

Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»  
Физтех-школа Прикладной Математики и Информатики  
Кафедра интеллектуальных систем

**Направление подготовки / специальность:** 03.04.01 Прикладные математика и физика

**Направленность (профиль) подготовки:** Математическая физика, компьютерные технологии и математическое моделирование в экономике

## **ВОССТАНОВЛЕНИЕ ФАЗЫ МИКРОСКОПИЧЕСКИХ ИЗОБРАЖЕНИЙ ПО ОБРАТНОЙ РЕШЕТКЕ**

(магистерская диссертация)

**Студент:**

Кириллов Егор Дмитриевич

---

*(подпись студента)*

**Научный руководитель:**

Стрижов Вадим Викторович,  
д-р физ.-мат. наук

---

*(подпись научного руководителя)*

**Консультант (при наличии):**

---

*(подпись консультанта)*

Москва 2021

## Аннотация

Рассмотрена задача восстановления фазы изображения по обратной решетке. Требуется получить изображение объекта по его дифракционной картине. В рентгенокопии фокусировка излучения является трудной задачей, что приводит к искажению изображения дифракционными эффектами. Поскольку в результате дифракции излучения, падающего на объект, в области детектора теряется фаза изображения, необходимо восстановить фазу. Для этого применяются специальные итерационные алгоритмы. Полагается, что использование моделей машинного обучения позволит упростить процесс восстановления фазы и улучшить получаемое изображение. Рассмотрены различные алгоритмы восстановления фазы, а также предложены модели машинного обучения для решения задачи восстановления фазы. Получены результаты работы алгоритмов восстановления фазы и моделей машинного обучения. Была обучена сеть, повторяющая работу преобразования Фурье, с помощью которой восстановлена фаза изображения. Обучить сверточную сеть восстанавливать исходное изображение по дифракционной картине не удалось, необходимы дальнейшие исследования.

## Содержание

1. Введение
2. Обзор
3. Постановка
4. Теория
  - 4.1. Автоэнкодер
  - 4.2. Сверточная сеть
  - 4.3. Алгоритм Герхберга-Сакстона
  - 4.4. Алгоритм уменьшения ошибки
  - 4.5. Функции ошибки
  - 4.6. Дифракция
  - 4.7. Симуляция дифракции
  - 4.8. Рентгеновская и электронная микроскопия
  - 4.9. Обратная решетка
  - 4.10. Преобразование Фурье
  - 4.11. Дискретное преобразование Фурье
5. Эксперимент
6. Заключение
7. Список литературы

## Введение

Это исследование нацелено на изучение возможности применения методов машинного обучения в задачах восстановления фазы изображения. Существующие методы решения этих задач достаточно сложны в реализации, поэтому создание конкурентоспособной модели помогло бы упростить исследования с применением рентгеноסקопии.

В рентгеноструктурном анализе применяется метод, основанный на дифракции рентгеновских лучей на кристаллических решетках или отдельных молекулах, поэтому лучи, при прохождении через образец не фокусируются на экране. Вместо изображения мы получаем дифракционную картину, которая сама по себе не дает представления об объекте исследования. Для получения полноценного микроскопического изображения, необходимо восстановить часть потерянной информации, а именно фазу изображения, поскольку на экране присутствует только распределение интенсивностей рентгеновского излучения. Существуют различные алгоритмы восстановления фазы изображения. Одним из самых известных среди них является алгоритм Герхберга-Сакстона.

Однако большинство этих алгоритмов являются или слишком неточными, или слишком сложными в реализации и требуют настройки разных параметров и априорных знаний об объекте. Появляется возможность повысить точность результатов восстановления фазы изображения по сравнению с точностью результатов применения существующих алгоритмов, а также скорость работы, с помощью методов машинного обучения.

Предполагается использование стека автоэнкодеров, вариационный автоэнкодер, а также другие, более простые модели для решения этой задачи.

## Обзор

1) T. Beuvier, I. Probert, L. Beaufort, B. Suchéras-Marx, Y. Chushkin, F. Zontone & A. Gibaud: X-ray nanotomography of coccolithophores reveals that coccolith mass and segment number correlate with grid size – <https://www.nature.com/articles/s41467-019-08635-x#data-availability>

Статья, содержащая изображения с рентгеновского микроскопа. Авторы использовали трехмерную когерентную рентгеновскую дифракцию на основе синхротрона для исследования кокколитов 7 видов. Была применена трехмерная рентгеновская когерентная дифракционная визуализация - это новый синхротронный метод, основанный на преобразовании Фурье численно-фазированной картины когерентного рассеяния с избыточной дискретизацией в дальнем поле, которая обеспечивает трехмерное распределение электронной плотности изолированных объектов.

2) Tatiana Latychevskaia: Iterative phase retrieval in coherent diffractive imaging: practical issues – <https://arxiv.org/ftp/arxiv/papers/1809/1809.04626.pdf>

Работа о восстановлении фазы изображения в когерентной дифракционной визуализации. Содержит описание алгоритма Герхберга-Сакстона и его улучшений, а также описание проблем при восстановлении фазы и способы их решений.

3) Sergei Grudinin, Maria Garkavenko, Andrei Kazennov: Pepsi-SAXS : an adaptive method for rapid and accurate computation of small-angle X-ray scattering profiles – <https://hal.inria.fr/hal-01516719>

Статья с описанием алгоритма восстановления изображений в когерентной дифракционной томографии Pepsi-Saxston. Это современный алгоритм, обладающий хорошей точностью.

4) O. Cherkas, T. Beuvier Orcid, Dag W. Breiby, Y. Chushkin, F. Zontone and A. Gibaud: Direct Observation of Microparticle Porosity Changes in Solid-State Vaterite to Calcite Transformation by Coherent X-ray Diffraction Imaging – <https://pubs.acs.org/doi/abs/10.1021/acs.cgd.7b00476>

Статья о наблюдении трансформации твердого фатерита в кальцит с помощью метода когерентной рентгеновской дифракции.

5) Christopher M. Bishop: Pattern Recognition and Machine Learning – <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>

Учебник по машинному обучению, содержащий теоретическое описание моделей машинного обучения.

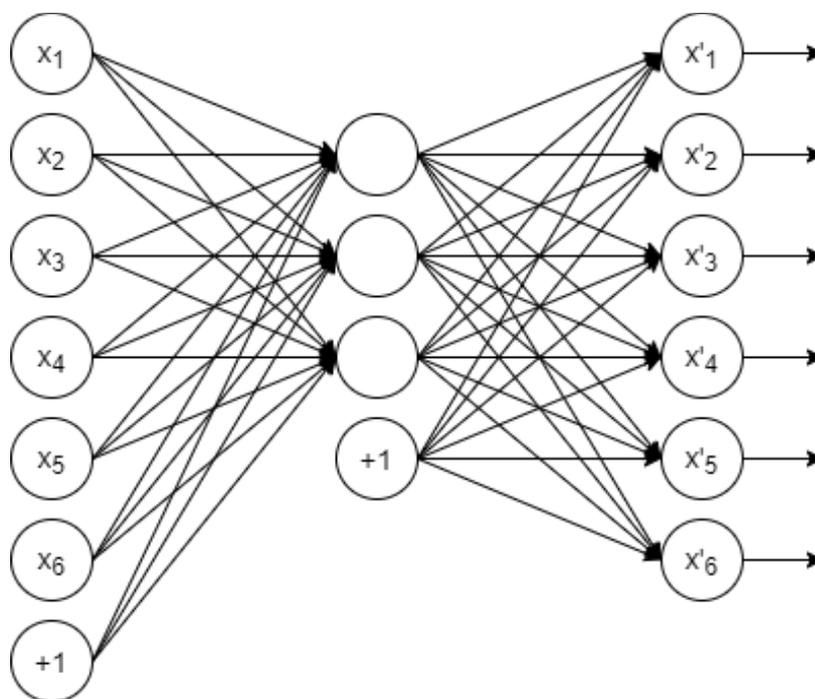
### **Постановка**

Имеется пара изображений, где есть обратная и прямая кристаллическая решетка. Необходимо восстановить изображение кристаллической решетки по обратному изображению. Для решения задачи предполагается использование автокодировщика и вариационного автокодировщика.

В алгоритме Герхберга-Сакстона, который является алгоритмом восстановления фазы изображения, применяется последовательно прямое и обратное преобразования Фурье, а в промежуточных этапах накладываются ограничения на интенсивность изображения. В данном алгоритме применяется дискретное преобразование Фурье. Оно трансформирует дискретно взятые значения функции в последовательность преобразований Фурье на соответствующих отрезках. Так же имеются ограничения, а именно совпадение амплитуды распределения с интенсивностью на экране, а также априорные знания о распределении объекта изучения. Это и представляет возможным использование автокодировщика в качестве модели.

## Теория

### Автокодировщик



Изображение слоев автоэнкодера(рис.1)

Автоэнкодер – это алгоритм обучения. Этот алгоритм нашел широкое применение в задачах обучения без учителя, однако он применяется и в задачах, где известен целевой вектор. В данном алгоритме используется нейронная сеть и метод обратного распространения ошибки, чтобы результат работы алгоритма равнялся входному вектору признаков, в случае обучения без учителя, или же целевому вектору, в случае обучения с учителем. Чтобы избежать тривиальных результатов необходимо обозначить два ограничения, накладываемые на архитектуру сети:

1. В скрытом слое мы будем активировать не все нейроны. Это означает, что в неактивные нейроны данные подаваться не будут.
2. Количество нейронов в этом слое мы будем ограничивать, чтобы их было меньше, чем количество нейронов в предыдущем слое.

С помощью этих ограничений автоэнкодер добивается сжатия входящих данных, а именно понижения их размерности, поскольку в первом случае размерность регулируется количеством одновременно активных нейронов, а во втором случае общим количеством нейронов в скрытом слое. Однако же снижение размерности подразумевает наличие корреляций и взаимосвязей в исходных данных. Таким образом сжимая и восстанавливая данные можно получить интересные результаты. Кроме того, автоэнкодер позволяет уменьшать зашумленность данных.

Автоэнкодер состоит из кодировщика и декодировщика. Кодировщик уменьшает размерность данных, а декодировщик пытается восстановить исходные данные. Вычислить значения промежуточного слоя можно следующим образом:

$$h = f(Wx + b), \tag{1}$$

где  $f$  – нелинейная функция,  $W$  – матрица с обучаемыми параметрами, а  $b$  – байес.

Вектор промежуточного слоя затем преобразуется следующим образом:

$$z = f(W'h + b), \quad (2)$$

где  $W'$ ,  $b'$  – это новая матрица и байес, которые не имеют ничего общего с  $W$ ,  $b$ . Но все-таки иногда применяются матрицы, транспонированные по отношению к начальной:  $W' = W^T$ . Функцией ошибки можно выбрать:

$$L(x, z) = - \sum_{n=1}^d [x_n \log z_n + (1 - x_n) \log(1 - z_n)] \quad (3)$$

Простейший случай автоэнкодера представляет собой нейронную сеть с прямой связью. Вход и выход сети обладают одинаковым количеством нейронов, соединенные одним скрытым слоем. Разница между входом и выходом автоэнкодера минимизируется.

$$f: X \rightarrow F$$

$$g: F \rightarrow X$$

$$f, g = \underset{f, g}{\operatorname{argmin}} \|X - (f \circ g)X\|^2 \quad (4)$$

$$\mathbf{h} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (5)$$

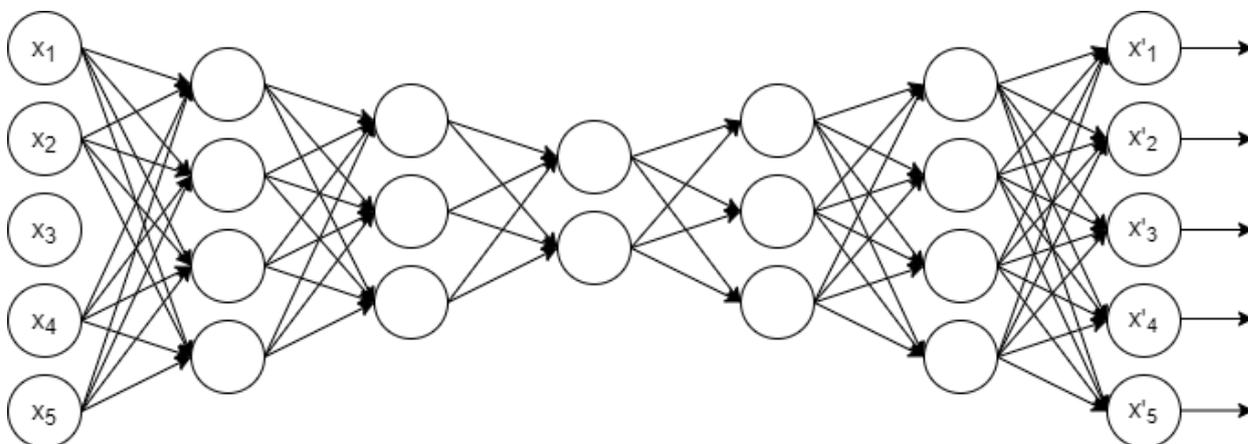
$$\mathbf{x}' = \sigma'(\mathbf{W}'\mathbf{h} + \mathbf{b}') \quad (6)$$

$$L(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 = \|\mathbf{x} - \sigma'(\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b})) + \mathbf{b}')\|^2 \quad (7)$$

Функция  $\sigma$  представляет из себя сигмоиду:

$$\sigma(x) = \frac{1}{1 + e^{-2\alpha x}}, \quad \alpha > 0 \quad (8)$$

Иногда обычного автоэнкодера не хватает для решения поставленной задачи, в этом случае применяется другой вид автоэнкодеров. Стек автоэнкодеров – это нейронная сеть, состоящая из нескольких автоэнкодеров, соединенных в один.



Стек автоэнкодеров(рис.2)

При таком соединении выход каждого скрытого слоя подается на вход следующего скрытого слоя. Процесс обучения такой модели необходимо разделить на два этапа. Сначала мы обучаем каждый автоэнкодер по отдельности, а затем обучаем всю модель целиком, рассматривая её как единый алгоритм[8].

### Сверточная сеть

Сверточная сеть – это архитектура нейронной сети, разработанная с целью обработки изображений.

Матричная операция, называемая сверткой представляет из себя операцию над парой матриц  $A$  и  $B$ , в результате которой получается третья матрица  $C$  ( $A * B = C$ ). Размер полученной матрицы определяется как:  $(n_x - m_x + 1)(n_y - m_y + 1)$ , где  $n_x \times n_y$  и  $m_x \times m_y$  размеры матриц  $A$  и  $B$  соответственно. Элементы матрицы  $C$

можно вычислить скалярным умножением матрицы B на подматрицу матрицы A:

$$C_{i,j} = \sum_{u=0}^{m_x-1} \sum_{v=0}^{m_y-1} A_{i+u,j+v} B_{u,v} \quad (9)$$

Перемещая матрицу B по матрице A, получаем все элементы матрицы C.

Матрица B является в некотором смысле фильтром, поскольку результат свертки тем больше, чем больше матрица B похожа на матрицу A в скалярном смысле. Матрица A в этом случае является изображением.

Сверточные слои (англ. *convolutional layer*), полносвязные слои (англ. *fully-connected layer*) и пулинговые слои (англ. *pooling layer*) являются основными слоями в сверточной сети.

Сверточный слой — это слой, где применяется матричная свертка, а коэффициенты матрицы свертки являются параметрами для обучения. Роль константного сдвига (англ. *bias*) выполняет ещё один весовой параметр. Помимо этого, в одном таком слое может применяться несколько сверток. Сверточные матрицы могут быть трехмерными, что, например, может быть полезно в случае, когда нужно усреднить информацию о цветах изображения. Так же можно заметить, что операция свертки приводит к уменьшению изображения, а элементы, находящиеся на краях изображения участвуют в меньшем количестве операций, чем остальные элементы, поэтому нередко применяется операция, которая добавляет элементы вокруг изображения. Помимо этого, существует такой параметр сверточного слоя как сдвиг, который отвечает за скорость перемещения сверточной матрицы по изображению.

Помимо обычных сверточных слоев, сверточные сети могут содержать так же пулинговые слои. Предназначением пулингового слоя является снижение размерности изображения. Это происходит за счет разделения исходного изображения на сегменты и объединения их в один нейрон следующего слоя посредством вычисления функции. Часто этими функциями являются функция максимума, выбирающая наибольший элемент в сегменте или функция среднего, усредняющая все элементы в сегменте. В этом слое отсутствуют обучаемые параметры. Пулинговый слой обладает следующими основными целями:

1. Уменьшение размера изображения. Это необходимо, чтобы следующие сверточные слои работали над большей областью изображения.
2. Увеличение независимости результата работы сети по отношению к малому сдвигу входных данных.
3. Ускорение процесса вычислений[4].

### **Алгоритм Герхберга-Сакстона**

Нам известно распределение интенсивностей в плоскости изображения и какая-то априорная информация об объекте. Мы используем эту информацию как ограничения для алгоритма Герхберга-Сакстона. Алгоритм Герхберга-Сакстона – это итерационный алгоритм для получения фазы пары распределений света (или любого другого математически допустимого распределения), связанных с помощью функции распространения, такой как преобразование Фурье, если их интенсивности в соответствующих оптических плоскостях известны. Часто необходимо знать только фазовое распределение от одной из плоскостей,

поскольку фазовое распределение на другой плоскости можно получить, выполнив преобразование Фурье на плоскости, фаза которой известна. Хотя алгоритм GS часто используется для двумерных сигналов, он также применим и для одномерных сигналов. Фазовые распределения в этих двух плоскостях восстанавливаются путем распространения комплексного волнового фронта между двумя плоскостями вперед и назад и замены обновленных амплитуд на каждой итерации измеренными амплитудами. С помощью прямого и обратного преобразования Фурье мы переходим от плоскости изображения в плоскость объекта и обратно. В плоскости детектора амплитуда волны заменяется на квадратный корень измеренной интенсивности. В области объекта распределение этого объекта должно быть окружено известными значениями, обычно нулями, то есть мы имеем представления о размерах объекта и зануляем все вокруг него.

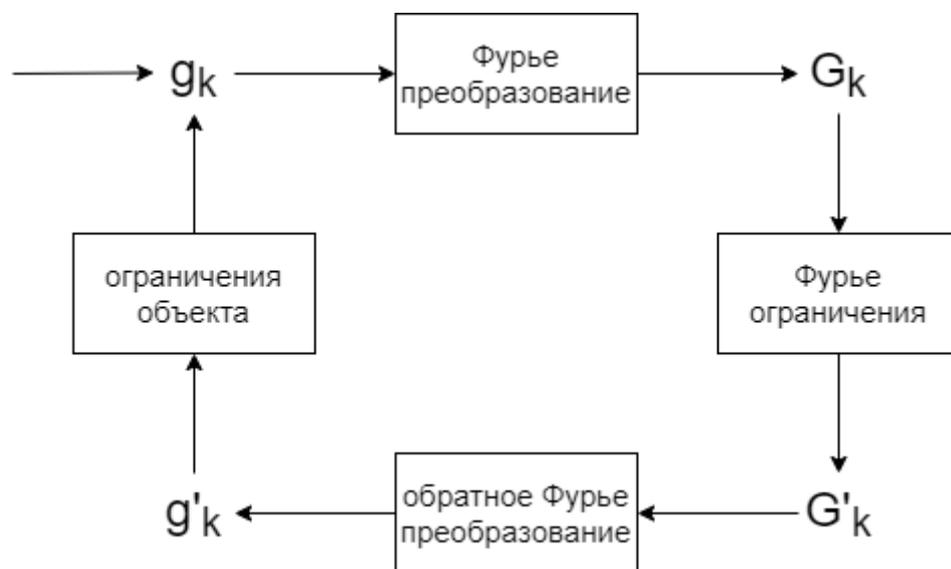


Схема алгоритма(рис.3)

## Алгоритм уменьшения ошибки

Это алгоритм восстановления фазы изображения, основанный на изначальном алгоритма Герхберга-Сакстона. Он заключается в следующем:

$$(1) G_k(v, w) = |G_k(v, w)| \exp[i\varphi_k(v, w)] = F[g_k(x, y)] \quad (10)$$

$$(2) G'_k(v, w) = |F(v, w)| \exp[i\varphi_k(v, w)] \quad (11)$$

$$(3) g'_k(x, y) = |g'_k(x, y)| \exp[i\vartheta'_k(x, y)] = F^{-1}[G'_k(v, w)] \quad (12)$$

$$(4) g_{k+1} = \begin{cases} g'_k(x, y), & \text{if } (x, y) \in \gamma \\ 0, & \text{if } (x, y) \notin \gamma \end{cases} \quad (13)$$

Так же существуют другие алгоритмы, получившиеся из базового алгоритма Герхберга-Сакстона.

Hybrid input-output алгоритм получается из алгоритма уменьшения ошибки путем модификации ограничений на амплитуду на 4ом шагу выполнения одной итерации алгоритма. Первые три шага k-той итерации алгоритма остаются такими же, а 4ый шаг выглядит следующим образом:

$$(4) g_{k+1}(x, y) = \begin{cases} g'_k(x, y), & \text{if } (x, y) \in \gamma \\ g_k(x, y) - \beta g'_k(x, y), & \text{if } (x, y) \notin \gamma \end{cases} \quad (14)$$

где  $\beta$  константа. Обычно выбирается  $\beta = 0.9$ , поскольку это позволяет достичь более плавных изменений реконструкции.

Shrinkwrap алгоритм является следующей модификацией Hybrid input-output алгоритма. Ограничения на амплитуду в плане объекта на 4ом шаге меняется каждую 20ю итерацию. Таким образом, ограничения на амплитуду постепенно начинают повторять форму объекта распределения, что обеспечивает лучшую точность работы алгоритма[2].

## Функции ошибки

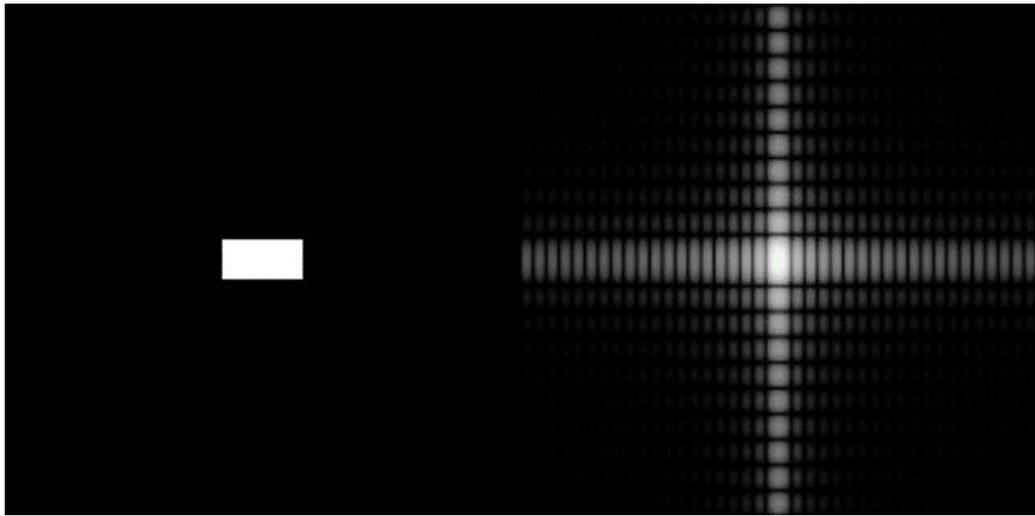
Для алгоритмов восстановления фазы, основанных на алгоритме Герхберга-Сакстона существуют функции ошибки, позволяющие отслеживать качество процесса восстановления фазы.

Для алгоритмов на основе алгоритма уменьшения ошибки функция ошибок оценивает, как хорошо последовательно восстановленные амплитуды соответствуют измеренным амплитудам в плоскости детектора:

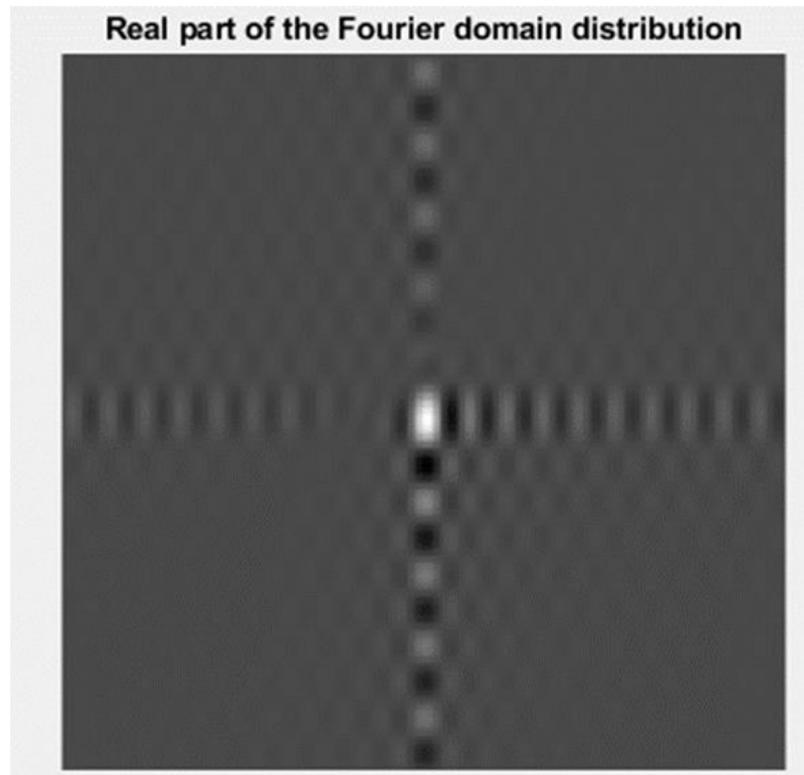
$$Error_{F_k} = \left\{ \frac{N^{-2} \sum_{v,w} [|G_k(v,w)| - |F(v,w)|]^2}{\sum_{v,w} |F(v,w)|^2} \right\}^{1/2} \quad (15)$$

Для алгоритмов на основе hybrid input-output алгоритма функция ошибок оценивает, насколько хорошо распределение восстановленного объекта удовлетворяет ограничениям объекта[2]:

$$Error_{f_k} = \left\{ \frac{\sum_{x,y \notin \gamma} |g'_k(x,y)|^2}{\sum_{x,y \in \gamma} |g'_k(x,y)|^2} \right\}^{1/2} \quad (16)$$



Исходное изображение и его дифракционная картина(рис.4)



Результат работы алгоритма восстановления фазы(рис.5)

## Дифракция

Дифракция света – отклонение лучей от прямолинейной траектории распространения. Поскольку свет обладает волновой природой, при столкновении волнового фронта с препятствием, на

краях этого препятствия возникает дифракция и волновой фронт его огибает. В данном случае можно воспользоваться правилом Гюйгенса-Френеля, которое гласит о том, что каждую точку волнового фронта можно рассматривать в качестве вторичного источника излучения.

Математическое описание дифрагированного волнового излучения выглядит следующим образом:

$$U(\vec{R}) = -\frac{i}{\lambda} \iiint \exp(ikz) o(\vec{r}) \frac{\exp(ik|\vec{R} - \vec{r}|)}{|\vec{R} - \vec{r}|} d\vec{r} \quad (17)$$

Распределение интенсивностей для дифракции на щели:

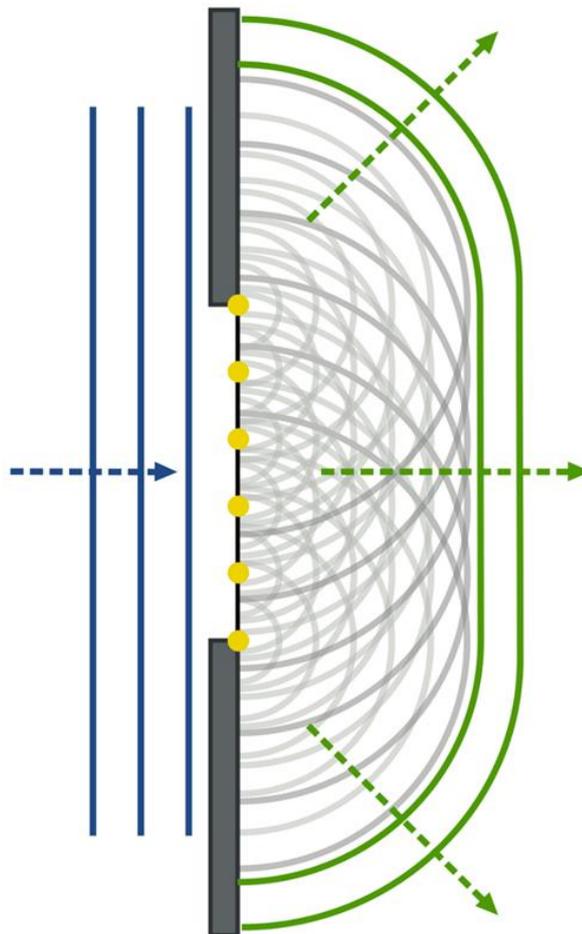
$$I(\theta) = I_0 \text{sinc}^2 \left( \frac{d\pi}{\lambda} \sin \theta \right) \quad (18)$$

Для того, чтобы получить изображение объекта съемки на экране, необходимо сфокусировать излучение, которое этот объект отражает в камеру. В случае рентгеновской микроскопии сфокусировать излучение не получается, поэтому одна точка объекта отражает лучи, падающие на неё в разных направлениях, что приводит к возникновению дифракции. При рентгеновской микросъемке происходит дифракция Френеля, кроме того предполагается, что излучение, попадающее на экран, обладает одной длинной волны. Электроны так же обладают волновыми свойствами при рассеянии на молекулах и кристаллических решетках.

Зная направления распространения и параметры излучения, а также форму препятствия, находящегося на его пути к экрану, можно воссоздать дифракционную картину. Изображение, полученное на экране не имеет ничего общего с препятствием, форму которого оно должно было бы принять в случае отсутствия дифракционных эффектов. Это происходит из-за потери части информации об

изображении, а точнее потери фазы изображения, что обычно называют проблемой фазы (англ. Phase problem).

При фокусировке изображения, каждый луч, исходящий из определенной точки попадает в другую точку, симметрично расположенную к ней относительно линзы. Но в случае дифракции лучи из одной точки плоскости препятствия попадают в разные точки плоскости экрана. Обнуление фазы изображения, приводит к тому, что все его гармонические составляющие сдвигаются к центру и накладываются друг на друга[6].



правило Френеля(рис.6)[14]

## Симуляция дифракции

Для упрощения получения данных, поскольку невозможно получить оригинальное и неискаженное изображение с микроскопа, предлагается использовать симуляцию дифракционной картины оригинального изображения. Таким образом можно сравнивать результат работы алгоритма с оригинальным изображением.

Симуляция дифракционной картины состоит в следующем. Происходит оцифровка распределения выборки, то есть её представление в пикселях. Далее, для каждого отдельного пикселя выполняется расчет дифрагированной комплексной волновой функции, как вычисление дифракции на отверстии квадратной формы.

Конечный комплексный волновой фронт дифракционной картины является суммой волновых фронтов всех пикселей. Для получения распределения интенсивностей дифракционной картины необходимо получить квадрат амплитуды конечного волнового фронта.

Согласно теореме Фурье  $E(x, y)$  можно разложить в ряд:

$$E(x) = \int_{-\infty}^{\infty} F(u, v) \exp[2\pi i(ux + vy)] du dv \quad (19)$$

Это означает, что непосредственно за экраном волновой фронт представляет собой суперпозицию плоских волн, которые обладают амплитудами  $F(u, v)$ :

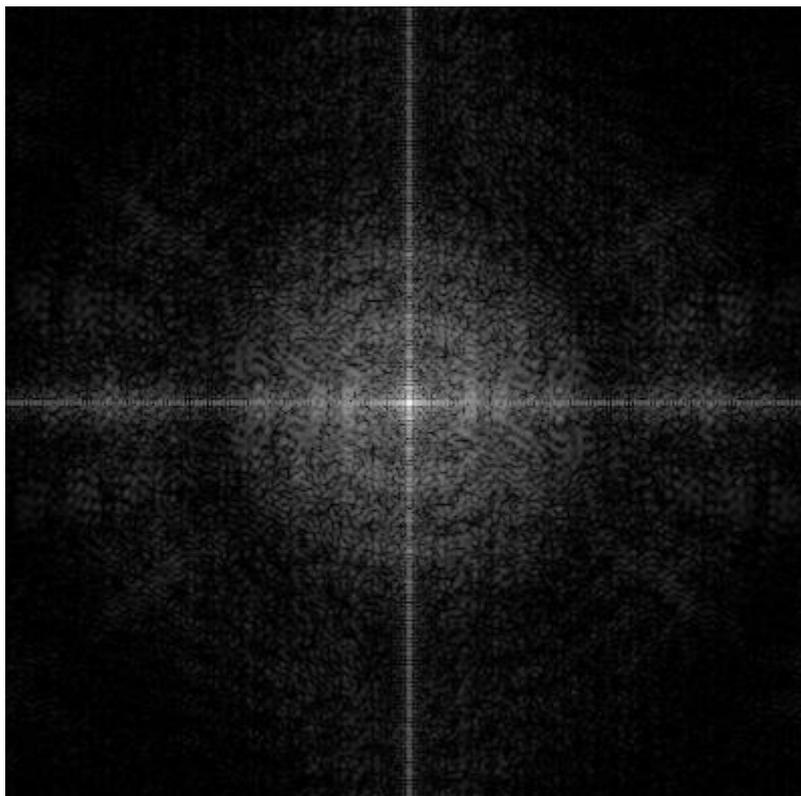
$$F(u, v) = \int_{-\infty}^{\infty} E(x, y) \exp[-2\pi i(ux + vy)] dx dy, \quad (20)$$

где  $u = \frac{\cos \alpha}{\lambda}$  и  $v = \frac{\cos \beta}{\lambda}$  – пространственные частоты, а  $\alpha$  и  $\beta$  это углы между осями  $X$  и  $Y$  и волновым вектором.

Таким образом распределение интенсивностей дифракционной картины может быть проведено с помощью преобразования Фурье следующим образом:

$$Diff = |FFT(Img)|^2, \quad (21)$$

где Diff – дифракционная картина, FFT – быстрое преобразование Фурье, Img – исходное изображение[2].



Симуляция дифракционной картины изображения(рис.7)

### **Рентгеновская и электронная микроскопия**

Рентгеновская микроскопия применяется в случаях, когда объект исследования по размерам сопоставим с длиной волны рентгеновского излучения, что соответствует длинам волн от  $\sim 10^{-2}$  до  $\sim 10^3$  нм. Поскольку фокусировка рентгеновского излучения при низких длинах волн нетривиальная задача, получившееся изображение будет

искажено дифракционными эффектами. Для устранения которых и применяются алгоритмы восстановления фазы.

В кристаллографии широко распространено применение рентгеновской микроскопии для получения изображения кристаллической решетки. Используя изображение обратной решетки, можно получить изображение прямой решетки, восстанавливая её фазу.

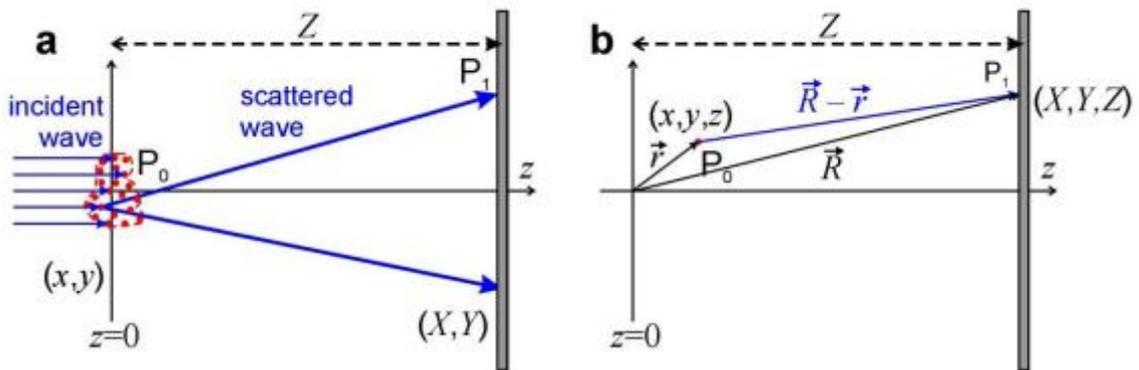
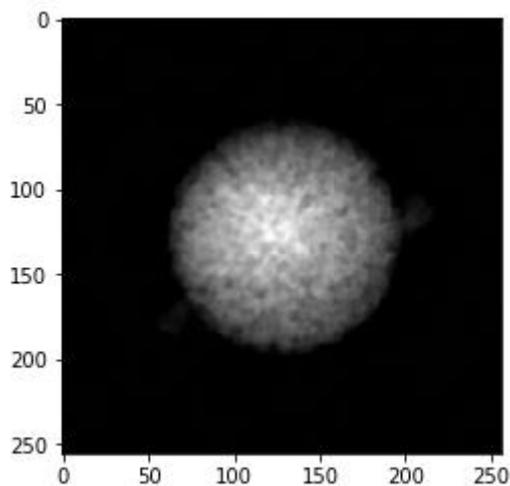


Схема распространения лучей(рис.8)[2]

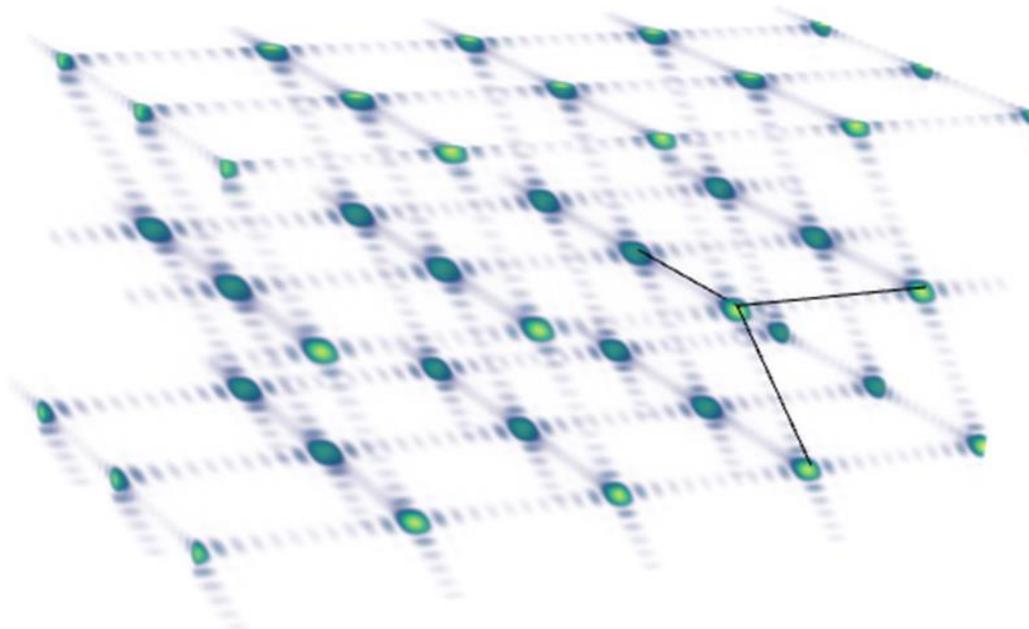


Снимок валорита с рентгеновского микроскопа(рис.9)[1]

### Обратная решетка

Обратная решетка, существующая в обратном пространстве, представляет собой Фурье преобразование прямой решетки, которая существует в реальном пространстве.

Обратное пространство, так же известное как импульсное пространство, это пространство, где единицей измерения расстояния является мера, обратно пропорциональная длине[6].



Обратная решетка[13](рис.10)

### **Преобразование Фурье**

Преобразование Фурье – это преобразование функции вещественной переменной, которая ставит ей в соответствие другую функцию в комплексном пространстве. Полученная функция представляет собой разложение изначальной функции на гармонические составляющие. Кроме того, функцию можно разложить в гармонический ряд, который представляют собой сумму синусов и косинусов с коэффициентами Фурье.

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)e^{-ix\omega} dx \quad (22)$$

В данном случае в результате преобразования происходит переход в пространство частот  $x \rightarrow \omega$ .

Преобразование Фурье обладает свойством линейности:

$$(\alpha \widehat{f} + \beta \widehat{g}) = \alpha \hat{f} + \beta \hat{g} \quad (23)$$

Кроме того, существует обратное преобразование Фурье, оно...

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(\omega)e^{ix\omega} d\omega \quad (24)$$

Поскольку обработка изображений требует работы с матрицами, необходимо задать многомерное преобразование Фурье.

Многомерное преобразование Фурье и его обращение имеют следующий вид:

$$\hat{f}(\omega) = \frac{1}{(2\pi)^{n/2}} \int_{R^n} f(x)e^{-ix\omega} dx \quad (25)$$

$$f(x) = \frac{1}{(2\pi)^{n/2}} \int_{R^n} \hat{f}(\omega)e^{ix\omega} d\omega \quad (26)$$

Преобразование Фурье часто применяется в обработке изображений, оно используется в фильтрации сигналов и уменьшения зашумленности, поскольку позволяет получить доступ к частотной составляющей сигнала.

## Дискретное преобразование Фурье

Дискретное преобразование Фурье часто применяется в цифровой обработке сигналов. Это преобразование является видом преобразования Фурье. Поскольку преобразование Фурье применяется только к непрерывным сигналам, для обработки дискретных данных можно применять данное преобразование.

Дискретное преобразование Фурье и его обратная форма имеют следующий вид:

$$Z_k = \sum_{n=0}^{N-1} z_n e^{-\frac{2\pi i}{N}kn} = \sum_{n=0}^{N-1} z_n \left[ \cos\left(\frac{2\pi kn}{N}\right) - i \sin\left(\frac{2\pi kn}{N}\right) \right], (k = 0, \dots, N - 1)$$

(28)

$$z_n = \frac{1}{N} \sum_{k=0}^{N-1} Z_k e^{\frac{2\pi i}{N}kn} = \frac{1}{N} \sum_{k=0}^{N-1} Z_k \left[ \cos\left(\frac{2\pi kn}{N}\right) + i \sin\left(\frac{2\pi kn}{N}\right) \right], (n = 0, \dots, N - 1)$$

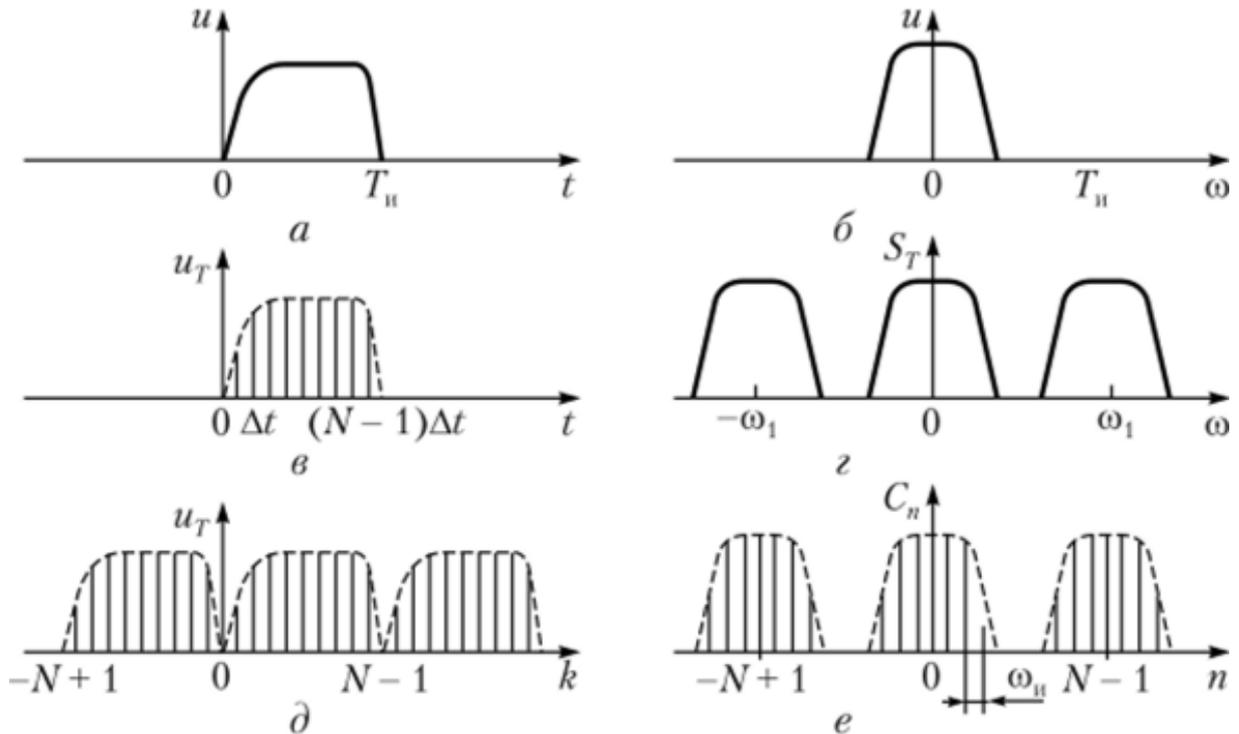
(29)

где

- $N$  — количество значений сигнала, за период, и количество компонент разложенного сигнала;
- $z_n$  — значения сигнала, являющиеся входными значениями для прямого преобразования и выходными для обратного преобразования;
- $Z_k$  — комплексные амплитуды, в количестве  $N$ , синусоидальных сигналов, составляющих исходный сигнал; они являются выходными значениями для прямого преобразования и входными для обратного; в связи с тем, что это комплексные

амплитуды, с их помощью можно вычислить как фазу, так и амплитуду;

- $|Z_k|/N$  — вещественная амплитуда  $k$ -го синусоидального сигнала;
- $Arg(Z_k)$  — фаза  $k$ -го синусоидального сигнала;
- $k$  — частотный индекс.



а, б — аналоговый сигнал и его спектр; в, г — дискретный сигнал и его спектр; д — периодическая последовательность дискретного сигнала; е — дискретное преобразование Фурье сигнала[15](рис.11)

Дискретное преобразование Фурье является преобразованием, которое производит переход от последовательности значений определенной функции на равных интервалах в другую последовательность, представляющую собой совокупность преобразований Фурье на соответствующих интервалах, что является последовательность комплексных значений:

$$Z(\omega) = \sum z_n e^{-i\omega n} \quad (30)$$

Учитывая этот факт и то, что дискретное преобразование Фурье обладает свойством линейности, оно принимает следующую форму:

$$Z = Az, \quad (31)$$

что является произведением матрицы преобразования на вектор преобразуемого сигнала.

Матрицу можно вычислить с помощью следующей формулы:

$$A_{k,l} = \exp\left(-2\pi i \frac{(k-1)(l-1)}{N}\right) \quad (32)$$

Матрица A имеет вид (33):

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-\frac{2\pi i}{N}} & e^{-\frac{4\pi i}{N}} & e^{-\frac{6\pi i}{N}} & \dots & e^{-\frac{2\pi i}{N}(N-1)} \\ 1 & e^{-\frac{4\pi i}{N}} & e^{-\frac{8\pi i}{N}} & e^{-\frac{12\pi i}{N}} & \dots & e^{-\frac{2\pi i}{N}2(N-1)} \\ 1 & e^{-\frac{6\pi i}{N}} & e^{-\frac{12\pi i}{N}} & e^{-\frac{18\pi i}{N}} & \dots & e^{-\frac{2\pi i}{N}3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{2\pi i}{N}(N-1)} & e^{-\frac{2\pi i}{N}2(N-1)} & e^{-\frac{2\pi i}{N}3(N-1)} & \dots & e^{-\frac{2\pi i}{N}(N-1)^2} \end{pmatrix}$$

Быстрое преобразование Фурье — это алгоритм, позволяющий ускорить выполнение операции преобразования. Дискретное преобразование Фурье выполняется за время  $O(N^2)$ , в связи с чем эффективнее использовать данный алгоритм вычисления преобразования, который выполняется за время  $O(N \log(N))$ .

Алгоритм быстрого преобразования Фурье основывается на известном принципе эффективного программирования, называемом «разделяй и властвуй». Этот принцип предполагает разбиение процесса вычислений, поэтому мы разбиваем входящий сигнал на несколько. Для простоты будем разбивать входящий дискретный сигнал на два, рекурсивно вычисляем дискретное преобразование

Фурье для каждого из них с последующим объединением в одно решение[7].

## Эксперимент

Датасетом для выполнения работы стали случайно генерируемые изображения геометрических фигур и преобразование этих изображений в дифракционную картину.

Были проведены эксперименты по восстановлению фазы с алгоритмом Герхберга-Сакстона. В результате получено изображение с восстановленной фазой. График ошибки алгоритма:

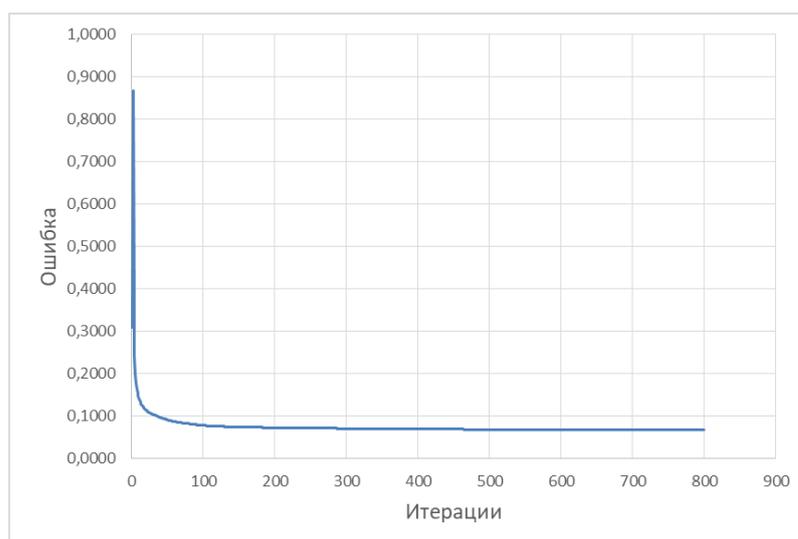
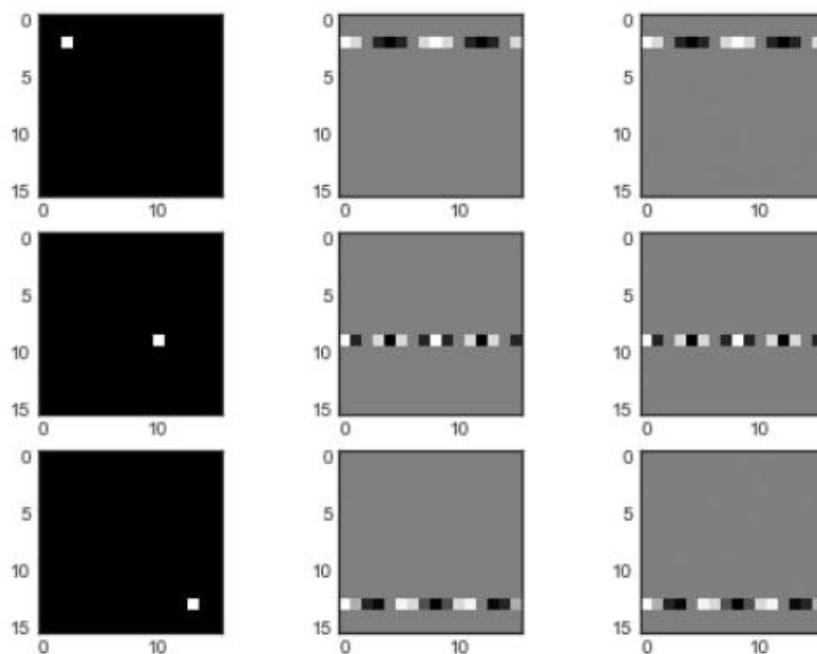
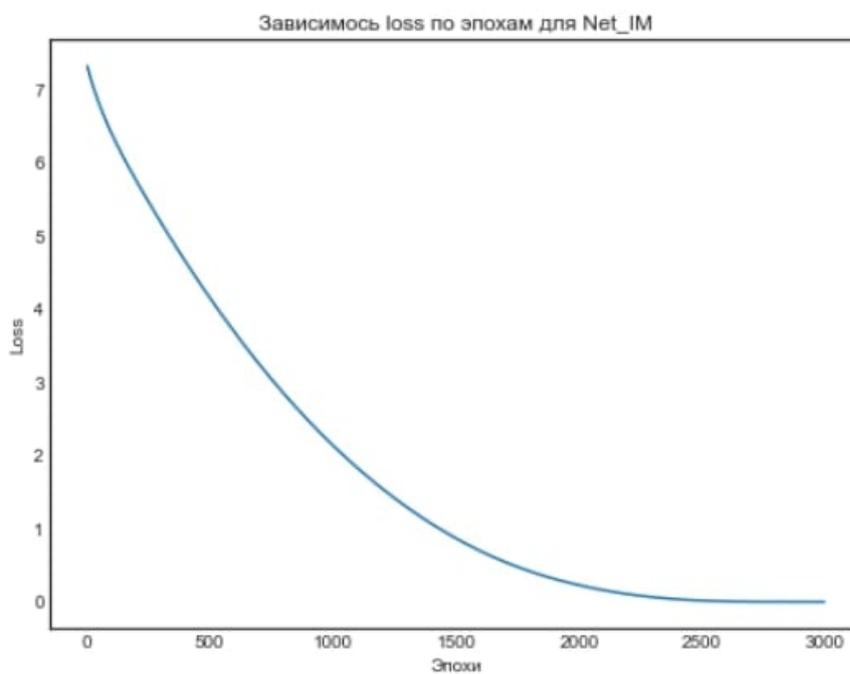


График ошибки алгоритма Герхберга-Сакстона(рис.12)

В алгоритме была проведена замена преобразования Фурье на автоэнкодер, поскольку дискретное преобразование Фурье является линейным и представимо в виде матрицы коэффициентов.

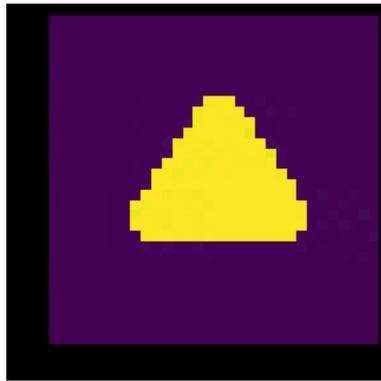


Нейросеть обученная воспроизводить дискретное двумерное преобразование Фурье(рис.13)

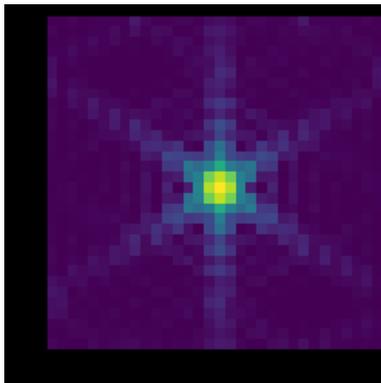


Функция ошибки для аппроксимации Фурье преобразования(рис.14)

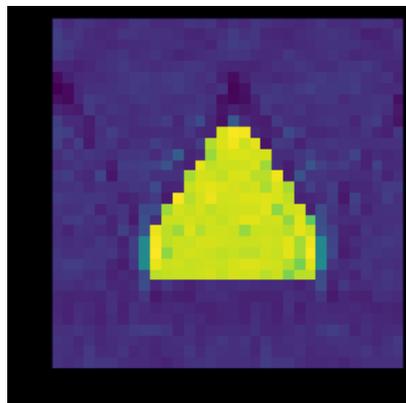
Результатом эксперимента стало следующее:



Исходное изображение(рис.15)

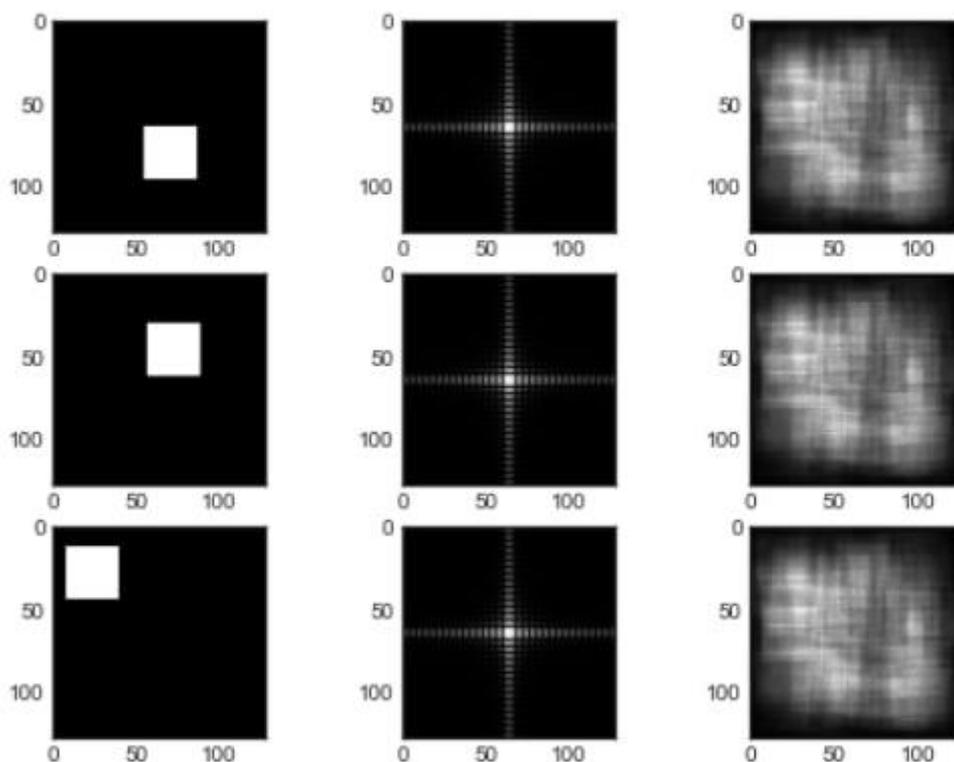


Дифракционная картина(рис.16)



Восстановленное изображение(рис.17)

Так же была составленная сверточная нейронная сеть, которая обучалась на изображениях с дифракционной картиной и оригинальных изображениях. Сеть не сошлась.



Сверточная сеть(рис.18)

В соответствии с темой настоящей работы основной задачей является серия экспериментов, имеющих целью улучшение качества реконструкции исходного изображения. К сожалению, основные результаты были получены в самом конце выполнения работы и не были учтены рецензентом, который высказал много обоснованных замечаний.

Основная гипотеза состояла в том, что аппроксимация алгоритма ГС может быть получена на пути нелинейной аппроксимации, которая теоретически возможна согласно известным результатам о нейросетях как универсальных аппроксиматорах. Получение такого результата было бы очень привлекательно. Однако теоретическая возможность не значит практическую достижимость, например, из-за невозможности определить оптимальную начальную инициализацию. Многочисленные попытки нелинейной аппроксимации не привели к положительному результату. Одной из причин этого

является принципиальное различие используемых в настоящее время базовых нейросетевых архитектур. Помимо базовых полносвязных (FC) как строительные элементы используются сверточные (CNN) и рекуррентные сети (RNN). Однако представить рекуррентные сети (RNN) через композицию FC/CNN сетей не представляется возможным или просто непрактично. Такой эксперимент по аппроксимации рекуррентных сетей полносвязными или сверточными интересен сам по себе, однако нет гарантии что он приведет к успеху. Природа алгоритма FC - итерация, рекурсия и для его аппроксимации естественно использовать рекуррентные (RNN) сети. В этом находит объяснение неудача попыток аппроксимации с помощью FC/CNN сетей

Однако попытки применения RNN сетей общего вида также оказались неуспешными по причине препятствий традиционного характера: экспоненциальное затухание градиентов. Для решения этих проблем успешно используются сети LSTM, GRU и т.п., однако в данном частном случае они неприменимы. Причина этого в том, что начальная инициализация параметров RNN сети осуществляется случайными небольшими значениями и при нескольких рекурсивных циклах алгоритма они экспоненциально уменьшаются до нуля и это состояние сети является стабильным, выйти из него не удастся с помощью разных методов регуляризации вроде Dropout.

Также была проведена разведочная работа по написанию автоэнкодеров и вариационных автоэнкодеров для моделирования скрытого пространства для тестовых изображений. В итоге достигнуто высокое качество восстановления изображений при минимальном размере скрытого пространства - 2x2.

Длительная серия неудачных экспериментов для построения нейросетевой аппроксимации привела к нескольким позитивным достижениям:

1. Построен и работает фреймворк для аппроксимации - нейросеть на PyTorch и ее окружение (Dataset, Dataloader, цикл обучение сети, генераторы пробных изображений, функции для визуализации).
2. Экспериментально определен минимально допустимый размер изображений - 16x16. Изначально эксперименты проводились с изображениями 400x400, однако даже при размерах 128x128 сложная нелинейная модель не помещалась в памяти 32 Гб и обучалась очень долго. Этот фактор является главным, потому что позволяет проводить быстрые эксперименты.
3. Алгоритм БПФ `fft2`, `ifft2` переписаны с нуля на PyTorch для встраивания в нейросеть.
4. Аргументирована (см. выше) необходимость использования рекуррентных сетей (или соответствующих циклов в `forward`).
5. Предложено решение проблемы инициализации нейросетевых слоев - весами с тензорами от БПФ.

В итоге для решения поставленной задачи было принято решение непосредственно реализовать подход, предлагаемый в постановке этой работы. Качество аппроксимации повышается за счет стека автоэнкодеров, непосредственно имитирующих алгоритм ГС, но за счет нейросетевых слоев позволяющий провести глубокую настройку на выбранный датасет из пробных изображений.

В данном случае автоэнкодером можно считать аналог шага итерации алгоритма ГС, поскольку на каждом шаге происходит прямое и обратное БПФ. Скрытым пространством является квази Фурье-образ

дифракционного изображения (с учетом изменений алгоритма). Обрезка образа в скрытом пространстве позволяет итерационно восстановить исходное изображение. Веса FC нейросетей реализующих стек автоэнкодеров обучаются в результате серии эпох обучения нейросети.

Предложенный подход реализован в коде на PyTorch, который приложен к данной работе.

К сожалению, каких-то неожиданных волшебных результатов и здесь добиться не удалось. Две FC-сети, заменяющие БПФ в результате работы стека автоэнкодеров формируют сложную функцию потерь, оптимизация которой требует дополнительных экспериментов. Всего на датасете из 256 пробных (из ячеек на центральном квадрате) изображений было сделано 30 эпох. Падение потерь средних на эпоху составило до 3-5%. Подробнее, потери `mse_loss` на первую эпоху (которая почти идентична оригинальному алгоритму ГС) - 7.18, потери на 10 эпоху - 7.07, на 13 эпоху - 7.07, на 29 эпоху = 7.21. На другом пробном датасете (линии на центральном квадрате) качество обучения было немного выше. Обучение сети на выбранном датасете происходит, сеть "живет", качество улучшается, но говорить о значимом повышении качества по сравнению с алгоритмом ГС не приходится.

Дальнейших экспериментов не проводилось, поскольку цель, поставленная перед работой полностью выполнена. Но можно предложить два пути улучшения качества аппроксимации. Во-первых, это усложнение структуры автоэнкодеров, введение нелинейности. Во-вторых требуется изучить распределение оригинальных целевых изображений, это может привести к использованию свойства этого распределения для улучшения аппроксимации. Также актуальным

остаётся изучение литературы в поиске новых идей для решения подобных задач нейросетевыми методами.

## **Заключение**

Рассмотрены различные алгоритмы восстановления фазы изображения. Получены результаты работы этих алгоритмов и предложена модель для решения задачи восстановления фазы изображения. Была составлена и обучена модель машинного обучения (нейросеть Net\_GS), имитирующая работу алгоритма Герхберга-Сакстона.

Предложенная нейросеть Net\_GS позволила улучшить качество восстановления изображений на модельных изображениях, однако это улучшение не является значимым. Это трудная задача, которая требует экспериментов. Доступная открыто в виде кода на PyTorch и воспроизводимая сеть Net\_GS может быть использована для дальнейших экспериментов. Предполагается, что дальнейшее улучшение качества возможно за счёт усложнения автоэнкодеров, изучение распределения актуальных изображений, а также поиска новых идей в области восстановления изображений.

## **Список литературы**

- 1) T. Beuvier, I. Probert, L. Beaufort, B. Suchéras-Marx, Y. Chushkin, F. Zontone & A. Gibaud: X-ray nanotomography of coccolithophores reveals that coccolith mass and segment number correlate with grid size – <https://www.nature.com/articles/s41467-019-08635-x#data-availability>

- 2) Tatiana Latychevskaia: Iterative phase retrieval in coherent diffractive imaging: practical issues – <https://arxiv.org/ftp/arxiv/papers/1809/1809.04626.pdf>
- 3) O. Cherkas, T. Beuvier Orcid, Dag W. Breiby, Y. Chushkin, F. Zontone and A. Gibaud: Direct Observation of Microparticle Porosity Changes in Solid-State Vaterite to Calcite Transformation by Coherent X-ray Diffraction Imaging – <https://pubs.acs.org/doi/abs/10.1021/acs.cgd.7b00476>
- 4) Christopher M. Bishop: Pattern Recognition and Machine Learning – <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>
- 5) Eliyahu Osherovich: Numerical methods for phase retrieval, 2011
- 6) И. Е. Иродов: Волновые процессы. Основные законы.
- 7) Павлидис Т.: Алгоритмы машинной графики и обработки изображений.
- 8) Andrew Ng: Sparse autoencoder
- 9) Hong Cheng II, Yong Liu, Rui Wang V, Quanbing Zhang II: Phase retrieval algorithm based on the neural network and the GS, 2019
- 10) Changliang Guo, S. Liu, J. Sheridan: Iterative phase retrieval algorithms. I: optimization.
- 11) Sergei Grudinin, Maria Garkavenko, Andrei Kazennov: Pepsi-SAXS: an adaptive method for rapid and accurate computation of small-angle X-ray scattering profiles – <https://hal.inria.fr/hal-01516719>
- 12) O. Cherkas, T. Beuvier Orcid, Dag W. Breiby, Y. Chushkin, F. Zontone and A. Gibaud: Direct Observation of Microparticle Porosity Changes in Solid-State Vaterite to Calcite Transformation by Coherent X-ray Diffraction Imaging – <https://pubs.acs.org/doi/abs/10.1021/acs.cgd.7b00476>

- 13) [https://en.wikipedia.org/wiki/Reciprocal\\_lattice](https://en.wikipedia.org/wiki/Reciprocal_lattice)
- 14) [https://en.wikipedia.org/wiki/Huygens%E2%80%93Fresnel\\_principle](https://en.wikipedia.org/wiki/Huygens%E2%80%93Fresnel_principle)
- 15) [https://studme.org/171393/tehnika/algorithmy\\_diskretnogo\\_bystrogo\\_preobrazovaniy\\_fure](https://studme.org/171393/tehnika/algorithmy_diskretnogo_bystrogo_preobrazovaniy_fure)