

Київський національний університет імені Тараса Шевченка

Інституту високих технологій

Завідувач кафедри нанофізики конденсованих середовищ

проф. Валерій Антонович Скришевський

Протокол № _____ засідання кафедри

від “ _____ ” _____ 2021 р.

**МАГНІТНО-РЕЗОНАНСНА РЕКОНСТРУКЦІЯ
СТАТИСТИЧНИМИ МЕТОДАМИ ДЛЯ РІЗНИХ ЗА
ЗВАЖЕНІСТЮ ТОМОГРАМ**

Магістерська робота

студента 2 курсу ОР «Магістр»

105 «Прикладна фізика та наноматеріали»

Лещенка Олександра Кийовича

Науковий керівник

к.ф.-м.н., доц. **Нетребя А.В.**

Оцінка захисту роботи

Київ - 2021

Зміст

| | |
|---|----|
| ВСТУП..... | 4 |
| 1. ФІЗИЧНІ ОСНОВИ ПОБУДОВИ ПРОСТОРОВИХ СПІНОВИХ ХАРАКТЕРИСТИК | 6 |
| 1.1 Актуальні дослідження в галузі МР-візуалізації..... | 6 |
| 1.2 Основи магнітно-резонансної томографії..... | 7 |
| 1.3 Побудова магнітно-томографічного зображення..... | 9 |
| 2. ТИПИ ДІАГНОСТИЧНИХ МЕДИЧНИХ ЗОБРАЖЕНЬ В ЯДЕРНІЙ МАГНІТНО-РЕЗОНАНСНІЙ ВІЗУАЛІЗАЦІЇ..... | 12 |
| 2.1 Різновиди МР-зображень за зваженістю..... | 12 |
| 3. ВІДНОВЛЕННЯ СПІНОВИХ РОЗПОДІЛІВ..... | 15 |
| 3.1 Побудова матриць сигналу дослідження її властивостей | 15 |
| 4. МАГНІТНО РЕЗОНАНСНА РЕКОНСТРУКЦІЯ СТАТИСТИЧНИМИ МЕТОДАМИ ДЛЯ РІЗНИХ ЗА СПІНОВОЮ ЗВАЖЕНІСТЮ ТОМОГРАМ..... | 21 |
| 4.1 Аналіз одновимірних та двовимірних розподілів | 21 |
| 4.2 Застосування методу реконструкції до зображень різного типу зваженості..... | 24 |
| 4.3 Деякі деталі програмної реалізації на мові Python..... | 27 |
| ВИСНОВКИ | 28 |
| ПЕРЕЛІК ПОСИЛАНЬ..... | 29 |
| Додаток А | 31 |
| Додаток Б..... | 39 |

Реферат

Бакалаврська робота: 42 с., 11 рис., 2 додатки, 10 джерел

Розглянуто теоретичні основи та принципи утворення зображення у магнітно-резонансній томографії. Наведено методи обробки томографічних зображень та оптимізації роботи з ними. Запропоновано власну реалізацію алгоритму відновлення зображення з фазо-частотного простору написану на програмній мові С. Якість роботи алгоритму оцінено за допомогою коефіцієнта кореляції цифрових зображень. Реалізовано метод пучка матриць із застосуванням його до реконструкції томографічних знімків. Викладені особливості застосування методу до цифрових зображень та порівняно якість застосування методу до зображень різного типу за зваженістю.

Ключові слова: ШВИДКЕ ПЕРЕТВОРЕННЯ ФУР'Є, ПРОТОННА ГУСТИНА, РЕКОНСТРУКЦІЯ МР-ЗОБРАЖЕНЬ, МЕТОД МАТРИЧНИХ ПУЧКІВ

ВСТУП

Основною метою дослідження є вивчення можливості застосування альтернативних спектральних методів в області цифрової обробки зображень та побудови інструменту для їх використання. Областю застосування є магнітно-резонансна томографія (МРТ). В першій частині викладається огляд фізичних основ та явищ на яких базується МРТ та яким чином стає можливим отримати інформацію про внутрішню будову тіла неінвазивним чином. У наступному розділі розглядається математичний апарат на якому базується метод матричних пучків та особливості його застосування до відновлення цифрових зображень, а саме МР-знімків. Створені програмні модулі, які можуть слугувати інструментом обробки великих масивів зображень оптимізованими алгоритмами. Викладене теоретичне обґрунтування методу матричних пучків та деталі його реалізації на програмній мові Python з застосуванням сучасних прикладних пакетів для наукових досліджень, а саме NumPy, SciPy та Pandas. Отримані результати відновлених зображень описаним методом. На основі отриманих результатів реконструкції зроблене порівняння початкового зображення та реконструйованого. Для візуалізації впродовж всієї роботи застосовувався програмний пакет Matplotlib. Запропоновані критерії оцінки якості реалізації цих алгоритмів на конкретних зображеннях. Отже, дана робота дає уявлення про підходи до реконструкції МРТ зображень та пропонує практичну реалізацію методів спектрального аналізу зображень.

Мета роботи: побудова програмного пакету для спектральної обробки цифрових зображень та реконструкції магнітно-томографічних знімків.

1. ФІЗИЧНІ ОСНОВИ ПОБУДОВИ ПРОСТОРОВИХ СПІНОВИХ ХАРАКТЕРИСТИК

1.1 Актуальні дослідження в галузі МР-візуалізації

Сучасний рівень медичної техніки дозволяє виявляти структурні і функціональні зміни одного і того ж об'єкту за допомогою пристроїв, які мають різний принцип роботи, при цьому достовірність даних буде порівняно однаковою. В подібних умовах на перший план виходить інформаційна складова досліджень.

На даному етапі одним з найбільш інформативних методів є томографія, яка дає набагато більше інформації про кожен елементарний об'єм досліджуваного об'єкта, ніж інші відомі методи діагностики. Існує декілька видів томографії: рентгенівська, електронно-променева, магнітно-резонансна, позитронно-емісійна, ультразвукова, оптична когерентна томографія та інші. Але суть усіх методів томографії одна: по сумарній інформації (наприклад, інтенсивності на детекторах або інтенсивності ехо-сигналу), отриманій від деякого зрізу речовини в кожній точці зрізу. Інформативність і достовірність кожного з них залежить від цілого ряду факторів, визначаючих результат дослідження, в тому числі і від фізичного принципу дії пристрою.

Якість томографічних зображень залежить від багатьох факторів: апаратного і програмного забезпечення, фізичних процесів, що беруть участь у зборі даних, параметрів сканування, властивостей речовини (густини, процесів релаксації дифузії), температури оточуючого середовища, перепади якої призводять до підвищення рівня шуму на зображенні і зниженню точності вимірювань.

Доволі часто при аналізі томографічних зображень доводиться стикатись з артефактами, які можуть призвести до постановки невірною діагнозу. В зв'язку з цим актуальними є методи і засоби, які дозволяють підвищити якість і достовірність зображень.

1.2 Основи магнітно-резонансної томографії

Магнітно-резонансна томографія (МРТ) - це томографічний метод дослідження внутрішніх органів і тканин з використанням фізичного явища ядерного магнітного резонансу (ЯМР). Метод ґрунтується на вимірюванні електромагнітного відклику атомних ядер, найчастіше ядер атомів водню, а саме на їхньому збудженні за допомогою певної комбінації електромагнітних хвиль у сталому магнітному полі високої напруженості. Цей метод дає змогу одержати висококонтрастне зображення тканин тіла, і тому його широко застосовують у медицині, у візуалізації тканин мозку, серця, м'язів, а також новоутворень, порівняно з іншими методами медичної візуалізації (такими, наприклад, як комп'ютерна томографія чи рентгенографія).

Вперше явище ЯМР було спостерігалось незалежно двома групами американських вчених фізиків, керівниками яких були Фелікс Блох та Едвард Парсел у 1945р [1]. Вони спостерігали ЯМР в твердих тілах та вперше отримали сигнали ядерного магнітного резонансу, продемонструвавши явище ЯМР в блочних матеріалах, за що обидва в 1952р. були удостоєні Нобелівської премії з фізики.

Для проведення ЯМР-досліджень потрібно помістити об'єкт в сильне магнітне поле, статичне та однорідне у просторі, яке створює в середині тканин зображуваного об'єкта макроскопічну намагніченість. Частіше за все в МРТ використовують протони водню з двох причин:

високої чутливості до МР-сигналу та їх високому вмісту в біологічних тканинах.

У постійному магнітному полі ядра обертаються навколо силових ліній поля. Це обертання називають прецесією. Воно характеризується певною частотою для різних типів ядер і залежить від напруженості магнітного поля. Відстань між енергетичними рівнями залежить від магнітного моменту ядра, індукції магнітного поля. Під час дії високочастотного імпульсу, якщо його частота збігається з частотою прецесії ядер, ці ядра переходять у збуджений стан. Це і є явище ЯМР. Проте такі переходи ядер у збуджений стан з поглинанням енергії відбуваються до певного моменту, до насичення. Якщо припинити подачу високочастотного сигналу, то ядра атомів речовини зразка будуть повертатися на нижчий рівень, випромінюючи енергію.

$$\Delta E = \hbar\omega_0 = \gamma\hbar B_0,$$

де γ - гіромагнітне співвідношення (залежить від розміру і форми ядра), $\hbar = h / 2\pi$ – стала Планка, ω_0 – частота електромагнітного випромінювання, необхідна для переходу з стану в стан [3].

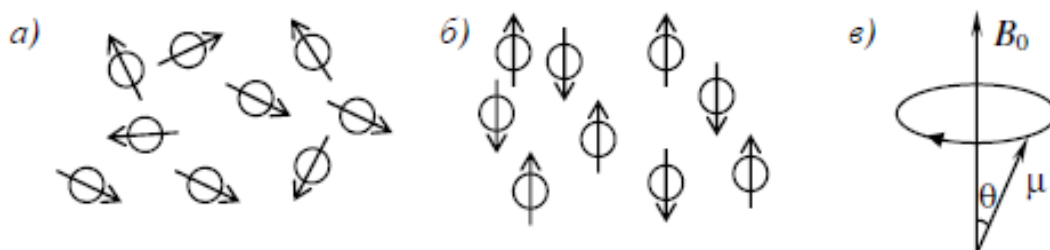


Рис 1.1 Розподіл магнітних моментів за відсутності (а) присутності (б) зовнішнього магнітного поля і прецесія магнітного моменту

На практиці для отримання сигналу від ядер необхідно опромінити об'єкт, поміщений в постійне магнітне поле, додатковим

радіочастотним-полем(РЧ).Якщо частота РЧ-сигналу співпадає з параметрами ядра та магнітного поля, то виникає резонанс – атоми елементу поглинають енергію імпульсу і переходять на більш високий рівень.

Після припинення дії РЧ-імпульсу створений магнітними моментами ядер сумарний вектор намагніченості M_z , відхилений від напрямку силових ліній основного поля, повертається в вихідний стан M_0 . Таким чином, після РЧ-імпульсу повздовжня складова намагніченості M_z повертається в стан M_0 , а поперечна намагніченість M_{xy} - приймає нульове значення. Під час цього процесу, названого релаксацією, ядра, що поглинули енергію випромінюють слабкі електромагнітні хвилі.

1.3 Побудова магнітно-томографічного зображення

МР-зображення по суті являється розрахованою картою або зображенням РЧ-сигналів, випромінюваних тілом людини. Сигнал являє собою одночасне отримання компонент намагніченості M_x та M_y як функції часу і реєструється за допомогою двох окремих каналів датчика, які даються інформацію компонент сигналу (амплітуду, фазу, частоту). В цьому фазочутливому методі комплексний демодульований сигнал розділений на 2 компоненти: дійсну і уявну, зміщену на 90° відносно першого. Сигнали обох каналів об'єднуються в один набір квадратурних дійсних та уявних спектрів та далі обробляються за допомогою перетворення Фур'є.

Кожна точка матриці «сирих» даних (k-простір) містить частину інформації про зображення і не відповідає точці матриці зображення; k-простір еквівалентний простору, визначеному напрямками фази та

частоти, кожна лінія даних якого відповідає оцифрованому МР-сигналу з унікальним рівнем кодування фази (рис. 1.2).

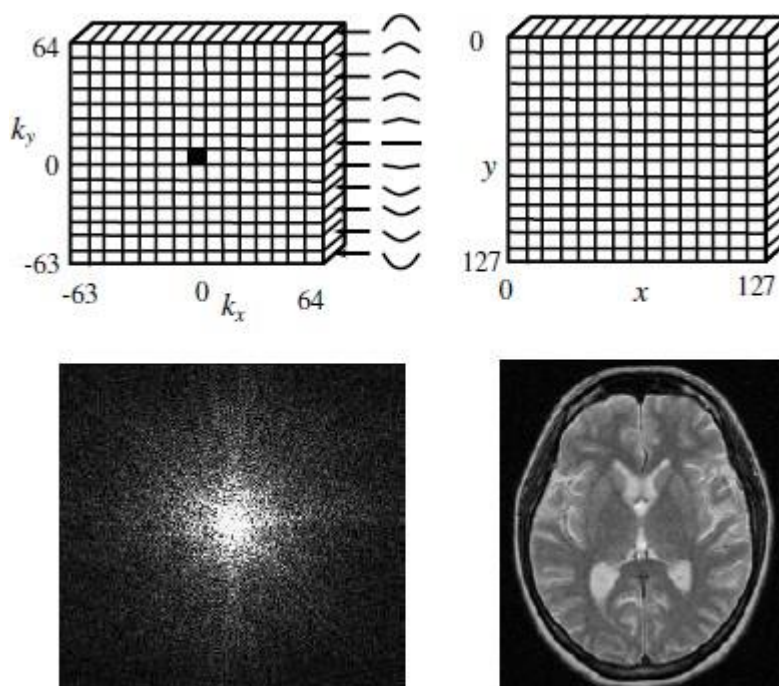


Рис 1.2 К-простір и реконструйоване зображення

Оскільки дані двох половин k -простору комплексно спряжені (сигнал має дійсну та уявну частину), то його зображення має симетричний вигляд. Корисно помітити, що k -простір крім корисного сигналу містить шум, який збільшується до країв матриці даних та спотворює їх симетричність. Центральна частина k -простору містить основну інформацію про об'єкт та визначає загальний контраст зображення. Зовнішні ряди матриці k -простору відповідають високочастотним складовим даних і несуть інформацію про границі і контури зображення або окремих структур, визначають роздільну здатність малих деталей. Також в області високих частот знаходяться дані, що відповідають шуму зображення.

Траєкторія k-простору – доріжка, в просторова-частотній області при зборі даних і визначена прикладеними градієнтами. Заповнення k-простору може виконуватись по рядкам або по спіралі в залежності від прикладених градієнтів та вибраних алгоритмів збору даних.

Інтенсивність кожного елементу МР-зображення пропорційна інтенсивності сигналу від відповідного елементу об'єму 3D простору для даної товщини зрізу. Розмір пікселя може бути менше фактичної просторової роздільної здатності і визначається розміром вибраної області простору і матрицею зображення. Пікселі часто використовуються для вимірів роздільної здатності(або точності) зображення [1].

2. ТИПИ ДІАГНОСТИЧНИХ МЕДИЧНИХ ЗОБРАЖЕНЬ В ЯДЕРНІЙ МАГНІТНО-РЕЗОНАНСНІЙ ВІЗУАЛІЗАЦІЇ

2.1 Різновиди МР-зображень за зваженістю

Інтенсивність зчитуваного МР-сигналу визначається чотирма основними параметрами :

- 1) протонною густиною (кількість протонів у досліджуваній тканині);
- 2) часом спін-граткової релаксації T_1 ;
- 3) часом спін-спінової релаксації T_2 ;
- 4) рухом або дифузією досліджуваних структур.

Після дії РЧ імпульсу поперечна складова намагніченості деякий час прецесує навколо напрямку основного поля, що обумовлено ефектами релаксації. Механізм релаксації повертає систему в її початковий рівноважний стан. Таким чином, після дії РЧ імпульсу повздовжня складова намагніченості M_z в напрямку статичного магнітного поля повертається в стан рівноваги M_0 в відповідності до постійної часу T_1 , а вектор поперечної намагніченості M_{xy} повертається в нульове значення.

Поперечна спін-спінова T_2 релаксація описує процес повернення намагніченості M_{xy} в рівноважний стан відповідно до сталої часу T_2 і залежить від обміну енергією між сусідніми спінами.

В кожному МР-зображенні існує контраст, обумовлений к протонною густиною, так і T_1 та T_2 релаксацією. В залежності від їх співвідношення говорять про T_1 - зважені зображення, T_2 -зважені або про зображення протонної густини (proton density), які також називають середньозваженими.

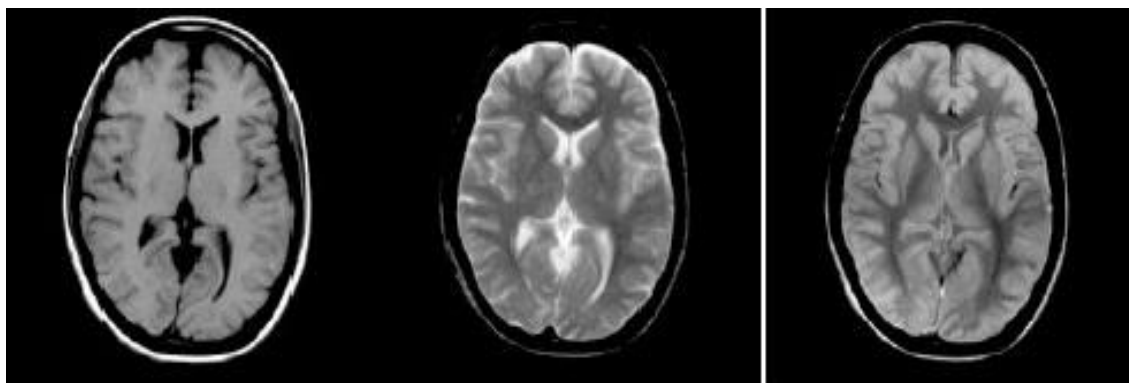


Рис. 2.1 зважене за T_1 (а), за T_2 (б) та за протонною густиною (в) зображення мозку

Спеціально розроблені імпульсні послідовності визначають вклад того чи іншого параметра в інтенсивність сигналу зображення для отримання оптимального контрасту між нормальними та паталогічними тканинами. Імпульсна послідовність (ІП) - це набір радіочастотних і градієнтних імпульсів заданої форми, амплітуди і інтервалу між ними, що повторюється багато раз під час сканування і визначають контраст зображення. Іншими словами, імпульсна послідовність це комп'ютерна програма, що керує сканером в процесі сканування.

Основними технічними параметрами ІП є час повтору (TR), час ехо (TE), час інверсії (TI), кут відхилення. В клінічній практиці використовують ІП які можна розділити на певні групи:

- 1) спін-ехо;
- 2) швидке спін-ехо;
- 3) градієнтне ехо;
- 4) інверсія відновлення.

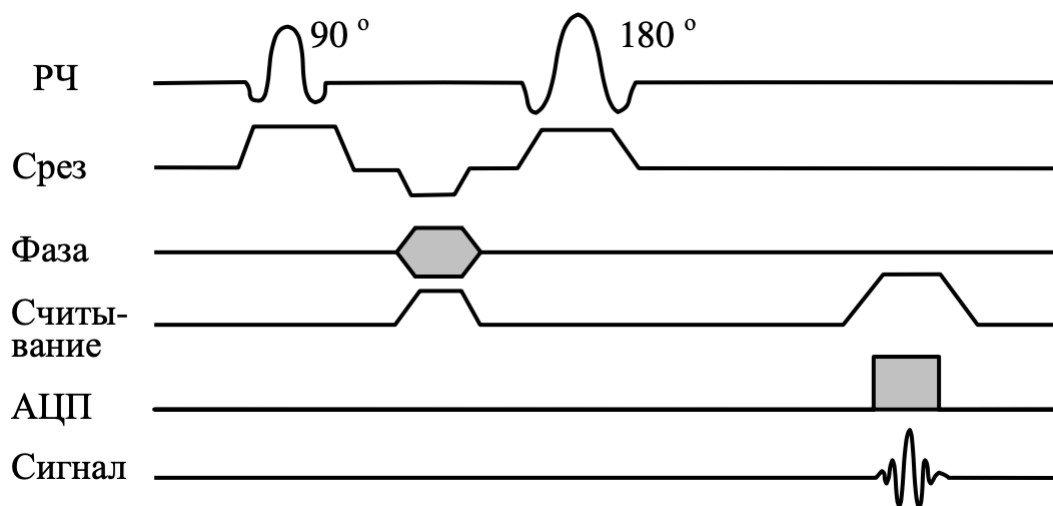


Рис 2.2 Приклад імпульсної послідовності спін-ехо

При створенні зображення зваженого за T_1 зазвичай часи TR та TE довші, а для T_2 навпаки використовують коротші. На T_2 кістковий мозок такої ж інтенсивності або більш інтенсивний ніж м'язи. При паталогічних процесах, як правило, збільшується кількість води в тканинах, що призводить до зниження інтенсивності сигналу на T_1 -зважених зображеннях і збільшення інтенсивності сигналу на T_2 – зважених зображеннях.

3. ВІДНОВЛЕННЯ СПІНОВИХ РОЗПОДІЛІВ

3.1 Побудова матриць сигналу дослідження її властивостей

Метод пучка матриць дозволяє розв'язувати задачу апроксимації дискретних даних $\vec{s} = [s_0, s_1, \dots, s_{N-1}]^T$, що становлять для нас інтерес, моделлю у вигляді суми комплексних експонент

$$s_n = \sum_{m=1}^M a_m \exp(j\omega_n t_n). \quad (3.1)$$

Метод забезпечує знаходження одного с параметрів моделі (3.1) – вектора $\vec{\omega} = [\omega_1, \omega_2, \dots, \omega_M]^T$ зі значеннями частот $\omega_m = 2\pi f_m$, що входить у модель нелінійним чином; при цьому передбачається, що амплітуди $\vec{a} = [a_1, a_2, \dots, a_M]^T$ є детермінованими, але невідомими постійними; M – порядок моделі. Параметр $\vec{\omega}$ входить у модель лінійно та може бути знайдений за допомогою стандартної процедури лінійного методу найменших квадратів. Здатність забезпечувати одержання параметрів моделі (3.1) відносить метод пучка матриць до класу методів параметричного спектрального аналізу. Передбачається, що значення $\vec{s} = [s_0, s_1, \dots, s_{N-1}]^T$ задані на дискретній еквідистантній сітці в моменти часу $t_n = t_0 + n\Delta t$, де Δt – крок дискретизації,

$n = 0, 1, \dots, N - 1$... Якщо прийняти, що $t_0 = 0$, та ввести позначення

$$z_m = \exp(j\omega_m \Delta t), \quad (3.2)$$

то розглянута модель у вигляді суми комплексних експонент може бути представлена в такому вигляді:

$$s_n = \sum_{m=1}^M a_m \exp(j\omega_m n \Delta t) = \sum_{m=1}^M a_m z_m^n \quad (3.3)$$

У цьому випадку шуканим вектором параметрів моделі можна вважати вектор $\vec{z} = [z_1, z_2, \dots, z_M]^T$.

У матричному вигляді застосовувана для апроксимації аналізованих даних модель може бути представлена як $\vec{s}_M = \mathbf{E}\vec{a}$, де \mathbf{E} – матриця експонент із M стовпцями $\vec{e}_m^N = [1, z_m, \dots, z_m^{N-1}]^T$, що містять N відліків експонент:

$$\mathbf{E} = [\vec{e}_1^N, \vec{e}_2^N, \dots, \vec{e}_M^N] = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ z_1^{N-1} & \dots & z_M^{N-1} \end{bmatrix} \quad (3.4)$$

Тоді задача апроксимації аналізованих даних \vec{s} адитивною експоненційною моделлю \vec{s}_M може бути зведена до задачі визначення значень вектора \vec{z} . Оцінка шуканого параметра \vec{z} , що отримана за однією реалізацією, з точки зору статистики є вибірковою. Щоб її усереднити процедуру оцінювання можна застосувати до сукупності вибірок, зроблених з вихідної послідовності \vec{s} .

Для реалізації зазначеного усереднення в класі матричних методів параметричного спектрального аналізу прийнято на базі вихідного сигналу \vec{s} формувати матрицю даних \mathbf{S} з L сегментів

$\vec{s}_i = [s_i, s_{i+1}, \dots, s_{i+K-1}]^T$ вихідної послідовності, що перекриваються, мають довжину K та зміщених один щодо одного на один відлік

($i = 0, 1, \dots, L-1$):

$$\mathbf{S} = [\vec{s}_0, \vec{s}_1, \dots, \vec{s}_{L-1}]^T = \begin{bmatrix} s_0 & \dots & s_{L-1} \\ \vdots & \ddots & \vdots \\ s_{K-1} & \dots & s_{N-1} \end{bmatrix} \quad (3.5)$$

Якщо вихідний сигнал повністю визначається сумою зважених експонент (3.3) то така матриця даних може бути представлена кістяковим розкладанням через матриці експонент:

$$\mathbf{S} = \mathbf{E}_K \mathbf{A} \mathbf{E}^T = \sum_{m=1}^M \mathbf{a}_m \vec{e}_m^K \vec{e}_m^L{}^T \quad (3.6)$$

Використання кістякового розкладання (3.6) дозволяє представити псевдообернену матрицю даних

$$\mathbf{S}^+ = (\mathbf{S}^H \mathbf{S})^{-1} \mathbf{S}^H \quad (3.7)$$

Таке подання матриці даних та її псевдооберненої матриці дозволяє показати, що обчислена з їхньою допомогою матриця проєктора

$$\mathbf{P} = \mathbf{S} \mathbf{S}^+ = \mathbf{E}_K \mathbf{E}_K^+ = \mathbf{E}_K (\mathbf{E}_K^H \mathbf{E}_K)^{-1} \mathbf{E}_K^H \quad (3.8)$$

визначає проєктор на підпростір експонент розмірності M («підпростір сигналу»). Власні значення матриці \mathbf{P} лежать у підпросторі експонент і це може бути використане для визначення їх частотних параметрів. Власні вектори матриці дорівнюють одиниці та не можуть бути використані для розв'язку цієї задачі. Більш інформативними є власні значення матриці зсуву даних експоненціальної моделі на відлік, що використовуються в методі пучка матриць.

Матриця \mathbf{S} також може бути представлена за допомогою свого сингулярного розкладання

$$\mathbf{S} = \mathbf{U} \mathbf{D} \mathbf{V}^H = \sum_{k=1}^M \sigma_k \mathbf{u}_k \mathbf{v}_k^H \quad (3.9)$$

де $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_K]$ - унітарна матриця лівих сингулярних векторів розміром $K \times K$, $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_L]$ - унітарна матриця правих сингулярних векторів розміром $L \times L$. Зв'яже їх квадратна діагональна матриця \mathbf{D} розміром $K \times L$, що містять сингулярні числа σ_k .

В загальному випадку розмір матриці лівих сингулярних векторів дорівнює $K \times K$, причому $K \geq M$. Тому, виконуючи впорядкування сингулярних чисел за убутанням $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_K$, прийнято розділяти вихідну матрицю сингулярних векторів $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_K]$ на сигнальну $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_M]$, сформовану базисом

підпростору сигналу, та шумову $\mathbf{U} = [\vec{u}_1, \vec{u}_2, \dots, \vec{u}_K]$, яка визначається ортогональним до сигналу підпростором шуму.

Таким чином використання сингулярного розкладання дозволяє виконати редукцію матриці даних або псевдооберненої до неї матриці шляхом усікання матриці сингулярних векторів з метою видалення шумових компонент і виділення сигнального підпростору.

Пучком матриць для деякої матричної пари $\{\mathbf{S}_0, \mathbf{S}_1\}$ називають конструкцію вигляду $\mathbf{S}_1 - z\mathbf{S}_0$, де z – параметр пучка. Метод пучка матриць заснований на пошуку узагальнених власних значень матричної пари $\{\mathbf{S}_0, \mathbf{S}_1\}$, що формується за допомогою сегментації вихідного сигналу $\vec{s} = [s_0, s_1, \dots, s_{N-1}]^T$. Для формування матричної пари з вихідної матриці даних \mathbf{S} виділяються дві матриці даних, що мають на один стовпець (сегмент) менше:

$$\mathbf{S}_0 = [\vec{s}_0, \vec{s}_1, \dots, \vec{s}_{L-1}] = \mathbf{E}_K \mathbf{A} \mathbf{E}_L^T; \quad \mathbf{S}_1 = [\vec{s}_1, \vec{s}_2, \dots, \vec{s}_L] = \mathbf{E}_K \mathbf{Z} \mathbf{A} \mathbf{E}_L^T \quad (3.10)$$

Друга матриця даних починається із другого сегмента й тому в її кістяковому розкладанні через матриці експонент повинна бути присутня матриця, що дає зсув на один відлік, а саме діагональна матриця $\mathbf{Z} = \text{diag}(z_1, z_2, \dots, z_M)$, утворена значеннями відношення двох сусідніх відліків сигнальних експонент $z_m = \exp(j\omega_n \Delta t)$.

За допомогою сформованих таким чином матриць даних у розгляд вводиться пучок цих матриць $\mathbf{S}_1 - z\mathbf{S}_0$, що з урахуванням (3.10) може бути представлений у вигляді $\mathbf{S}_1 - z\mathbf{S}_0 = \mathbf{E}_K (\mathbf{Z} - z\mathbf{I}) \mathbf{A} \mathbf{E}_L^T$. Аналіз показує, що простір стовпців матриць \mathbf{S}_1 і \mathbf{S}_0 є однаковим. Отже ранг матриці пучка дорівнює рангу її складових, за винятком $z = z_m$, тому що в цьому випадку m -ий рядок матриці $\mathbf{Z} - z\mathbf{I}$, обертається в нуль. Тому для узагальненої задачі на власні значення пучка матриць $\mathbf{E}_K (\mathbf{Z} - z\mathbf{I}) \mathbf{A} \mathbf{E}_L^T \vec{p}_m = 0$, розв'язок буде досягтися при $z = z_m$. Тут \vec{p}_m –

це узагальнений власний вектор розглянутого пучка матриць, що відповідає власному числу z_m . Виходячи з цього, можна зробити висновок, що числа $\{z_1, z_2, \dots, z_m\}$, які дають інформацію про частоти сигнальних експонент, будуть являти собою набір узагальнених власних значень матричної пари $\{\mathbf{S}_0, \mathbf{S}_1\}$. Розв'язок узагальненої проблеми для пучка матриць, що відповідає власному числу можна знайти, знайшовши власні значення вигляду

$$\mathbf{S}_0^+ \mathbf{S}_1 \vec{\mathbf{p}}_m = z_m \vec{\mathbf{p}}_m \quad (3.11)$$

де $\mathbf{S}_0^+ = (\mathbf{S}_0^H \mathbf{S}_0)^{-1} \mathbf{S}_0^H$, - це псевдообернена до \mathbf{S}_0 матриця. Задача (3.11)

визначає пошук власних векторів і значень матриці $\mathbf{P}_{01} = \mathbf{S}_0^+ \mathbf{S}_1$, таким чином метод пучка матриць зводиться до розв'язку задачі на власні значення $\mathbf{P}_{01} \vec{\mathbf{p}}_m = z_m \vec{\mathbf{p}}_m$.

Насамперед псевдообернену матрицю \mathbf{S}_0^+ в методі знаходять за допомогою сингулярного розкладання відповідної матриці

$\mathbf{S}_0^+ = \mathbf{V} \mathbf{D}^{-1} \mathbf{U}^H$, \mathbf{U} – усічена матриця лівих сингулярних векторів розміру $K \times M$, \mathbf{V} – усічена матриця правих сингулярних векторів розміру $L \times M$, $\mathbf{D} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_M)$ – діагональна матриця сингулярними числами розміру $M \times M$. Використовуючи отриману таким чином матрицю \mathbf{S}_0^+ , матрицю \mathbf{P}_{01} , щодо якої розглядається задача на власні значення, можна обчислити як $\mathbf{P}_{01} = \mathbf{S}_0^+ \mathbf{S}_1 = \mathbf{V} \mathbf{D}^{-1} \mathbf{U}^H \mathbf{S}_1$. Потім за рахунок множення на усічену матрицю правих сингулярних векторів вихідна задача на власні значення перетворюється на задачу на власні значення розмірності моделі сигналу M

$$\mathbf{G} = \mathbf{V} \mathbf{P}_{01} \mathbf{V} = \mathbf{D}^{-1} \mathbf{U}^H \mathbf{S}_1 \mathbf{V} \quad (3.12)$$

Саме на розв'язку цієї задачі в методі пучка матриць засноване знаходження шуканих значень $\{z_1, z_2, \dots, z_m\}$, що дозволяють

визначити частоти $\omega_m = \ln z_m / j \Delta t$, гармонійних коливань за відліками $\{s_0, s_1, \dots, s_{N-1}\}$, які їх представляють. Використовуючи знайдені значення z_m для обчислення матриці експонент \mathbf{E} із (3.4), можна легко знайти значення вектора амплітуд $\vec{\mathbf{a}} = [a_1, a_2, \dots, a_m]^T$, що доповнюють шукані параметри моделі, застосувавши для цього метод найменших квадратів щодо невідомих амплітуд

$$\vec{\mathbf{a}} = (\mathbf{E}^H \mathbf{E})^{-1} \mathbf{E}^H \vec{\mathbf{s}}. \quad (3.13)$$

4. МАГНІТНО РЕЗОНАНСНА РЕКОНСТРУКЦІЯ СТАТИСТИЧНИМИ МЕТОДАМИ ДЛЯ РІЗНИХ ЗА СПІНОВОЮ ЗВАЖЕНІСТЮ ТОМОГРАМ

4.1 Аналіз одновимірних та двовимірних розподілів

Для імітації роботи магнітно-резонансного томографа використано швидке двовимірне перетворення Фур'є реалізоване на програмній мові С. Таким чином вхідне зображення перетворюється з просторового двовимірного сигналу у фазо-частотний простір, що є певним аналогом вихідного сигналу магнітно-резонансного сканування.

Спершу зчитане зображення було нормоване на проміжку значень $[0, 1]$. Далі взявши один рядок, було вирішено задачу відновлення сигналу рядка томографічного зображення.

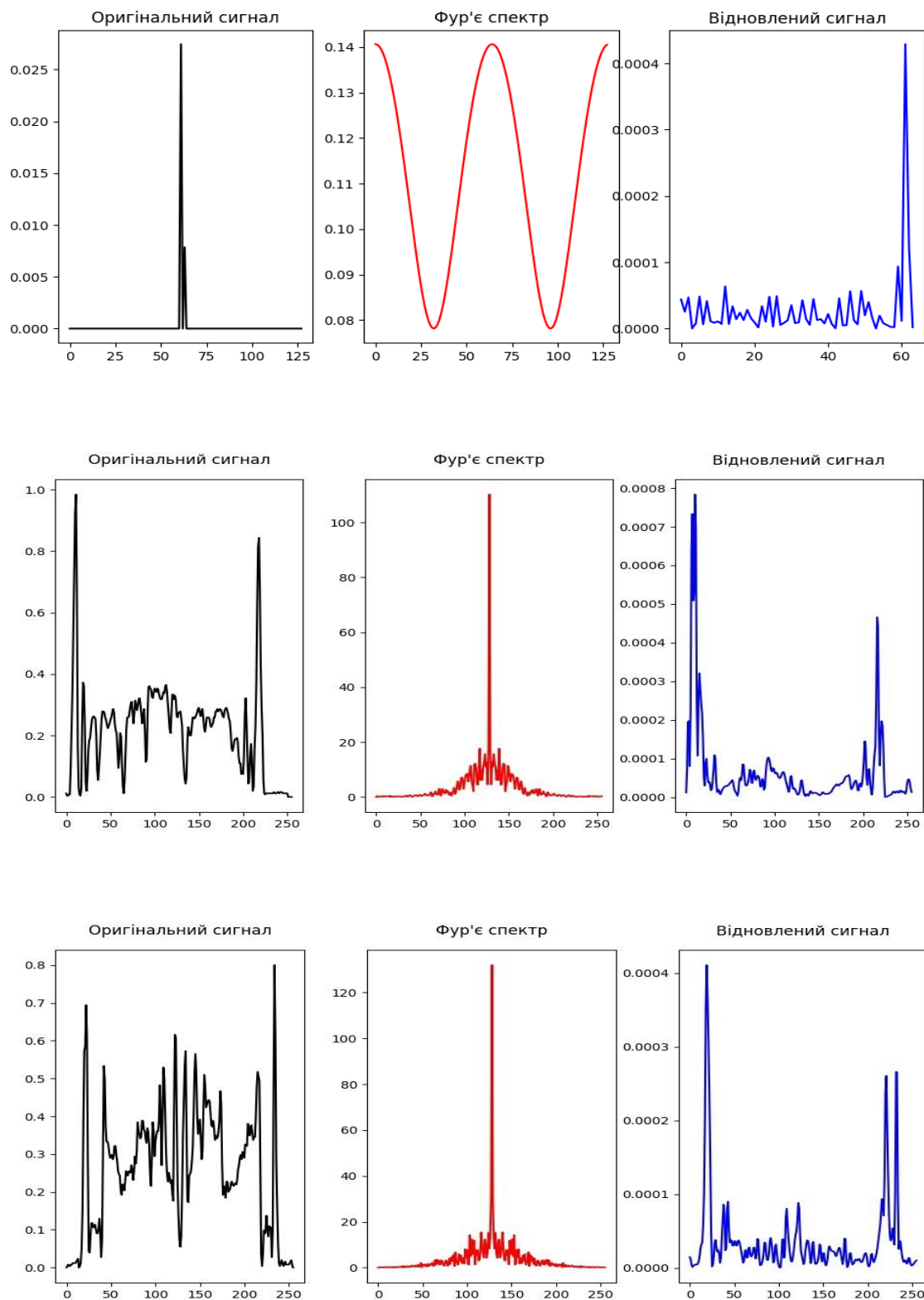


Рис 4.1 Приклади відновлених рядків

На графіках що розташовані зліва (див. рис. 4.1) відображений оригінальний сигнал з томографічного знімку, в центрі - Фур'є спектр цього сигналу, в правій частині – відновлений за допомогою методу матричних пучків сигнал.

З результатів добре видно, що метод відновлює первинну інформацію закладену в зображення, хоча має значні спотворення щодо складових з меншими амплітудами.

Наступним кроком була поставлена задача відновлення справжнього знімку томографічного знімку. Був запропонований порядковий спосіб застосування методу до масиву спектру зображення, що дало наступні результати (див. рис. 4.2 та 4.3).

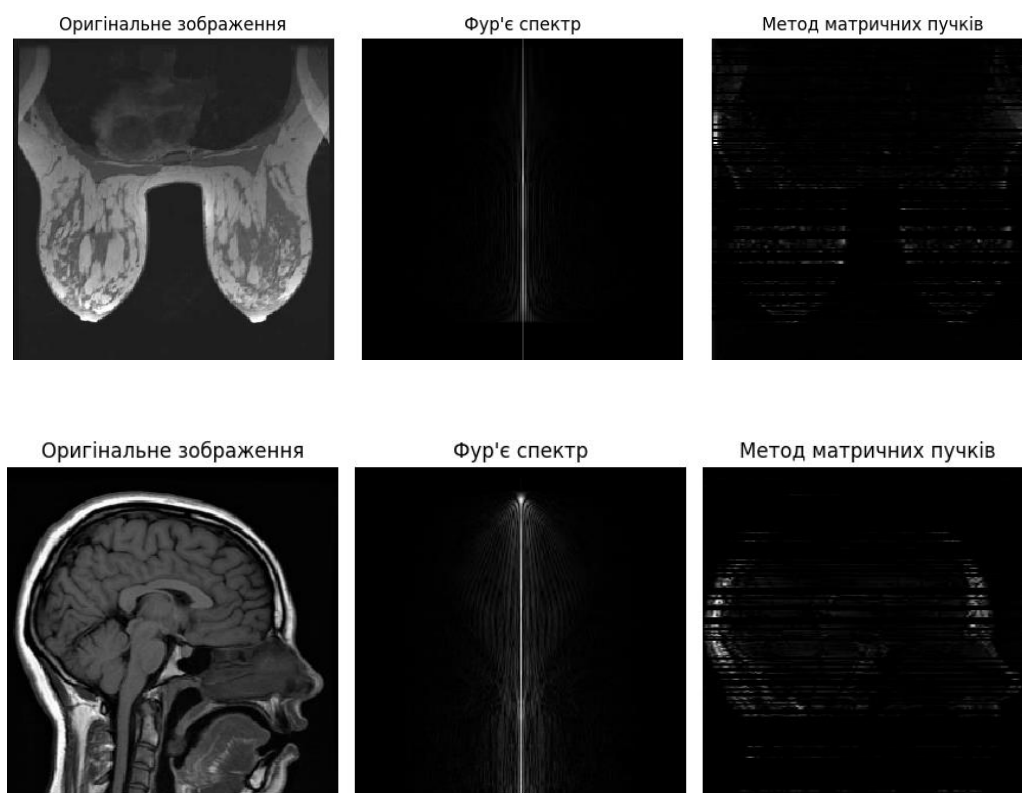


Рис. 4.2 Приклади зображень до яких застосовано алгоритм відновлення

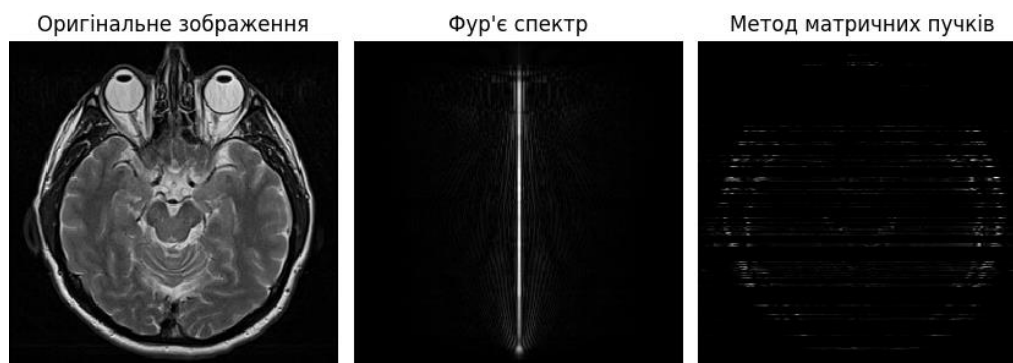


Рис. 4.3 Приклад реконструкції іншої томограми

В наведених прикладах зображень значно краще відновлюються загальні контури аніж дрібні деталі. Це можна пояснити тим, що зображення як двовимірний сигнал має набір з безлічі частот, що значно ускладнює роботу для методу матричних пучків, оскільки він значно краще працює з вузькосмуговими сигналами.

4.2 Застосування методу реконструкції до зображень різного типу зваженості

Далі розглянуто можливості реконструкції МР-зображень для різних типів зваженості, а саме: середньозважені (протонна густина), зважені за T_1 та зважені за T_2 (рис. 4.4 - 4.6). З огляду на результат реконструкції можна сказати, що алгоритм не має особливої чутливості до конкретного типу зваженості. Тому можна говорити про деяку універсальність щодо типу відновлюваного зображення.

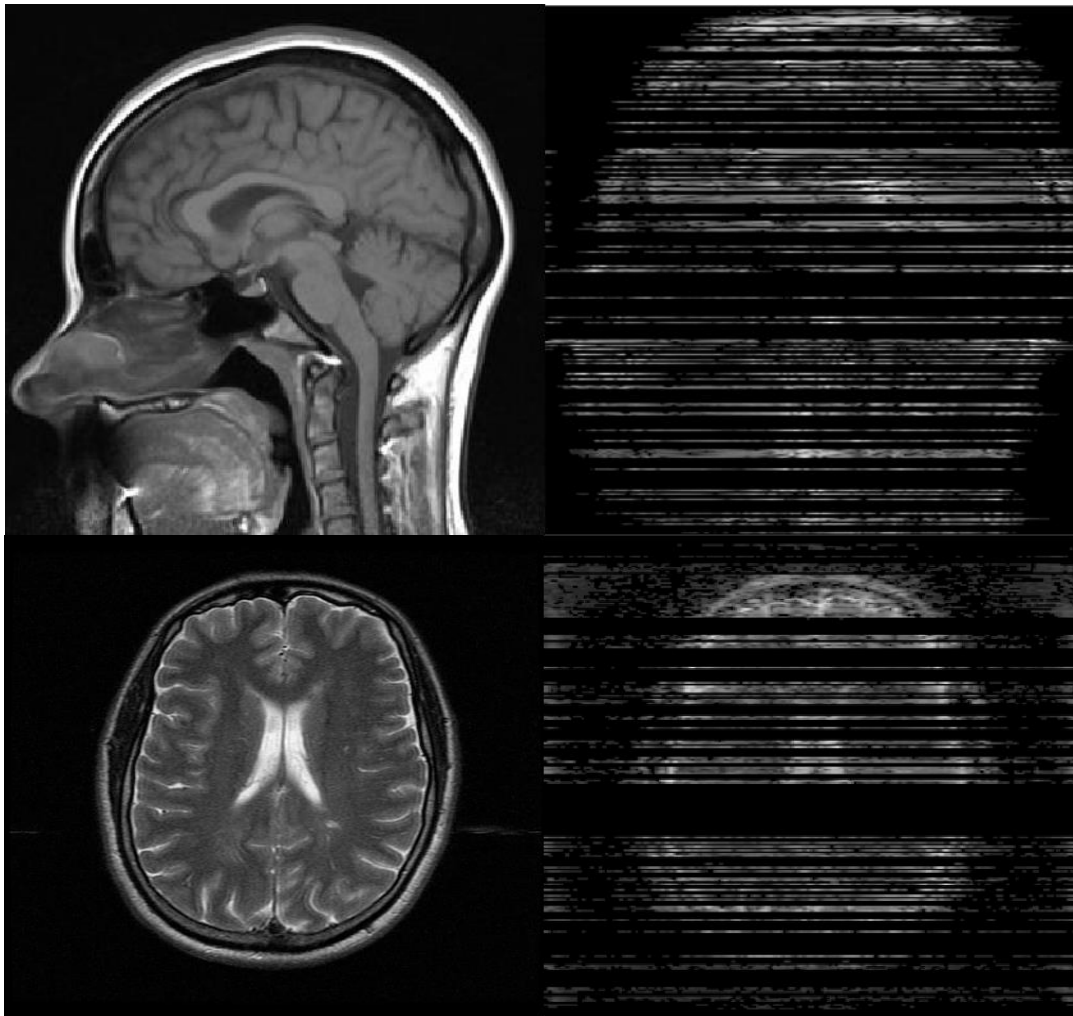


Рис. 4.4 Приклади реконструкції зображення зваженого за протонною густиною

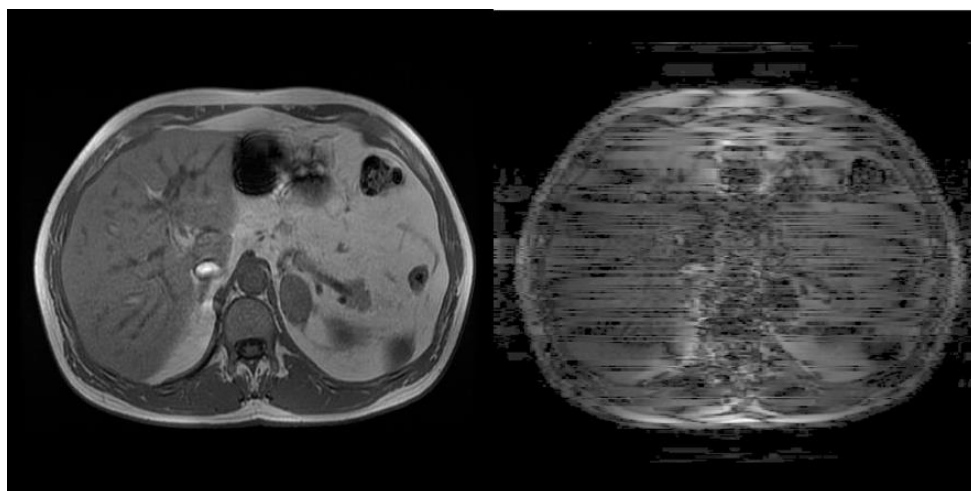


Рис. 4.5 Приклади реконструкції зображення зваженого за T_1

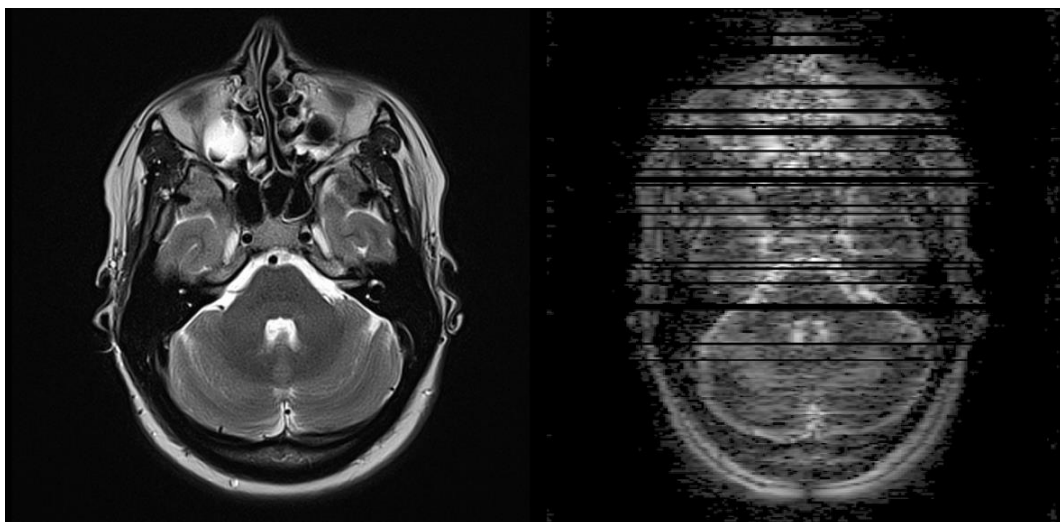


Рис. 4.6 Приклад реконструкції зображення зваженого за T_2

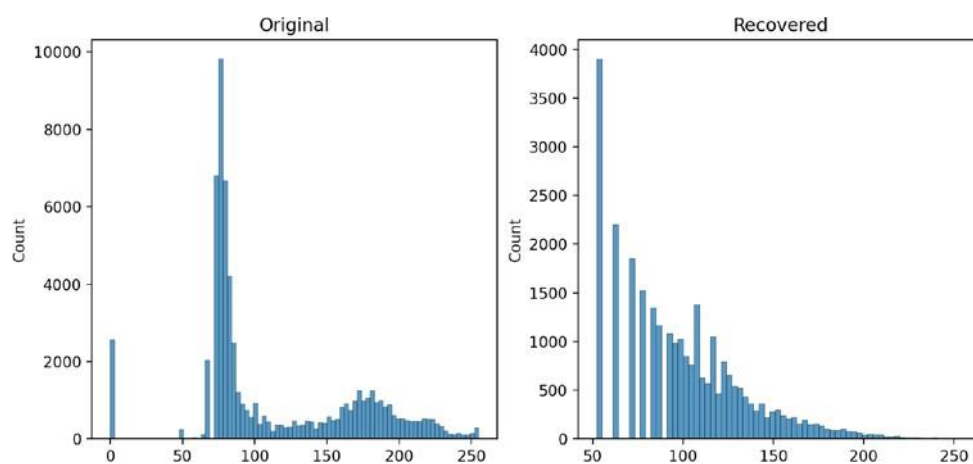


Рис. 4.7 розподіл інтенсивності у початковому та відновленому зображенні

Проте порівнюючи розподіл інтенсивності в оригінальному та у відновленому зображенні, видно що існує деяка втрата інформації, що і є причиною спотворення в результаті реконструкції. Хоча варто зазначити що більшість присутньої в початкових даних інформації зберігається. Подальші покращення методу можливі шляхом збільшення розмірів реконструйованих знімків та вирішення проблеми переповнення числових типів при деяких перетвореннях.

4.3 Деякі деталі програмної реалізації на мові Python

Програмна реалізація методу матричних пучків виконана за допомогою мови Python 3.7. Оскільки для спектральної обробки зображень потрібний певний специфічний набір інструментів, був зроблений вибір на користь застосування бібліотек для наукових розрахунків NumPy та SciPy останніх версій на момент написання роботи. Для візуалізації графіків та зображень використовувався графічний пакет Matplotlib, що має широкі та гнучкі можливості налаштування. Приклади зображень використані в роботі були знайдені у вільному доступі у відповідних джерелах таких як openmri.org.

ВИСНОВКИ

При виконанні роботи реалізовані етапи:

- вивчені особливості методів магнітно-резонансної візуалізації просторових розподілів спінової густини, часів спін-спінової та спін-граткової релаксації;
- реалізована реконструкція томограм за умов оптимізації процедури вирішення оберненої задачі;
- продемонстрована можливість реконструкції одновимірних сигналів зі збереженням їх початкової форми;
- проведено оцінювання якості відновлення томограми в залежності від типу імпульсної послідовності за допомогою якого формувалось зображення
- запропонована методика оцінювання просторових розподілів спінових характеристик за допомогою одного з методів параметричного спектрального аналізу;
- реалізована процедура обробки двовимірних розподілів шляхом послідовного аналізу лінійних масивів, отримане із плоских перерізів;
- продемонстрована універсальність методу щодо типу відновлюваного МР-зображення;
- проаналізовано гістограму розподілу інтенсивності у початковому та відновленому двовимірному сигналі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Марусина М.Я Сучасні види томографії. Навчальний посібник. / А.О. Казначєєва, – / СПбДУ ІТМО, 2006. – 152 с.
2. Марплд-молодший Цифровий спектральний аналіз і його застосування. Навчальний посібник /. – / Пер. з англ. — Москва: Мир, 1990. – 584 с.
3. Холін О.В. Магнітно-резонансна томографія при захворюваннях і травмах центральної нервової системи / О.В. Холін, – М.: МЕДпрес-інформ., 2017. – 256 с.
4. Вербицький Є. В. Швидке перетворення Фур'є модульованих сигналів, представлених рядом Фур'є двох змінних / Є. В. Вербицький // Вісник НТУ «ХПІ», Серія: Нові рішення в сучасних технологіях. – Харьків: НТУ «ХПІ». – 2018. – № 16 (1292). – С. 102-106. – doi:10.20998/2413-4295.2018.16.15
5. Черногор М. С. Моделювання систем комп'ютерного відтворення та розпізнавання цифрових зображень. / Максим Черногор 2018 – 42 с.
6. Ібряєва О. Л. Модифікація методу матричних пучків, з використанням спільного оцінювання полюсів сигналу і обернених до них / О. Л. Ібряєва, Д. Д. Салов. – Челябинськ: Вісник ЮУрГУ, 2017. – 37 с. – (Обчислювальна математика та інформатика; № 1).
7. Тапан С. Використання методу матричних пучків для оцінки параметрів сум комплексних експонент [Електронний ресурс] / С. Тапан, О. Перейра // Сиракузьський університет. – 1995. – Режим доступу до ресурсу: http://krein.unica.it/~cornelis/private/IEEE/IEEEAntennasPropagMag_37_48.pdf.

8. Хорнак Д. Основи МРТ [Електронний ресурс] / Джозеф П. Хорнак // Російський державний медичний університет. – 2007. – Режим доступу до ресурсу: <https://www.cis.rit.edu/htbooks/mri/inside-r.htm>.
9. Zhou, B. and Zhou, S.K., 2020. DuDoRNet: Learning a dual-domain recurrent network for fast MRI reconstruction with deep T1 prior. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4273-4282).
10. Triphan, S.M., Breuer, F.A., Gensler, D., Kauczor, H.-U. and Jakob, P.M. (2015), Oxygen enhanced lung MRI by simultaneous measurement of T1 and T2* during free breathing using ultrashort TE. J. Magn. Reson. Imaging, 41: 1708-1714. <https://doi.org/10.1002/jmri.24692>

Додаток А

Код власного програмного пакету:

1. Метод для обрахунку швидкого перетворення Фур'є

```
#include "header.h"

static int n2ln( int n )
{
    int ln = 0;
    while (n >>= 1)
        ln++;
    return (ln);
}

static void w_mac(w_type* cc, w_type a, w_type w, w_type b)
{
    cc->real = a.real + w.real * b.real - w.im * b.im;
    cc->im = a.im + w.real * b.im + w.im * b.real;
}

void fft0(w_type *InOut, int n, int mode)
{
    int i;
    w_type w, *A, *B;

    if (n == 1)
        return ;
    A = (w_type*)malloc(sizeof(w_type) * n / 2 );
    B = (w_type*)malloc(sizeof(w_type) * n / 2 );
    for (i = 0; i < n / 2; i++)
    {
        A[i] = InOut[i * 2];
        B[i] = InOut[i * 2 + 1];
    }
    fft0( A, n / 2 , mode);
    fft0( B, n / 2 , mode);
    for (i = 0; i < n; i++)
    {
        w.real = cos(2 * M_PI * i / n);
```

```

    w.im = mode * sin(2 * M_PI * i / n);
    w_mac(&InOut[i], A[i % (n / 2)], w, B[i % (n / 2)]);
}
free(A);
free(B);
}

```

2. Методи запису та зчитування bmp-файлів

```
#include "header.h"
```

```

void impl_saveBMP (IMPL_PIXELMAP *image_info, const char
*filename)
{

```

```

    FILE * pFile = fopen (filename, "wb");
    if (pFile == NULL)
        return;

```

```

    uint16_t tmpUInt16 = 0;
    uint32_t tmpUInt32 = 0;
    int32_t tmpInt32 = 0;

```

```

    fputc ('B', pFile);
    fputc ('M', pFile);

```

```

    tmpUInt32 = 14U + 40U + image_info->biWidth *
                image_info->biHeight * 32U;
    fwrite (&tmpUInt32, sizeof(uint32_t), 1, pFile);

```

```

    fputc (0, pFile);
    fputc (0, pFile);
    fputc (0, pFile);
    fputc (0, pFile);

```

```

    tmpUInt32 = 14 + 40;
    fwrite (&tmpUInt32, sizeof(uint32_t), 1, pFile);

```

```

    tmpUInt32 = 40;
    fwrite (&tmpUInt32, sizeof(uint32_t), 1, pFile);

```

```

    tmpInt32 = (int32_t) image_info->biWidth;
    fwrite (&tmpInt32, sizeof(int32_t), 1, pFile);

```

```

tmpInt32 = (int32_t) image_info->biWidth;
fwrite (&tmpInt32, sizeof(int32_t), 1, pFile);

tmpUInt16 = 1;
fwrite (&tmpUInt16, sizeof(uint16_t), 1, pFile);

tmpUInt16 = image_info->biBitCount;
fwrite (&tmpUInt16, sizeof(uint16_t), 1, pFile);

tmpUInt32 = 0;
fwrite (&tmpUInt32, sizeof(uint32_t), 1, pFile);

tmpUInt32 = 0;
fwrite (&tmpUInt32, sizeof(uint32_t), 1, pFile);

tmpInt32 = 2835;
fwrite (&tmpInt32, sizeof(int32_t), 1, pFile);

tmpInt32 = 2835;
fwrite (&tmpInt32, sizeof(int32_t), 1, pFile);

tmpUInt32 = 0;
fwrite (&tmpUInt32, sizeof(uint32_t), 1, pFile);

tmpUInt32 = 0;
fwrite (&tmpUInt32, sizeof(uint32_t), 1, pFile);

size_t i = 0;
for (; i < image_info->biWidth * image_info->biHeight; ++i)
{
    fwrite (&(image_info->data[i].b), sizeof(uint8_t), 1, pFile);
    fwrite (&(image_info->data[i].g), sizeof(uint8_t), 1, pFile);
    fwrite (&(image_info->data[i].r), sizeof(uint8_t), 1, pFile);
}

fclose (pFile);
}

IMPL_PIXELMAP *impl_loadBMP (const char *filename) {
    FILE *pFile;

```

```

IMPL_PIXELMAP *image_info;

pFile = fopen (filename, "rb");
    if (pFile == NULL)
        return NULL;

    image_info = (IMPL_PIXELMAP *) malloc
(sizeof(IMPL_PIXELMAP));
    if (image_info == NULL)
        return NULL;

    fread (&image_info->bfType, sizeof(uint8_t), 2, pFile);
    if (image_info->bfType[0] != 'B' || image_info->bfType[1] != 'M')
        return NULL;

    fread(&image_info->bfSize, sizeof(uint32_t), 1, pFile);

        fread(&image_info->bfReserved, sizeof(uint16_t), 2, pFile);
        fread (&image_info->pixelDataOffset, sizeof(uint32_t), 1, pFile);

        fread(&image_info->biSize, sizeof(uint32_t), 1, pFile);
        fread (&image_info->biWidth, sizeof(int32_t), 1, pFile);
        fread (&image_info->biHeight, sizeof(int32_t), 1, pFile);

        fread(&image_info->biPlanes, sizeof(uint16_t), 1, pFile);
        fread (&image_info->biBitCount, sizeof(uint16_t), 1, pFile);

        fread(&image_info->biCompression, sizeof(uint32_t), 1, pFile);
        fread(&image_info->biSizeImage, sizeof(uint32_t), 1, pFile);
        fread(&image_info->biXPelsPerMeter, sizeof(int32_t), 1, pFile);
        fread(&image_info->biYPelsPerMeter, sizeof(int32_t), 1, pFile);

        fread(&image_info->biClrUsed, sizeof(uint32_t), 1, pFile);
        fread(&image_info->biClrImportant, sizeof(uint32_t), 1, pFile);

    if (image_info->biBitCount == 8)
    {
        image_info->data = (IMPL_PIXEL*)malloc(image_info->biWidth *
image_info->biHeight * sizeof(IMPL_PIXEL));
        if (!image_info->data)

```

```

    {
        fclose (pFile);
        return NULL;
    }

    fseek (pFile, image_info->pixelDataOffset, SEEK_SET);
    size_t i = 0;
    for (; i < image_info->biWidth * image_info->biHeight; ++i)
    {
        fread(&image_info->data[i].r, sizeof(uint8_t), 1, pFile);
        image_info->data[i].b = 0;
        image_info->data[i].b = 0;
    }
    fclose (pFile);

    return (image_info);
}

if (image_info->biBitCount == 24)
{
    image_info->data = (IMPL_PIXEL*)malloc(image_info->biWidth *
image_info->biHeight * sizeof(IMPL_PIXEL));
    if (!image_info->data)
    {
        fclose (pFile);
        return NULL;
    }

    fseek (pFile, image_info->pixelDataOffset, SEEK_SET);
    size_t i = 0;
    for (; i < image_info->biWidth * image_info->biHeight; ++i)
    {
        fread(&image_info->data[i].r, sizeof(uint8_t), 1, pFile);
        fread(&image_info->data[i].g, sizeof(uint8_t), 1, pFile);
        fread(&image_info->data[i].b, sizeof(uint8_t), 1, pFile);
    }
    fclose (pFile);

    return (image_info);
}

    fclose (pFile);
return NULL;
}

```

3. Хедер файл

```

#ifndef HEADER_H
#define HEADER_H

#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include <string.h>
#include <math.h>

#define STRAIGHT_MODE -1
#define INVERSE_MODE 1

typedef struct _IMPL_COLOR_TABLE_CELL
{
    uint8_t r,g,b, reserved;
}          COLOR_CELL;

typedef struct _IMPL_PIXEL
{
    uint8_t r, g, b;
}          IMPL_PIXEL;

typedef struct _IMPL_PIXELMAP {
    //BITMAPFILEHEADER
    uint8_t bfType[2];
    uint32_t bfSize;
    uint16_t bfReserved[2];
    uint32_t pixelDataOffset;

    //BITMAPINFOHEADER
    uint32_t biSize;
    int32_t biWidth, biHeight;
    uint16_t biPlanes;
    uint16_t biBitCount;
    uint32_t biCompression;
    uint32_t biSizeImage;
    int32_t biXPelsPerMeter;
    int32_t biYPelsPerMeter;
    uint32_t biClrUsed;
    uint32_t biClrImportant;
}

```

```

    COLOR_CELL *color_table;
    uint32_t *pixel_data;
    IMPL_PIXEL *data;

}          IMPL_PIXELMAP;

// impl_pixel functions
IMPL_PIXELMAP *impl_copyPixelmap(IMPL_PIXELMAP * source);
void impl_freePixelmap(IMPL_PIXELMAP *);

IMPL_PIXEL *impl_createPixel_RGB(uint8_t r, uint8_t g, uint8_t b,
uint8_t a);
void impl_freePixel(IMPL_PIXEL *);
IMPL_PIXEL *impl_getPixel(IMPL_PIXELMAP *, size_t, size_t);
IMPL_PIXEL *impl_setPixel(IMPL_PIXELMAP * pixelmap, size_t x,
size_t y,
    uint8_t r, uint8_t g, uint8_t b, uint8_t a, uint8_t grey);

// impl_bmp functions
void impl_saveBMP (IMPL_PIXELMAP*, const char *);
IMPL_PIXELMAP *impl_loadBMP (const char *);

// FFT function

typedef struct
{
    double real;
    double im;
} w_type;

void fft0(w_type InOut[], int n, int mode);

#endif

```

4. Мейн програми для перетворення зображень:

```

#include "header.h"

int main (int argc, char **argv)
{
    printf ("Loading BMP: %s\n", argv[1]);

```

```

IMPL_PIXELMAP *picture = impl_loadBMP (argv[1]);
printf ("BMP loaded\nSaving BMP: %s\n", argv[1]);

int size = picture->biWidth * picture->biHeight;
w_type *signal = (w_type*)malloc(sizeof(w_type) * size);
// FFT
for (int i = 0; i < size; i++)
{
    signal[i].real = picture->data[i].r;
    signal[i].im = 0;
}

fft0(signal, size, STRAIGHT_MODE);
for (int i = 0; i < size; i++)
{
    picture->data[i].r = (uint8_t)(0.5 * (signal[i].real * signal[i].real
        + signal[i].im * signal[i].im));
    picture->data[i].g = (uint8_t)(0.5 * (signal[i].real *
        signal[i].real + signal[i].im * signal[i].im));
    picture->data[i].b = (uint8_t)(0.5 * (signal[i].real *
        signal[i].real + signal[i].im * signal[i].im));
}
impl_saveBMP (picture, "../images/FFT_brain.bmp");

// FFT(-1)
fft0(signal, size, INVERSE_MODE);
for (int i = 0; i < size; i++)
{
    printf("%f\n", signal[i].real / size);
    picture->data[i].r = (uint8_t)(signal[i].real / size);
    picture->data[i].g = (uint8_t)(signal[i].real / size);
    picture->data[i].b = (uint8_t)(signal[i].real / size);
}
impl_saveBMP (picture, "../images/FFT(-1)_brain.bmp");

return (0);}

```

1. Реалізація методу пучка матриць на програмній мові Python3.7 з допомогою бібліотек NumPy, SciPy та CV2:

```

def matrix_beam(X):
    N = X.shape[1]
    L = N // 2
    K = N - L
    M = L
    print(f"N: {N}, L: {L}, K: {K}, M:{M}")

    recovered = []

    for i in range(X.shape[0]):
        x = X[i, :]

        if np.sum(x == 0) > 0.95 * x.size:
            recovered.append(np.zeros(M).tolist())
            continue

        S = linalg.hankel(x[:K], x[K - 1:])
        S_0 = S[:, :L]
        S_1 = S[:, 1:L + 1]

        U, D, V = np.linalg.svd(S_0, full_matrices=False)

        V = np.conj(V.T)

        # D = D[:-1]
        # U = U[:, :-1]
        # V = V[:, :-1]

        G = np.linalg.multi_dot([np.linalg.inv(np.diag(D)),
np.conj(U.T), S_1, V])
        eigenvalues = np.linalg.eigvals(G)

        freq = np.angle(eigenvalues)
        indexes = freq.argsort()[::-1]

        freq = freq[indexes]
        eigenvalues = eigenvalues[indexes]

```

```

E = np.zeros((N, M), dtype=np.complex128)

for k in range(N):
    for m in range(M):
        E[k, m] = eigenvalues[m] ** k

magnitude = np.linalg.lstsq(E, x, rcond=None)[0]

# print(f"line {i}\t{magnitude}")
recovered.append(magnitude.tolist())
return np.array(recovered)

```

2. Мейн програми з використанням методу пучка матриць для штучного комплексного сигналу з двома гармоніками:

```

if __name__ == "__main__":
    N = 500
    delta_t = 5 * 10**(-6)

    n = np.linspace(0.0, 20, N)

    signal = np.exp(1j * 0.2 * np.pi * n) + np.exp(1j * 0.8 * np.pi * n)

    magnitude, freq = matrix_beam(signal)

    print(f"\n\nmagnitude: {magnitude}, frequency: {freq}")

    // outputs magnitude: [1.-0.0026j 1.+0.0026j],
    // frequency: [2.0408e-06 5.1021e-07]

```

3. Мейн програми для відновлення рядка томографічного зображення:

```

if __name__ == "__main__":
    IMAGE_DIR = "./"
    IMAGE_NAME = "brain.png"
    IMG_SIZE = 128

    image_array = cv2.imread(os.path.join(IMAGE_DIR,
                                           IMAGE_NAME),
                             cv2.IMREAD_GRAYSCALE)

```

```

image_array = cv2.resize(image_array, (IMG_SIZE,
                                       IMG_SIZE))
image_array = image_array.astype(np.float) / 255.0
# scale = 1/(IMG_SIZE*IMG_SIZE)

fft_image = np.zeros(image_array.shape,
                     dtype=np.complex128)

for i in range(image_array.shape[0]):
    fft_image[i] = np.fft.fftshift(np.fft.fft(image_array[i])) / IMG_SIZE

recovered = matrix_beam(fft_image, abs_or_ln=True)

fft_power_spectr = 2 * np.abs(fft_image * 255)
recovered_image = 2 * np.abs(recovered)

nrow = 100

plot_row([image_array[nrow], fft_power_spectr[nrow],
          recovered_image[nrow]])

plot_images([image_array, fft_power_spectr,
             recovered_image])

```

4. Мейн програми для відновлення зображення:

```

if __name__ == "__main__":
    images_path = ["breast.jpg", "face.png", "other_brain.jpg"]
    # images_path = ["high_res.jpg"]

    for IMAGE_NAME in images_path:
        IMAGE_DIR = "./images/"
        IMG_SIZE = 256
        image_array=cv2.imread(os.path.join(IMAGE_DIR,IMAGE_NAME),
                               cv2.IMREAD_GRAYSCALE)
        image_array = cv2.resize(image_array, (IMG_SIZE, IMG_SIZE))
        image_array = image_array.astype(np.float) / 255.0

        fft_image = np.zeros(image_array.shape, dtype=np.complex128)

        for i in range(image_array.shape[0]):
            fft_image[i] = np.fft.fftshift(np.fft.fft(image_array[i])) /
                IMG_SIZE

```

```
recovered = matrix_beam(fft_image)

fft_power_spectr = 2 * np.abs(fft_image * 255)
# recovered_image = 2 * np.abs(recovered)
recovered_image = np.abs(np.real(recovered))

flipped_recovered_image = np.zeros(recovered_image.shape)
half = recovered_image.shape[1] // 2

flipped_recovered_image[:, half:] = recovered_image[:, :half]
flipped_recovered_image[:, :half] = recovered_image[:, half:]

flipped_recovered_image = cv2.resize(flipped_recovered_image,
                                     (IMG_SIZE, IMG_SIZE))
plot_images({"Оригінальне зображення": image_array,
            "Фур'є спектр": fft_power_spectr,
            "Метод матричних пучків": flipped_recovered_image,
            })
```