovariate(MU):
                return None

```

```

        x0 = x1
        height = new_height

    return x0 if x0 < len(cross_section) - 1 else None

def get_new_coordinate(self, coord: Coordinate, height: int) -> Union[Coordinate,
None]:
    """Отримання нових координат після розсіювання кванта."""
    phi = random.choice((1, -1))
    theta = self.theta_dist.rvs(size=1)[0]
    cross_section = self.get_cross_section(coord, phi)
    length = self.trace_calculation(height, theta, cross_section)
    if length is not None:
        dy = -phi * length
        return coord.move(0, dy)
    else:
        return None

def run_quantum(self, coord: Coordinate) -> Tuple[int, int, Coordinate]:
    """Обробка одного кванта."""
    height = self.obj[coord.get()]
    ref = 0
    pred = 0

    if height == 0:
        ref += 1
        pred += 1
    else:
        height_entry = random.expovariate(MU)
        if height < height_entry:
            height_scattering = random.expovariate(K)
            if height < height_scattering:

```

```

        ref += 1
        pred += 1
    else:
        new_coord = self.get_new_coordinate(coord, height - height_scattering)
        if new_coord is not None:
            pred += 1
            coord = new_coord

    return ref, pred, coord

def process_row(self, i: int) -> Tuple[np.ndarray, np.ndarray]:
    """Обробка одного рядка зображення."""
    ref = np.zeros((self.width, self.width), dtype=np.int64)
    pred = np.zeros((self.width, self.width), dtype=np.int64)
    for j in range(self.width):
        for _ in range(MAX_QUANTUM):
            ref_new, pred_new, coord = self.run_quantum(Coordinate(i, j))
            ref[coord.get()] += ref_new
            pred[coord.get()] += pred_new
    return ref, pred

def run_simulation(self):
    """Запуск симуляції."""
    random.seed()
    with Pool() as pool:
        for ref, pred in tqdm(pool.imap(self.process_row, range(self.width)),
total=self.width, desc='Processing rows'):
            self.ref += ref
            self.pred += pred

    np.savetxt('ref.csv', self.ref)
    np.savetxt('pred.csv', self.pred)
    cv2.imwrite('ref.png', norm_array(self.ref))

```

```
cv2.imwrite('pred.png', norm_array(self.pred))
```

```
class ImageProcessing:
```

```
def __init__(self, path: str = './') -> None:
```

```
    """Ініціалізація обробки зображень."""
```

```
    path = path.rstrip('/')
```

```
    self.ref = np.loadtxt(f'{path}/ref.csv')
```

```
    self.pred = np.loadtxt(f'{path}/pred.csv')
```

```
    self.res = np.zeros_like(self.ref, dtype=np.int64)
```

```
    self.scats = self.pred - self.ref
```

```
    cv2.imwrite('scats.png', norm_array(self.scats))
```

```
def convolve(self):
```

```
    """Зменшення розмірності зображень шляхом згортки."""
```

```
    new_size = len(self.ref) // 2
```

```
    ref = np.zeros((new_size, new_size), dtype=np.int64)
```

```
    pred = np.zeros((new_size, new_size), dtype=np.int64)
```

```
    for i in range(new_size):
```

```
        for j in range(new_size):
```

```
            ref[i, j] = np.sum(self.ref[2*i:2*i+2, 2*j:2*j+2])
```

```
            pred[i, j] = np.sum(self.pred[2*i:2*i+2, 2*j:2*j+2])
```

```
    self.ref = ref
```

```
    self.pred = pred
```

```
    self.scats = pred - ref
```

```
    cv2.imwrite('ref_convolve.png', norm_array(self.ref))
```

```
    cv2.imwrite('pred_convolve.png', norm_array(self.pred))
```

```
    cv2.imwrite('scats_convolve.png', norm_array(self.scats))
```

```
def get_alpha_beta(self, mode: int = 0) -> Tuple[np.ndarray, np.ndarray]:
```

```
    """Обчислення коефіцієнтів альфа та бета для відновлення  
зображення."""
```

```
    if mode == 0:
```

```

        return self.pred / 10, 9 * self.pred / 10
elif mode == 1:
    return self.ref / MAX_QUANTUM, self.scat / MAX_QUANTUM

def restore_image(self):
    """Відновлення зображення."""
    alpha, beta = self.get_alpha_beta(1)
    for i in range(len(self.pred)):
        for j in range(len(self.pred)):
            self.res[i, j] = (self.pred[i, j] * alpha[i, j]) / (alpha[i, j] + beta[i, j])

    np.savetxt('res.csv', self.res)
    cv2.imwrite('res.png', norm_array(self.res))
    cv2.imwrite('diff_res_ref.png', norm_array(self.ref - self.res))
    cv2.imwrite('diff_res_pred.png', norm_array(self.pred - self.res))

class ImageComparison:
    def __init__(self, path: str = './'):
        """Ініціалізація класу для порівняння зображень."""
        path = path.rstrip('/')
        self.ref = np.loadtxt(f'{path}/ref.csv')
        self.pred = np.loadtxt(f'{path}/pred.csv')
        self.res = np.loadtxt(f'{path}/res.csv')

    @staticmethod
    def mse(arr1: np.ndarray, arr2: np.ndarray) -> float:
        """Обчислення середньоквадратичної помилки."""
        return np.sqrt(np.mean((arr1 - arr2) ** 2))

    @staticmethod
    def psnr(arr1: np.ndarray, arr2: np.ndarray) -> float:
        """Обчислення пікового співвідношення сигнал/шум."""
        arr_max = max(np.max(arr1), np.max(arr2))

```

```

    return 20 * log10(arr_max / ImageComparison.mse(arr1, arr2))

    @staticmethod
    def ssim(arr1: np.ndarray, arr2: np.ndarray) -> float:
        """Обчислення структурної подібності зображень."""
        score, _ = structural_similarity(arr1, arr2, full=True)
        return score

    def get_mse(self) -> Tuple[float, float]:
        """Отримання середньоквадратичної помилки для різних
зображень."""
        return ImageComparison.mse(self.ref, self.res), ImageComparison.mse(self.ref,
self.pred)

    def get_psnr(self) -> Tuple[float, float]:
        """Отримання пікового співвідношення сигнал/шум для різних
зображень."""
        return ImageComparison.psnr(self.ref, self.res), ImageComparison.psnr(self.ref,
self.pred)

    def get_ssim(self) -> Tuple[float, float]:
        """Отримання структурної подібності для різних зображень."""
        return ImageComparison.ssim(self.ref, self.res), ImageComparison.ssim(self.ref,
self.pred)

```