



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Национальный исследовательский университет «МЭИ»

Институт

РЭ

Кафедра

Физики им В.А. Фабриканта

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)

Направление **11.03.04 Электроника и нанoeлектроника**
(код и наименование)

Направленность (профиль) **Квантовая и оптическая электроника**

Форма обучения **Очная**
(очная/очно-заочная/заочная)

Тема: **Оценка качества оптических систем методом анализа картины
распределения интенсивности**

Студент **ЭР-03-16** **Пинчуков В.В.**
группа подпись фамилия и инициалы

**Научный
руководитель** **Инж. иссл. 1 кат.** **Макашов Д.А.**
уч. степень должность подпись фамилия и инициалы

«Работа допущена к защите»

Зав. кафедрой **Д.т.н** **Доцент** **Скорнякова Н.М.**
уч. степень звание подпись фамилия и инициалы

Дата _____

Москва, 2020



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Национальный исследовательский университет «МЭИ»

Институт РЭ
Кафедра Физики им В.А. Фабриканта

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
(бакалаврскую работу)

Направление 11.03.04 Электроника и нанoeлектроника
(код и наименование)

Направленность (профиль) Квантовая и оптическая электроника

Форма обучения Очная
(очная/очно-заочная/заочная)

Тема: Оценка качества оптических систем методом анализа картины
распределения интенсивности

Студент ЭР-03-16 Пинчуков В.В.
группа подпись фамилия и инициалы

**Научный
руководитель** Инж. иссл. 1 кат. Макашов Д.А.
уч. степень должность подпись фамилия и инициалы

Зав. кафедрой Д.т.н Доцент Скорнякова Н.М.
уч. степень звание подпись фамилия и инициалы

Место выполнения работы Акционерное общество
«Научно-производственная корпорация «Системы прецизионного
Приборостроения» (АО «НПК «СПП»)

СОДЕРЖАНИЕ РАЗДЕЛОВ ЗАДАНИЯ И ИСХОДНЫЕ ДАННЫЕ

Изучение и реализация итерационной процедуры для восстановления
волнового фронта по распределению интенсивности.

1. Провести обзор литературы по теме исследования.
2. Изучить теоретическую базу.
3. Реализовать решение прямой и обратной задачи.
4. Выполнить проверку решения.

5.	Применить полученный алгоритм для оценки качества объектива.
----	--

ПЕРЕЧЕНЬ ГРАФИЧЕСКОГО МАТЕРИАЛА

Количество листов 0

Количество слайдов в презентации 9

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Braat J., Dirksen P., Haver S., Janssen A. Detailed description of the ENZ approach [electronic resource]. 2013. P. 39. Available at:
http://www.nijboerzernike.nl/_PDF/NijboerZernike_webpage_detailed_20151023.pdf

Теребиж В.Ю. Введение в статистическую теорию обратных задач. – М.: ФИЗМАТЛИТ, 2005. С. 226–249.

Smith C.S., Marinica R., Verhaegen M. Real-time wavefront reconstruction from intensity measurements // 3rd AO4ELT Conf. - Adapt. Opt. Extrem. Large Telesc. 2013. P. 1–8.

Селивестрова Е.В., Щербина Г.А., Шурыгин Б.М. Алгоритм нахождения абберационных коэффициентов в космической оптико-электронной аппаратуре наблюдения // Журн. Радиоэлектроники 2017. №12. С. 1–18.

РЕФЕРАТ

Отчёт содержит 78 страниц, 2 части, 24 рисунка, 88 источников.

Ключевые слова: фазовая проблема, полиномы Цернике, прямая задача, обратная задача, абберрации, метод максимального правдоподобия, дифракция.

Объектом исследования является метод восстановления волнового фронта.

Цель работы – изучение и реализация итерационной процедуры для восстановления волнового фронта по распределению интенсивности.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) провести обзор литературы по теме исследования;
- 2) изучить теоретическую базу;
- 3) реализовать решения прямой и обратной задачи;
- 4) выполнить проверку решения;
- 5) применить полученный алгоритм для оценки качества объектива.

Область применения: восстановление волнового фронта, оценка качества оптических систем.

ОГЛАВЛЕНИЕ

Введение.....	6
1 Обзор литературы. Теоретическое введение	8
1.1. Измерение функции рассеяния.....	8
1.2. Измерение частотной контрастной характеристики	9
1.3. Интерференционный метод	11
1.4. Восстановление волнового фронта по распределению интенсивности	16
1.4.1. Полиномы Цернике	16
1.4.2. Прямая и обратная задача	21
1.4.3. Матрица Фишера	27
2 Практическая часть	30
2.1. Реализация прямой задачи	30
2.2. Реализация обратной задачи	35
2.3. Проверка алгоритма по синтезированным изображениям	47
2.4. Стенд для получения изображений.....	62
2.5. Применение алгоритма к полученным изображениям и оценка качества оптической системы	64
Заключение	70
Список источников	71

ВВЕДЕНИЕ

В широком смысле работа посвящена решению обратной задачи, которая более распространена чем прямая задача. Прямыми называют задачи, в которых по известным параметрам и структуре необходимо получить результат действия. Например, получение изображения заданного объекта оптической системой с известными параметрами. Обратными называют задачи, в которых по каким-либо проявлениям объекта необходимо восстановить его определённые параметры. Так, если мы знаем каков объект и его изображение, то обратной задачей будет получение параметров оптической системы. По вопросам решения таких задач, как принципиального подхода к анализу и выбора математического аппарата для нахождения оптимального ответа, до сих пор ведутся работы [1–6].

К классу таких задач относится фазовая проблема. Для понимания этой проблемы представим источник излучения, от которого распространяется электромагнитное поле, характеризуемое комплексным сигналом с амплитудой и фазой. При приёме излучения обычно регистрируется интенсивность электромагнитного поля, что является квадратом модуля комплексного сигнала, то есть теряется информация о фазе этого сигнала. Для решения этой проблемы применяются различные системы измерения и анализа сигнала: интерференционные, интерферометр сдвига, наклона, вращения и так далее; датчики Шока-Гартмана [7,8], измеряющий наклоны волнового фронта; итерационные методы подгонки волнового фронта и другие.

Немало исследований проводится по фазовой проблеме в рентгеновской кристаллографии [9–13]. Реконструирование пространственного строения кристалла возможно при знании амплитуды и фазы. И если значение амплитуды можно оценить по интенсивности вторичного рентгеновского излучения, то фазу необходимо рассчитывать, с привлечением дополнительных средств. Такая проблема возникает при визуализации в голографии [14], при изготовлении дифракционных оптических элементов ДОО [15].

Для описания формы волнового фронта помимо полиномов Цернике, которым будет уделено отдельное внимание, используют модальный метод

Фурье [16]. Этот метод имеет более простую реализацию и высокую скорость расчёта, но его применение ограничено.

Так как данная работа не останавливается только на решение фазовой проблемы, то ниже рассматриваются различные способы оценки качества получаемого изображения, то есть качества оптической системы. Эти методы имеют как свои преимущества, так и недостатки, поэтому не являются взаимозаменяемыми.

1 ОБЗОР ЛИТЕРАТУРЫ. ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

1.1. Измерение функции рассеяния

Этот способ оценки качества изображения состоит в наблюдении изображения тестового объекта, структура которого фотометрически исследуется, то есть определяется функция рассеяния, например, функция рассеяния точки ФРТ или функция рассеяния линии ФРЛ. Эти функции рассеяния связаны с такими величинами, как: оптическая передаточная функция, функция передачи модуляции и концентрация энергии в изображении, что является количественными характеристиками качества оптической системы. Такой подход применяется для оценки высококачественных систем, имеющих малые величины aberrаций.

Универсальным и наиболее простым тестовым объектом является точечный источник, изображение которого будет ФРТ (рисунок 1). Для реализации такого источника используют диафрагму малого диаметра.

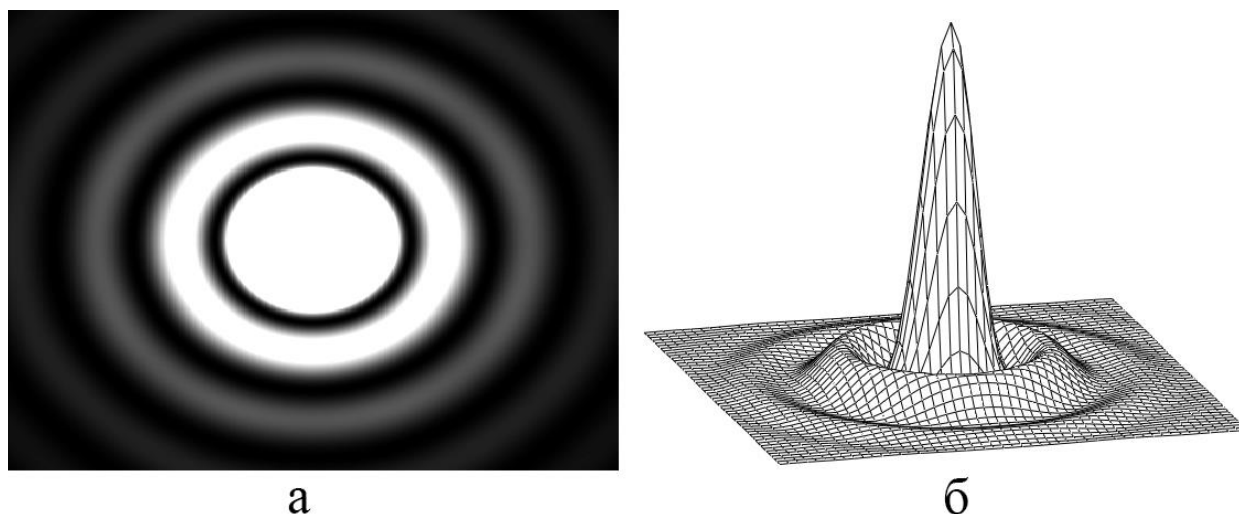


Рисунок 1 – Примеры ФРТ: а – переэкспонированная ФРТ; б – график поверхности ФРТ

Измерение ФРТ используют для оценки качества телескопов [17–19], оптимизации расфокусировки [20] при оптическом восстановлении трёхмерных

объектов, оценки качества изображения камер [21], постобработки изображений [22], в машинном зрении [23] при исследовании деградации изображения.

Влияние aberrаций на ФРТ описывают следующими величинами: число Штреля, критерий Релея для малых aberrаций, допуск Марешаля для малых aberrаций. Число Штреля равняется отношению интенсивности в максимуме ФРТ при наличии aberrаций к максимуму ФРТ при отсутствии aberrаций. Определение интенсивности ФРТ без aberrаций проводится путём расчётов, а с aberrациями непосредственно экспериментальными измерениями. Значение этого числа лежит в диапазоне от нуля до единицы. Если число Штреля равно единице, то система безабберационная, если больше или равно 0,8 – система практически безабберационная, поскольку уменьшение максимума на 20% слабо влияет на форму волнового фронта. Критерий Релея для малых aberrаций заключается в том, что если величина aberrаций меньше или равна $\lambda/4$, то число Штреля больше или равно 0,8. Но это справедливо не для всех типов aberrаций. Допуск Марешаля для малых aberrаций позволяет приблизительно оценить число Штреля, если известен средний квадрат деформации волнового фронта. Этот допуск должен быть меньше или равен $\lambda/14$. Марешалевский допуск справедлив для любых типов aberrаций малой величины [24].

1.2. Измерение частотной контрастной характеристики

Частотная контрастная характеристика ЧКХ или в англоязычный аналог modulation transfer functions MTF является характеристикой оптической системы, определяемая отношением величины контраста изображения к контрасту изображаемого объекта в зависимости от пространственной частоты (рисунок 2). Величина контраста лежит в интервале от нуля до единицы (рисунок 3).

Измерение ЧКХ проводится для получения информации о разрешающей способности оптической системы, но так как разрешение не одинаково по всему полю изображения и отличается для различных длин волн, то снимают

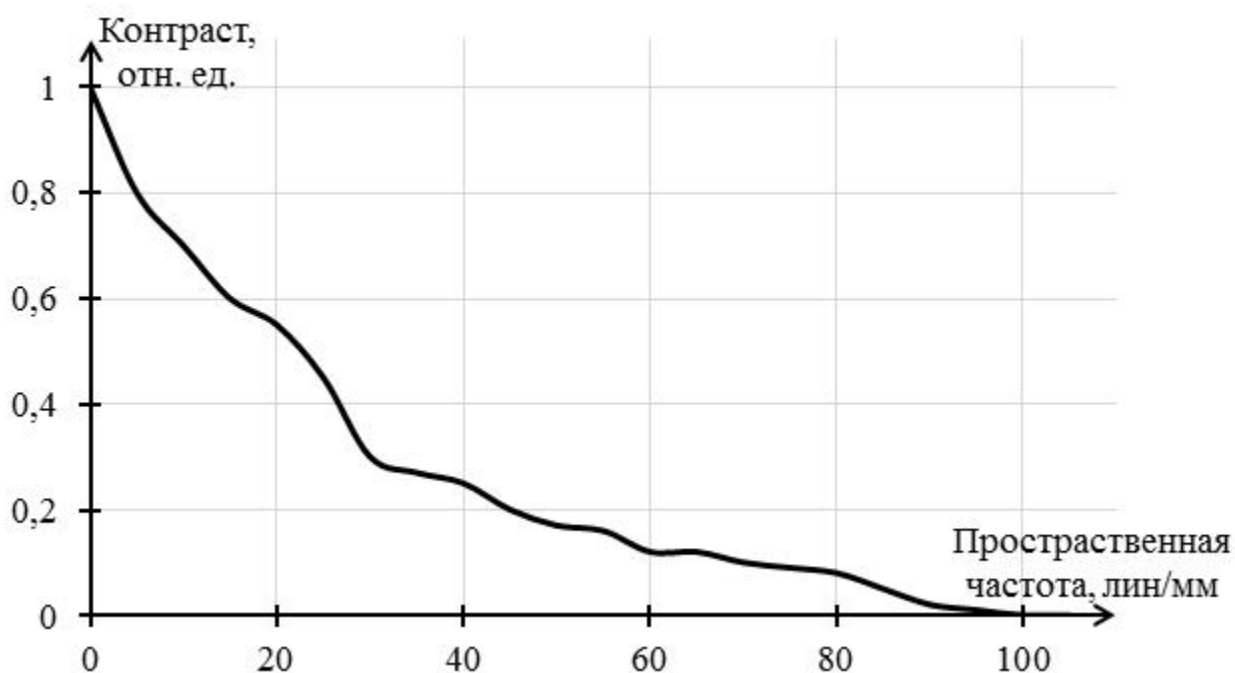


Рисунок 2 – Пример ЧКХ

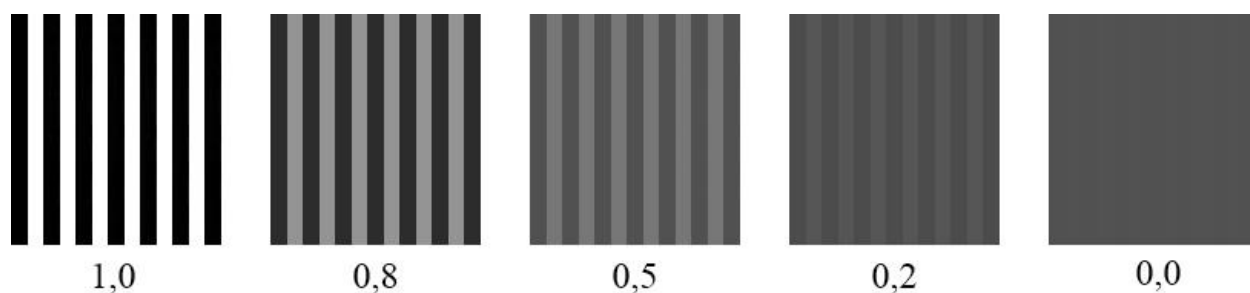


Рисунок 3 – Иллюстрация контраста

ЧКХ в зависимости удаления объекта от центра изображения при разных цветах объекта и разной ориентации в пространстве. Такой метод оценки качества оптики широко применяется при массовом производстве фотографических объективов.

Интересен случай ЧКХ для системы с разнесённой или сегментированной апертурой. Их существенным недостатком является провал контраста на средних пространственных частотах (рисунок 4), но обычно изготавливают подобные системы для того, чтобы улучшить предел разрешения по пространственным частотам, так как эффективный размер апертуры будет больше, чем для системы

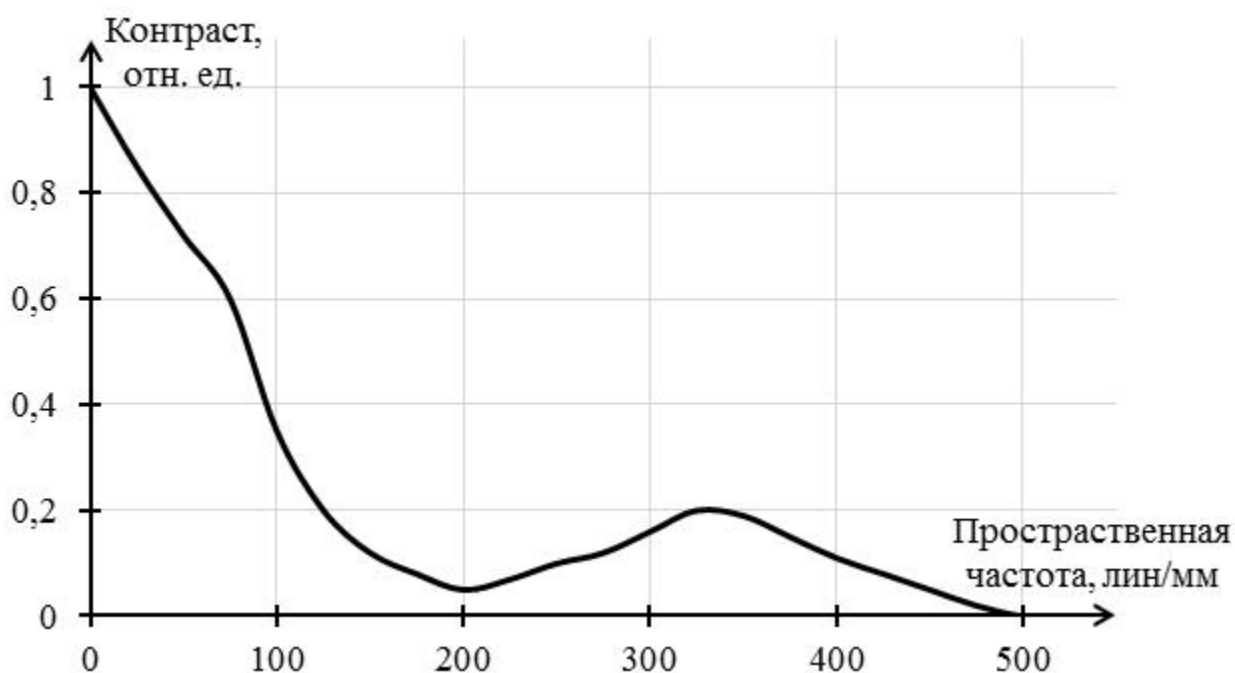


Рисунок 4 – Пример ЧКХ с провалом контраста на средних частотах

с единой апертурой, но при частотах близких к предельной значение контраста мало. В статье [25] предложены методы компенсации среднечастотной ЧКХ: метод оптимизации диафрагмы и метод восстановления изображения.

Измерение ЧКХ используются во многих областях: в спутниковой оптике для динамической оценки искажений изображения, вызванных эффектом движения [26]; в оптическом проектировании [27], телевизионных системах [28], в медицине при разработке аддитивных линз [29], коррекции аберраций асферической оптики [30], в компьютерной томографии [31,32].

Исследование по улучшению способов и качества измерения ЧКХ актуальны и по сей день [33,34]. А также создаются специальные установки для измерения ЧКХ объективов различной конфигурации [35].

1.3. Интерференционный метод

Данный метод широко используется для контроля кривизны, формы, шероховатости и локальных неоднородностях поверхности. Форма контролируемой поверхности может быть различной: сферической [36], цилиндрической [37] и любой произвольной формы [38,39]. Интерференционный метод позволяет измерять отклонения поверхности от

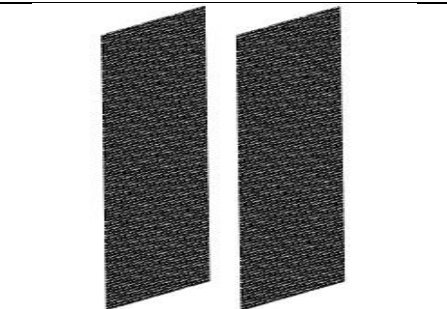

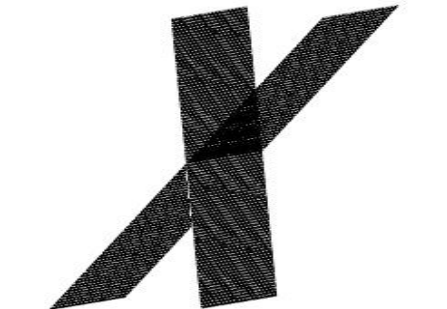

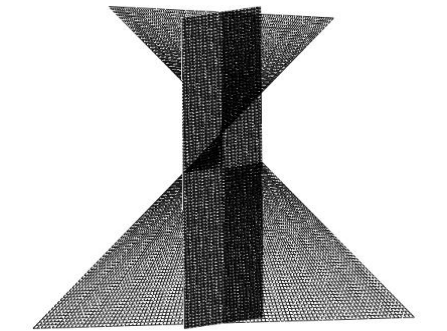
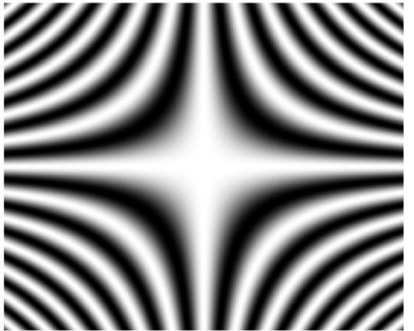
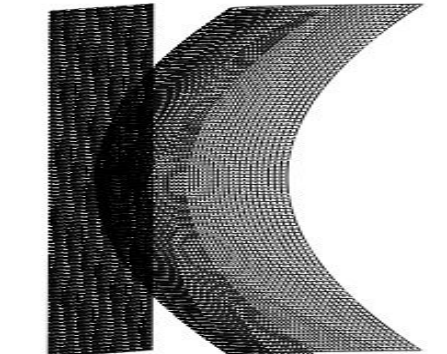
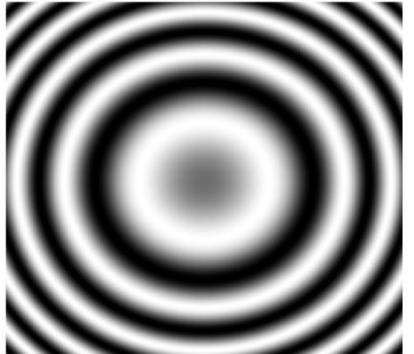
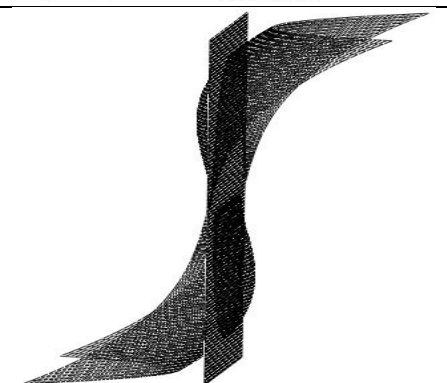
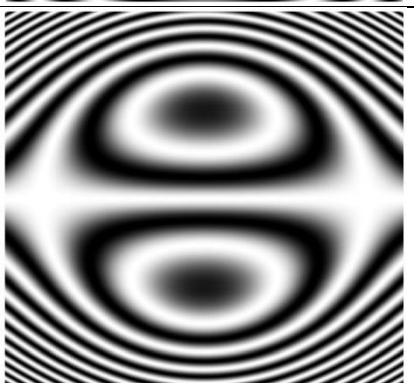
заданной с высокой точностью до единиц нанометров [40]. Суть этого метода состоит в измерении числа, формы и расстояния между интерференционными полосами, полученные в результате наложения опорного и измеряемого волновых фронтов.

Опорный фронт формируется при помощи пробных зеркал или дифракционных оптических элементов. Для измерения высокой точности необходимо, чтобы опорная волна повторяла номинальную форму измеряемой поверхности. Использование плоской волны в качестве опорной возможно не только для плоских поверхностей, но для сферических и асферических. Но в этом случае можно будет определить в лучшем случае форму поверхности, а из-за значительного отклонения измеряемого фронта информация о местных ошибках будет утеряна, так как замерить интерференционные полосы будет невозможно.

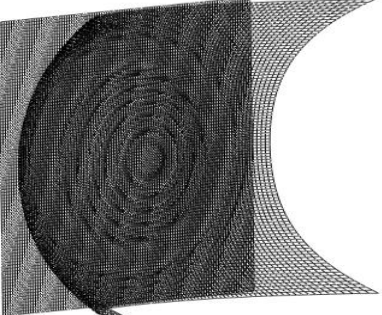

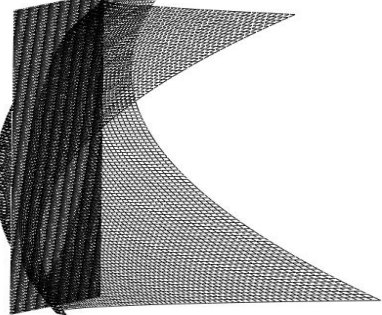

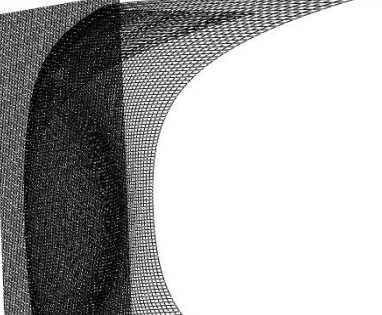

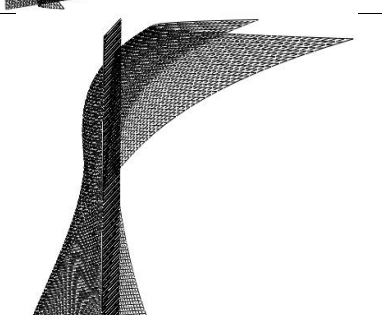
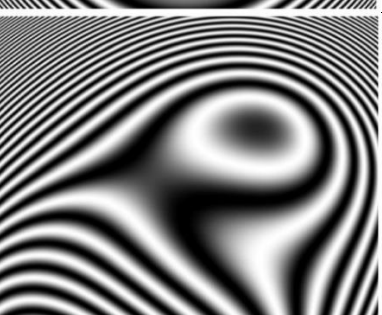
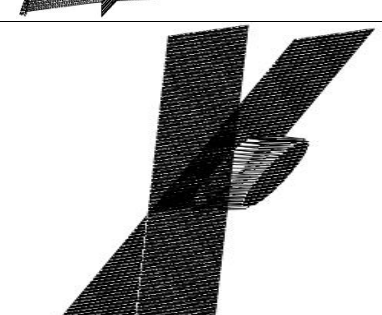
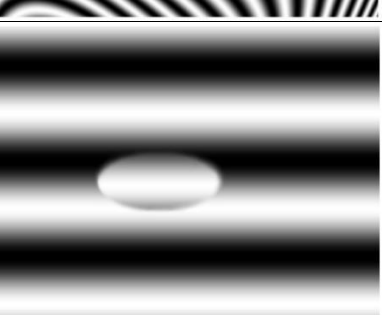
Важным параметрами при измерении является ширина и число интерференционных полос, которые возможно наблюдать в рабочей области. Поэтому для смещения наблюдаемой области создают наклон опорного фронта, если картина интенсивностей имеет симметрию. В результате чего становится возможным посчитать большее количество полос. Конструируют пробные стёкла так, чтобы размер интерференционных колец, на основании практического опыта, был больше 0,05 мм.

В таблице 1 представлены различные формы волнового фронта и получаемые в результате картины распределения интенсивностей. Для визуализации измеряемый волной фронт сдвинут относительно опорного на постоянную величину кратную 2π , поэтому сдвиг не влияет на распределение интенсивности. Видно, что при постоянном сдвиге интерференционная картина не образуется, то есть равномерно засвеченный экран. При наклоне образуются параллельные светлые и тёмные полосы. При расфокусировке образуются кольца. В случае действия нескольких аберраций картина интенсивности становится сложной.

Таблица 1 – Примеры интерференционных картин

№ п/п	Тип отклонения	Опорный и измеряемый волной фронт	Интерференционная картина
1	Постоянный сдвиг		
2	Постоянный наклон		
3	Астигматизм		
4	Расфокусиров- ка		
5	Кома		

Продолжение таблицы 1

№ п/п	Тип отклонения	Опорный и измеряемый волной фронт	Интерференционная картина
6	Сферическая		
7	Астигматизм и расфокусировка		
8	Кома и сферическая		
9	Сумма всех выше перечисленных (2, 3, 4, 5, 6)		
10	Локальная неоднородность		

При помощи ДОЭ можно сформировать опорную волну любой формы. А также на одном элементе создавать несколько различных опорных волн. Поэтому ДОЭ широко используются при высококачественной сборке сложных оптических систем, имеющих внеосевые элементы [41] и асферические поверхности [42]. Интересно отметить, что в статье [42] обнаружена вырожденная ситуация, заключающаяся в том, что при контроле поверхности одного из элементов, во время сборки системы, при нахождении в системе нескольких не юстированных оптических элементов возникает компенсация искажений в результате наложения погрешностей положения элементов. Чтобы предотвратить появление такой ситуации ведут последовательную сборку оптической системы, то есть сначала устанавливают первый элемент и юстируют его, потом второй и также юстируют его и так далее. Это говорит о том, что одна конфигурация волнового фронта многокомпонентной системы может соответствовать нескольким групповым отклонениям элементов от идеального положения.

Применяют интерференционный метод для: определения зависимости радиуса кривизны поверхности от температуры [43], калибровки адаптивных зеркал [44], тестировании кольцевых субапертур [45]. При работе с большой оптикой возникает проблема с измерением всей поверхности, поэтому промеряют поверхность частями и потом сшивают интерферометрические картины [46–48].

Существует интерференционный метод без использования опорной волны [49]. Основан он на том, что измеряемый фронт дублируется и его копию сдвигают или вращают относительно исходного, а потом складывают с измеряемым в результате получается интерференционная картина.

1.4. Восстановление волнового фронта по распределению интенсивности

1.4.1. Полиномы Цернике

Полиномы Цернике – это система круговых полиномов ортогональных внутри единичного круга. Подобных систем полиномов может быть множество, но выбор пал на полиномы Цернике по следующим причинам:

а) Коэффициенты Цернике являются независимыми друг от друга (свойство ортогональности разложения), поэтому вклад любого из членов ряда невозможно частично или полностью заменить другими членами ряда. Из-за этого коэффициенты Цернике имеют физический смысл, то есть при использовании в оптике каждому полиному соответствует какая-либо аберрация. Это означает, что при разложении аберрации по полиномам Цернике каждый из коэффициентов даёт представление о качестве изображения или оптической системы независимо от других коэффициентов.

б) Значения коэффициентов уменьшаются увеличением степени полиномов, то есть ряд полиномов является сходящимся.

в) Полиномы Цернике образуют устойчивый базис к погрешностям вычислений.

Из-за описанных выше преимуществ полиномов Цернике они нашли широкое применение в оптике и других областях: реконструкция и коррекция волнового фронта систем с круговой апертурой [50–52] и кольцевой апертурой [53], классификации изображений [54,55], описания фотометрических характеристик светодиодов [56], моделировании поверхностей и деформации поверхностей [57–64], в анализе интерферограмм [65], формировании лазерного фронта [66], в зрачковых фильтрах [67,68], для расчёта и фильтрации пространственных частот поверхности [69,70].

Работы по исследованию возможностей применения и различных свойств полиномов Цернике ведутся и по сей день [71–78].

Рассмотрим представление волнового фронта W как ряд полиномов:

$$W(\rho, \varphi) = \sum_{n,m} C_n^{\pm m} Z_n^{\pm m}(\rho, \varphi). \quad (1)$$

Учитывая возможные комбинации индексов n и m можно записать последовательность полиномов в следующем виде: $Z_0^0, Z_1^{-1}, Z_1^1, Z_2^{-2}, Z_2^0, Z_2^2, Z_3^{-3}, Z_3^{-1}, Z_3^1, Z_3^3$ и так далее. Отсутствие комбинации индексов, например, $Z_1^2, Z_2^1, Z_3^2, Z_3^{-2}$ и подобных, является следствием способа математической реализации данных полиномов, то есть $n - m$ всегда чётное число. Для удобства обозначения изменим способ индексации соответствующий записанному выше: $Z_0, Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8, Z_9$ и так далее. В результате формула (1) примет вид:

$$W(\rho, \varphi) = \sum_i C_i Z_i(\rho, \varphi), \quad (2)$$

где C – коэффициенты Цернике, являющиеся константами, Z – полиномы Цернике, ρ и φ – полярные координаты.

Коэффициенты C служат для количественного описания аберраций. Сами же полиномы являются произведением радиальных полиномов на синус или косинус в зависимости от знака индекса m :

$$\begin{aligned} Z_n^{-m}(\rho, \varphi) &= R_n^{-m}(\rho) \sin(-m\varphi), \text{ при } m < 0 \text{ (нечётные полиномы),} \\ Z_n^m(\rho, \varphi) &= R_n^m(\rho) \cos(m\varphi), \text{ при } m \geq 0 \text{ (чётные полиномы).} \end{aligned} \quad (3)$$

Радиальные полиномы R можно рассчитать двумя способами:

$$R_n^m(\rho) = \frac{1}{\left(\frac{n-m}{2}\right)! \rho^m} \left\{ \frac{d}{d(\rho^2)} \right\}^{\frac{n-m}{2}} \left\{ (\rho^2)^{\frac{n+m}{2}} (\rho^2 - 1)^{\frac{n-m}{2}} \right\}, \quad (4)$$

$$R_n^m(\rho) = \sum_{s=0}^{\left(\frac{n-m}{2}\right)} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+m}{2} - s\right)! \left(\frac{n-m}{2} - s\right)!} \rho^{n-2s}, \quad (5)$$

где n – степень полинома, m – параметр угловой зависимости ($n \geq m$), ρ – нормированное радиальное расстояние ($0 \leq \rho \leq 1$).

В формуле (4) используется производная, что усложняет её использование при моделировании полиномов в отличие от формулы (5).

Полиномы Цернике вследствие ортогональности внутри единичного круга могут принимать значения от -1 до 1:

$$|Z_n^m(\rho, \varphi)| \leq 1. \quad (6)$$

Ортогональность полиномов в радиальной части записывается равенством:

$$\int_0^1 R_n^m(\rho) \rho \sqrt{2n+2} R_{n'}^m(\rho) \rho \sqrt{2n'+2} d\rho = \delta_{n,n'}, \quad (7)$$

где δ – символ Кронекера, n' - независимый индекс от n .

Ортогональность в угловой части записывается набором равенств:

$$\begin{aligned} \int_0^{2\pi} \cos(m\varphi) \cos(m'\varphi) d\varphi &= \varepsilon_m \pi \delta_{|m|,|m'|}, \\ \int_0^{2\pi} \sin(m\varphi) \sin(m'\varphi) d\varphi &= (-1)^{m+m'} \pi \delta_{|m|,|m'|}, \quad m \neq 0, \\ \int_0^{2\pi} \cos(m\varphi) \sin(m'\varphi) d\varphi &= 0, \end{aligned} \quad (8)$$

где параметр ε_m полагают равным 2, если $m = 0$, и равным 1, если $m \neq 0$.

Ортогональность полиномов Цернике, являющихся произведением радиальной и угловой части, записывается следующим образом:

$$\int Z_n^m(\rho, \varphi) Z_{n'}^{m'}(\rho, \varphi) \rho d\rho d\varphi = \frac{\varepsilon_m \pi}{2n+2} \delta_{n,n'} \delta_{m,m'}, \quad (9)$$

где интегрирование производится по единичному кругу.

Предположим, что имеется волновой фронт, являющийся суперпозицией полиномов Цернике с собственными весовыми коэффициентами. Смысл ортогональности состоит в том, что интеграл произведения двух ортогональных функций по независимым параметрам отличен от нуля только в том случае, когда функция умножается сама на себя. Отталкиваясь от этого свойства, найдём интеграл произведения произвольной функции (1) на некоторый полином Цернике $Z_{n'}^{m'}$ [79].

$$\begin{aligned}
& \iint Z_{n'}^{m'}(\rho, \varphi) W(\rho, \varphi) \rho d\rho d\varphi = \\
& = \iint Z_{n'}^{m'}(\rho, \varphi) \sum_{n,m} (C_n^m Z_n^m(\rho, \varphi)) \rho d\rho d\varphi = \\
& = \sum_{n,m} C_n^m \iint Z_{n'}^{m'}(\rho, \varphi) Z_n^m(\rho, \varphi) \rho d\rho d\varphi = C_{n'}^{m'} \frac{\varepsilon_{m'} \pi}{2n'+2}.
\end{aligned} \tag{10}$$

Из полученной формулы (10) можно получить значение весового коэффициента при полиноме Цернике $Z_{n'}^{m'}$:

$$C_{n'}^{m'} = \frac{2n'+2}{\varepsilon_{m'} \pi} \iint Z_{n'}^{m'}(\rho, \varphi) W(\rho, \varphi) \rho d\rho d\varphi. \tag{11}$$

В таблице 2 представлены первые 28 полиномов Цернике и соответствующие им абберрации. Обозначение $G(\varphi)$ отвечает за угловую часть полинома, а $N_{n,m}$ – нормировочная константа, которая умножается на полином и определяется следующим образом: если $m = 0$, то $N_{n,m} = \sqrt{n+1}$, а если $m \neq 0$, то $N_{n,m} = \sqrt{2(n+1)}$. Название высокопорядковых абберраций взяты из работы [71].

Таблица 2 – Примеры первых 28 полиномов Цернике

№ п/п	n	m	$N_{n,m}$	$R(\rho)$	$G(\varphi)$	Название
0	0	0	1	1	1	Сдвиг oz
1	1	-1	2	ρ	$\sin(\varphi)$	Дисторсия ox
2	1	1	2	ρ	$\cos(\varphi)$	Дисторсия oy
3	2	-2	$\sqrt{6}$	ρ^2	$\sin(2\varphi)$	45° Астигматизм
4	2	0	$\sqrt{3}$	$2\rho^2 - 1$	1	Расфокусировка
5	2	2	$\sqrt{6}$	ρ^2	$\cos(2\varphi)$	90° Астигматизм
6	3	-3	$\sqrt{8}$	ρ^3	$\sin(3\varphi)$	Трилистник
7	3	-1	$\sqrt{8}$	$3\rho^3 - 2\rho$	$\sin(\varphi)$	Кома ox
8	3	1	$\sqrt{8}$	$3\rho^3 - 2\rho$	$\cos(\varphi)$	Кома oy
9	3	3	$\sqrt{8}$	ρ^3	$\cos(3\varphi)$	Трилистник

Продолжение таблицы 2

№ п/п	n	m	$N_{n,m}$	$R(\rho)$	$G(\varphi)$	Название
10	4	-4	$\sqrt{10}$	ρ^4	$\sin(4\varphi)$	Четырёхлистник
11	4	-2	$\sqrt{10}$	$4\rho^4 - 3\rho^2$	$\sin(2\varphi)$	5-го порядка астигматизм 45°
12	4	0	$\sqrt{5}$	$6\rho^4 - 6\rho^2 + 1$	1	Сферическая
13	4	2	$\sqrt{10}$	$4\rho^4 - 3\rho^2$	$\cos(2\varphi)$	5-го порядка астигматизм 90°
14	4	4	$\sqrt{10}$	ρ^4	$\cos(4\varphi)$	Четырёхлистник
15	5	-5	$\sqrt{12}$	ρ^5	$\sin(5\varphi)$	Пятилистник
16	5	-3	$\sqrt{12}$	$5\rho^5 - 4\rho^3$	$\sin(3\varphi)$	Высокого порядка трилистник
17	5	-1	$\sqrt{12}$	$10\rho^5 - 12\rho^3 + 3\rho$	$\sin(\varphi)$	5-го порядка кома ох
18	5	1	$\sqrt{12}$	$10\rho^5 - 12\rho^3 + 3\rho$	$\cos(\varphi)$	5-го порядка кома оу
19	5	3	$\sqrt{12}$	$5\rho^5 - 4\rho^3$	$\cos(3\varphi)$	Высокого порядка трилистник
20	5	5	$\sqrt{12}$	ρ^5	$\cos(5\varphi)$	Пятилистник
21	6	-6	$\sqrt{14}$	ρ^6	$\sin(6\varphi)$	Шестилистник
22	6	-4	$\sqrt{14}$	$6\rho^6 - 5\rho^4$	$\sin(4\varphi)$	Высокого порядка четырёхлистник
23	6	-2	$\sqrt{14}$	$15\rho^6 - 20\rho^4 + 6\rho^2$	$\sin(2\varphi)$	7-го порядка астигматизм 45°
24	6	0	$\sqrt{7}$	$20\rho^6 - 30\rho^4 + 12\rho^2 - 1$	1	5-го порядка сферическая
25	6	2	$\sqrt{14}$	$15\rho^6 - 20\rho^4 + 6\rho^2$	$\cos(2\varphi)$	7-го порядка астигматизм 90°
26	6	4	$\sqrt{14}$	$6\rho^6 - 5\rho^4$	$\cos(4\varphi)$	Высокого порядка четырёхлистник
27	6	6	$\sqrt{14}$	ρ^6	$\cos(6\varphi)$	Шестилистник

Существуют другие способы порядковой нумерации последовательности полиномов Цернике, но последующей работе будет использоваться нумерация, приведённая в таблице 2.

1.4.2. Прямая и обратная задача

Прямая задача – это задача, в которой по известным начальным данным и по известной процедуре необходимо получить результат. В данном случае по известным коэффициентам при полиномах Цернике нужно получить распределение интенсивности в плоскости изображения. Решение этой задачи используется как один из шагов обратной задачи.

В работах [80,81] приведены примеры решения прямой задачи. В них описан общий подход и результаты моделирования, но техническая реализация опущена. Поэтому реализация прямой задачи описана в практической части. В статье [82] описан метод преобразования коэффициентов полиномов Цернике при масштабировании, вращении и перемещении зрачка системы, что может быть полезным при моделировании. Ещё важно отметить, что обычно в литературе для описания полиномов Цернике используют полярную или цилиндрическую систему координат, что неудобно при работе с прямоугольными изображениями и при дискретизации выходного зрачка. Поэтому лучше использовать запись полиномов в декартовой системе координат [83]:

Если $m > 0$, то

$$Z_n^m(x, y) = \sqrt{2(n+1)} \sum_{s=0}^{\frac{n-|m|}{2}} \sum_{j=0}^{\frac{n-|m|}{2}-s} \sum_{k=0}^{\frac{|m|-1}{2}} \frac{(-1)^{s+k} (n-s)!}{s! \left(\frac{n+|m|}{2} - s \right)! \left(\frac{n-|m|}{2} - s \right)!} \cdot \left(\frac{\frac{n-|m|}{2} - s}{j} \right) \binom{|m|}{2k} x^{n-2(s+j+k)} y^{2(j+k)}. \quad (12)$$

Если $m < 0$, то

$$Z_n^m(x, y) = \sqrt{2(n+1)} \sum_{s=0}^{\frac{n-|m|}{2}} \sum_{j=0}^{\frac{n-|m|}{2}-s} \sum_{k=0}^{\frac{|m|-1}{2}} \frac{(-1)^{s+k} (n-s)!}{s! \left(\frac{n+|m|}{2} - s \right)! \left(\frac{n-|m|}{2} - s \right)!} \cdot \left(\frac{\frac{n-|m|}{2} - s}{j} \right) \binom{|m|}{2k+1} x^{n-2(s+j+k)-1} y^{2(j+k)+1}. \quad (13)$$

Если $m = 0$, то

$$Z_n^m(x, y) = \sqrt{n+1} \sum_{s=0}^{\frac{n}{2}} \sum_{j=0}^{\frac{n-s}{2}} \frac{(-1)^s (n-s)!}{s! \left(\frac{n}{2} - s \right)! \left(\frac{n}{2} - s \right)!} \cdot \binom{\frac{n}{2} - s}{j} x^{n-2(s+j)} y^{2j}, \quad (14)$$

где $\binom{\frac{n-|m|}{2} - s}{j} \binom{|m|}{2k}$, $\binom{\frac{n-|m|}{2} - s}{j} \binom{|m|}{2k+1}$, $\binom{\frac{n}{2} - s}{j}$ в формулах (12-14)

соответственно являются биномиальными коэффициентами.

При этом должно соблюдаться условие $x^2 + y^2 \leq 1$.

Первый метод формирования распределения интенсивности в плоскости изображения описывается дифракционным интегралом Кирхгофа:

Нормированная интенсивность, нормировка производилась на интенсивность, полученную при построении изображения точки в параксиальной области, в цилиндрической системе координат f, ν, ψ [84]

$$I(f, \nu, \psi) = \frac{1}{\pi^2} \left| \int_0^1 \int_0^{2\pi} \exp[-if\rho^2] \exp[-i\nu\rho \cos(\varphi - \psi)] P(\rho, \varphi) \rho d\rho d\varphi \right|^2, \quad (15)$$

где

$$f = \frac{2\pi}{\lambda} \left(\frac{a}{R} \right)^2 z, \quad (16)$$

$$\nu = \frac{2\pi}{\lambda} \left(\frac{a}{R} \right) \sqrt{x^2 + y^2}, \quad (17)$$

$P(\rho, \varphi)$ зрачковая функция определяемая как

$$P(\rho, \varphi) = A(\rho, \varphi) \exp[iW(\rho, \varphi)], \quad (18)$$

λ – длина волны излучения, a – радиус выходного зрачка, R – радиус опорной сферы гаусса, x, y, z – декартовы координаты в плоскости изображения, $A(\rho, \varphi)$ – передаточная функция, описывающая прозрачность и форму зрачка.

В декартовой системе координат, нормированное таким образом, что интеграл по всей плоскости изображения равен 1 независимо от величины продольной координаты z

$$I(x, y, z) = \frac{1}{(\lambda D z)^2 t_0} \left| \iint \exp \left[-i2\pi \left\{ \frac{xx' + yy'}{\lambda z} - W(x'', y'') - A_0 \rho^2 \right\} \right] t(x', y') dx' dy' \right|^2, \quad (19)$$

где D – диаметр выходного зрачка, x' и y' – ненормированные декартовы координаты в плоскости выходного зрачка,

$$\rho = \frac{2\sqrt{x'^2 + y'^2}}{D}, \quad (20)$$

x'' и y'' – нормированные декартовы координаты в плоскости выходного зрачка, то есть $x''^2 + y''^2 \leq 1$,

$$t_0 = \frac{1}{D^2} \iint t^2(x', y') dx' dy', \quad (21)$$

$t(x', y')$ – амплитудная функция пропускания выходного зрачка, тоже самое что $A(\rho, \varphi)$, написанное выше, только в декартовой системе координат,

$$A_0 = -\frac{\delta z}{8\kappa r_c}, \quad (22)$$

$\delta z = z - F$, F – фокусное расстояние системы, $k = F/D$ – относительное фокусное расстояние системы, $r_c = \lambda k$ – критический радиус для данной длины волны, то есть пространственные частоты выше частоты среза $f_c = 1/r_c$ не пропускаются даже идеальной оптической системой.

Множитель A_0 отвечает за величину расфокусировки изображения, то есть при $z = F$ множитель $A_0 = 0$. Такую же функцию выполняет $\exp[-if \rho^2]$ в формуле (15).

Запишем условие нормировки:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y) dx dy = 1. \quad (23)$$

Использование интеграла Кирхгофа даёт наиболее точную картину, но использование данного метода является очень ресурсозатратным по вычислениям. Второй метод – это приближение Нижбера-Цернике, которой учитывает дифракцию, но является менее точным методом и средним по сложности вычислений. Третий метод – это метод геометрической оптики, не учитывающий дифракции на элементах, выполняемый при помощи трассировки лучей по входному зрачку системы. Так как не ставится задача обеспечения скорости вычислений, то используется в дальнейшем первый метод.

Набор получаемых данных с любой оптической системы оцифровывается и поэтому является дискретным, то есть изображение записывается приёмником в 8-битном, 16-битном диапазоне и т.д. Можно аппроксимировать получаемые измерения непрерывными функциями, но это будет вносить дополнительные погрешности. Поэтому запишем дифракционный интеграл Кирхгофа в дискретной форме [5]:

$$I_{k,l} = I(x_k, y_l, z) x_{step} y_{step}, \quad k, l = 0, 1, \dots, N-1;$$

$$P_{m,n} = \begin{cases} P(x'_m, y'_n, z), & m, n = 0, 1, \dots, M-1, \\ 0, & m, n \geq M, \end{cases} \quad (24)$$

$$I_{k,l} = \frac{g^2}{t_0 N^4} \left| \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} \exp \left[-i \frac{2\pi}{N} (km + nl) + i 2\pi (W_{m,n} + A_0 \rho_{m,n}^2) \right] t_{m,n} \right|^2, \quad (25)$$

где $N = Mg$ – количество пикселей по ширине или длине изображения, M – количество разбиений стороны квадрата, описанного вокруг выходного зрачка, тогда шаг на зрачке равен $x'_{step} = D/M$, g – приблизительно равно количеству пикселей на радиусе дифракционного изображения точки (для

уверенной передачи информации о интенсивности необходимо чтобы $g \geq 2$), определяемое как

$$g = \frac{\lambda z}{Dx_{step}}, \quad (26)$$

x_{step} – шаг в плоскости изображения.

Суть обратной задачи в данной работе состоит в том, что по распределению интенсивности необходимо получить форму волнового фронта, порождающего это распределение, без привлечения дополнительных систем, то есть интерферометров, датчиков волнового фронта и так далее. В работах [85,86] рассматривается решение в приближении малости аберраций. Данный подход вследствие приближения сильно ограничен в применении, поэтому основой для решения задачи без приближения малости аберраций является работа [5].

Выберем линейную модель формирования изображения, то есть

$$Q(c) = S \cdot I(c) + b, \quad (27)$$

где $Q = \{Q_{k,l}\}$ – количество отсчётов фотодетектора, S – нормировочная константа, которая соответствует максимальной интенсивности на фотодетекторе, например, если нормировать I так чтобы максимум был равен единице, то при 8-ми битном детекторе $S = 255$, $b = \{b_{k,l}\}$ – матрица фона, $I = \{I_{k,l}\}$ – ФРТ, которая определена выше, c – вектор коэффициентов Цернике.

Учитывая стохастичность отсчётов при регистрации излучения полагаем, что отсчёты в различных пикселях являются независимыми пуассоновскими случайными величинами. При этом вероятность получить на опыте некоторую реализацию отсчётов $Y = \{Y_{k,l}\}$ равна

$$f(Y | c) = \prod_{k,l=1}^N \frac{\exp[-Q_{k,l}(c)] \cdot Q_{k,l}(c)^{Y_{k,l}}}{Y_{k,l}!}. \quad (28)$$

В этом случае удобно рассматривать функцию противоположную функции правдоподобия представимую в виде

$$J(Y | c) = -\ln f(Y | c). \quad (29)$$

Максимально правдоподобная оценка вектора c_{inv} по наблюдаемой реализации Y определяется из поиска минимума функции $J(Y|c)$

$$c_{inv} = \arg \min_c J(Y | c), \quad (30)$$

где c_{inv} – инверсная оценка вектора c .

Поиск минимума функции многих переменных является задачей многомерной оптимизации (минимизации). Для решения данной задачи выбран алгоритм Бroyдена – Флетчера – Гольдфабра – Шанно (БФГШ). Этот метод является квазиньютоновским, то есть совмещает в себе два преимущества. Первое, обладает скоростью сходимости равной методу Ньютона. Второе, по сложности вычисления подобен градиентным методам, то есть не вычисляет на прямую обратную матрицу Гессе, а использует приближённую формулу. Запишем последовательность шагов алгоритма [87]:

- 1) Выбираем стартовую точку x_1 , начальную обратную матрицу Гессе H_1 – единичную, точность $\varepsilon > 0$, $k = 1$, вычисляем $\nabla f(x_k)$.
- 2) Определяем направление спуска $p_k = -H_k \nabla f(x_k)$.
- 3) Найдём λ_k минимизируя функцию $\min f(x_k + \lambda p_k)$, то есть выполняем одномерную минимизацию функции. Определяем $x_{k+1} = x_k + \lambda_k p_k$.
- 4) Вычисляем $f(x_{k+1})$, если выполнены условия останова, например $f(x_{k+1}) < \varepsilon_1$ или $\lambda_k p_k < \varepsilon_2$, где ε_1 и ε_2 константы, выбранные заранее, то завершаем поиск $x_{inv} = x_{k+1}$, иначе следующий шаг.
- 5) Обновляем H_k , $H_{k+1} = H_k + A_k + B_k$, по формулам (31), $k = k + 1$, переходим на шаг 1.

$$A_k = -\frac{u_k \cdot (u_k)^T}{u_k \cdot v_k}, \quad B_k = -\frac{H_k \cdot v_k \cdot (v_k)^T \cdot (H_k)^T}{H_k \cdot v_k \cdot v_k}, \quad (31)$$

где $u_k = x_{k+1} - x_k$, $v_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.

Метод БФГШ обладает локальной сходимостью, но на практике хорошо сходится и для невыпуклых функций.

На этапе три необходимо провести одномерную минимизацию, для нахождения оптимального шага. Для этой задачи выбран метод золотого сечения [88], так как обладает лучшей устойчивостью по сравнению с методами использующими информацию о первой или второй производной, но при этом

меньшей скоростью сходимости. Запишем последовательность шагов алгоритма:

- 1) Задать $\varepsilon > 0$, границы интервала a_1 и b_1 , в котором будет осуществляться поиск, $k = 1$. Вычислить λ_1 и μ_1 по формулам (32), $f(\lambda_1)$ и $f(\mu_1)$.
- 2) Если $b_k - a_k \leq \varepsilon$, то $x_{inv} = (b_k + a_k)/2$. Иначе: если $f(\lambda_k) > f(\mu_k)$, то перейти на шаг 3, если $f(\lambda_k) \leq f(\mu_k)$, то на шаг 4.
- 3) Положить $a_{k+1} = \lambda_k$, $b_{k+1} = b_k$, $\lambda_{k+1} = \mu_k$, $\mu_{k+1} = a_{k+1} + \tau(b_{k+1} - a_{k+1})$, где $\tau = (\sqrt{5} - 1)/2$. Вычислить $f(\mu_{k+1})$, перейти на шаг 5.
- 4) Положить $a_{k+1} = a_k$, $b_{k+1} = \mu_k$, $\mu_{k+1} = \lambda_k$, $\lambda_{k+1} = a_{k+1} + (1 - \tau)(b_{k+1} - a_{k+1})$. Вычислить $f(\lambda_{k+1})$, перейти на шаг 5.
- 5) $k = k + 1$, переходим на шаг 2.

$$\begin{aligned}\lambda_k &= a_k + (1 - \tau)(b_k - a_k), \\ \mu_k &= a_k + \tau(b_k - a_k).\end{aligned}\tag{32}$$

В методе БФГШ необходима информация о производной функции правдоподобия. Нахождение производной численным дифференцированием вносит большие погрешности значение производной. Поэтому лучше представить производную в аналитическом виде [5]

$$\frac{\partial}{\partial c_m} J(Y | c) = \sum_{k=1}^N \sum_{l=1}^N \left(\frac{Y_{k,l}}{Q_{k,l}} - 1 \right) G_{k,l,m},\tag{33}$$

Где $m = 1, 2, \dots, K$, K – число коэффициентов Цернике, расчёт $G_{k,l,m}$ приведён в следующей главе.

1.4.3. Матрица Фишера

Точность оценки значений c определяется матрице Фишера, которая в данном случае принимает вид [5]

$$F_{i,j}(c) = \frac{\partial^2}{\partial c_i \partial c_j} J(Y | c), \quad i, j = 1, \dots, K.\tag{34}$$

Расчёт в случае дискретных величин записывается следующим образом

$$F_{i,j}(c) = (4\pi\beta)^2 \sum_{k=1}^{N^2} \frac{1}{Q_k} \text{Im}(L_k^* \cdot \Phi_{k,i}) \cdot \text{Im}(L_k^* \cdot \Phi_{k,j}),\tag{35}$$

где $\beta = S \cdot g^2 / (t_0 \cdot N^4)$, $\text{Im}(z)$ – мнимая часть z , звёздочка означает комплексное сопряжение, Q_k – векторизованная матрица $Q_{k,l}$, то есть вектор-столбец составленный из последовательных столбцов матрицы, L_k – есть векторизованная матрица $L_{k,l}$ формула (25),

$$\Phi_{k,l,t} = \frac{g^2}{t_0 N^4} \left| \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} \exp \left[-i \frac{2\pi}{N} (km + nl) + i 2\pi (W_{m,n} + A_0 \rho_{m,n}^2) \right] t_{m,n} \cdot Z_{m,n,t} \right|^2, \quad (36)$$

$t = 1, \dots, K,$

$\Phi_{k,i}$ и $\Phi_{k,j}$ являются векторизацией матрицы $\Phi_{k,l,t}$.

Согласно спектральной теореме получаем

$$F = V \Lambda V^T, \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n), \quad V = [v_1, \dots, v_n], \quad (37)$$

где λ – собственные значения матрицы, v – собственные вектора матрицы.

Минимальная дисперсия оцениваемого коэффициента σ_m^2 равна

$$\sigma_m^2 = \frac{1}{\lambda_m}. \quad (38)$$

При расчётах целесообразно представить матрицу Фишера таким образом

$$F = A^T A, \quad (39)$$

где

$$A_m = G_m \cdot / \sqrt{Q}, \quad (40)$$

где «./» означает поэлементное деление матриц,

$$G_m = 4\pi\beta \text{Im}(L^* \cdot * \Phi_m), \quad (41)$$

где «.*» означает поэлементное умножение матриц.

Тогда, выполнив разложение по сингулярным значениям матрицы A , получим

$$A = U \Lambda V^T, \quad (42)$$

где $\Lambda = \Lambda^2$.

Используя разложение по сингулярным значениям собственные значения λ нумеруются в порядке убывания, что может быть удобным при оценивании общего количества восстанавливаемых коэффициентов.

Зная V^T , можно найти главные компоненты

$$p_n = \nu_n^T \cdot c_{inv}. \quad (43)$$

Главные компоненты позволяют найти корреляцию оценок коэффициентов Цернике.

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1. Реализация прямой задачи

Программная реализация выполнена в среде Matlab. Сначала необходимо сформировать набор полиномов Цернике, не умноженных на коэффициенты. Эти полиномы будут использоваться как константы для последующих вычислений. Входными данными являются M – дискретизация выходного зрачка, то есть количество разбиений стороны квадрата, описанного вокруг выходного зрачка; k – количество полиномов Цернике. Выходными данными являются *VectorPolinoms* – вектор полиномов, где каждому из значений вектора соответствует матрица значений по координатам x и y , то есть получается трёхмерная матрица; R – матрица значений радиус вектора полярной системы по координатам декартовой системы. Матрица R нужна будет для того, чтобы последующих функциях ограничить область вычислений и не рассчитывать точки за пределами круга (за пределами выходного зрачка).

Листинг 1 – Функция *Polinoms*

```
function [VectorPolinoms,R] = Polinoms(M,k)

% Декларация и инициализация переменных
VectorPolinoms = zeros(k,M,M);
% Декартова система координат
x = zeros(1,M);
y = zeros(1,M);
% Полярная система координат
R = zeros(M);
tt = zeros(M);

for a = 1:M
    for b = 1:M
        x(a) = -1 + 2*(a-1)/(M-1);
        y(b) = -1 + 2*(b-1)/(M-1);
        % Перевод полярной системы координат в декартовую
        R(a,b) = sqrt(x(a)^2 + y(b)^2); % Перевод радиус вектора
        if R(a,b) <= 1 % За пределами круга единичного радиуса
            % расчёт не ведётся
            % Перевод полярного угла
            if (x(a) > 0 && y(b) >= 0) || (x(a) > 0 && y(b) <= 0)
                tt(a,b) = atan(y(b)/x(a));
            elseif (x(a) < 0 && y(b) >= 0) || (x(a) < 0 && y(b) <= 0)
                tt(a,b) = pi + atan(y(b)/x(a));
            elseif (x(a) == 0 && x(a-1) > 0 && y(b) > 0) || ...
                (x(a) == 0 && x(a-1) < 0 && y(b) < 0)
```

```

end
end

```

и (5) в порядке, указанном в таблице 2, то есть идёт перевод полярной системы в

декартовую и последующий расчёт. Формулы (12-14) для расчёта непосредственно в декартовой системе не используются, потому что расчёт таким образом занимает больше времени, а результат тот же, например, для числа полиномов не выше 50 в среднем расчёт занимает в 7 раз больше времени, вероятно, из-за большего числа использования математической операции факториал. Но это не является критичным, так как данные полиномы рассчитываются всего один раз и в дальнейшем используются как константы.

Также в этой программе расчёт полиномов начинается с дисторсии, то есть первый (или нулевой) полином Цернике, отвечающий за смещение всего волнового фронта, не учитывается, так как он не влияет на распределение интенсивности. Поэтому определить его значение исходя из распределения интенсивности невозможно.

Дальше необходимо задать коэффициенты Цернике $C0$ и выполнить умножение на полученные полиномы. Выходные данные *Front* – форма волнового фронта.

Листинг 2 – Функция WaveFront

```
function [Front] = WaveFront(VectorPolinoms,R,M,C0)

% Декларация и инициализация переменной
Front = zeros(M);

% Определение размера вектора коэффициентов, является ли вектором
% столбцом или вектором строкой
Size_C = size(C0);

% Если вектор столбец, то транспонируем
if Size_C(1) > Size_C(2)
    C0 = C0';
end

% Суммирование полиномов умноженных на соответствующие
% коэффициенты
for a = 1:M
    for b = 1:M
        if R(a,b) <= 1
            Front(a,b) = C0*VectorPolinoms(:,a,b);
        end
    end
end
end

end
```


Для расчёта распределения интенсивности в плоскости изображения необходимо задать параметры, такие как; Dp – диаметр выходного зрачка; F – фокусное расстояние системы, z – расстояние от главной плоскости системы до плоскости изображения, приближенно можно считать от крайнего оптического элемента системы, отвечает за расфокусировку изображения, при $z = F$ плоскость изображения находится в фокусе и расфокусировка равна нулю; $wave$ – длина волны излучения; N – дискретизация в плоскости изображения, то есть число разбиений стороны квадрата; $step$ – расстояние между точками расчёта в плоскости изображения (реальный размер пикселя на приёмном устройстве). Произведение $N \cdot step$ равняется размеру, построенного изображения. Выходные данные I – распределение интенсивности, нормированное так, чтобы максимум равнялся единице; L – комплексное распределение, необходимо будет для решения обратной задачи; $beta$ – нормировочная константа, необходимо для решения обратной задачи. Входные параметры описанные выше записываются в вектор для удобства обращения к функции.

Листинг 3 – Функция *DirectTask*

```
function [I,L,beta] = DirectTask(C0,VectorPolinoms,R,Set)

% Раскрытие вектора параметров
M = Set(1);
N = Set(2);
Dp = Set(3);
z = Set(4);
wave = Set(5);
step = Set(6);
F = Set(7);

% Формирование волнового фронта
Front = WaveFront(VectorPolinoms,R,M,C0);

% Декларация инициализация переменных
L = zeros(N);
I = zeros(N);

% Расчёт параметров g и A (см. формулы (22) и (26))
g = wave*z/(Dp*step);
A = -(z-F)*(Dp^2)/(8*wave*F^2);

for k = 1:N
```

```

    for l = 1:N
        for a = 1:M
            for b = 1:M
                if R(a,b) <= 1
                    L(k,l) = L(k,l) + exp(-sqrt(-1)*2*pi*...
((k-(N-1)/2-1)*(a-(M-1)/2-1)+(l-(N-1)/2-1)*(b-(M-1)/2-1))/(M*g) -
Front(a,b)-A*R(a,b)^2));
                end
            end
        end
        I(k,l) = (abs(L(k,l)))^2;
    end
end

% Расчёт коэффициента нормировки
beta = 1/max(I(:));

% Нормировка матрицы
I = beta*I;

end

```

В программе выше не используется нормировка, описанная в формуле (23), так как для дискретных отсчётов камер такая нормировка плохо подходит, по причине того, что не позволяет задать нужный диапазон значений (8 *bit*, 16 *bit* и т.д.) простым умножением на соответствующий предел диапазона. Также функция выше выдает только квадратные изображения.

Полученное распределение I осталось только умножить на нужный диапазон значений и прибавить шумы. Входные параметры $Norm$ – предел диапазона значений, например, для 8 *bit* $Norm = 255$; $Noise$ – матрица шума, можно задавать как число. Выходные данные Q – распределение интенсивности в плоскости изображение.

Листинг 4 – Функция *Model*

```

function [Q] = Model(I, Norm, Noise)

% Формирование модели изображения
Q = Norm*I + Noise;

% Округление до целых значений
Q = round(Q);

end

```

Приведём пример вызова функции и результатов:

Листинг 5 – Пример вызова функций

```
M = 100; k = 15;  
[V,R] = Polinoms(M,k);  
C0 = zeros(1,k);  
C0(1) = 0.5; C0(7) = 0.5;  
Front = WaveFront(V,R,M,C0);  
surf(Front)  
N = 100; Dp = 10*10^(-3); z = 240*10^(-3); wave = 0.63*10^(-6);  
step = 2*10^(-6); F = 240*10^(-3);  
Set = [M,N,Dp,z,wave,step,F];  
I = DirectTask(C0,V,R,Set);  
Q = Model(I,255,0);  
imshow(Q,[0,255])
```

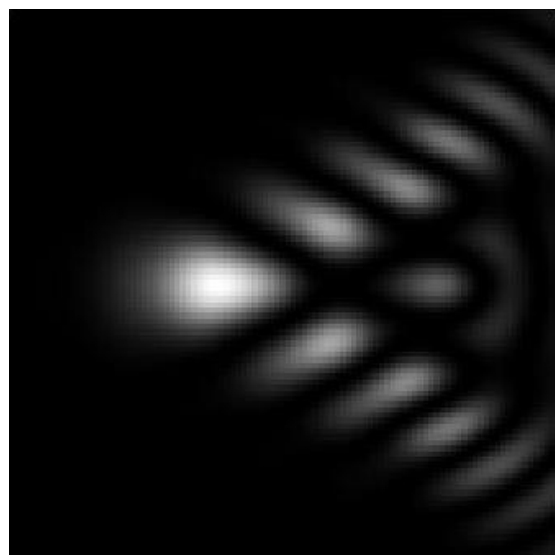
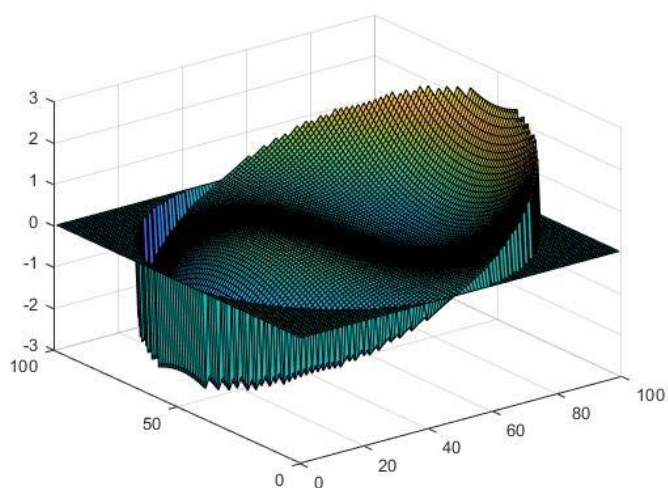


Рисунок 5 – Форма волнового фронта и распределение интенсивности

2.2. Реализация обратной задачи

Начнём с основной функции *BFGS*, а потом опишем вызываемые в ней функции. Эта функция использует метод многомерной минимизации функции БФГШ, математический алгоритм которого описан ранее. Входные данные *C0Start* – начальное приближение коэффициентов; *Function* – функция минимум которой будет рассчитываться, в данном случае это будет функция правдоподобия; *SetInverse* – вектор параметров обратной задачи; *MaxIters* –

максимальное количество итераций функции *BFGS*; *MaxStep* – максимальный шаг приращения коэффициента Цернике; *LeftBord* – минимально допустимое значения коэффициента Цернике; *RightBord* – максимально допустимое значения коэффициента Цернике; *MSD* – пороговое значение СКО реального и рассчитанного изображения; *ErrorCoeff* – граничное значение приращения коэффициентов Цернике. *SetDirect* – тоже самое, что и *Set*, описанное в прямой задаче; *KC0* – тоже самое, что и *k*, описанное в прямой задаче; *RealQ* – реально изображение, к которому идёт приближение.

Листинг 6 – Функция *BFGS*

```
function [C0Inv,Iters,MSDInv] =
BFGS(C0Start,Function,SetInverse,VectorPolinoms,R,SetDirect,Norm,
Noise,KC0,RealQ)

% Раскрытие вектора параметров
MaxIters = SetInverse(1);
MaxStep = SetInverse(2);
LeftBord = SetInverse(3);
RightBord = SetInverse(4);
MSD = SetInverse(5);
ErrorCoeff = SetInverse(6);

% Формирование нового вектора коэффициентов, чтобы не
% переписывать входные коэффициенты
C0Inv = C0Start;

% Формирование единичной матрицы, которая выполняет роль обратной
% матрицы Гессе
H = eye(KC0);

Iters = 0;

% Расчёт градиента функции правдоподобия
Grad1 =
Derivative(C0Inv,KC0,RealQ,VectorPolinoms,R,SetDirect,Norm,Noise);

flag1 = 0;

while flag1 == 0
    Iters = Iters + 1;
    if Iters > MaxIters
        break
    end

    % Формирование вектора направлений движения изменения
    % коэффициентов
    B = -1*H*Grad1;
```

```

% Определение границ коэффициента шага, для того, чтобы не
% выйти за диапазон допустимых значений
Lmax = Borders(Grad1,MaxStep,10^(-3));

if Lmax == 0
    break
end

Lleft = -Lmax;
Lright = Lmax;

% Расчёт коэффициента шага, методом одномерной минимизации
% функции
Lambda =
LineSearch(10,10^(10),Lleft,Lright,C0Inv,VectorPolinoms,R,
SetDirect, Norm, Noise, RealQ, B, Function);

% Формирование вектора приращений коэффициентов C0Inv
d = Lambda*B;

% Ограничение на приращение коэффициентов
for t = 1:KC0
    if abs(d(t)) > MaxStep && d(t) > 0
        d(t) = MaxStep;
    elseif abs(d(t)) > MaxStep && d(t) < 0
        d(t) = -MaxStep;
    end
end

% Формирование новых коэффициентов
C0Inv = C0Inv + d;

flag2 = 0;

% Если хоть один из коэффициентов достигнет границы, то
% завершаем работу
for t = 1:KC0
    if C0Inv(t) <= LeftBord
        C0Inv(t) = LeftBord;
        flag2 = 1;
    elseif C0Inv(t) >= RightBord
        C0Inv(t) = RightBord;
        flag2 = 1;
    end
end

if flag2 == 1
    break
end

% Расчёт среднего квадратичного отклонения (СКО) двух
% изображений: реального и рассчитанного

```

```

MSDInv =
MSDFunction(C0Inv,VectorPolinoms,R,SetDirect,Norm,Noise,RealQ);

% Если СКО меньше заданного, то завершаем работу
if MSDInv <= MSD
    break
end

% Расчёт нового градиента
Grad2 =
Derivative(C0Inv,KC0,RealQ,VectorPolinoms,R,SetDirect,Norm,Noise);

DeltaGrad = Grad2 - Grad1;
Grad1 = Grad2;
Indicator = 0;

% Если шаг изменения всех коэффициентов меньше заданного, то
% завершаем работу
for t = 1:KC0
    if abs(d(t)) <= ErrorCoeff
        Indicator = Indicator + 1;
    end
end

if Indicator == KC0
    break
end

% Рассчитываем новую приближенную обратную матрицу Гессу
ddT = d*d';
dTD = d'*DeltaGrad;
if dTD == 0
    break
end

DTHD = DeltaGrad'*H*DeltaGrad;
dDTH = d*DeltaGrad'*H;
HDdT = H*DeltaGrad*d';
H = H + (1 + DTHD/dTD)*ddT/dTD - dDTH/dTD - HDdT/dTD;

% На 4, 8, 12 и 16 итерации обновляем обратную матрицу Гессе,
% то есть формируем снова единичную матрицу. Оптимально при
% максимальном количестве итераций равным 20.
if Iters == 4 || Iters == 8 || Iters == 12 || Iters == 16
    H = eye(KC0);
end
end
end

```

В конце программы используется обновление обратной матрицы Гессе на определённых итерациях. Значения этих итераций получены опытным путём.

Такую операцию приходится выполнять потому, что в результате расчётов набегают погрешность вычисления приближённой обратной матрицы, что приводит к неверным шагам и удалению от минимума функции. Этот процесс можно автоматизировать для любого значения максимального количества итераций. Для этого необходимо на каждой итерации проверять на сколько хорошо обусловлена обратная матрица и обновлять её, когда теряет свою обусловленность.

Первая функция, которая вызывается в *BFGS* это *Derivative*. Она выполняет расчёт градиента функции правдоподобия, по формуле (33). Переменная $Ff \equiv G$, расчёт которой производится по формулам (36) и (41).

Листинг 7 – Функция *Derivative*

```
function [Deriv] =
Derivative(C0,KC0,RealQ,VectorPolinoms,R,Set,Norm,Noise)

% Раскрытие вектора параметров
M = Set(1);
N = Set(2);
Dp = Set(3);
z = Set(4);
wave = Set(5);
step = Set(6);
F = Set(7);

% Декларация и инициализация переменных
Ff = zeros(KC0,N,N);
Deriv = zeros(KC0,1);

% Формирование волнового фронта
Front = WaveFront(VectorPolinoms,R,M,C0);

% Расчёт распределения интенсивности
[I,L,beta] = DirectTask(C0,VectorPolinoms,R,Set);

% Формирование модели изображения
Q = Model(I,Norm,Noise);

% Расчёт параметров g и A
g = wave*z/(Dp*step);
A = -(z-F)*(Dp^2)/(8*wave*F^2);

for k = 1:N
    for l = 1:N
        for a = 1:M
            for b = 1:M
                for t = 1:KC0
```

```

        if R(a,b) <= 1
            Ff(t,k,l) = Ff(t,k,l)+ exp(-sqrt(-1)*2*...
pi*((k-(N-1)/2-1)*(a-(M-1)/2-1)+(1-(N-1)/2-1)*(b-(M-1)/2-1))/...
(M*g)-Front(a,b)-A*R(a,b)^2))*VectorPolinoms(t,a,b);
        end
    end
end
end
end
end

% Сопряжение матрицы
L = conj(L);

for t = 1:KC0
    for k = 1:N
        for l = 1:N
            Ff(t,k,l) = 4*pi*beta*imag(L(k,l)*Ff(t,k,l));
            Deriv(t) = Deriv(t)+((RealQ(k,l)+1)/(Q(k,l)+1)-1)*...
Ff(t,k,l);
        end
    end
end
end
end

```

В этой функции формула (33) немного изменена для избавления от нулей в знаменателе и числителе прибавлением единицы, то есть выполнено смещение динамического диапазона 0 – 255 на 1 – 256.

Следующей идёт программа *Borders*. Её задача состоит в определении граничного значения L_{max} переменной $Lambda$, суть которой будет пояснена позже. Входные данные $Grad$ – градиент минимизируемой функции; $MinLimit$ – минимальное значение предела переменной $Lambda$. В функции *BFGS* значение $MinLimit = 10^{-3}$, оно необходимо только для того, что избежать нулевого значения L_{max} , такое возможно, когда функция находится вблизи точки локального минимума. Его можно изменить исходя из вида минимизируемой функции и значения $MaxStep$.

Листинг 8 – Функция *Borders*

```

function [Lmax] = Borders(Grad,MaxStep,MinLimit)

if abs(max(Grad)) >= abs(min(Grad))
    Max = abs(max(Grad));
else

```



```

        Max = abs(min(Grad));
end

if Max < MinLimit
    Lmax = 0;
else
    Lmax = MaxStep/Max;
end

end

```

В методе многомерной минимизации БФГШ для определения наилучшего шага изменения искомых параметров используется коэффициент *Lambda*, расчёт которого ведётся методом одномерной минимизации. В данном случае используется метод золотого сечения, описанный ранее. Входные данные *LimitIters* – максимальное число итераций, в функции *BFGS* это значение равно 10, определено опытным путём, но значение можно изменять исходя из вида минимизируемой функции; *ErrorLambda* – минимально возможный шаг изменения коэффициента *Lambda*, погрешность вычисления, выбрано значение равное 10^{-10} ; *Lleft* и *Lright* – границы диапазона в пределах которого осуществляется поиск минимума функции;

Листинг 9 – Функция *LineSearch*

```

function [Lambda] =
LineSearch(LimitIters,ErrorLambda,Lleft,Lright,C0,VectorPolinoms,
R,Set,Norm,Noise,RealQ,B,Function)

% Расчёт коэффициента золотого сечения
tau = (sqrt(5) - 1)/2;

Iters = 0;
k = 1;

% Декларация и инициализация переменных
Left = zeros(LimitIters,1);
Right = zeros(LimitIters,1);
L1 = zeros(LimitIters,1);
R1 = zeros(LimitIters,1);

% Определение начальных границ
Left(k) = Lleft;
Right(k) = Lright;

while abs(Right(k)-Left(k)) > ErrorLambda
    Iters = Iters + 1;

```

```

    if Iters > LimitIters
        break
    end

    L1(k) = Left(k) + (1-tau)*(Right(k)-Left(k));
    R1(k) = Left(k) + tau*(Right(k)-Left(k));

    if
IncrementFunction(C0,VectorPolinoms,R,Set,Norm,Noise,RealQ,L1(k),
B,Function) > IncrementFunction(C0,VectorPolinoms,R,Set,Norm,
Noise,RealQ,R1(k),B,Function)
        Left(k+1) = L1(k);
        Right(k+1) = Right(k);
        L1(k+1) = R1(k);
        R1(k+1) = Left(k+1) + tau*(Right(k+1)-Left(k+1));
    else
        Left(k+1) = Left(k);
        Right(k+1) = R1(k);
        R1(k+1) = L1(k);
        L1(k+1) = Left(k+1) + (1-tau)*(Right(k+1)-Left(k+1));
    end
    k = k + 1;
end

Lambda = (Right(k) + Left(k))/2;

end

```

В программе выше идёт вызов функции *IncrementFunction*, которая рассчитывает приращений всех параметров (коэффициентов Цернике) минимизируемой функции.

Листинг 10 – Функция IncrementFunction

```

function [Increment] =
IncrementFunction(C0,VectorPolinoms,R,Set,Norm,Noise,RealQ,Lambda,
B,Function)

C0 = C0 + Lambda*B;
Increment = Function(C0,VectorPolinoms,R,Set,Norm,Noise,RealQ);
end

```

Как говорилось уже раньше минимизируемой функцией в нашем случае является функция правдоподобия формула (29), под названием *LikelihoodFunction*.

Листинг 11 – Функция LikelihoodFunction

```

function [Likelihood] =
LikelihoodFunction(C0,VectorPolinoms,R,Set,Norm,Noise,RealQ)

N = Set(2);

```

```

% Расчёт распределения интенсивности
I = DirectTask(C0,VectorPolinoms,R,Set);
Q = Model(I, Norm, Noise);

% Расчёт константы
const = Factorial(RealQ + 1,N);
Likelihood = const;

for k = 1:N
    for l = 1:N
        Likelihood = Likelihood + ...
            (Q(k,l) + 1) - (RealQ(k,l) + 1)*log(Q(k,l) + 1);
    end
end
end

```

В программе используется прибавление единицы к распределениям Q и $RealQ$, чтобы избавиться от нулей, аналогичная операция есть в функции *Derivative*. Это сделано в первую очередь для того, чтобы исключить ноль в логарифме.

При расчёте величины *const* используется функция *Factorial*. Связано это с тем, что встроенной функцией расчёт логарифма от факториал величины больше 170 невозможен, так программа сначала пытается рассчитать факториал 170, а потом взять логарифм от него, но из-за ограничения максимально возможного числа (10^{307}) выдаёт ошибку. Поэтому написана данная программа с использованием свойства логарифма. Порог в 100, а не в 170 выбран субъективно.

Листинг 12 – Функция *Factorial*

```

function [const] = Factorial(Q,N)

const = 0;
for k = 1:N
    for l = 1:N
        if Q(k,l) <= 100
            const = const + log(factorial(Q(k,l)));
        else
            for t = 1:(Q(k,l)-100)
                const = const + log(Q(k,l)-(t-1));
            end
            const = const + log(factorial(100));
        end
    end
end
end
end

```

Крайней вызываемой функцией является *MSDFunction*. Она рассчитывает среднее квадратичное отклонение двух изображений. Её задача состоит в том, чтобы оценить насколько близко (похоже) рассчитанное изображение на реально, помимо функции правдоподобия. Количественная оценка значений функции правдоподобия мало о чём говорит, а среднее квадратичное отклонение интенсивности в пикселе уже лучше отражает количественную оценку. Например, если СКО равняется двум, то ясно что в среднем по всему изображению отклонения интенсивности в пикселях равняется два. Конечно, данный критерий не гарантирует, что рассчитанное распределение повторяет реальное, но чем ближе он к нулю, тем ниже вероятность того, что рассчитанные коэффициенты не совпадают с реальными, но распределения интенсивности схожи. Также такой критерий, зная максимум шума на изображении, позволяет довольно просто определить верхнюю границу СКО, то есть на сколько близки могут быть два изображения в результате расчёта.

Листинг 13 – Функция MSDFunction

```
function [MSD] =
MSDFunction(C0,VectorPolinoms,R,Set,Norm,Noise,RealQ)

N = Set(2);

% Расчёт распределения интенсивности
I = DirectTask(C0,VectorPolinoms,R,Set);
Q = Model(I,Norm,Noise);

MSD = 0;

for k = 1:N
    for l = 1:N
        MSD = MSD + (Q(k,l) - RealQ(k,l))^2;
    end
end

MSD = sqrt(MSD)/N;

end
```

Перед началом решения обратной задачи надо оценить сколько коэффициентов Цернике возможно восстановить исходя из величины расфокусировки. И после решения обратной задачи желательно проверить,

сколько коэффициентов позволяет восстановить полученное распределение интенсивности. Для этой оценки используется представление информации в виде матрицы Фишера формула (35). Для расчёта собственных значений и собственных векторов матрицы будем использовать её представление по формуле (39) и расчёт по формуле (40). Данные вычисления проводит функция *FisherMatrix*. По сути эта программа повторяет программу *Derivative*, с небольшими изменениями в конце.

Листинг 14 – Функция *FisherMatrix*

```
function [Ff] =
FisherMatrix(C0,KC0,VectorPolinoms,R,Set,Norm,Noise)

% Раскрытие вектора параметров
M = Set(1);
N = Set(2);
Dp = Set(3);
z = Set(4);
wave = Set(5);
step = Set(6);
F = Set(7);

% Декларация и инициализация переменной
Ff = zeros(KC0,N,N);

% Формирование волнового фронта
Front = WaveFront(VectorPolinoms,R,M,C0);

% Расчёт распределения интенсивности
[I,L,beta] = DirectTask(C0,VectorPolinoms,R,Set);

% Формирование модели изображения
Q = Model(I,Norm,Noise);

% Расчёт параметров g и A
g = wave*z/(Dp*step);
A = -(z-F)*(Dp^2)/(8*wave*F^2);

for k = 1:N
    for l = 1:N
        for a = 1:M
            for b = 1:M
                for t = 1:KC0
                    if R(a,b) <= 1
                        Ff(t,k,l) = Ff(t,k,l)+ exp(-sqrt(-1)*2*...
pi*((k-(N-1)/2-1)*(a-(M-1)/2-1)+(1-(N-1)/2-1)*(b-(M-1)/2-1))/...
(M*g)-Front(a,b)-A*R(a,b)^2))*VectorPolinoms(t,a,b);
                    end
                end
            end
        end
    end
end
```

```

end
end
end
end

% Сопряжение матрицы
L = conj(L);

for t = 1:KC0
    for k = 1:N
        for l = 1:N
            Ff(t,k,l) = 4*pi*beta*Norm*imag(L(k,l)*Ff(t,k,l));
            Ff(t,k,l) = Ff(t,k,l)/sqrt(Q(k,l) + 1);
        end
    end
end
end
end

```

Дальше для того, чтобы перемножить матрицы по формуле (39) или воспользоваться вложенной функцией разложения матрицы по сингулярным значениям по формуле (42) необходимо векторизовать полученную матрицу Ff . Функция *VectorizationFf* из трёхмерной матрицы Ff векторизацией по первым двум индексам формирует двумерную матрицу.

Листинг 15 – Функция VectorizationFf

```

function [VectorM] = VectorizationFf(Matrix,N,KC0)

VectorM = zeros(N^(2),KC0);

for t = 1:KC0
    a = 1;
    for k = 1:N
        for l = 1:N
            VectorM(a,t) = Matrix(t,k,l);
            a = a + 1;
        end
    end
end
end
end

```

После получения векторизованной матрицы перемножим транспонированную матрицу на обычную, тогда получим матрицу Фишера. Для оценки общего количества восстанавливаемых коэффициентов можно

воспользоваться встроенной функцией *svd* для разложения матрицы по формуле (42). Но данная функция требует много оперативной памяти, поэтому для больших изображений (больше 200x200 пикселей) и количестве коэффициентов больше 50 возможны трудности с расчётов. Если возникнет данная проблема, то появляется необходимость переписать встроенную функцию самостоятельно. Второе, использовать способ расчёта матрицы Фишера по формуле (35) или (39), и потом сортировать собственные значения матрицы в порядке убывания.

2.3. Проверка алгоритма по синтезированным изображениям

Для формирования первого синтезированного изображения сначала выберем параметры расчёта. Пусть $M = 50$; $N = 100$; $Dp = 300$ мм; $z = 6003$ мм; $wave = 0,6328$ мкм; $step = 3$ мкм; $F = 6000$ мм. То есть расфокусировка $\delta z = z - F = 3$ мм. Диапазон интенсивности от 0 до 255, шум равен нулю. Зададим не нулевые коэффициенты Цернике: $C0_1 = -0,4$; $C0_2 = 0,4$; $C0_4 = -0,2$; $C0_7 = 0,3$; $C0_8 = -0,2$. В результате получим изображение рисунок 6.

При этом параметр $g = 4,22$ – приблизительное количество пикселей на размере дифракционного изображения точки и коэффициент расфокусировки равен $A = -1,48$.

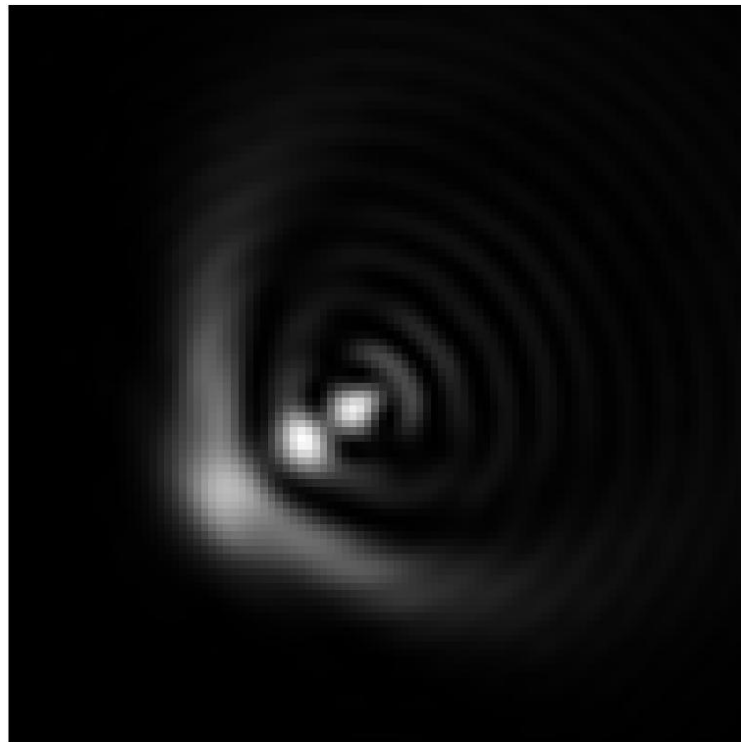


Рисунок 6 – Результат синтезированного изображения

Теперь, когда имеем синтезированное изображение до начала решения обратной задачи необходимо оценить сколько коэффициентов Цернике возможно восстановить при данной расфокусировке и предположении не искривлённого волнового фронта. Рассчитаем собственные значения матрицы Фишера до 50-ого коэффициента, рисунок 7.

Выберем только первые 20 коэффициентов, тогда максимальная погрешность из минимальных не превысит $9,04 \cdot 10^{-4}$. Возможно восстановить и больше коэффициентов, но так как мы заранее знаем, что остальные коэффициенты должны быть нулевые, то это особого смысла не имеет. Если коэффициент расфокусировки был равен нулю $A = 0$, то есть картина в фокусе, то невозможно было бы восстановить 50 коэффициентов, в лучшем случае только 26 ($\sigma^2 = 0,0027$), так как в этом случае дисперсия для 27-го $\sigma^2 = 1,01 \cdot 10^{12}$, то есть очень велика, рисунок 9.

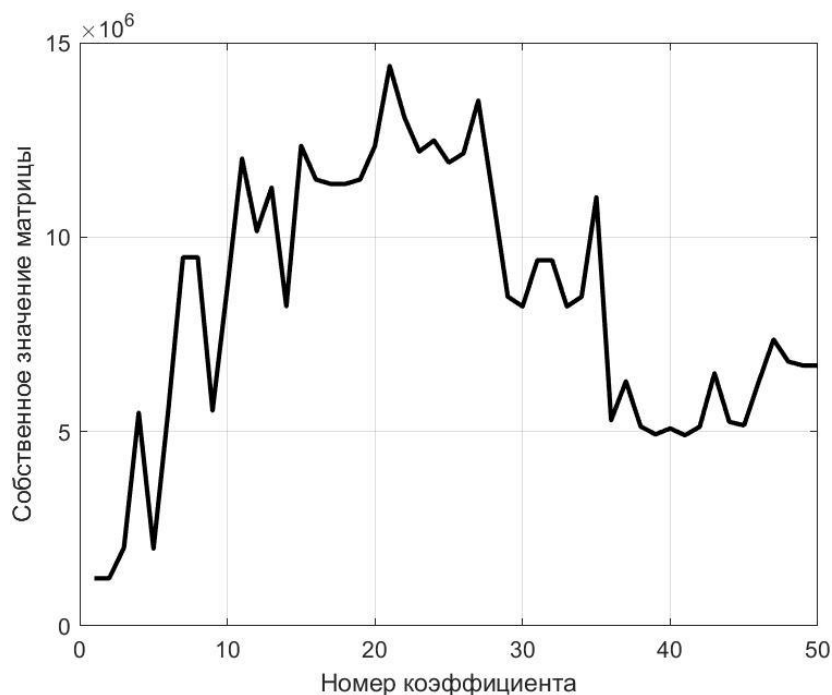


Рисунок 7 – Собственные значения матрицы Фишера для расфокусировки в 3 мм

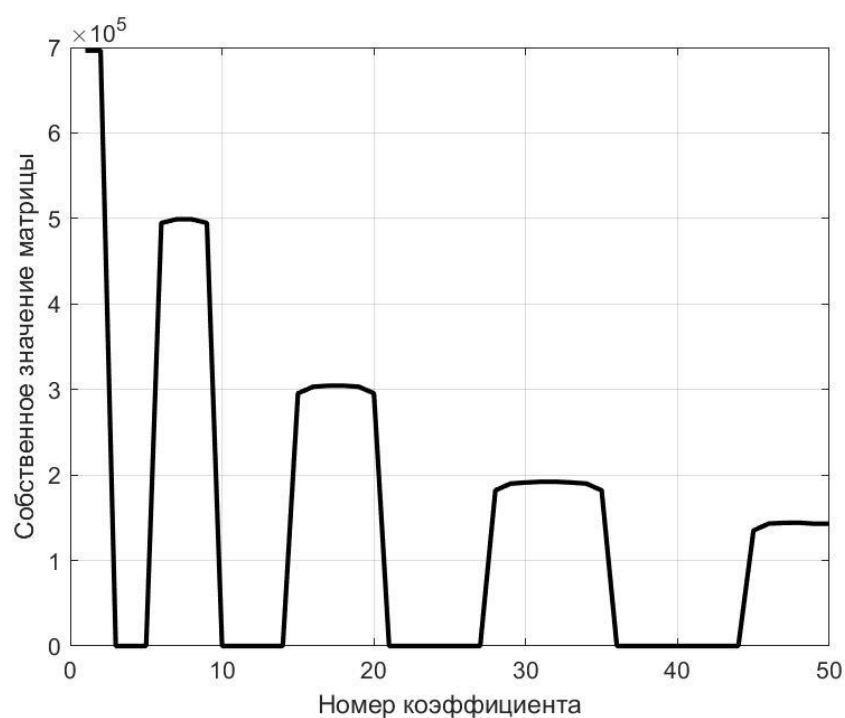


Рисунок 8 – Собственные значения матрицы Фишера для картины в фокусе

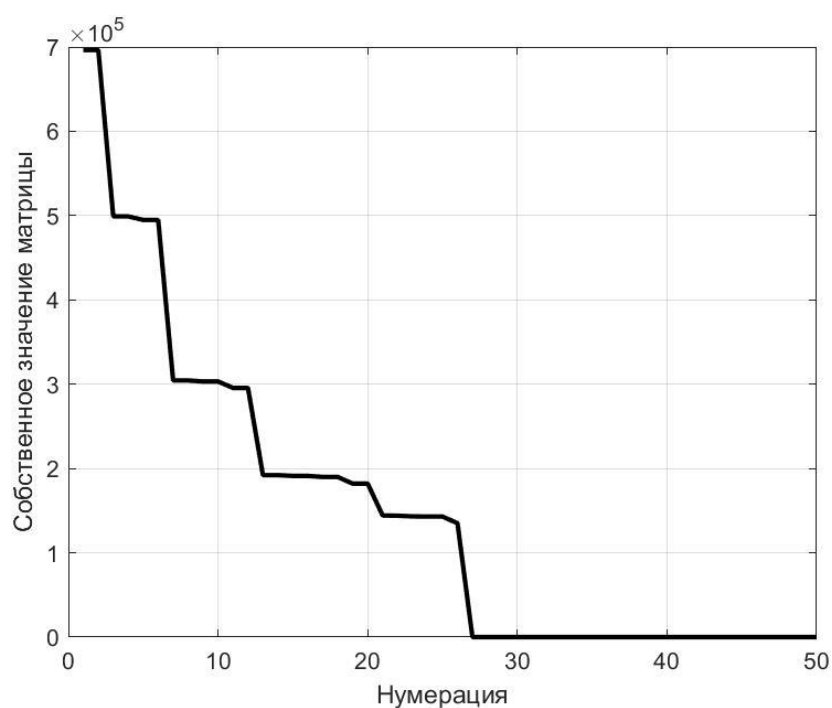


Рисунок 9 – Собственные значения матрицы Фишера для картины в фокусе, нумерация в порядке убывания собственных значений

Как видно из рисунка 8 осесимметричные aberrации имеют очень высокую дисперсию, то есть остаются неизвестными. Поэтому фокальная картина имеет меньше информации о aberrациях, чем расфокусированная.

Зададим параметры обратной задачи. $MaxIters = 20$; $MaxStep = 0,2$; $LeftBord = -1$; $RightBord = 1$; $MSD = 1$; $ErrorCoeff = 10^{-7}$. Так как шум на синтезированном изображении отсутствует, то в результате расчёта в теории можно получить $CKO(MSD)$ равное нулю, но из-за овражного характера функции правдоподобия в области минимума и вычислительной погрешности это едва ли удастся сделать. Поэтому CKO выбрано равное единице.

Стартовые коэффициенты для расчёта выберем нулевые. Тогда начальное значение CKO равно 32,30. Ниже представлено пошаговое описание о результатах обратной задачи.

Запуск функции *BFGS*:

Значения коэффициентов округлены до трёх знаков после запятой.

1) $Iters = 21$; $MSD = 10,79$; $C0(1) = -0,216$; $C0(2) = 0,186$; $C0(3) = 0,113$;
 $C0(4) = -0,427$; $C0(5) = -0,022$; $C0(6) = 0,003$; $C0(7) = 0,279$;
 $C0(8) = -0,197$; $C0(9) = -0,060$; $C0(10) = -0,004$; $C0(11) = 0,016$;
 $C0(12) = -0,052$; $C0(13) = 0,012$; $C0(14) = -0,016$; $C0(15) = -0,010$;
 $C0(16) = -0,011$; $C0(17) = 0,052$; $C0(18) = -0,031$; $C0(19) = -0,004$;
 $C0(20) = -0,025$;

Прерывание работы программы из-за достижения $MaxIters$. Запускаем программу снова используя полученные коэффициенты, округляя их до трёх знаков после запятой.

2) $Iters = 21$; $MSD = 4,35$; $C0(1) = -0,274$; $C0(2) = 0,267$; $C0(3) = 0,092$;
 $C0(4) = -0,336$; $C0(5) = -0,016$; $C0(6) = 0,019$; $C0(7) = 0,359$;
 $C0(8) = -0,259$; $C0(9) = -0,026$; $C0(10) = 0,031$; $C0(11) = 0,020$;
 $C0(12) = -0,038$; $C0(13) = 0,002$; $C0(14) = -0,030$; $C0(15) = 0,028$;
 $C0(16) = -0,004$; $C0(17) = 0,017$; $C0(18) = -0,010$; $C0(19) = 0,000$;
 $C0(20) = -0,001$;

Если продолжить расчёт с выше указанными коэффициентами, то это приведёт к медленному схождению цикла, то есть следующее через 21 итерацию $MSD = 2,69$ и решение будет сходиться всё медленнее к желаемому CKO или вовсе расходиться. Поэтому, чтобы «расшевелить» решение можно использовать

грубое округление, полученных значений, до одного знака после запятой. Тогда СКО откатится до значения 6,73, но посмотрим следующий на результат.

3) $Iter = 21$; $MSD = 1,95$; $C0(1) = -0,302$; $C0(2) = 0,330$; $C0(3) = 0,075$;
 $C0(4) = -0,270$; $C0(5) = 0,022$; $C0(6) = 0,019$; $C0(7) = 0,349$;
 $C0(8) = -0,236$; $C0(9) = 0,037$; $C0(10) = -0,007$; $C0(11) = 0,031$;
 $C0(12) = -0,020$; $C0(13) = 0,006$; $C0(14) = 0,018$; $C0(15) = -0,001$;
 $C0(16) = 0,005$; $C0(17) = -0,002$; $C0(18) = -0,001$; $C0(19) = 0,006$;
 $C0(20) = 0,003$;

4) $Iter = 16$; $MSD = 0,72$; $C0(1) = -0,391$; $C0(2) = 0,392$; $C0(3) = 0,007$;
 $C0(4) = -0,207$; $C0(5) = 0,001$; $C0(6) = 0,001$; $C0(7) = 0,300$;
 $C0(8) = -0,201$; $C0(9) = 0,002$; $C0(10) = -0,001$; $C0(11) = 0,001$;
 $C0(12) = 0,001$; $C0(13) = 0,000$; $C0(14) = 0,002$; $C0(15) = -0,002$;
 $C0(16) = 0,000$; $C0(17) = -0,001$; $C0(18) = 0,001$; $C0(19) = 0,000$;
 $C0(20) = 0,000$;

Как видно достигнуто желаемое СКО меньше единицы и полученные коэффициенты очень близки к заданным. Измени порог СКО до нуля и запустим функцию.

5) $Iter = 21$; $MSD = 0,51$; $C0(1) = -0,399$; $C0(2) = 0,400$; $C0(3) = 0,001$;
 $C0(4) = -0,201$; $C0(5) = -0,001$; $C0(6) = 0,000$; $C0(7) = 0,300$;
 $C0(8) = -0,200$; $C0(9) = 0,000$; $C0(10) = 0,000$; $C0(11) = 0,000$;
 $C0(12) = 0,000$; $C0(13) = 0,001$; $C0(14) = 0,000$; $C0(15) = 0,000$;
 $C0(16) = 0,001$; $C0(17) = -0,001$; $C0(18) = 0,000$; $C0(19) = 0,000$;
 $C0(20) = 0,000$;

Смысла продолжать расчёт нет, так как этот процесс будет очень медленно сходиться к определённому значению СКО не равному нулю.

В результате погрешность вычисления коэффициентов не превысила по модулю 0,001.

Так как искривление волнового фронта влияет на собственные значения матрицы Фишера, то произведём их расчёт снова, но уже с полученными коэффициентами.

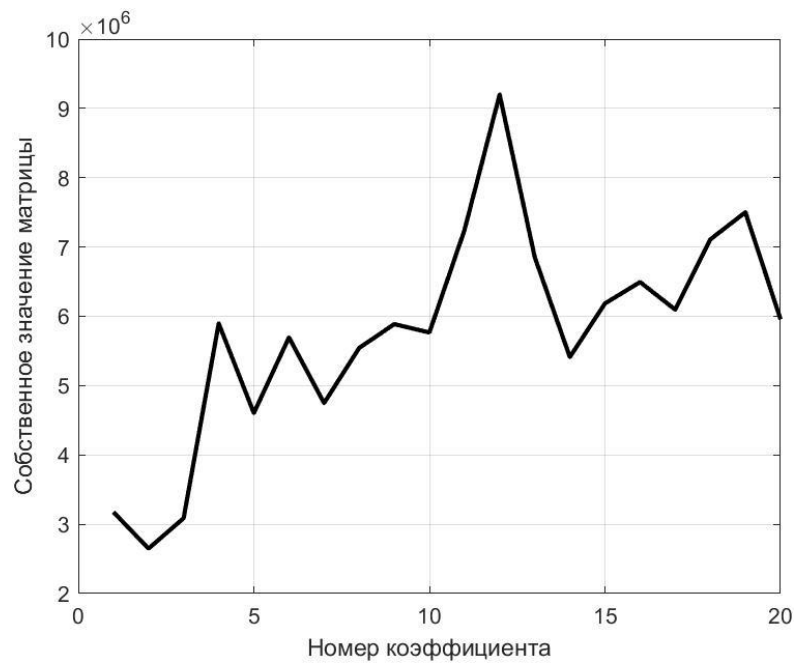


Рисунок 9 – Собственные значения матрицы Фишера для искажённого волнового фронта

Погрешность не превысила $6,14 \cdot 10^{-4}$. Видно, что аберрации слабо влияют на минимальную погрешность коэффициентов.

Отобразим стартовое изображение, то есть распределение интенсивности при всех нулевых коэффициентах.

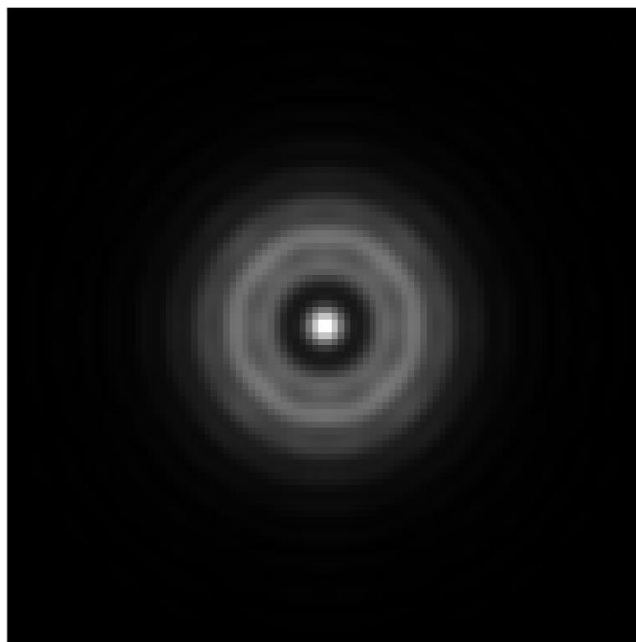


Рисунок 10 – Стартовое для расчёта распределение

На рисунке 10 заметно 8-ми угольная форма колец. Это связано с небольшой дискретизацией выходного зрачка ($M = 50$). Если увеличить дискретизацию, например, до 100, то подобных искажений, вызванных погрешностью расчётов, можно избежать. Но увеличение дискретизации зрачка или размера изображения приводит к квадратичному росту времени расчёта.

На рисунке 11 показано изображение, построенное с помощью полученных коэффициентов. Если сравнить его с рисунком 6, то визуальных отличий не наблюдается.

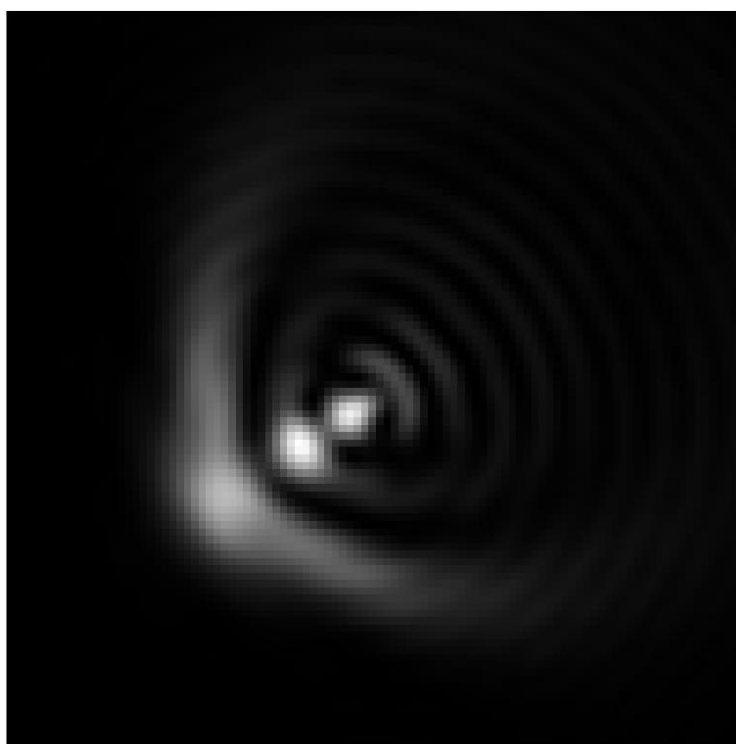


Рисунок 11 – Изображение, полученное в результате расчёта

Для того, чтобы визуализировать отличия двух изображений вычтем из реального рассчитанное и возьмём модуль. Тогда максимальное отклонение в пикселе равно 6. Отобразим эту разность, где максимум равен 6.

Подобная разность, вероятно, вызвана небольшим отклонением дисторсии и расфокусировки.

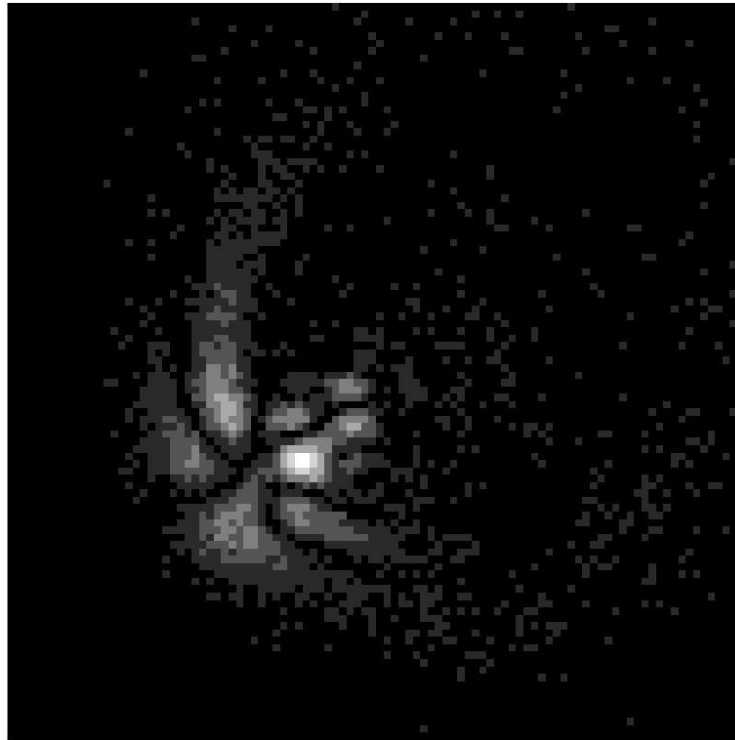


Рисунок 12 – Матрица разности двух изображений

Для формирования второго синтезированного изображения используем те же параметры что и выше, кроме $z = 6005$ мм. То есть расфокусировка $\delta z = z - F = 5$ мм. Диапазон интенсивности от 0 до 240, так как в реальных условиях не всегда получается стабилизировать и удерживать максимум на 255. Шум зададим в виде матрицы с помощью встроенной команды *rand*, которая распределяет значения по равномерному закону, и округлим до целых значений. Тогда среднее значение шума равно $1,0006 \approx 1$, а максимум равен 2. Зададим ненулевые коэффициенты Цернике: $C0_1 = 0,2$; $C0_2 = -0,1$; $C0_3 = -0,1$; $C0_4 = -0,1$; $C0_5 = 0,1$; $C0_7 = -0,1$; $C0_8 = 0,15$; $C0_{12} = 0,15$; $C0_{23} = -0,05$. В результате получим изображение

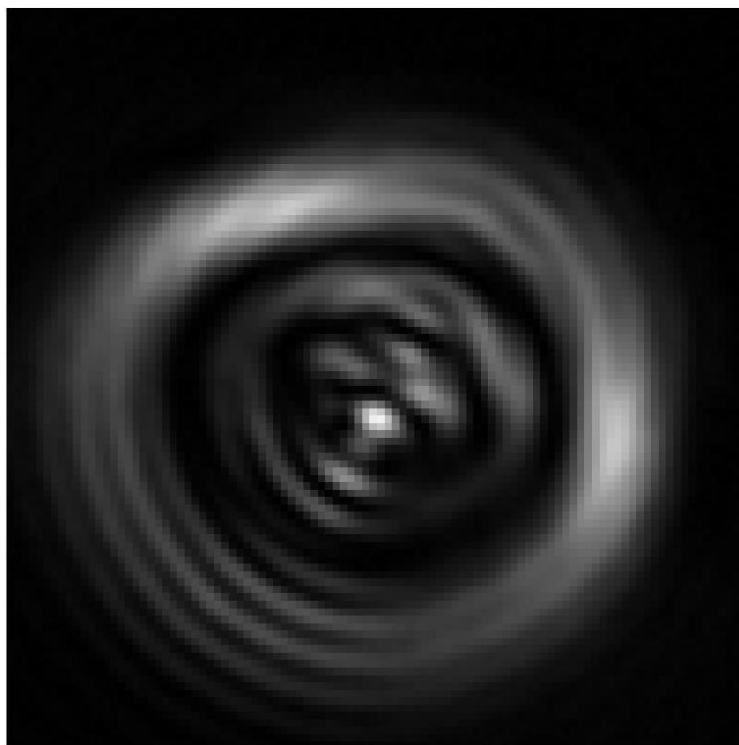


Рисунок 13 – Второе синтезированное изображение

При этом параметр $g = 4,22$ и $A = -2,47$. Максимум на изображении равен 242.

Стартовые коэффициенты выберем нулевые и сформируем новую матрицу шума. Тогда среднее значение шума равно $0,9975 \approx 1$, а максимум равен 2. Это сделано для того, чтобы уменьшить СКО двух изображений в случае совпадения коэффициентов Цернике. Тогда минимально возможный СКО равен $1,0064 \approx 1$. Диапазон интенсивностей для расчёта выберем от 0 до 241, так как в реальном случае невозможно гарантировать, что в максимуме интенсивности максимальное значение шума, поэтому будем считать, что в максимуме среднее значение. Из-за этого минимально возможный СКО будет больше единицы.

Рассчитаем собственные значения матрицы Фишера до 50-ого коэффициента и выберем только первые 30, тогда погрешность не превысит $7,60 \cdot 10^{-4}$.

Параметры обратной задачи остаются те же, кроме $MSD = 3$. Начальное значение СКО равно 37,27.

Ниже представлено пошаговое описание о результатах обратной задачи.

1) $Iter = 21$; $MSD = 4,94$; $C_0(1) = 0,177$; $C_0(2) = -0,107$; $C_0(3) = -0,098$;
 $C_0(4) = -0,108$; $C_0(5) = 0,100$; $C_0(6) = 0,005$; $C_0(7) = -0,083$;
 $C_0(8) = 0,142$; $C_0(9) = 0,003$; $C_0(10) = 0,001$; $C_0(11) = -0,002$;
 $C_0(12) = 0,152$; $C_0(13) = 0,002$; $C_0(14) = 0,006$; $C_0(15) = -0,001$;
 $C_0(16) = 0,000$; $C_0(17) = 0,000$; $C_0(18) = -0,004$; $C_0(19) = 0,002$;
 $C_0(20) = 0,001$; $C_0(21) = 0,000$; $C_0(22) = 0,001$; $C_0(23) = -0,058$;
 $C_0(24) = 0,002$; $C_0(25) = -0,003$; $C_0(26) = 0,000$; $C_0(27) = 0,002$;
 $C_0(28) = 0,001$; $C_0(29) = 0,000$; $C_0(30) = 0,002$;

2) $Iter = 2$; $MSD = 2,95$; $C_0(1) = 0,176$; $C_0(2) = -0,101$; $C_0(3) = -0,099$;
 $C_0(4) = -0,106$; $C_0(5) = 0,101$; $C_0(6) = -0,001$; $C_0(7) = -0,086$;
 $C_0(8) = 0,149$; $C_0(9) = 0,001$; $C_0(10) = -0,001$; $C_0(11) = 0,001$;
 $C_0(12) = 0,149$; $C_0(13) = 0,002$; $C_0(14) = 0,000$; $C_0(15) = 0,000$;
 $C_0(16) = 0,000$; $C_0(17) = 0,001$; $C_0(18) = -0,001$; $C_0(19) = 0,004$;
 $C_0(20) = -0,001$; $C_0(21) = 0,001$; $C_0(22) = 0,000$; $C_0(23) = -0,052$;
 $C_0(24) = 0,002$; $C_0(25) = 0,000$; $C_0(26) = 0,000$; $C_0(27) = 0,000$;
 $C_0(28) = 0,000$; $C_0(29) = 0,000$; $C_0(30) = 0,000$;

Уменьшим MSD до нуля и попробуем добиться большей точности.

3) $Iter = 21$; $MSD = 1,18$; $C_0(1) = 0,200$; $C_0(2) = -0,101$; $C_0(3) = -0,100$;
 $C_0(4) = -0,100$; $C_0(5) = 0,100$; $C_0(6) = 0,000$; $C_0(7) = -0,100$;
 $C_0(8) = 0,150$; $C_0(9) = 0,000$; $C_0(10) = 0,000$; $C_0(11) = 0,000$;
 $C_0(12) = 0,150$; $C_0(13) = 0,000$; $C_0(14) = 0,000$; $C_0(15) = 0,000$;
 $C_0(16) = 0,000$; $C_0(17) = 0,000$; $C_0(18) = 0,000$; $C_0(19) = 0,000$;
 $C_0(20) = 0,001$; $C_0(21) = 0,000$; $C_0(22) = 0,000$; $C_0(23) = -0,050$;
 $C_0(24) = 0,000$; $C_0(25) = 0,000$; $C_0(26) = 0,000$; $C_0(27) = 0,000$;
 $C_0(28) = 0,000$; $C_0(29) = 0,000$; $C_0(30) = 0,000$;

В результате погрешность вычисления коэффициентов не превысила по модулю 0,001, а минимальная погрешность стала равна $4,41 \cdot 10^{-4}$.

Стартовое распределение показано на рисунке 14.

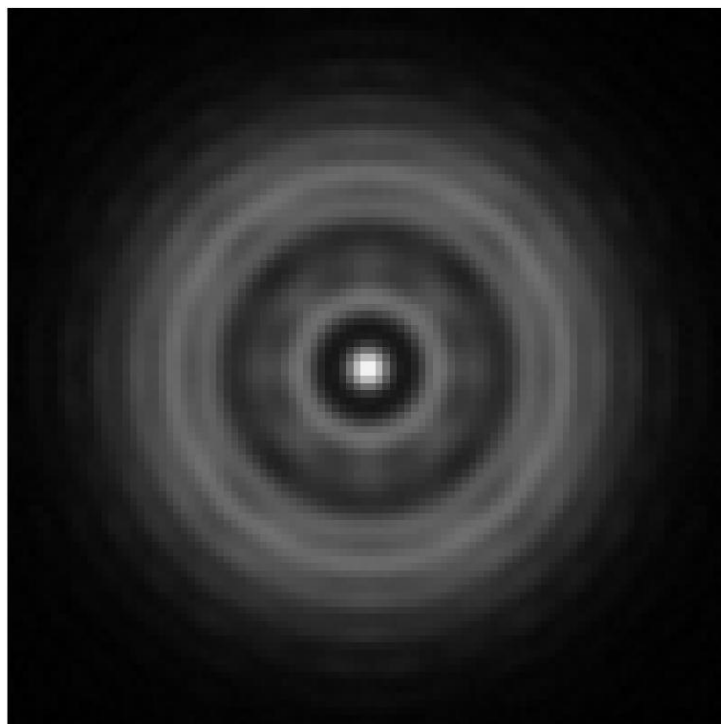


Рисунок 14 – Стартовое для расчёта распределение

Распределение, полученное в результате расчёта.

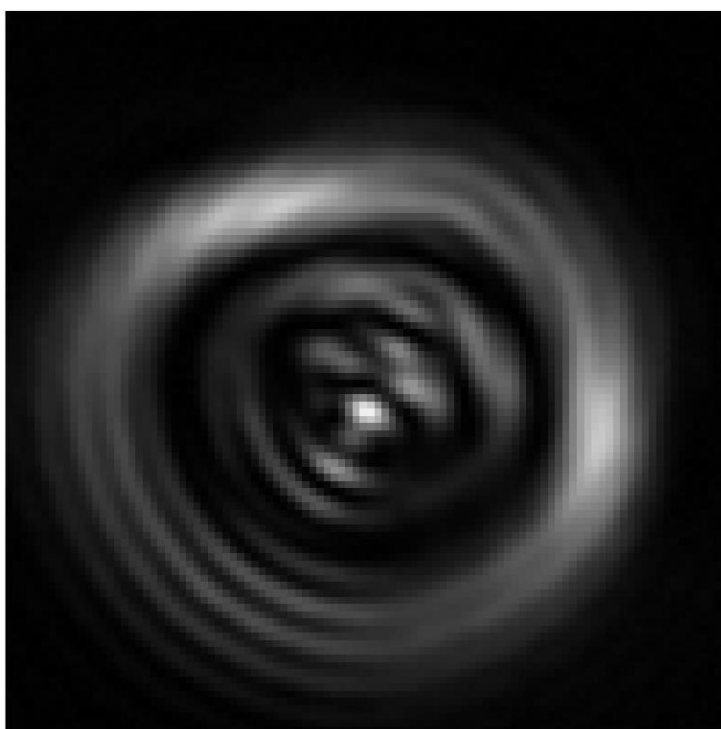


Рисунок 15 – Изображение, полученное в результате расчёта

Максимальное отклонение в пикселе равно 5. Отобразим эту разность, где максимум равен 5.

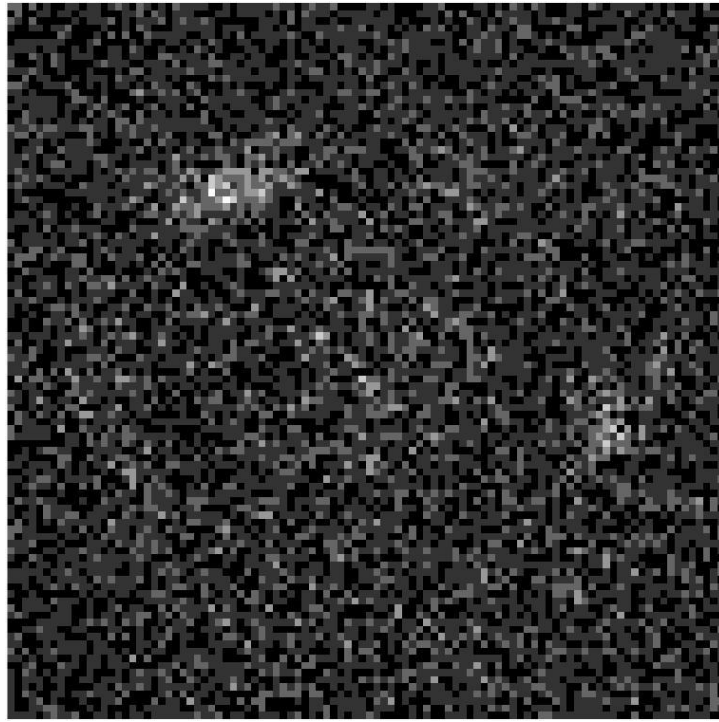


Рисунок 16 – Матрица разности двух изображений

Вид матрицы разности обусловлен тем, что на двух изображениях разные значения шума (*Noise*), хотя обе матрицы шума распределены по равномерному закону.

Для формирования третьего синтезированного изображения используем те же параметры что и выше, кроме $z = 6004$ мм. То есть расфокусировка $\delta z = z - F = 4$ мм. Диапазон интенсивности от 0 до 255, шум равен нулю. Зададим не нулевые коэффициенты Цернике: $C0_3 = -0,08$; $C0_8 = 0,13$. В результате получим изображение, показанное на рисунке 17.

При этом параметр $g = 4,22$ и $A = -1,98$.

Рассчитаем собственные значения матрицы Фишера до 50-ого коэффициента и выберем только первые 10, тогда погрешность равна $6,00 \cdot 10^{-4}$.

Стартовые значения коэффициентов выберем нулевые. Параметры обратной задачи остаются те же, что для первого синтезированного изображения. Начальное значение СКО равно 58,31.

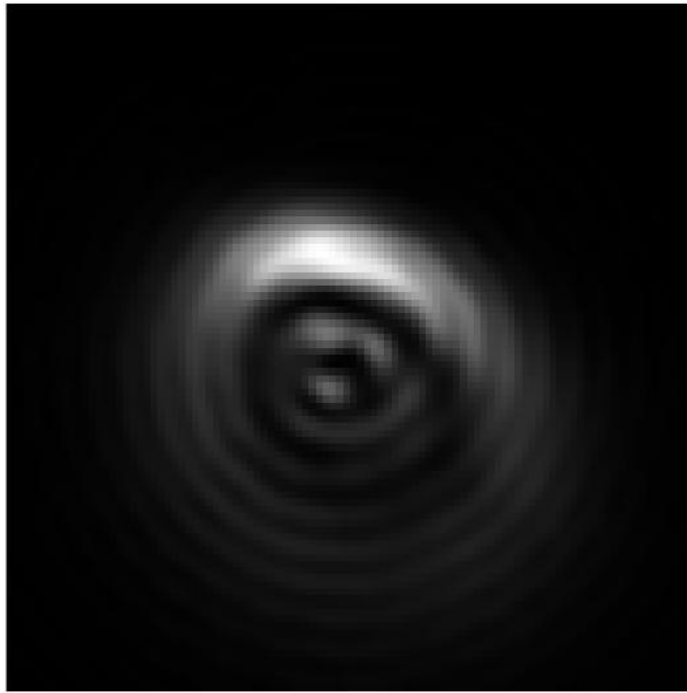


Рисунок 17 – Третье синтезированное изображение

Ниже представлено пошаговое описание о результатах обратной задачи.

1) $Iters = 6$; $MSD = 0,94$; $C0(1) = -0,006$; $C0(2) = 0,006$; $C0(3) = -0,072$;
 $C0(4) = -0,001$; $C0(5) = -0,005$; $C0(6) = -0,005$; $C0(7) = -0,001$;
 $C0(8) = 0,130$; $C0(9) = 0,002$; $C0(10) = 0,002$;

Уменьшим MSD до нуля и попробуем добиться большей точности.

2) $Iters = 21$; $MSD = 0,30$; $C0(1) = 0,000$; $C0(2) = 0,000$; $C0(3) = -0,080$;
 $C0(4) = 0,000$; $C0(5) = 0,000$; $C0(6) = 0,000$; $C0(7) = 0,000$;
 $C0(8) = 0,130$; $C0(9) = 0,000$; $C0(10) = 0,001$;

В результате погрешность вычисления коэффициентов не превысила по модулю 0,001, а минимальная погрешность стала равна $9,43 \cdot 10^{-4}$.

Стартовое распределение показано на рисунке 18.

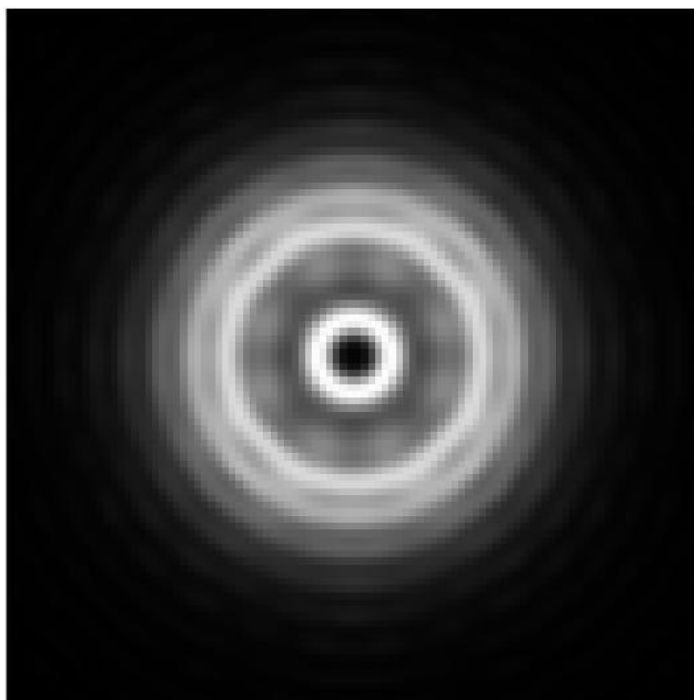


Рисунок 18 – Стартовое для расчёта распределение

Распределение, полученное в результате расчёта.

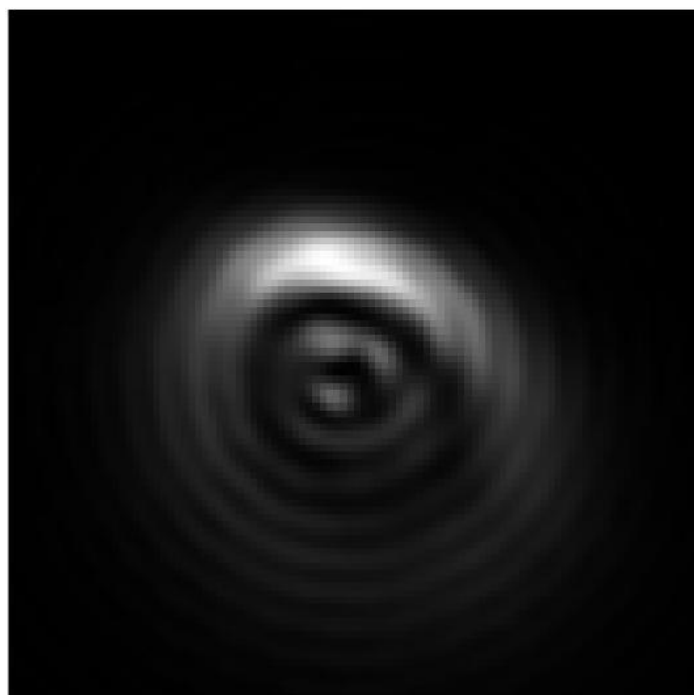


Рисунок 19 – Изображение, полученное в результате расчёта

Максимальное отклонение в пикселе равно 2. Отобразим эту разность, где максимум равен 2. Ясно что эти отклонения из-за 10-го коэффициента.

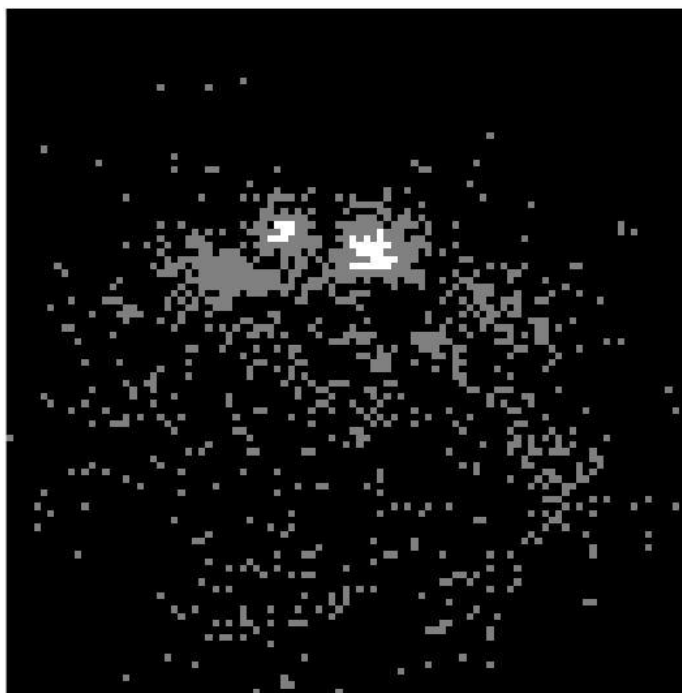


Рисунок 20 – Матрица разности двух изображений

По результатам выше описанных теории и проверок функции можно сделать некоторые выводы. Первое, как выглядит функция правдоподобия, которая минимизируется, неизвестно и её форма может сильно меняться от коэффициентов Цернике. Так как в примерах приведены сходящиеся решения при нулевых стартовых коэффициентах, то может показаться, что функция правдоподобия является выпуклой, но это совершенно не так. Она имеет множество локальных минимумов, помимо глобального. Второе, предложенная программа отлично работает при величинах аббераций, не превышающих по модулю $0,2 - 0,3$, при нулевых стартовых значениях. Из-за использованной нормировки полиномов малые значения коэффициентов высоких порядков вносят значительные искажения, поэтому значения в $0,3$ в реальных распределениях маловероятны. Третье, в предложенном решении есть критерий глобальности найденного минимума – это СКО. Зная максимальное значение шума, можно определить при каком СКО итерационный процесс окажется в области глобального минимума, то есть будет близок нему. Четвёртое, если процесс не сходится при выборе нулевых начальных значениях, то есть попадет в локальный минимум, то необходимо задать каким-либо образом приближение,

например, провести расчёт идеальной оптической системы и полученные коэффициенты использовать как стартовые значения для функции.

Есть замечание по поводу 4-го коэффициента Цернике – расфокусировка. В примерах выше коэффициент A задавался теоретически без погрешности, но в реальных ситуациях, есть погрешность определения фокусного расстояния F и расстояния z , следовательно, и δz . В такой постановке задачи возможно только определение суммы коэффициентов A и CO_4 . Поэтому если погрешность определения A значительно выше погрешности итерационной процедуры, то CO_4 будет далёк от реального значения, но их сумма будет близка к действительности.

2.4. Стенд для получения изображений

Для снятия изображения точечного источника (ФРТ) в данной работе использован лабораторный стенд, показанный на рисунке 21.

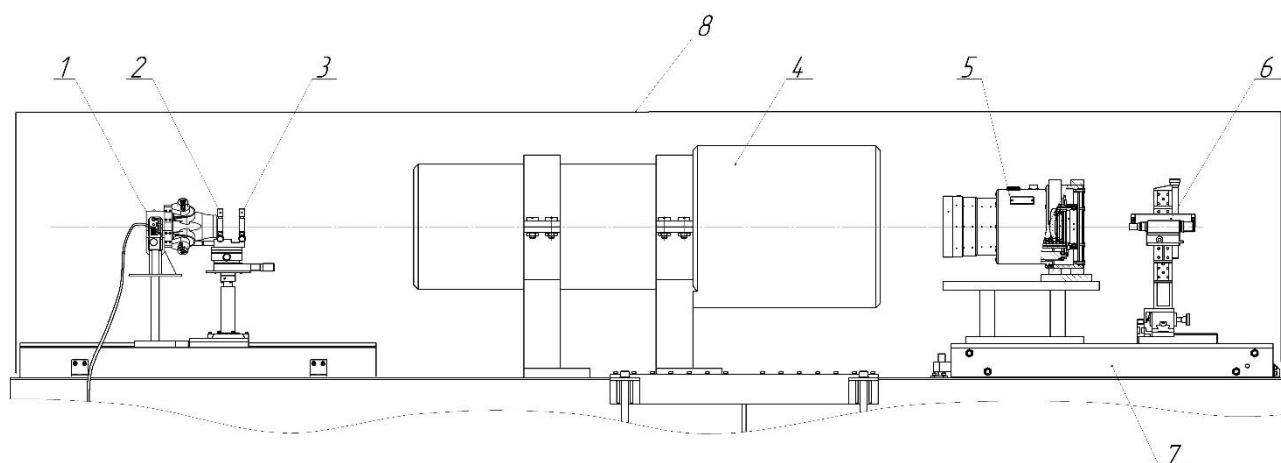


Рисунок 21 – Схема лабораторного стенда: 1 – источник излучения;
2 – конденсор; 3 – тест-объект; 4 – объектив коллиматора «Телегоир-12»;
5 – исследуемый объектив; 6 – система регистрации изображения;
7 – поворотный узел; 8 – светозащитный чехол.

В качестве источника 1 используется оптико-электронный узел с диагональным зеркалом, позволяющим выводить на оптическую ось системы один из четырех источников излучения – светодиодов. Тем самым он позволяет без дополнительной переюстировки производить контроль качества оптических

систем на разных длинах волн с использованием узкоспектральных светодиодов видимого и ближнего инфракрасного диапазонов. Свет источника, отраженный диагональным зеркалом под 90° , выводится на оптическую ось системы и равномерно засвечивает апертуру конденсора 2, представляющего собой 20-кратный микрообъектив, который в свою очередь направляет сигнал на тест-объект 3. В роли тест-объекта используется точечная диафрагма. На тест-объекте свет дифрагирует, образуя дифракционную картину Эйри, которая впоследствии попадает на коллиматор 4.

Далее сколлимированное излучение попадает на испытуемый объектив 5, после которого происходит регистрация кадра с помощью системы регистрации 6. В системе регистрации 6 используется микроскоп измерительный АЛ5.902.020, который позволяет увеличивать получаемое изображение до 20-ти крат с последующей регистрацией на ПЗС-матрице. Тестируемый объектив и система регистрации расположены на едином поворотном основании 7, что позволяет производить контроль качества изображений объектива не только по центру поля (на оси), но и на зонах (до $\pm 25^\circ$). Изображение с системы регистрации передается на компьютер, где происходит дальнейшая обработка информации. Все оборудование располагается на оптических столах с пневмоопорами, позволяющими полностью исключить влияние вибрации на регистрацию изображений.

В данном стенде моделируется точечный источник при помощи тест-объекта и коллиматора, что ведёт к появлению дополнительных искажений волнового фронта, помимо тех, что привносит тестируемый объектив. Поэтому на выходе получается ФРТ всей системы, а не тестируемого объектива.

Будем считать, что аберрации других оптических элементов (коллиматора и увеличительная система микроскопа) пренебрежимо малы по сравнению с тестируемой оптикой.

Для более точного расчёта необходимо учитывать погрешности юстировки системы, то есть соосность элементов стенда, которая в первую очередь влияет

на точность определения aberrации дисторсия. Но в целях упрощения этот момент опущен.

Стоит учесть шумы ПЗС-матрицы и шум фона. Совокупность этих шумов оценивается по значениям интенсивности в пикселях матрицы при отсутствии источника излучения. Фотонным шумом пренебрегаем.

2.5. Применение алгоритма к полученным изображениям и оценка качества оптической системы

Объектом исследования является объектив «Сова-5», оптическая схема которого представлена на рисунке 22, являющийся полностью линзовым с 13 оптическими элементами. Также рисунке представлены ход лучей по краю входного зрачка, диаметр которого равен 49,93 мм. Диаметр выходного зрачка равен $Dp = 65,27$ мм, фокусное расстояние равно $F = 99,86$ мм.

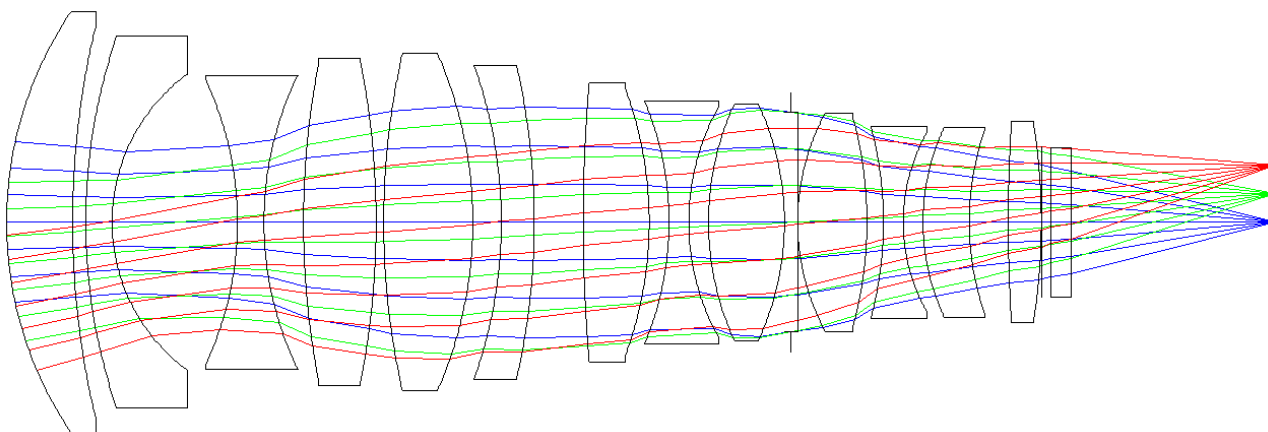


Рисунок 22 – Оптическая схема объектива «Сова-5» с лучами по полю 5° и 10°

Длина волны излучения источника равна $\lambda = 630$ нм, размер пикселя ПЗС-матрицы равен 3,45 мкм, увеличение приёмной системы равно 20 крат, тогда с учётом увеличения размер пикселя для расчёта равен $\text{step} = 3,45/20$ мкм = 0,1725 мкм. Съёмка изображения проводилась в фокусе, то есть $\delta z = z - F = 0$ мм, так как данная оптическая система чувствительна к расфокусировке, при этом погрешность определения плоскости z равна 0,01 мм. Тогда параметр $A = 0$, погрешность $\Delta A = 0,85$. Такая высокая погрешность обусловлена малым отношением фокусного расстояния к выходному зрачку $F/Dp \approx 2$. Параметр $g = 5,60$.

Со стенда сняты изображения под углом 5° и 10° , представленные на рисунке 23. Изображение на оптической оси (0°) практически без абберрационное, поэтому не представляет особого интереса.

Максимум суммарного шума на ПЗС-матрице составил в отсчётах 2. Размер стороны изображения $N = 70$ пикселей.

Для получения идеальных изображений и расчёта стартовых коэффициентов использовалась программа *Zemax*. Теоретические изображения представлены на рисунке 24 и в таблице 3 записаны коэффициенты, соответствующие им, также стоит учесть, что в *Zemax* используется другая нумерация коэффициентов, поэтому их необходимо перенумеровать для сравнения с расчётными.

Расчёт матрицы Фишера до 60-го коэффициента для 5° показал, что возможно восстановить все 60 коэффициентов. Выберем все 60 коэффициентов, тогда погрешность не превысит $9,36 \cdot 10^{-4}$.

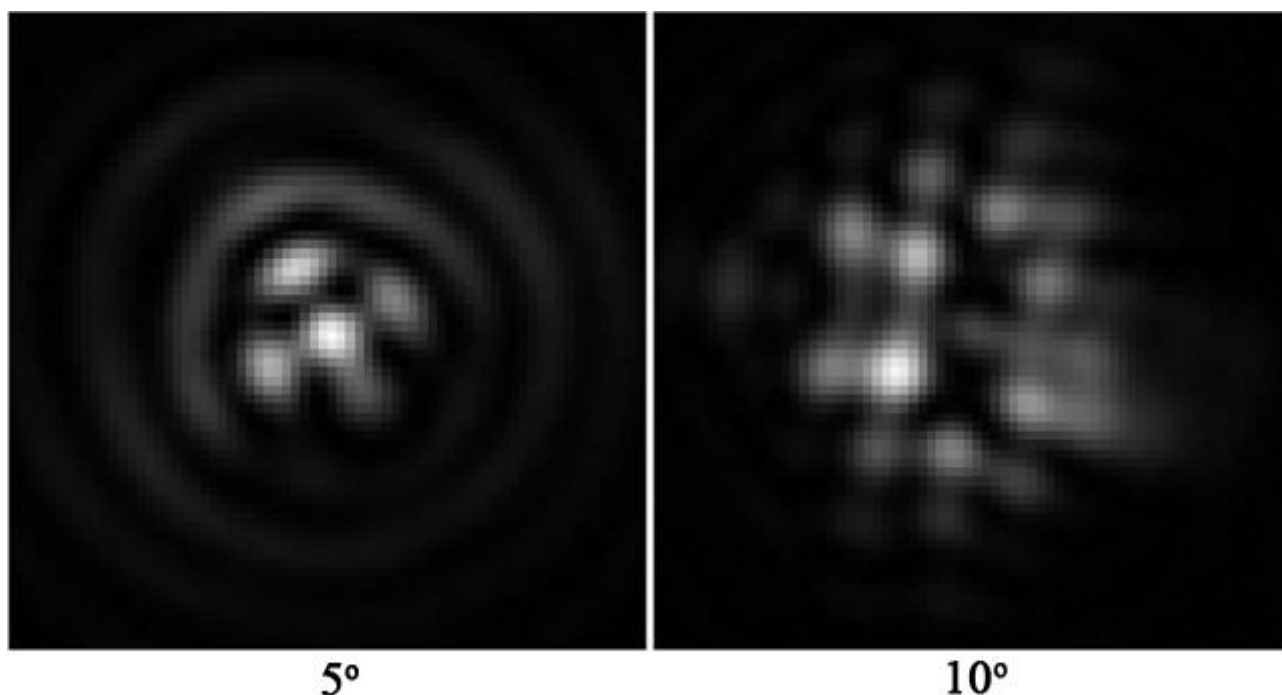


Рисунок 23 – Полученные со стенда изображения

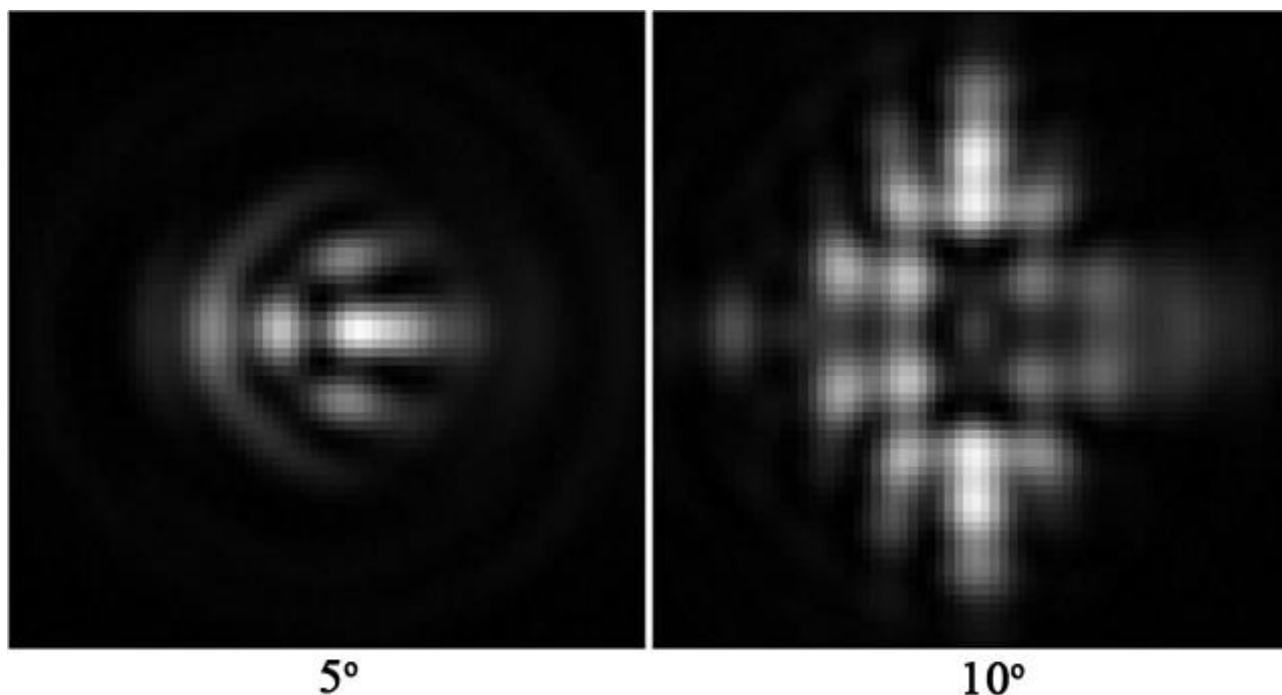


Рисунок 24 – Теоретические изображения

Расчёт матрицы Фишера до 60-го коэффициента для 10° показал, что возможно восстановить все 60 коэффициентов. Выберем все 60 коэффициентов, тогда погрешность не превысит $6,62 \cdot 10^{-4}$.

В результате расчёта полученные коэффициенты, соответствующие реальным изображениям, представлены в таблице 3. СКО для 5° составил 1,09 и для 10° составил 1,17.

Таблица 3 – Теоретические и рассчитанные коэффициенты Цернике

№	Теоретические коэффициенты		Рассчитанные коэффициенты	
	5°	10°	5°	10°
1	0,1066	0	-0,0004	0,0005
2	0	0	-0,0008	-0,0003
3	0	0	0,0987	-0,1297
4	-0,2272	0,0307	-0,1833	-0,0221
5	0,1160	0,3725	0,0540	0,3015
6	0,0012	0,0137	0,0016	0,0134
7	-0,0334	-0,0072	-0,0110	-0,0095
8	0	0	-0,0467	-0,0108
9	0	0	0,0017	0,0110

Продолжение таблицы 3

№	Теоретические коэффициенты		Рассчитанные коэффициенты	
	5°	10°	5°	10°
10	0	0	-0,0004	0
11	0	0	-0,0331	0,0610
12	0,1215	0,0954	0,2336	0,0331
13	-0,0307	-0,0761	-0,0355	-0,0900
14	0	-0,0203	0,0004	-0,0189
15	0	0	0,0004	0,0004
16	0,0007	-0,0063	-0,0002	-0,0072
17	-0,0279	-0,0477	-0,0363	-0,0506
18	0	0	-0,0212	0,0013
19	0	0	0,0008	0,0006
20	0	0	-0,0005	0,0005
21	0	0	0,0006	-0,0006
22	0	0	0	-0,0007
23	0	0	-0,0053	0,0006
24	-0,0747	-0,0647	-0,0542	-0,0584
25	-0,0058	-0,0117	-0,0068	-0,0127
26	0	0	0,0002	0,0003
27	0	0	0,0004	0,0008
28	0	0	0,0001	-0,0003
29	0	0	0,0002	0,0007
30	0,0001	0	0,0007	-0,0003
31	0,0264	0,0139	0,0207	0,0127
32	0	0	0,0224	0,0006
33	0	0	-0,0004	0,0001
34	0	0	-0,0002	0,0001
35	0	0	0,0011	0,0008
36	0	0	0,0003	0,0003
37	0	0	0,0005	-0,0008
38	0	0	0,0003	0
39	0	0	-0,0014	0,0007
40	0,0398	0,0226	0,0301	0,0208

Продолжение таблицы 3

№	Теоретические коэффициенты		Рассчитанные коэффициенты	
	5°	10°	5°	10°
41	-0,0007	0,0138	-0,0009	0,0148
42	0	0	0,0004	0,0002
43	0	0	0,0002	0,0005
44	0	0	-0,0001	-0,0001
45	0	0	0,0004	0
46	0	0	-0,0003	0,0002
47	0	0	-0,0002	0,0002
48	0	0	-0,0001	0,0085
49	0,0060	0,0024	0,0062	0,0035
50	0	0	0,0037	-0,0003
51	0	0	-0,0001	0,0006
52	0	0	-0,0006	0,0111
53	0	0	0,0009	-0,0001
54	0	0	-0,0004	0,0001
55	0	0	0,0003	-0,0001
56	0	0	0,0004	0,0004
57	0	0	0,0003	-0,0013
58	0	0,0040	0,0004	-0,0008
59	0	0	-0,0001	0,0038
60	0,0112	0,0060	0,0101	0,0084

Рассмотрение коэффициента 4 (расфокусировка) смысла не имеет, так как погрешность определения плоскости z велика. Точность определения дисторсии (коэффициентов 1 и 2) зависит от погрешности юстировки элементов стенда по оптической оси, так как дисторсия при изображении точечного источника только смещает картину по плоскости изображения.

Из рисунков 23 и 24 видно, что структура распределения в целом схожа. Основные отличия для 5° заключаются в аберрациях сферической, кома и астигматизм. Для 10° существенны отклонения аберраций сферической, астигматизм и аберрации под номером 11. Такие различия в расчётных и

реальных изображениях вызваны, во-первых, качеством сборки объектива (соосность, пережатие и угловые отклонения оптических элементов), и, во-вторых, качеством изготовления этих элементов (однородность, двулучепреломление и т.д.). Отличие коэффициентов в данном случае, вероятно, вызваны погрешностью расстояний между линзами и пережатием некоторых линз.

Для того, чтобы получить больше информации о системе можно снять наборы картин по полю зрения с малым шагом и построить графики зависимостей реальных аберраций от угла для последующего сравнения с теоретическими.

На данный момент связь величин аберраций и взаиморасположения оптических элементов является малоизученной, потому что при большом количестве элементов эта связь очень сложна из-за быстрорастущего количества степеней свободы всей системы, и неоднозначна, так как существуют вырожденные случаи.

ЗАКЛЮЧЕНИЕ

В данной работе произведён обзор актуальных способов оценки качества оптических систем: измерение функции рассеяния, измерение частотной контрастной характеристики, интерференционный и по распределению интенсивности в плоскости изображения. Изучению последнего посвящена эта работа. Приведены особенности решения прямой и обратной задачи и оценка погрешности решения. Проведена проверка реализованного алгоритма как на синтезированных изображениях, так и на реальных.

Особенности применимости алгоритма описаны в третьей главе практической части. По полученным результатам можно заключить, что данный метод возможно успешно применять для получения информации о качестве оптических систем. Сложности могут возникнуть, если съёмка производилась в фокусе оптической системы, мало отношение сигнал/шум или приёмник излучения имеет малый динамический диапазон.

Для поиска более устойчивого и быстрого решения можно попробовать другие методы многомерной и одномерной минимизации функции такие, как: Нелдера – Мида, Пауэлла, сопряжённых градиентов, Пирсона, Фибоначчи, Ньютона и другие. Интересными в этом плане являются алгоритмы стохастические, например, Монте – Карло, или эволюционные, например, генетические алгоритмы.

СПИСОК ИСТОЧНИКОВ

1. Fernandez-Martinez J.L., Fernandez-Muniz Z. The curse of dimensionality in inverse problems // J. Comput. Appl. Math. 2019. Vol. 369.
2. Anastasio M.A., Schoonover R.W. Basic Principles of Inverse Problems for Optical Scientists // Opt. Encycl. 2016. P. 1–24.
3. Aster R.C., Borchers B., Thurber C.H. Nonlinear Inverse Problems // Param. Estim. Inverse Probl. 2019. Vol. 8. P. 257–278.
4. Isakov V. Inverse problems // Appl. Math. Sci. 2017. Vol. 127. P. 1–22.
5. Теребиж В.Ю. Введение в статистическую теорию обратных задач. – М.: ФИЗМАТЛИТ, 2005. С. 226–249.
6. Richter M., Reiss K., Korn R. Inverse Probleme. Springer Spektrum, 2015. P. 128.
7. Liu R., Zhao S., Zhang P., Gao H., Li F. Complex wavefront reconstruction with single-pixel detector // Appl. Phys. Lett. 2019. Vol. 114, № 16.
8. Bonnefond S., Tallon M., Louarn M.L., Madec P.Y. Wavefront reconstruction with pupil fragmentation: study of a simple case // Adapt. Opt. Syst. V. 2016. Vol. 9909 990972. P. 1–6.
9. Arnal R.D., Millane R.P. The phase problem with structured sampling // Int. Conf. Image Vis. Comput. New Zealand. 2016.
10. Arnal R.D., Millane R.P. The phase problem for two-dimensional crystals. I. Theory // Acta Crystallogr. Sect. A Found. Adv. 2017. Vol. 73. P. 438–448.
11. Arnal R.D. et al. The phase problem for two-dimensional crystals.II. Simulations // Acta Crystallogr. Sect. A Found. Adv. 2018. Vol. 74. P. 537–544.
12. Millane R.P. The phase problem for one-dimensional crystals // Acta Crystallogr. Sect. A Found. Adv. 2017. Vol. 73. P. 140–150.
13. Tajiri H. Progress in surface X-ray crystallography and the phase problem // Jpn. J. Appl. Phys. 2019. P. 1–40.
14. Situ G., Wang H. Phase problems in optical imaging // Front. Inf. Technol. Electron. Eng. 2017. Vol. 18, № 9. P. 1277–1287.
15. Казанский Н.Л., Харитонов С.И., Козлова И.Н., Моисеев М.А. СВЯЗЬ

- фазовой проблемы в оптике, фокусировки излучения и задачи Монжа–Кантровича // Компьютерная оптика. 2018. Т. 42, № 4. С. 574–587.
16. Liang P. et al. Fourier modal method for two-dimensional wavefront reconstruction // Second Int. Conf. Photonics Opt. Eng. 2017. Vol. 10256 1025642. P. 1–11.
 17. Tani L. Monitoring the optical quality of the FACT Cherenkov Telescope // Master's Thesis. 2019.
 18. Jia P. et al. PSF-NET: A Non-parametric Point Spread Function Model for Ground Based Optical Telescopes. 2020.
 19. Beltramo-Martin O. et al. Pushing Point spread function reconstruction to the next level. Application to SPHERE/ZIMPOL // Mnras. 2019. Vol. 000. P. 1–15.
 20. Wang Y. et al. Defocusing parameter selection strategies based on PSF measurement for square-binary defocusing fringe projection profilometry // Opt. Express. 2018. Vol. 26, № 16. P. 20351-20367.
 21. Damian C., Coltuc D. Measurement of Non-Circular PSFs in Single Pixel Cameras // 2018 12th Int. Conf. Commun. COMM 2018 - Proc. IEEE. 2018. P. 121–124.
 22. Zheng Y. et al. Optimal PSF estimation for simple optical system using a wide-band sensor based on psf measurement // Sensors. 2018. Vol. 18, № 3552. P. 1–11.
 23. Wittpahl C. et al. Realistic image degradation with measured PSF // IS T Int. Symp. Electron. Imaging Sci. Technol. 2018.
 24. Сайт по прикладной оптике университета ИТМО – Режим доступа: http://aco.ifmo.ru/el_books/basics_optics/glava-9/glava-9-5-s.html, дата обращения 19.02.2020.
 25. Zhou C., Wang Z. Mid-frequency MTF compensation of optical sparse aperture system // Opt. Express. 2018. Vol. 26, № 6. P. 6973–6992.
 26. Serief C., Ghelamallah Y. Dynamic MTF estimate of the optical imager onboard Alsat-1B satellite // Int. Conf. Space Opt. 2018. Vol. 11180 111807S. P. 1–7.
 27. Burns P.D. et al. Application of ISO Standard Methods to Optical Design for

- Image Capture // Int. Symp. Electr. Imaging. 2020 Vol. 240. P. 1–6.
28. Koksai S., Canarslan I., Coskun O.D. Image Quality Characterization of Earth Observation Electro-optic Imagers through PSF and MTF Analysis // J. Chem. Inf. Model. 2019. Vol. 19. P. 429–434.
 29. Yu J., Hua F., Li D. MTF measurement of progressive addition lens // Optik – Int. J. Light Electron Opt. 2019. Vol. 193. P. 163000.
 30. Wei X., Han J., Wan X. Aberration correction based on MTF testing for aspheric optical system // Optik – Int. J. Light Electron Opt. 2019 Vol. 185. P. 1089–1095.
 31. Han S., Choi K., Yoo S.W. A subband-specific deconvolution model for MTF improvement in CT // J. Healthcare. Eng. 2017. Vol. 2017.
 32. Takanashi T., Kawamura H. Evaluation of Spatial Resolution of MRI, Optical CT and X-Ray CT Using MTF for Gel Dosimeter // Springer Nature Singapore Pte Ltd. 2019. P. 35–38.
 33. Haefner D.P. Best Practices for Imaging System MTF Measurement // Int. Symp. Electr. Imaging. 2019. Vol. 319. P. 1–6.
 34. Fang Y.C., Tsay H.L. A study of multi-angles knife-edge method applied to digital modulation transfer function measurement // Microsyst. Technol. Springer Berlin Heidelberg, 2019. Vol. 9.
 35. «Уран» ЗАО НПФ. Контроль качества изображения оптических систем на приборах серии ImageMaster производства Trioptics // «Уран». 2017. Vol. 8.
 36. Pokorny P. et al. Method for interferometric testing of optical surfaces based on evaluation of similarity of nominal and measured interferograms // Opt. Meas. Conf. 2014. Vol. 9442 944211. P. 1–6.
 37. Poleshchuk A.G., Nasyrov R.K., Asfour J.M. Interferometric testing of steep cylindrical surfaces with on-axis CGHS // Comput. Opt. 2016. Vol. 40, № 5. P. 625–628.
 38. Xue S. et al. Adaptive null interferometric test using spatial light modulator for free-form surfaces // Opt. Express. 2019. Vol. 27, № 6. P. 8414–8428.
 39. Zang Z. et al. Misalignment correction for free-form surface in non-null interferometric testing // Opt. Commun. Elsevier Ltd., 2019. Vol. 437.

P. 204–213.

40. Hibino K., Hanayama R., Kim Y. Absolute interferometric test for high numerical-aperture spherical concave surfaces: Gravitational effect // *Meas. J. Int. Meas. Confed. Elsevier Ltd*, 2016. Vol. 94. P. 920–926.
41. Колючкин В.Я., Одинокоев С.Б., Горевой А.В. Методика контроля внеосевых оптических систем интерференционным методом // *Общественная академия «Контенант» Электр. журн.* №3. 2015. С. 12–24.
42. Насыров Р.К., Полещук А.Г., Сокольский М.Н., Трегуб В.П. Интерферометрический метод контроля качества сборки оптической системы с эксцентрично расположенной асферической линзой // *Журн. Автометрия*. 2017. Т. 53, №5. С. 124–130.
43. Jie M., Xi H., Fan W. The effects of thermal field on radius of curvature interferometric testing // *Opt. Rev.* 2015. Vol. 22, № 2. P. 299–307.
44. Pariani G. et al. Optical calibration of the ELT adaptive mirror M4: Design, alignment and verification of the interferometric test tower // *Adapt. Opt. Extrem. Large Telesc.* 2017. Vol. 2017. P. 1–12.
45. Wen Y., Cheng H. Further investigations of stitching model for annular subaperture interferometric testing based on Zernike annular polynomials // *Optik. Elsevier GmbH*, 2015. Vol. 126, № 19. P. 2236–2241.
46. Wu X. et al. Dynamic stitching interferometric testing for large optical plane // *Model. Asp. Opt. Metrol. V.* 2015. Vol. 9526 952614. P. 1–4.
47. Wu X. et al. Non-overlap subaperture interferometric testing for large optics // *Opt. Commun.* 2017. Vol. 396, P. 191–198.
48. Wu X. et al. In-situ stitching interferometric test system for large plano optics // *Adv. Manuf. Shanghai University*, 2018. Vol. 6, № 2. P. 195–203.
49. Liu Y. et al. Extended shift-rotation method for absolute interferometric testing of a spherical surface with pixel-level spatial resolution // *Appl. Opt.* 2017. Vol. 56, № 16. P. 4886–4891.
50. Lai X. et al. Automatic Modal Wavefront Estimation and Correction for Optical Aberration // *IEEE Photonics J.* 2019. Vol. 11, № 1. P. 1–9.

51. Kirilenko M.S., Khorin P.A., Porfirev A.P. Avefront analysis based on Zernike polynomials // CEUR Workshop Proc. 2016. Vol. 1638. P. 66–75.
52. Wei L., Urbach H.P. Shaping the focal field of radially/azimuthally polarized phase vortex with Zernike polynomials // J. Opt. IOP Publishing, 2016. Vol. 18, № 6. P. 1–8.
53. Mochi I., Goldberg K.A. Modal wavefront reconstruction from its gradient // Appl. Opt. 2015. Vol. 54, № 12. P. 3780–3785.
54. Asli B.H.S., Flusser J., Zhao Y. 2-D Generating Function of the Zernike Polynomials and their Application for Image Classification // 2019 9th Int. Conf. Image Process. Theory, Tools Appl. IPTA 2019. IEEE, 2019. P. 1–6.
55. Asli B.H.S. et al. Filter-generating system of Zernike polynomials // Automatica. Elsevier Ltd, 2019. Vol. 108. P. 108498.
56. Velazquez J.L. et al. Zernike polynomials for photometric characterization of LEDs // J. Opt. IOP Publishing, 2016. Vol. 18, № 2. P. 25605.
57. Chen Z. et al. Membrane Reflector Surface Reconstruction Using Zernike Polynomials and Two-stage Algorithm // IFAC-PapersOnLine. Elsevier, 2015. Vol. 48, № 28. P. 632–637.
58. Cong B., Yi T.S. Simulation of surface deformation for the lithographic object lens by Zernike polynomials // Sel. Pap. from Conf. Photoelectron. Technol. Comm. Chinese Soc. Astronaut. 2014, Part II. 2015. Vol. 9522. P. 952206.
59. Ma X., Li T. Surface reconstruction of deformable reflectors by combining Zernike polynomials with radio holography // AIAA J. 2019. Vol. 57, № 6. P. 2544–2552.
60. Lyu H. et al. Absolute optical flatness testing by surface shape reconstruction using Zernike polynomials // Opt. Eng. 2018. Vol. 57, № 9. P. 094103.
61. Hang Z. et al. Design of Uniform Square Spot Lens Based on Smooth Optimization of Zernike Polynomials // Laser and Optoelectronics Progress. 2018. Vol. 55. P. 102202.
62. Ramos-López D. et al. Optimal sampling patterns for Zernike polynomials // Appl. Math. Comput. 2016. Vol. 274. P. 247–257.

63. Schilke M., Liebl J., Wunsche C. Surface reconstruction by using Zernike polynomials // Third Eur. Semin. Precis. Opt. Manuf. 2016. Vol. 10009. P. 1000910.
64. Yang T., Zhu J., Jin G. Nodal aberration properties of coaxial imaging systems using Zernike polynomial surfaces // J. Opt. Soc. Am. A. 2015. Vol. 32, № 5. P. 822–836.
65. Gu W. et al. Model-based multi-fringe interferometry using Zernike polynomials // Opt. Lasers Eng. Elsevier Ltd, 2018. Vol. 105. P. 198–200.
66. Houzet J. et al. Ultrafast laser spatial beam shaping based on Zernike polynomials for surface processing // Opt. Express. 2016. Vol. 24, № 6. P. 6542.
67. Liu J. et al. Phase only pupil filter design using zernike polynomials // J. Opt. Soc. Korea. 2016. Vol. 20, № 1. P. 101–106.
68. Liu J. et al. Depth of field extending and super-resolving with phase pupil filter of zernike polynomials // Acta Opt. Sin. 2015. Vol. 35, № 12. P. 1211002.
69. Hosseinimakarem Z. et al. Considering a Zernike polynomial representation for spatial frequency content of optical surfaces // Imaging Appl. Opt. 2015. P. FT2B.2.
70. Hosseinimakarem Z., Davies A.D., Evans C.J. Zernike polynomials for mid-spatial frequency representation on optical surfaces // Reflection, Scatt. Diffr. from Surfaces V. 2016. Vol. 9961. P. 99610P.
71. Rolt S. Optical Engineering Science, First Edition. 2020 John Wiley & Sons Ltd. P. 95–109.
72. Aharmim B. et al. Generalized Zernike polynomials: operational formulae and generating functions // Integr. Transform. Spec. Funct. 2015. Vol. 26, № 6. P. 395–410.
73. Hamyani A. El, Ghanmi A., Intissar A. Generalized Zernike polynomials: Integral representation and Cauchy transform. 2016. P. 1–7.
74. Krishna P.R., Sagar D.K. Aberrations in Theories of Optical Aberrations // IOSR J. Appl. Phys. 2017. Vol. 9, № 4. P. 37–43.
75. Area I., Dimitrov D.K., Godoy E. Recursive computation of generalised Zernike

- polynomials // J. Comput. Appl. Math. Elsevier, 2017. Vol. 312. P. 58–64.
76. Rodrigues M.M., Vieira N. Integral Methods in Science and Engineering, Volume 1 // Integr. Methods Sci. Eng. 2017. Vol. 1. P. 265–276.
 77. Svechnikov M. V. et al. Resolving capacity of the circular Zernike polynomials // Opt. Express. 2015. Vol. 23, № 11. P. 14677.
 78. Bentley J., Olson C. Zernike Polynomials // F. Guid. to Lens Des. 2013. Vol. 1. P. 888–906.
 79. Лексин А.Ю., Кутровская С.В. Физические и математические принципы адаптивной оптики. Методические указания к лабораторным занятиям для студентов ВлГУ. С. 33.
 80. Masalehdan H. et al. Modeling of zernike optical aberrations by MTF and PSF // Opt. InfoBase Conf. Pap. 2010. P. 3–5.
 81. Jallod U.E. Simulations of Four Types of Optical Aberrations using Zernik Polynomials // Iraqi J. of Sci. 2017. Vol. 58, № 1C. P. 583–591.
 82. Li L. et al. Analytical method for the transformation of Zernike polynomial coefficients for scaled, rotated, and translated pupils // Appl. Opt. 2018. Vol. 57, № 34. P. F22–F30.
 83. Li Y. et al. Freeform surface of progressive addition lens represented by Zernike polynomials // 8th Int. Symp. Adv. Opt. Manuf. Test. Technol. Adv. Opt. Manuf. Technol. 2016. Vol. 9683. P. 96830W.
 84. Селивестрова Е.В., Щербина Г.А., Шурыгин Б.М. Алгоритм нахождения абберационных коэффициентов в космической оптико-электронной аппаратуре наблюдения // Журн. Радиоэлектроники 2017. №12. С. 1–18.
 85. Smith C.S., Marinica R., Verhaegen M. Real-time wavefront reconstruction from intensity measurements // 3rd AO4ELT Conf. - Adapt. Opt. Extrem. Large Telesc. 2013. P. 1–8.
 86. Braat J., Dirksen P., Haver S., Janssen A. Detailed description of the ENZ approach [electronic resource]. 2013. P. 39. Available at: http://www.nijboerzernike.nl/_PDF/NijboerZernike_webpage_detailed_20151023.pdf

87. Методы многомерной оптимизации : методические указания и задания к выполнению лабораторных работ по дисциплине «Методы оптимизации» для студентов направления «Прикладная математика»/ сост. Т. М. Попова. – Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2012. – 44 с.
88. Методы одномерной оптимизации : методические указания и задания к выполнению лабораторных работ по дисциплине «Методы оптимизации»/ сост. Т. М. Попова. – Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2011. – 26 с.