

Санкт–Петербургский государственный университет

*Куранова Ирина Юрьевна*

Выпускная квалификационная работа

*Реставрация фотоизображений*

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2015 «Прикладная математика, фундаментальная информатика и программирование»

Профиль «Исследование и проектирование систем управления и обработки сигналов»

Научный руководитель:

доцент, кафедра компьютерного моделирования  
и многопроцессорных систем, к.т.н.,  
Гришкин В.М.

Рецензент:

доцент, кафедра космических технологий  
и прикладной астродинамики, к.ф.-м.н.,  
Степенко Н.А.

Санкт-Петербург

2019 г.

# Оглавление

Введение.....	4
Постановка задачи.....	6
Глава 1. Анализ проблемы .....	7
1.1 Общий обзор литературы.....	7
1.2 Классификация дефектов и их детектирование.....	9
1.2.1 Дефекты старых фотографий.....	9
1.2.2 Цифровые дефекты .....	10
1.2.3 Шумы.....	10
1.2.4 Дефекты, возникающие в процессе фотосъемки.....	11
1.2.5 Детектирование дефектов .....	11
1.3 Обзор существующих методов восстановления изображений .....	12
1.3.1 Структурные методы .....	12
1.3.2 Тектурные методы.....	13
1.3.3 Гибридные методы.....	14
1.3.4 Полуавтоматические методы и методы быстрой реконструкции.....	14
1.3.5 Поиск по экземпляру .....	15
1.3.6 Нейронные сети.....	16
1.3.7 Методы на основе решения дифференциальных уравнений в частных производных.....	18
Глава 2. Методы .....	19
2.1 Самоорганизующиеся карты Кохонена.....	19
2.1.1 Основная информация .....	19
2.1.2 Алгоритм обучения.....	21
2.1.3 Построение модели .....	22
2.1.4 Восстановление изображений.....	25

2.1.5 Способы улучшения алгоритма.....	26
2.2 Метод Криминиси .....	27
2.3 Метод Мамфорда – Шаха.....	29
Глава 3. Практическая реализация .....	34
3.1 Структура программного обеспечения.....	34
3.2 Сравнение методов .....	34
3.3 Сравнение с медианным фильтром.....	36
Выводы .....	40
Заключение .....	41
Список литературы .....	42
Приложение А Иллюстративный материал .....	47

## Введение

За время существования фотографии было создано огромное количество объектов, имеющих высокую культурную ценность. Большинство из них на бумажной основе, и по разным причинам их состояние с годами ухудшается. Сохранение и повсеместное их использование требуют в качестве первого шага перевода в цифровой формат и последующей корректировки для дальнейшей работы с ними.

Среди основных задач компьютерной графики выделяют три основных направления: визуализация, распознавание и обработка изображений. Обработка изображений — форма обработки информации, для которой входные и выходные данные являются изображениями. Задачей обработки может быть улучшение изображения по определенному критерию или преобразование, кардинально меняющее изображение.

В данной работе будет рассматриваться задача восстановления изображений. В последнее время ей уделяется пристальное внимание как одной из главных проблем цифровой обработки. Предполагается, что существовало некоторое исходное изображение, которое было испорчено, полученное изображение хуже воспринимается человеком. Задача — заполнить поврежденные области с использованием информации из оставшихся участков изображения.

Эта задача имеет практическое применение во многих областях. Например, редактирование пользовательских фотографий; удаление изображений людей или надписей; реставрация старых фильмов, картин и фотографий; обработка цифровых аэрокосмических изображений (АКИ), восстановление зашумленных изображений. Также восстановление служит промежуточным этапом для дальнейшего распознавания изображений.

Для успешного решения многих задач важна чистота данных. Требуется обработать добываемые данные, например АКИ, чтобы

максимально приблизить их к исходным. Виды и причины возникновения дефектов будут рассмотрены в следующей главе. С чистым изображением проще и эффективнее работать, улучшается результативность методов обработки, улучшается визуальная составляющая, понижается шанс ложных срабатываний алгоритмов обработки.

Помимо этих, достаточно очевидных способов применения алгоритмов восстановления, можно также привести нетривиальный способ их использования для анонимизации фотоснимков. Бывают случаи, когда нужно скрыть информацию о том, какой камерой была сделана фотография. В работе [1] предлагается заменить некоторые пиксели, используя алгоритмы восстановления, что не ухудшит качества изображения, но позволит исказить признаки, по которым опознаются фотокамеры.

Исторически задача реконструкции изображений в утерянных областях решалась художниками и реставраторами вручную. Как правило, это подразумевало устранение незначительных дефектов вроде царапин и трещин, а также нежелательных эффектов, возникающих в процессе фотосъемки. Это сложно, дорого и приемлемо только для очень важных фотографий. Старые фотографии оказалось проще восстановить после их оцифровки. С развитием цифровых технологий появилась возможность автоматизации этих действий, что облегчит жизнь непрофессионалам. Автоматизированное восстановление даст более быстрый и дешевый результат. Можно предложить следующую схему работы: автоматическое устранение дефектов; коррекция автоматической обработки; устранение реставратором в диалоговом режиме редких и сложных дефектов.

## Постановка задачи

Восстановление фотоизображений также называют врисовыванием отсутствующих областей (inpainting). Целью является "врисовка" поврежденной области на изображении таким образом, чтобы результат восстановления был незаметен для взгляда наблюдателя.

В литературе описано множество различных методов и подходов к реставрации, но все они обладают определенными недостатками. Большинство из них хорошо работает с небольшими повреждениями, так что вопрос о восстановлении крупных областей остается открытым.

Проанализируем уже существующие методы и выберем для подробного рассмотрения некоторые из них, потенциально способные справиться с задачей восстановления больших повреждений. Также рассмотрим нестандартное применение метода карт Кохонена для задачи восстановления изображений и сравним полученные результаты.

Для проведения экспериментов поставим более узкую задачу — восстановить область на растровом цветном изображении, обозначенную пользователем. Например, нужно удалить с изображения мешающий восприятию объект и восстановить фоновое пространство за ним.

Сравним результаты экспериментов, полученные различными методами, для изображений с различной степенью повреждений, применим различные критерии оценивания итогового результата. Кроме того, проведем сравнительный анализ результатов работы карт Кохонена и алгоритмов шумоподавления (фильтров).

# Глава 1. Анализ проблемы

## 1.1 Общий обзор литературы

Проблеме восстановления изображений посвящено множество монографий и статей. Рассмотрим наиболее интересные из них.

Монография Г. Винклера [2] является одной из наиболее информативных книг, в ней описываются подходы синтеза и анализа текстур, основы вероятностного подхода к задачам обработки изображений. Эта книга не требует специальных знаний и дает достаточно полное представление о перспективном, с точки зрения автора, Байесовском подходе к задаче восстановления изображений.

В ряде работ [3-9] предлагается использование нейронных сетей для задач врисовки. Возможности их применения очень широки. Им присущи определенные недостатки: сложная структура, плохая интерпретируемость и долгое время обучения [10]. Однако их преимущества, такие как, высокая допустимость к зашумленным данным и низкий коэффициент ошибок, отказоустойчивость, адаптация к изменениям, непрерывное усовершенствование и оптимизация различных алгоритмов обучения сетей, делают их все более популярными. Авторы статей используют разнообразные архитектуры сетей, от генеративно-состязательных и сверточных до совершенно необученных. Некоторые из подходов оказываются весьма нетривиальными.

Ряд статей [11-15], описывающих применение карт Кохонена, стал вдохновением для данной работы. Это необычный и любопытный способ применения многим известному алгоритму для несвойственной ему задачи.

Хотя восстановление видеодокументов не является областью интереса данной работы, невозможно было не уделить внимание некоторым способам реконструкции отдельного видеокadra [16, 17]. Хотя эти задачи смежные и

кажутся очень похожими, методы их восстановления зачастую неприменимы друг к другу.

Одной из сложностей разработки алгоритмов является оценивание результата. Ряд исследований особенностей восприятия человека [17–21] показывает, что некорректно реконструированные области привлекают внимание людей, заставляя их тщательно рассматривать такой участок изображения. Работа [22] показывает, что качество реконструкции в большей степени зависит от формы невидимой области, чем от ее размера или количества таких областей.

Работу [23] можно назвать настольной книгой обработчика изображений, в ней рассматриваются задачи восстановления, сегментации, сжатия изображений, распознавания объектов, различные модели фильтрации, описывается применение преобразования Фурье и вейвлет-преобразования для поставленных задач. Автор диссертации [24] однако уверена, что необходимы новые альтернативные спектральные методы для цифровой обработки, отличные от методов прошлого века для обработки аналоговых сигналов.

Достаточно большое количество работ [25–28] посвящены использованию метода, основанного на дифференциальных уравнениях в частных производных, и его различным модификациям, однако результаты этого подхода не так впечатляют, как, например, результаты нейронных сетей.

Немало работ посвящено алгоритму поиска по экземпляру, но основной среди них является работа [29], положившая начало подобному подходу. В работе [30] проводится сравнение двух методов: Bertalmio [25], основанного на продлении изофотов внутрь отсутствующей области, и поиска по экземпляру. В результате исследований результаты второго метода были признаны неудовлетворительными, а сам алгоритм требовал значительно больше времени.

## 1.2 Классификация дефектов и их детектирование

### 1.2.1 Дефекты старых фотографий

Старые фотографии — это листы бумаги с желатином и нитратом серебра на поверхности. На них может появляться множество дефектов под воздействием различных факторов. Некоторые из них могут привести к полной потере информации. Повреждение может быть глобальным, если оно затрагивает каждый пиксель изображения, диффузным, если распределено по всему изображению, не затрагивая все пиксели, и локальным, если оно расположено в ограниченной части изображения. Рассмотрим классификацию повреждений по причинам их возникновения [31, 32].

Механические (физические) повреждения:

- Пятна от приклеившихся материалов;
- Трещины;
- Загибы;
- Кракелюр;
- Потертости;
- Деформация самой бумаги;
- Царапины;
- Действия человека: оторванные части, пометки, штампы, надписи, следы клея;
- Отслоение фотографического слоя.

Химические повреждения:

- Полупрозрачные пятна;
- "Фоксинг" — лисьи пятна;
- Однородные серые пятна;
- Выцветание;
- Пожелтение.

Биологические повреждения:

- Цветные пятна плесени из-за грибковых инфекций;
- Повреждения, вызванные насекомыми или мелкими грызунами, питающимися фотоматериалами.

При ручном восстановлении каждый тип дефекта имеет стандартную цифровую маркировку, и каждый дефект удаляется наиболее подходящим способом (рис. А1 Приложения А). При оцифровке изображения все эти дефекты останутся.

### 1.2.2 Цифровые дефекты

Некоторые дефекты могут появляться после перевода изображений в цифровой вид. Примером является муар (moire) — это оптический интерференционный эффект, возникающий на изображении при наложении двух растровых сеток (и любых двух изображений, имеющих периодическую структуру). При сканировании печатных оригиналов, можно столкнуться с проблемой появления "муара". Он может отображаться в виде полос, ряби или кривых интенсивности и цветовых разнообразий, наложенных на фотографию. Этот эффект серьезно повреждает визуальное качество изображения.

### 1.2.3 Шумы

Шумы могут возникать по многим причинам, например, при передаче изображения по зашумленному каналу, для аэрокосмических снимков из-за мешающих электронных средств бортового комплекса, например, устройств электропитания, бортовых систем связи, в процессе съемок из-за особенностей матрицы.

Типичными примерами моделей шумов являются:

- аддитивный гауссовский шум;
- импульсный шум;
- спекл-шум;
- шум типа "соль и перец";
- белый шум;

- и прочие.

#### 1.2.4 Дефекты, возникающие в процессе фотосъемки

- смаз изображения;
- расфокусировка изображения;
- смаз движущегося объекта на изображении;
- и прочие.

На рис. А2 приложения А представлены некоторые из описанных выше дефектов.

#### 1.2.5 Детектирование дефектов

Данные дефекты сложно обнаружить автоматически, так как нет определенных правил, как их отличить от реальных особенностей изображения. Они могут указываться пользователем в виде масок или обнаруживаться специальными алгоритмами.

В работе [33] описаны общие методы детектирования механических повреждений. В статье [34] предлагается метод для поиска чернильных пометок и пятен. Обнаружить чернила, используя цветовую информацию, почти невозможно. Используются фильтр по интенсивности и форма дефектов. Изменение интенсивности чернил обычно плавное и устойчивое. При уменьшении интенсивности фотографии часть чернил останется, в то время как другие части будут отфильтрованы до некоторого порогового значения. Запоминается количество пикселей, обнаруженных для различных пороговых значений. Если дисперсия числа пикселей между двумя различными шагами настройки низкая, следовательно, пиксели не зависят от текущих настроек и могут быть пикселями чернил из-за устойчивой интенсивности.

Одна из немногих работ [35] посвящена удалению эффекта муара. Предлагается способная работать с различными разрешениями входного изображения (multiresolution) сверточная сеть для автоматического удаления дефекта. Сеть преобразует входное изображение в несколько карт, каждая

карта объектов подается в стек каскадных сверточных слоев, эти слои отвечают за корректирование определенной частоты, потом результаты работы всех слоев сливаются.

Для детектирования повреждений на видео можно использовать методы оценки движения, модели Кокарама, но в общем случае они не пригодны для работы с единственным изображением.

Во многих работах предполагается, что отметить область для восстановления должен пользователь. Это значительно проще полностью ручного восстановления, но лишает процесс полной автоматизации.

### **1.3 Обзор существующих методов восстановления изображений**

Существует множество методов для восстановления испорченных изображений, но все они обладают определенными недостатками. Активно разрабатывается задача усовершенствования и повышения эффективности и быстродействия уже имеющихся методов, но до сих пор не изобретено универсального метода.

Весьма условно можно выделить несколько классов:

- Структурные методы
- Текстурные методы
- Гибридные методы
- Полуавтоматические методы и методы быстрой реконструкции
- Поиск по экземпляру
- Нейронные сети
- Методы на основе решения дифференциальных уравнений в частных производных.

#### **1.3.1 Структурные методы**

Структурные методы [36] основаны на наличии априорной информации о структуре данных в поврежденной области. Один из способов

получения этой информации — анализ блоков изображения, смежных с поврежденным. Структурные методы восстановления можно разделить на два подкласса:

- 1) на основе взвешенного усреднения значений пикселей (weighted averaging);
- 2) на основе направленного восстановления контуров (directional interpolation).

Алгоритмы первого типа способны хорошо восстанавливать гладкие области на изображении, методы второго типа — области, содержащие контуры.

Алгоритм можно описать следующим образом:

- 1) оценка направления контуров в поврежденной области за счет анализа пикселей из неискаженных блоков;
- 2) восстановление контуров на основе информации, содержащейся в соседних блоках;
- 3) восстановление оставшихся пикселей изображения;
- 4) постобработка всего восстановленного региона целиком.

Человеческое восприятие очень чувствительно к контурам объектов, поэтому восстанавливать их нужно точно и в максимально четком, не размытом виде.

### 1.3.2 Текстурные методы

Самые первые подходы предлагали заполнение повреждений за счет копирования блоков из неповрежденных областей. Данная идея получила развитие — новая текстура синтезируется из уже существующей, а после производится уточнение значений соседних пикселей. Далее обеспечивается условие непрерывности между границами области и восстановленным содержимым. Такие методы показывают неплохие результаты лишь для некоторых изображений. Эти идеи развиваются, например, в исследовании [37] предполагается использование сверточной нейронной сети для синтеза текстур, но применение только текстурных методов дает хорошие результаты для простейших случаев, так как в большинстве случаев нужно учитывать и

структуру изображения. Поэтому получили развитие идеи гибридных алгоритмов.

### 1.3.3. Гибридные методы

Гибридные алгоритмы учитывают и текстурные, и структурные особенности. В работе Bertalmio "Simultaneous structure and texture image inpainting" [38] предлагается декомпозиция изображения на структурную и текстурную части, применяются соответствующие алгоритмы. Во многих работах именно пользователю предлагается указать линии структур, так как автоматический процесс испытывает трудности с разнообразными кривыми и Т-соединениями. В работе [39] структурные элементы заполняются первыми, потом область заполняется текстурой. Этот подход показывает хорошие результаты при предположениях, что наиболее заметные отсутствующие структуры могут быть аппроксимированы несколькими четко определенными кривыми, и каждый структурный фрагмент имеет соседнюю неповрежденную область. Таким образом, выполняется три этапа: пользовательское взаимодействие для задания кривых, распространение структуры, распространение текстуры для заполнения оставшихся неизвестных областей. Такой подход позволяет сохранить структурную целостность изображения. В работе [40] предлагается проводить реконструкцию контуров с помощью сплайн-интерполяции, что показывает результаты лучше, чем методы на основе дифференциальных уравнений и поиска по экземпляру.

### 1.3.4. Полуавтоматические методы и методы быстрой реконструкции

В основном, участие пользователя сводится к помощи в структурной реконструкции границ: нужно наметить их приблизительное расположение. После этого применяется алгоритм для синтеза текстур. Если граница описывается одной кривой, то можно использовать технику динамического программирования.

В работе [41] задача пользователя — указать диффузионный барьер на изображении, потом происходит свертка изображения с Гауссовым ядром. Этот метод работает достаточно быстро, что является его основным преимуществом, но дает хорошие результаты только для небольших повреждений.

Для ускорения работы методов реконструкции был разработан ряд техник, получивших название быстрых алгоритмов реконструкции, но такие ускоренные методы не подходят для реконструкции больших областей на изображениях, так как уступают в точности определения границ классическим методам и чаще порождают размытые области.

### 1.3.5. Поиск по экземпляру

Внутри класса можно выделить две методологии.

Первый способ восстановления больших фрагментов изображений был предложен в работе [29]. Точки, лежащие на границе, имеют наивысший приоритет и будут заполняться в первую очередь. Из возможных кандидатов для заполнения будет выбран один наиболее подходящий экземпляр из неповрежденной области. Метод показывает неплохие результаты работы, но его недостатком является риск неверного определения наиболее подходящего экземпляра, так как будет происходить итерационное накопление ошибки. Существуют усложненные варианты с выбором ориентации и разрешения для наиболее подходящего фрагмента, что требует значительных вычислительных затрат. Кроме того, не удастся создать новые объекты, отсутствующие в изображении.

Второй подход основан на предположении, что невозможно получить всю требующуюся информацию из одного неполного изображения, поэтому будет логичным использовать базу изображений, осуществляя в ней поиск наиболее близких изображений с помощью дескриптора и потом с помощью преобразования Пуассона обеспечить незаметную, бесшовную вставку [6]. В работе [5] используется такой же подход, но на выходе пользователю предоставляются на выбор несколько наилучших результатов. Но главная

мысль этой статьи, что результаты работы алгоритма улучшаются по мере увеличения базы; что для любого входного изображения можно найти в базе наиболее подходящее, и при этом количество изображений не будет бесконечно большим. Однако при всех этих плюсах, время работы алгоритма может быть весьма продолжительным. Для подобных методов критерии качества, основанные на сравнении с опорным изображением, уже не будут работать. Нужно, чтобы наблюдатель просто не заметил восстановленную область.

В качестве примера такого метода можно привести сервис, разработанный компанией Mail.ru . Он находит и устраняет дефекты, затем раскрашивает фотографию. Для обучения использовалось множество фотографий и фильмов. Результат работы сервиса представлен в приложении А на рис. А3. Однако результат работы данного сервиса заметно уступает ручному восстановлению.

### 1.3.6. Нейронные сети

В последнее время активно развиваются идеи применения нейронных сетей для восстановления изображений. Причем, существует множество разнообразных подходов к их применению. В работе [7] был предложен подход использования предобученной классифицирующей сети для восстановления фотоизображений. Испытания проводились на двух типах данных: рукописных цифрах и человеческих лицах. Важным фактором обучения сети является обучение ее не на целых изображениях, а на их фрагментах, что позволяет сети лучше усваивать визуальные характеристики изображения. Технически это осуществлялось за счет использования одного и того же тренировочного набора, подающего каждый пример много раз со случайными нулевыми масками, чтобы удалить из него большие части. Идея метода — минимизировать отклонение между меткой исходного изображения и меткой результата классификации путем дорисовки недостающих пикселей. Для естественности изображения применяется метод полной вариации.

Алгоритм использует метод градиентного спуска, поэтому его начальная точка критична. Удалось значительно улучшить результаты, заполнив недостающие части линейной интерполяцией (для цифровых рисунков) и усредненным значением (для лиц). Результаты работы алгоритма названы впечатляющими, но он не срабатывает для небольших повреждений, которые просто не замечаются сетью, и, очевидно, что такой подход достаточно узкоспециализирован.

Одна из недавних идей применения нейронных сетей в восстановлении изображений — применение генеративно-сопоставительных сетей (Generative adversarial network, GAN). Эта сеть состоит из двух дочерних нейронных сетей: сети-генератора и сети-дискриминатора. Генератор получает на вход случайный вектор (точку из скрытого пространства (hidden space)) и переводит его в искусственное изображение, а дискриминатор получает на вход изображение и пытается определить настоящее ли оно (взято из обучающего датасета) или сгенерировано. Нейронные сети такого типа могут иметь любую структуру генератора и дискриминатора, однако на практике в обработке изображений повсеместно используются сверточные сети, являющиеся удобным и хорошо изученным средством для работы с изображениями. Этот вид сетей позволяет заполнять даже большие поврежденные области с потрясающей точностью.

Однако для данного виде сетей существуют несколько проблем. Они связаны с проблемой нестабильности в процедуре обучения: исчезающие градиенты и коллапс режима.

Если дискриминатор способен отлично различать реальные и сгенерированные изображения, он достигает своего оптимального значения, и, таким образом, генератор больше не улучшает сгенерированные данные. С другой стороны, коллапс режима происходит, когда генератор продолжает производить выходы, уже обманувшие дискриминатор, не производя ничего нового. В некоторых работах, например, [4] предлагается использовать

модифицированную сеть Вассерштайна, чтобы избежать проблемы исчезающих градиентов.

Можно использовать более сложную архитектуру сети [3]. Она состоит из трех подсетей: кодер–декодер, дискриминатор и сеть синтаксического анализа (parsing network). Задача дискриминатора — судить, соответствует ли заполненная область изображению. Сеть кодер–декодер является прямой сверточной нейронной сетью, которая стремится заполнить недостающую область максимально правдоподобно, чтобы обмануть дискриминатор. Сеть синтаксического анализа — это предварительно обученная сеть для захвата различий между синтезированной областью и истиной.

### 1.3.7. Методы на основе решения дифференциальных уравнений в частных производных

Идея метода основана на принципах работы профессиональных реставраторов и уравнении теплопроводности. Опыт работы реставраторов привел к итеративному процессу продления структурных границ областей объектов на изображении (изофотов) внутрь отсутствующей области. Эта методика получила название анизотропной диффузии, так как она учитывает специфическую информацию о направлении изофотов. Однако процесс диффузии приводит к размытию, поэтому у данного метода возникнут проблемы с заполнением текстурированной области.

При использовании методов вариационного исчисления, ставится задача найти функцию, минимизирующую функционал. Однако этот подход также не способен восстанавливать текстурную информацию, но неплохо справляется с небольшими областями и в некоторых случаях с задачей шумоподавления.

## Глава 2. Методы

Для подробного рассмотрения выберем следующие методы: карты Кохонена, метод Криминиси, метод Мамфорда – Шаха.

### 2.1 Самоорганизующиеся карты Кохонена

#### 2.1.1 Основная информация

Рассмотрим изображение как массив статистически упорядоченных данных. Так как группы пикселей связаны между собой, то возможно восстановление с помощью статистического и кластерного анализа окружающих пикселей или изображения в целом. Следовательно, для восстановления фотографий можно использовать самоорганизующиеся карты Кохонена.

Самоорганизационный процесс — процесс обучения без учителя, то есть, на выходе нет сравнения с желаемыми значениями.

Идея карты была предложена финским ученым Т. Кохоненом в 1984 году. Самоорганизующаяся карта Кохонена — это самообучающаяся нейросетевая архитектура, в которой учитывается информация о взаимном расположении нейронов, которые образуют решетку. Т. Кохонен предложил два типа решеток — прямоугольный и гексагональный (рис. 1). Отличие состоит в том, что в прямоугольной сетке каждый узел соединен с четырьмя соседними узлами (слева, справа, сверху, снизу), а в гексагональной — с шестью ближайшими (по циферблату часов – 2, 4, 6, 8, 10, 12 часов). Процесс построения карты для сеток отличается способом перебора соседей, но шестиугольные ячейки более корректно отображают декартово расстояние

между объектами на карте, так как для смежных ячеек расстояние между центрами одинаковы.

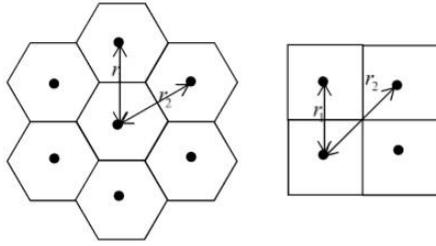


Рис. 1: Два типа решеток: гексагональная и прямоугольная

Самоорганизующиеся карты Кохонена широко применяются для моделирования, прогнозирования, классификации, кластеризации, поиска закономерностей в больших массивах данных, выявления наборов независимых признаков и сжатия информации. При правильной настройке карты и корректно выбранных критериях обучения и восстановления этот метод показывает неплохие результаты и для задачи восстановления. Решение данной задачи состоит из двух основных этапов: обучения карты Кохонена и восстановления по обученной карте.

Сигнал в сеть поступает сразу на все нейроны, веса синапсов интерпретируются как координаты положения узла, и выходной сигнал формируется по принципу WTA (winner takes all, победитель забирает все), то есть ненулевой выходной сигнал имеет ближайший к поданному на вход объекту нейрон. В процессе обучения карты настраиваются веса не только нейрона-победителя, но и группы его соседей из некоторой окрестности карты. Нейрон принадлежит окрестности нейрона-победителя, если расстояние между ним и нейроном-победителем на карте меньше определенной величины (в процессе обучения изменяются веса нейронов, но не их положение). Такой алгоритм называют алгоритмом типа WTM (winner takes most, победитель получает больше). Цель — минимизировать ошибку замены близких в смысле используемой метрики входных векторов вектором весов. В результате близкие по некоторой метрике входные векторы в карте Кохонена могут относиться к разным близко расположенным на сетке

нейронам. Итогом работы алгоритма является обученная карта (рис. А4 приложения А), то есть будут определены веса всех нейронов. Очевидно, что обучение неконтролируемое, мы можем задать лишь входные значения.

Рассмотренная разновидность нейронных сетей называется сетью с конкурентным обучением. Изначально она была предложена как математическая модель, но позже удалось найти некоторые ее аналоги в нейрофизиологии.

Данный метод можно также применять для обработки отдельного видеокadra. Например, ставится задача реконструировать фоновое пространство за движущимся объектом. Эта задача решается за счет выборки смежных кадров и получения дополнительной информации из них, построения трехмерной сети Кохонена, а также отслеживания объекта [16].

### 2.1.2 Алгоритм обучения

Алгоритм обучения производится по итерациям. Пусть  $t$  – номер итерации.

1. Выбираем случайный вектор  $x(t)$  из набора входных значений.
2. Находим расстояния до всех векторов весов всех нейронов карты.

Для данной операции выбирается метрика  $d(x,y)$ , необходимая для выбора нейрона-победителя (best matching unit, BMU). Ищем нейрон, наиболее близкий к входному значению  $x(t)$ :

$$d(x(t), m_c(t)) \leq d(x(t), m_i(t)),$$

где  $m_c(t)$  – вектор веса нейрона победителя,  $m_i(t)$  – вектор веса  $i$ -го узла в карте,  $d(x,y)$  – выбранная мера отклонения. Если этому условию удовлетворяет несколько нейронов, то победитель выбирается случайным образом.

3. Определение меры соседства нейронов и изменение весов нейронов в карте.

Выбираем меру соседства — функцию  $h(t)$  — функция, возвращающая расстояние между нейронами и зависящая от номера итерации  $t$ . Лучший результат дает Гауссова функция.

#### 4. Вычисление ошибки карты

Изменяем вектора весов по формуле:

$$m_i(t) = m_i(t - 1) + h(t)(x(t) - m_i(t - 1)). \quad (1)$$

Таким образом, все узлы, являющиеся соседями ВМУ, становятся еще ближе к рассматриваемому наблюдению.

#### 5. Критерий остановки

Основными критериями выступают:

- максимальное количество итераций, при котором система считается обученной
- минимальная точность, рассчитываемая как сумма средних изменений весов в итерации обучения
- исчерпание всех возможных наборов векторов наблюдаемых данных (предложен в работе [14]).

### 2.1.3 Построение модели

#### 1. Преобразование изображения в массив векторов

Один из подходов к преобразованию — разбить изображение на блоки определенной аналитиком размерности и сформировать его векторное представление. Поврежденные пиксели не должны учитываться при формировании вектора данных. Вектор строится по формулам:

$$\begin{aligned} \omega_i(x, y) &= P[l + i \oplus S, t + i \div S], \\ l &= x - [S/2], \\ t &= y - [S/2], \end{aligned} \quad (2)$$

где  $S$  — размер блока пикселей;  $P[i, j]$  — пиксель, расположенный в  $i$ -м столбце и  $j$ -й строке на изображении;  $x, y$  — координаты, для которых строится вектор.

При использовании представленного выше метода построения векторов необходимо учитывать следующие ограничения:

$$[S/2] \leq x \leq W - [S/2], \quad (3)$$

$$[S/2] \leq y \leq H - [S/2],$$

где  $W$  – ширина изображения в пикселях,  $H$  – высота изображения в пикселях.

Условия (3) вводятся, так как размер рассматриваемого блока больше одного пикселя, и количество таких блоков будет меньше, чем количество пикселей в изображении. Дополнительно накладывается условие, что  $S$  должно быть нечетным для симметричности окружности пикселя.

## 2. Инициализация

Перед процессом обучения производится инициализация сети — первоначальное задание векторов весов. В простейшем случае задаются случайные значения весов.

Одним из недостатков карт Кохонена является зависимость от инициализации. Начальное распределение весов существенно влияет на процесс обучения и может сказываться на процессе. Установлено, что любое упорядочение начального состояния карты сокращает время обучения сети. Поэтому вторым популярным подходом к инициализации является инициализация стартовыми блоками.

## 3. Этап грубой подгонки

Для этих блоков производится обучение карты. Изначально карта должна нести максимальное количество информации из изображения. Поэтому из изображения выбираются блоки без повреждений и максимально отличные друг от друга.

## 4. Выбор метрики

В работе [44] предложены четыре метрики, основанные на различиях пикселей.

- Метрика Минковского,
- Среднеквадратическая ошибка,
- Метрика разницы с соседями,
- Метрика многомерного расстояния.

Можно использовать любую из них, но в общем случае используется второй вид, во многом из-за его простоты.

## 5. Выбор функции соседства

Самыми распространенными являются следующие функции: Гаусса (4), "французская шляпа" (5) и дискретная (6):

$$h(p, t) = e^{-\frac{p^2}{2\sigma^2(t)}}, \quad (4)$$

$$h(t) = \begin{cases} 1, & |p| \leq a, \\ -\frac{1}{3}, & a < |p| \leq 3a, \\ 0, & |p| > 3a. \end{cases} \quad (5)$$

$$h(t) = \begin{cases} 1, & |p| \leq 1, \\ \frac{1}{2}, & 1 < |p| \leq 2, \\ \frac{1}{4}, & 2 < |p| \leq 3, \\ \frac{1}{8}, & 4 < |p| \leq 4. \end{cases} \quad (6)$$

где  $p$  – расстояние между узлами карты;  $t$  – номер итерации;  $a$  – заранее определенная константа (обычно  $a = 2$ ).

Для лучшего результата предлагается видоизменить вид  $h(t)$  следующим образом:

$$h(t) = \alpha(t) \cdot e^{-\frac{d(c,i)^2}{2\sigma(t)}}, \quad (7)$$

где  $\alpha(t)$  — функция скорости обучения, монотонно убывающая с каждой последующей итерацией (определяет приближение значения векторов веса ВМУ и его соседей к наблюдаемому).

Из опытов предлагается следующий вид функции:

$$\alpha(t) = \begin{cases} 1, & t < 10, \\ \frac{1}{(t-9)^{0.2}}, & t \geq 10. \end{cases} \quad (8)$$

Функция (8) на первых 9 итерациях имеет значение 1. Это сделано для того, что бы максимально равномерно инициализировать карту в начале обучения. Первые итерации являются инициализацией, и в ходе этих шагов не происходит поиск ВМУ. Номера нейронов на этом этапе задаются явно и расположены максимально равномерно по всей карте.

Функция (9) является убывающей функцией от номера итерации. Наиболее часто используется линейно убывающая функция. Из опытов было установлено, что функция вида

$$\sigma(e) = 5 \frac{\sqrt{Q}}{\sqrt{t}}, \quad (9)$$

где  $Q$  – количество нейронов в карте, позволила максимизировать воздействие на нейроны на начальных итерациях вне зависимости от количества нейронов.

Рассмотрим  $d(c, i)$ .

$$d(c, i) = \min_{\substack{d_x \in \{-Q_x, 0, Q_x\} \\ d_y \in \{-Q_y, 0, Q_y\}}} \sqrt{(m_c\{x\} - m_i\{x\} + d_x)^2 + (m_c\{y\} - m_i\{y\} + d_y)^2}, \quad (10)$$

где  $Q_x$  – ширина карты в нейронах,  $Q_y$  – высота карты в нейронах,  $m_c\{x\}$  – горизонтальная координата местоположения нейрона  $m_c$ ,  $m_c\{y\}$  – вертикальная координата местоположения нейрона  $m_c$ .

Функция (10) позволяет избежать проблемы попадания ВМУ на край карты и максимизировать воздействие обучения. При использовании, например, евклидова расстояния, при попадании ВМУ на край карты количество нейронов, которые изменяются при обучении, уменьшается в 4 раза по сравнению с ситуацией попадания в центр карты.

#### 2.1.4 Восстановление изображений

После остановки процесса обучения можно приступить к самой процедуре восстановления. Восстановление также производится итерационно. При восстановлении выбирается случайный вектор, в котором поврежденным является только один пиксель. Для данного вектора ищется ВМУ таким образом, чтобы найти наиболее "похожий" вектор, без учета поврежденного пикселя. После того, как ВМУ найден, поврежденный пиксель на изображении заменяется соответствующим из вектора веса нейрона. Данная процедура повторяется, пока есть поврежденные пиксели.

### 2.1.5 Способы улучшения алгоритма

В работах [11, 12, 15] предложен мультиагентный подход для увеличения эффективности, предоставляющий возможность распараллелить процесс решения задачи восстановления текстуры и получить оптимальные результаты точности. Агент по определению может воспринимать свою среду с помощью датчиков и воздействовать на эту среду. В случае единственного агента, он имеет все необходимые способности, знания и ресурсы для решения проблемы. При создании мультиагентных систем (МАС) предполагается, что отдельный агент может иметь лишь частичное представление о задаче и решает одну подзадачу. Следовательно, для решения сложной проблемы требуется взаимодействие агентов (рис. А5 приложения А).

Мультиагентный подход позволяет увеличить точность получаемых результатов реконструкции и производительность процесса реконструкции. Механизм распределенного решения задачи позволил достичь 90–110 % увеличения быстродействия системы для четырех агентов, что является их оптимальным числом. Больше их количество увеличивает коммуникационные затраты на их взаимодействие. На менее сложных изображениях результаты многоагентной и одноагентной конфигурации близки, однако с увеличением размерности и структурной сложности изображений прирост точности становится заметен.

В работах [11, 15] сделан вывод: одна обученная карта Кохонена в большинстве случаев не покрывает 50 % цветовой палитры исходника, поэтому мультикартирование, то есть построение сразу нескольких карт, является оптимальным механизмом для восстановления, так как это дает возможность параллельно восстанавливать изображение по нескольким картам с выбором оптимального значения для каждого пикселя. Оптимальное количество карт — четыре, так как достигается более 92 % покрытия палитры без существенной потери производительности.

Еще больше улучшить результаты можно, восстанавливая каждую смежную текстуру на стыке по отдельно обученной карте. Для этого требуется дополнительный шаг предварительной сегментации изображения.

Недостатком метода является необходимость в точной подборке параметров, что является задачей эксперта. В частности, можно исходить из следующих предположений:

1. Чем больше размер  $S$ , тем менее четким получается изображение.
2. Чем больше нейронов в сети, тем больше палитра цветов.
3. Чем больше итераций обучения, тем точнее восстановленные цвета соответствуют цветам реальной картинке и тем лучше видны детали.

## 2.2 Метод Криминиси

Работа Криминиси [29] положила начало новому подходу к восстановлению изображений, а именно поиску по экземпляру. Алгоритм предназначен как для обработки больших областей, так и для тонких царапин.

Дубликация фрагментов изображения позволяет учесть одновременно и текстурные, и структурные особенности. Алгоритм качественно восстанавливает текстуру и линейные структуры. В отличие от других алгоритмов, в подходе не используется диффузия, так как она порождает размытые регионы.

Ядро алгоритма (рис. 2-3) разбито на две части. Первая часть заполняет отверстия в изображениях путем распространения линейных структур в целевую область. Во второй части предлагается идея, что заполнить большие области изображения можно, повторяя текстуры, окружающие целевую область.

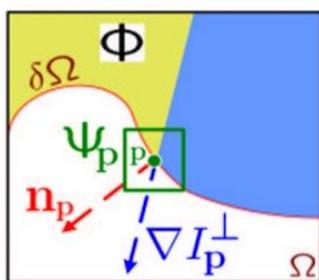


Рис. 2: Схема обозначений:  $n_p$  – единичный вектор, ортогональный границе  $\delta\Omega$  в точке  $p$ ,  $\nabla I_p^\perp$  – вектор изофотов,  $\Phi$  – известная область изображения,  $\Psi_p$  – блок с наибольшим приоритетом.



Рис. 3: Слева направо: исходное изображение с отсутствующей областью  $\Omega$ ; заполняемый участок, содержащий блок с наивысшим приоритетом; наиболее подходящие кандидаты для заполнения лежат на границе между двух текстур в  $\Phi$ ; блок  $\Psi_p$  с наивысшим приоритетом заполнен наиболее подходящим экземпляром.

Для большей эффективности используются блоки, а не пиксели, размер этого блока должен быть больше чем самый большой различимый элемент текстуры. Пользователь выбирает область для удаления. Алгоритм повторяет следующие три шага, пока всем значениям пикселей в целевой области не будет присвоен цвет.

Первым шагом является вычисление приоритетов для блоков, блоки, прилегающие к границе области имеют наивысший приоритет. Приоритеты блоков имеют решающее значение, поскольку они определяют, какой из блоков заполняется первым. Алгоритм заполняет блоки с наивысшим приоритетом и по этой причине известен как алгоритм наилучшего заполнения. Приоритет блока нужен, чтобы сбалансировать распространение линейных структур с заполнением текстур. В результате он определяется как произведение доверительного и информационного сомножителей.

$$P(p) = C(p)D(p), \quad (11)$$

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \bar{\Omega}} C(q)}{|\Psi_p|}, D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha},$$

где  $|\Psi_p|$  – площадь  $\Psi_p$ ,  $\alpha$  – коэффициент нормализации. Множитель  $C(p)$  имеет значение 0 для целевого региона и значение 1 для неповрежденной области. Он гарантирует, что первыми будут залиты блоки, прилежащие к границе. Второй сомножитель отвечает за распространение линейных структур.

Второй шаг включает распространение информации о текстуре и структуре. После того, как все приоритеты были рассчитаны, берется блок с наивысшим приоритетом и заполняется информацией из исходного региона. Информация из области  $\Phi$  отбирается непосредственно, а не распространяется с помощью диффузии. Сходство блоков измеряется с помощью суммы квадратов расстояний уже заполненных пикселей в целевом блоке.

Третий шаг включает в себя обновление значений  $C(p)$  для блока, который только что был заполнен.

Такой способ реконструкции показывает хорошие результаты для разнообразных изображений, однако, испытывает сложности при восстановлении искривленных структурных границ. Также могут возникнуть проблемы, если объект для удаления был выделен не полностью, в этом случае возникают различные деформации, как показано на рис. А6 приложения А.

### 2.3 Метод Мамфорда – Шаха

В работах [27, 28] предложен подход, основанный на вариационном методе и теории эволюции кривых, позволяющий одновременно решать задачи сегментирования и восстановления изображений. Этот подход

позволяет обнаружить границы объектов и эффективно восстановить изображение.

Принцип вариационных методов заключается в построении некоторого функционала, минимум которого дает наилучшую сегментацию. Рассмотрим классический функционал Мамфорда – Шаха:

$$E_{MS}(u, \Gamma) = \beta \int_{\Omega \setminus \Gamma} (u - g)^2 dx + \alpha \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx + \gamma \int_{\Gamma} ds, \quad (12)$$

где  $\Omega$  – некоторая область,  $g$  – функция, заданная в  $\Omega$ ,  $\Gamma$  – совокупность границ,  $u$  – кусочно-гладкое приближение к заданному изображению  $g$ ,  $\alpha$  и  $\beta$  – неотрицательные константы.

Задача — найти пару  $(u, \Gamma)$ , минимизирующую функционал (12). Одновременно ищется кусочно-гладкое приближение  $u$  и некоторое множество границ  $\Gamma$ , на котором функция, приближающая изображение, терпит разрыв, исходя из соображений, что интенсивность внутри каждого объекта изображения является гладкой функцией с небольшим градиентом, но на границе между объектами она может иметь большой градиент или быть разрывной. Тогда сегментацией изображения считается разбиение области на максимальные подобласти, в каждой из которых приближенное изображение представлено как гладкая функция, а множество границ сегментированного изображения  $\Gamma$  должно быть минимальным.

Первое слагаемое функционала отвечает за точность аппроксимации заданного изображения приближенным изображением, второе — за гладкость приближенного изображения в подобластях, а третье представляет собой суммарную длину кривых, составляющих  $\Gamma$ , и отвечает за минимизацию общей длины границ. При отсутствии одного из слагаемых задача имеет тривиальные решения. Например, без первого слагаемого решение  $u = \text{const}$ ,  $\Gamma = \emptyset$ , без второго —  $u = g$ ,  $\Gamma = \emptyset$ , без третьего — функционал можно максимально приблизить к 0, взяв в качестве  $u$  кусочно-постоянную функцию, хорошо аппроксимирующую  $g$ .

Сама по себе задача минимизации этого функционала не является тривиальной, так как неизвестно множество границ  $\Gamma$ . Для того, чтобы облегчить эту задачу, можно построить аппроксимации функционала (Амброзио – Торторелли, эллиптическая) [28] или искать решение в пространстве кусочно-постоянных функций (тогда будет отсутствовать второе слагаемое функционала). Рассмотрим применение функционала для восстановления изображений.

Рассмотрим модель поставленной задачи.

$$g(x, y) = u(x, y) * h(x, y),$$

где  $h(x, y)$  – известная модель повреждений. Ей соответствует дискретная версия  $g = Hu$ , где  $H$  – матрица повреждений.

Для решения задачи предлагается итеративная схема, где каждая итерация состоит из двух шагов.

На первом шаге вычисляется  $u$ , которая минимизирует  $E(u, \Gamma)$ , при заданном  $\Gamma$ . На втором шаге формируем кривую  $\Gamma_t$ , которая является антиградиентом  $E(u(\Gamma), \Gamma)$ . Алгоритм можно представить в виде псевдокода.

*initialize  $\Gamma$ ;*

*while (not converged) {*

*solve  $u = \arg \min_u E(u, \Gamma)$  for fixed  $\Gamma$ ;*

*evolve the curve with  $\Gamma$ ;*

*}*

где условие сходимости –  $\frac{\|u_{prev} - u_{new}\|}{\|u_{prev}\|} < \varepsilon$ ,  $\varepsilon$  – малое неотрицательное число.

1) Введем обозначения (рис. 4).

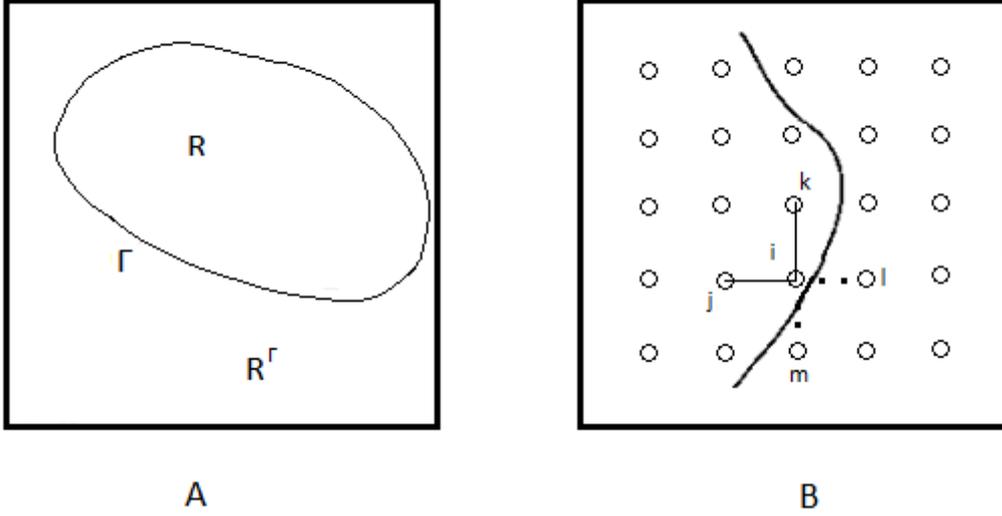


Рис. 4: Схема обозначений: часть А:  $\Gamma$  – замкнутая кривая,  $R$  – область внутри кривой,  $R^\Gamma$  – внешняя область; часть В: наложенная дискретная решетка

2) Найдем необходимое условие, вычислив первую производную от  $E$  относительно  $u_i$  для каждого  $i$ . Пусть  $h_i$  – столбец матрицы  $H$ , то есть

$Hu = \sum_i h_i u_i$ , тогда получаем

$$\frac{\partial E}{\partial u_i} = 2\beta \sum_j h_i^T h_j u_j - 2\beta h_i^T g + 2\alpha \sum_{j \text{ s.t. } (i,j) \in S} (u_i - u_j) = 0, \quad (13)$$

$S = \{(i, j) \mid i, j \text{ находятся по одну сторону от } \Gamma \text{ и являются соседями в решетке}\}$ . Получаем условие для  $i$ -ой строки:

$$\sum_j h_i^T h_j u_j - \frac{\alpha}{\beta} \sum_{j \text{ s.t. } (i,j) \in S} (u_j - u_i) = h_i^T g. \quad (14)$$

Таким образом, необходимое условие (13) выражено следующим линейным уравнением:

$$H^T Hu - \frac{\alpha}{\beta} Au = H^T g, \quad (15)$$

где

$$A_{ij} = \begin{cases} 1, & \text{если } (i, j) \in S, \\ -\sum_{k \text{ s.t. } (i,k) \in S} A_{ik}, & \text{если } i = j, \\ 0, & \text{иначе.} \end{cases}$$

Матрица  $A$ , которая зависит от текущего положения кривой, является дискретной реализацией оператора  $\nabla^2$ . Это линейное уравнение является дискретной аппроксимацией дифференциального уравнения в частных

производных. Один из возможных методов его решения – метод сопряженного градиента.

3) Теперь рассмотрим вариацию  $E$  относительно кривой  $\Gamma$ . Эта вариация похожа на производную  $\frac{\partial E}{\partial u_i}$ . Идея метода состоит в том, чтобы переместить кривую вокруг близкой к кривой точки индекса  $i$  (ее глубина равна  $u_i$ ). В точке, смежной с границей, существует конкуренция между областями  $R$  и  $R^\Gamma$ . Мы должны решить, поместить точку  $i$  в  $R$  или  $R^\Gamma$ .

Формула изменения кривой задается следующим образом:

$$\Gamma_t = \beta DN + \alpha PN - \gamma kN, \quad (16)$$

где

$$D = \|Hu - g\|_{|u_i=u_{R^\Gamma}}^2 - \|Hu - g\|_{|u_i=u_R}^2,$$

$$P = \sum_{\substack{j \in R^\Gamma \\ j \text{ is adjacent to } i}} (u_i - u_j)^2 - \sum_{\substack{j \in R \\ j \text{ is adjacent to } i}} (u_i - u_j)^2,$$

$K$  – кривизна,  $N$  – внешний нормальный вектор кривой  $\Gamma$ .

Таким образом, получили формулу (16) эволюции кривой для минимизации функционала Мамфорда – Шаха. При использовании данного метода для задач восстановления изображения получаются близкие к идеальным границы, хорошо восстанавливаются небольшие фрагменты изображения, но недостатком метода являются трудности с воспроизведением текстуры.

## Глава 3. Практическая реализация

### 3.1 Структура программного обеспечения

Для программной реализации была выбрана среда MATLAB, так как она предоставляет средства для работы с изображениями и нейронными сетями. Использовались пакеты MATLAB Image Processing Toolbox и Deep Learning Toolbox. Кроме того, для некоторых задач был использован встроенный C компилятор.

Был реализован программный комплекс, в котором пользователь может загрузить изображение, указать с помощью мышки объект для удаления, после выбранным способом производилось восстановление фотографии. В результате работы пользователь получает восстановленное изображение и при желании может посмотреть оценку результатов.

Интерфейс рабочего приложения приведен на рис. 5.

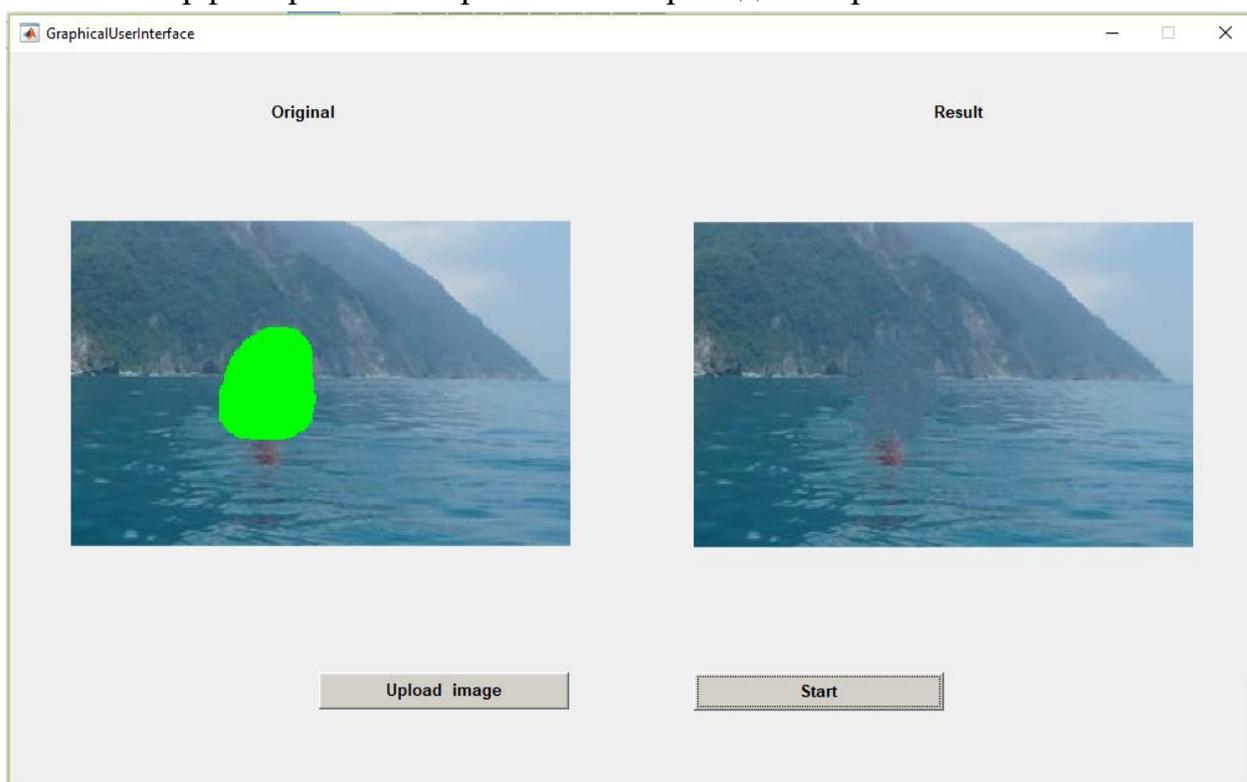


Рис. 5: Интерфейс и результат работы приложения

### 3.2 Сравнение методов

Проведем ряд экспериментов. Рассмотрим 2 два типа повреждений:

1 – прямоугольная область  $8 \times 32$  пикселя;

2 – область произвольной формы.

Для каждого типа проводилась процедура восстановления несколькими способами. Результаты экспериментов приведены в таблицах (1–3).

Для оценивания использовалось несколько метрик: mean squared error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM).

Одна из наиболее часто используемых ошибок MSE — среднеквадратическая ошибка, она не имеет физического смысла, чем меньше, тем лучше.

Ошибка PSNR — отношение сигнал–шум, описывает соотношение между максимумом возможного значения сигнала и мощностью шума, искажающего значения сигнала. Обычно варьируется от 20 до 40, но абсолютного значения не имеет.

Формула для расчета выглядит следующим образом:

$$PSNR = 10 \log \frac{MAX^2}{MSE}, \quad (17)$$

где  $MAX$  – максимальное значение, принимаемое пикселем изображения.

Однако из-за особенностей восприятия человека, эти метрики не могут адекватно характеризовать результаты восстановления. Поэтому были изобретены более сложные метрики, учитывающие эти особенности, например, SSIM — индекс структурного сходства, который оценивает сходство структуры в целом. Ее значение лежит в промежутке от -1 до 1, причем 1 достигается при полном совпадении изображений. Рассчитывается метрика следующим образом:

$$SSIM = \frac{(2\mu_a\mu_b+c_1)(2\sigma_{ab}+c_2)}{(\mu_a^2+\mu_b^2+c_1)(\sigma_a^2+\sigma_b^2+c_2)}, \quad (18)$$

Где  $\mu_a$  – среднее значение для первого изображения,  $\mu_b$  – среднее для второго,  $\sigma_a$  – среднеквадратическое отклонение для первого изображения,  $\sigma_b$  – среднеквадратическое отклонение для второго,  $\sigma_{ab}$  – ковариация.  $c_1$  и  $c_2$  – поправочные коэффициенты, которые нужны вследствие малости знаменателя.

	1 тип	2 тип
MSE	36.2478	466.6571
PSNR	40.0665	26.7069
SSIM	0.9789	0.9689

Таблица 1: Оценка точности: метод карты Кохонена

	1 тип	2 тип
MSE	17.4639	450.6015
PSNR	45.5682	23.3170
SSIM	0.9989	0.9519

Таблица 2: Оценка точности: метод Криминиси

	1 тип	2 тип
MSE	46.5186	147.0146
PSNR	34.4837	35.9794
SSIM	0.9011	0.9463

Таблица 3: Оценка точности: метод Мамфорда – Шаха

Рис. А7 в приложении А иллюстрирует результаты работы для всех представленных выше методов.

Первые два метода показали неплохие, сопоставимые результаты. Результаты работы третьего метода визуально оцениваются хуже из-за размытия заполненной области.

### 3.3 Сравнение с медианным фильтром

Могло показаться, что задачи шумоподавления и восстановления изображения близки, однако они отличаются друг от друга. Алгоритмы врисовки пытаются воссоздать поврежденные области, используя оставшуюся информацию. Задача алгоритма шумоподавления состоит в восстановлении исходного изображения путем построения моделей шума и

статистических оценок. Алгоритмы шумоподавления напрямую, как правило, неприменимы при решении задачи. Примеры различных исходных изображений для задач шумоподавления и восстановления приведены на рис.6.

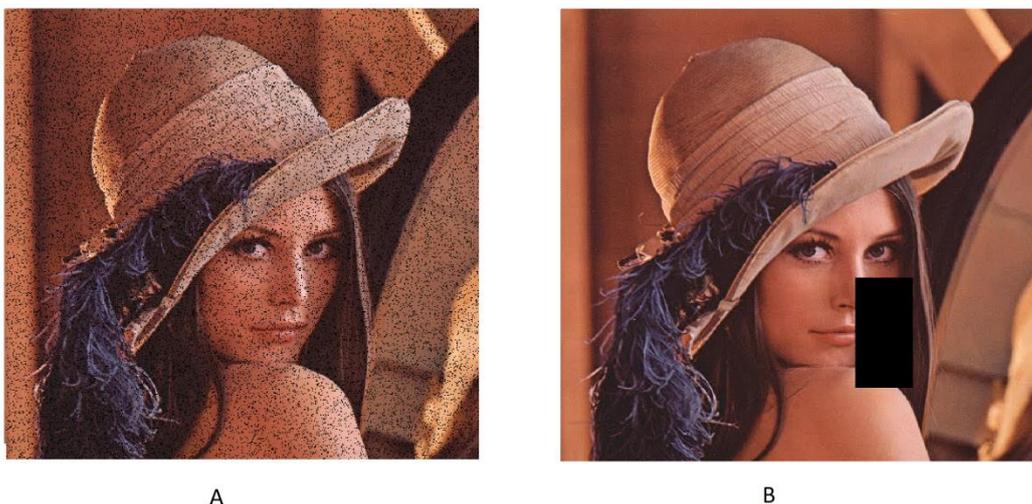


Рис. 6: Различные исходные изображения: часть А: исходное изображение для задачи шумоподавления, зашумленное на 10%; часть В: исходное изображение с поврежденной областью, которую требуется восстановить

Медианный фильтр — один из наиболее часто применяемых фильтров для обработки фотоизображений. Его преимущества состоят в простоте реализации, он хорошо обрабатывает мелкие рассредоточенные вкрапления, однако пропускает более крупные области сосредоточенных помех и при увеличении окна фильтра вызывает размытие изображения. Данный фильтр показывает хорошие результаты при удалении импульсного шума, но не совсем подходит для подавления белого и гауссовского шумов.

Медианный фильтр — это скользящее окно заданной размерности, которое получает массив значений цветов пикселей внутри него, сортирует, на выход выдает среднее значение — медиану. Рис. 7 иллюстрирует этот процесс.

Медианой дискретной последовательности  $a_1, a_2, \dots, a_N$  для нечетного  $N$  является тот ее элемент, для которого существуют  $(N - 1)/2$  элементов, меньших или равных ему по величине, и  $(N - 1)/2$  элементов, больших или

равных ему по величине. Для четного  $N$  за медиану можно принять среднее арифметическое двух чисел, стоящих посередине.

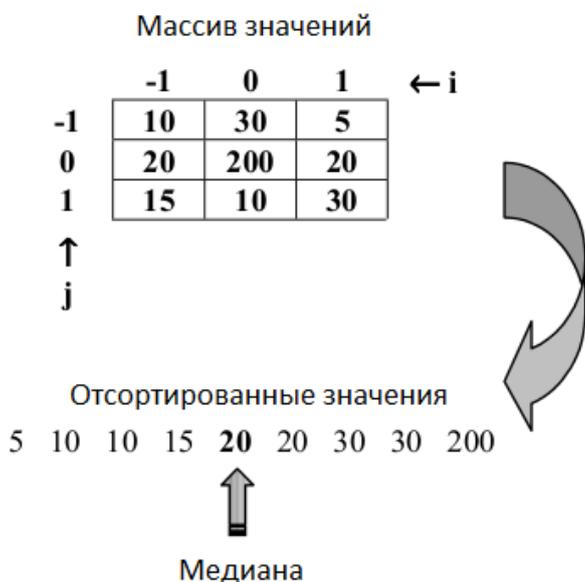


Рис. 7: Иллюстрация работы медианного фильтра для размера окна  $3 \times 3$

Базовый алгоритм медианного фильтра предполагает наличие только одного цветового канала, поэтому простейший вариант использования фильтра для RGB изображений — отдельно обрабатывать каждый массив, соответствующий одной компоненте RGB [44].

Есть несколько подходов применения медианного фильтра. Первый из них предполагает начать с окна  $3 \times 3$ , расширить окно можно до тех пор, пока фильтрация больше улучшает изображение, чем ухудшает его. Вторым вариантом — использовать фильтр итерационно, при необходимости меняя размер окна.

Помимо использования непосредственно фильтров для шумов, можно применить другой подход. Было высказано предположение о существовании закономерностей, связывающих каждый элемент матрицы с его ближайшими соседями, которые являются свойствами всего изображения в целом. Проанализировав информацию о взаимном расположении пикселей с различными значениями цвета, можно попытаться найти эти закономерности. Зная их, можно выявить те пиксели, которые выпадают из найденных зависимостей. С большой вероятностью они окажутся поврежденными

шумом [45]. Это предположение дает обоснование для использования алгоритмов врисовки для фильтрации шумов.

Были рассмотрены три типа повреждений:

- 1 – 100 случайных окрашенных пикселей;
- 2 – наложенный на изображение шум, повреждено 25% изображения;
- 3 – наложенный на изображение шум, повреждено более 80% изображения.

Сравним результаты работы 2 методов: медианного фильтра и самоорганизующейся карты Кохонена, они приведены в таблицах 4 и 5.

	1 тип	2 тип	3 тип
MSE	164.1984	92.2594	1.5623e+03
PSNR	16.2578	30.8658	18.9463
SSIM	0.8795	0.8723	0.2288

Таблица 4: Оценка точности: метод карта Кохонена

	1 тип	2 тип	3 тип
MSE	100.7468	906.5695	1.9558e+04
PSNR	30.8535	20.9346	26.3529
SSIM	0.8794	0.4944	0.0352

Таблица 5: Оценка точности: метод медианный фильтр

Рис. А8 приложения А иллюстрирует применение 2 разных методов для восстановления фотографии, где было повреждено 25% пикселей. На рис. А9 из приложения А — применение методов для восстановления фотографии с более чем 80% повреждений.

Визуально алгоритм Кохонена неплохо справляется с задачей фильтрации, даже для очень поврежденных данных получен приемлемый результат.

## Выводы

Из полученных в ходе экспериментов количественных и визуальных оценок можно сделать вывод, что метод карт Кохонена почти настолько же эффективен, как и методы, классически применяющиеся для задач восстановления изображений, а при визуальном сравнении демонстрирует даже лучший результат, что также подтверждает и метрика SSIM, которая наиболее точно из представленных характеризует восприятие человека.

Кроме того, он показал хорошие результаты при использовании для задач шумоподавления. Его преимуществами в сравнении с медианным фильтром оказались точное отображение многообразия цветов и сохранение четких границ объектов даже для сильно поврежденных изображений.

## Заключение

В данной работе были рассмотрены различные виды дефектов и причины, их вызывающие, проведен обзор существующих методов восстановления фотоизображений, их особенностей и недостатков. Были подробно рассмотрены несколько разнотипных методов и проведен ряд экспериментов, доказывающих возможность и результативность использования карт Кохонена для задач реконструкции и шумоподавления. Несмотря на некоторые недостатки, такие как достаточно долгое время работы алгоритма и необходимость в точной настройке параметров, метод дает хорошие результаты по различным критериям оценивания. Кроме того, имеется потенциал для улучшения эффективности и быстродействия метода.

## Список литературы

1. Mandelli S., Bondi L., Lameri S., Lipari V., Bestagini P., Tubaro S. Inpainting-Based Camera Anonymization // IEEE International Conference on Image Processing (ICIP), 2017. P. 1522 – 1526.
2. Винклер Г. Анализ изображений, случайные поля и методы Монте-Карло. Математические основы. Новосибирск: Филиал "Гео" Издательства СО РАН, 2002. 343 с.
3. Afreen N., Singh S., Kumar S. Semantic Deep Image Inpainting// International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, P.1190 – 1195.
4. Carrera P. V., Sintes J., Ballester C. Semantic Image Inpainting through Improved Wasserstein Generative Adversarial Networks// 14th International Conference on Computer Vision Theory and Applications, 2019. P. 249 – 260.
5. Dale K., Johnson M.K., Sunkavalli K., Matusik W., Pfister H. Image Restoration using Online Photo Collections// IEEE 12th International Conference on Computer Vision (ICCV), 2009. P. 2217 – 2224.
6. Hays J., Efros A.A. Scene completion using millions of photographs //ACM Transactions on Graphics (SIGGRAPH), 2007. Vol. 26. № 3.
7. Kerzhner Y., Elad A., Romano Y. Image Inpainting via Pre-trained Classification-CNN  
[https://www.researchgate.net/publication/325470762\\_Image\\_Inpainting\\_via\\_Pre-trained\\_Classification-CNN](https://www.researchgate.net/publication/325470762_Image_Inpainting_via_Pre-trained_Classification-CNN)
8. Sidorov O., Hardeberg J. Y. Deep Hyperspectral Prior: Denoising, Inpainting, Super-Resolution  
[https://www.researchgate.net/publication/330844782\\_Deep\\_Hyperspectral\\_Prior\\_Denoising\\_Inpainting\\_Super-Resolution](https://www.researchgate.net/publication/330844782_Deep_Hyperspectral_Prior_Denoising_Inpainting_Super-Resolution)
9. Ulyanov D., Vedaldi A., and Lempitsky V. Deep image prior, 2017.

10. Манжула В.Г., Федяшов Д.С. Нейронные сети Кохонена и нечеткие нейронные сети в интеллектуальном анализе данных// *Фундаментальные исследования*, 2011. №4. С. 108 – 114.
11. Болгов А. Н. Мультиагентный подход в реконструкции текстуры на изображениях// *Вестник СибГАУ*, 2014. №5(57). С. 44 – 50.
12. Болгов А. Н. Применение карт Кохонена для восстановления слабоструктурированных изображений// *Решетневские чтения*, 2014. Т.2. С. 227 – 229.
13. Кохонен Т. Самоорганизующиеся карты. М.: БИНОМ. Лаборатория знаний, 2008. 655 с.
14. Ларионов И. Б., Карты Кохонена как способ восстановления мультимедийной информации// *Журнал радиоэлектроники*, 2010. №10.
15. Фаворская М.Н., Болгов А.Н. Использование карт Кохонена для восстановления фоновой текстуры на изображении// *Техническое зрение*, 2014. №1. С.9 – 17.
16. Болгов А. Н., Баранов Р. П., Казмирук Е. С. Многомерные сети Кохонена для реконструкции видеопотока// *Решетневские чтения*, 2015. Т.2. С. 18 – 19.
17. Karpagam V., Sanjushri. R., Sujitha.S., Sumithra.M. Digital restoration of torn films using filtering techniques, 2014. Vol.1. Iss.1.
18. Elazary L., Itti L. Interesting objects are visually salient // *Journal of Vision*, 2008. Vol.8. №.3. P. 1 – 15.
19. Girod B. What's wrong with mean-squared error // *Digital Images and Human Vision*. MA, USA:MIT Press Cambridge, 1993. P. 207 – 220.
20. Itti L., Baldi P. F. Bayesian surprise attracts human attention // *Advances in Neural Information Processing Systems*, 2006. Vol. 19. P. 547 – 554.
21. Щербаков О.В. Реконструкция изображений в отсутствующих или повреждённых областях на основе критерия алгоритмической вероятности: дис. ... канд. тех. наук: 05.13.01/ О.В. Щербаков. СПб., 2015. 128 с.

22. Chan T.F., Kang S.H. Error analysis for image inpainting // *Journal of Mathematical imaging and Vision*, 2006. Vol. 26. №.1-2. P. 85 – 103.
23. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с.
24. Саблина В.А. Разработка и исследование алгоритмов восстановления изображений методами секвентного анализа: автореф. дис. ... канд. тех. наук: 05.13.01/ В.А. Саблина. Рязань, 2009. 19 с.
25. Bertalmio M., Sapiro G., Caselles V., and Ballester C. Image inpainting// *Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, 2000, P. 417 – 424.
26. Chan T. F., Shen J. Morphologically invariant PDE inpaintings, UCLA CAM Report 01-15, 2001.
27. Kim J., Tsai A., Cetin M., Willsky A.S. A curve evolution-based variational approach to simultaneous image restoration and segmentation"// *IEEE ICIP*, 2002. Vol. 1, P. 109 – 112.
28. Schönlieb C.-B. Partial differential equation methods for image inpainting// Cambridge: Cambridge University Press, 2015. 250 p.
29. Criminisi A., Perez P., Toyama K. Region filling and object removal by exemplar-based image inpainting// *IEEE Transactions on Image Processing*, 2004. Vol. 13. Iss. 9. P. 1200 – 1212.
30. Hazra T. K., Mitra R., Bhattacharya J. A comparative study of Bertalmio and exemplar based image inpainting// *International Journal of Science and Research (IJSR)*, 2013. Vol.4. Iss.6. P.953 – 958.
31. Ardizzone E., Dindo H., Mazzola G. A knowledge based architecture for the virtual restoration of ancient photos// *Journal Pattern Recognition*, 2018. Vol. 74. Iss. C. P. 326 – 339.
32. Stanco F., Ramponi G., de Polo A. Towards the automated restoration of old photographic prints: a survey// *The IEEE Region 8 EUROCON*, 2003. Vol.2. P. 370 – 374.

33. Садыков С.С., Канунова Е.Е., Варламов А.Д. Автоматизированная реставрация изображений архивных текстовых и фотографических документов// Автоматизация и современные технологии, 2007. №7. С. 10 – 15.
34. Chang R.-C., Sie Y.-L., Chou S.-M., Shih T.K. Photo defect detection for image inpainting// Seventh IEEE International Symposium on Multimedia (ISM'05), 2005. P. 403 – 407.
35. Sun Y., Yu Y., Wang W. Moiré Photo Restoration Using Multiresolution Convolutional Neural Networks// IEEE Transactions on Image Processing, 2018. Vol.27. Iss. 8. P. 4160 – 4172.
36. Ястребов В. А., Веселов А.И., Гильмутдинов М. Р. Базовые методы и принципы построения алгоритмов восстановления регионов изображений// Информационно-управляющие системы, 2015. №5. С. 34 – 42.
37. Laube P., Grunwald M., Franz M. O., Umlauf G. Image inpainting for high-resolution textures using cnn texture synthesis [https://www.researchgate.net/publication/321719025\\_Image\\_Inpainting\\_for\\_High-Resolution\\_Textures\\_using\\_CNN\\_Texture\\_Synthesis](https://www.researchgate.net/publication/321719025_Image_Inpainting_for_High-Resolution_Textures_using_CNN_Texture_Synthesis)
38. Bertalmio M., Vese L., Sapiro G., Osher S. Simultaneous structure and texture image inpainting // IEEE Transactions on Image Processing, 2003. Vol. 12. №8. P. 882 – 889.
39. Sun J., Yuan L., Jia J., Shu H.-Y. Image completion with structure propagation // ACM Transactions on Graphics (TOG), 2005. Vol. 24. Iss. 3. P. 861 – 868.
40. Воронин В.В., Семенищев Е.А., Франц В.А., Свирин И.С. Восстановление искаженных пикселей изображений на основе текстурно-геометрической модели// Фундаментальные исследования, 2015. №12. С. 17 – 21.

41. Oliveira M. M., Bowen B., McKenna R., Chang Y.-S. Fast digital image inpainting// International Conference on. Visualization, Imaging and Image Processing (VIIP), 2001. P. 261 – 266.
42. Ларионов И.Б. Кластеризация матриц с пропусками как метод восстановления графической информации // Математические структуры и моделирование, 2009. Вып. 20. С. 97-106.
43. Бирюков Е.Д. Использование медианного фильтра в системе обработки изображений реалистичной компьютерной графики// Новые информационные технологии в автоматизированных системах, 2014. С. 216 – 220.
44. Белим С. В., Майоров-Зильбернагель А. О. Алгоритм поиска поврежденных пикселей и удаление импульсного шума на изображениях с использованием метода ассоциативных правил// Наука и Образование. МГТУ им. Н.Э. Баумана, 2014. № 12. С. 716 – 737.

# Приложение А

## Иллюстративный материал

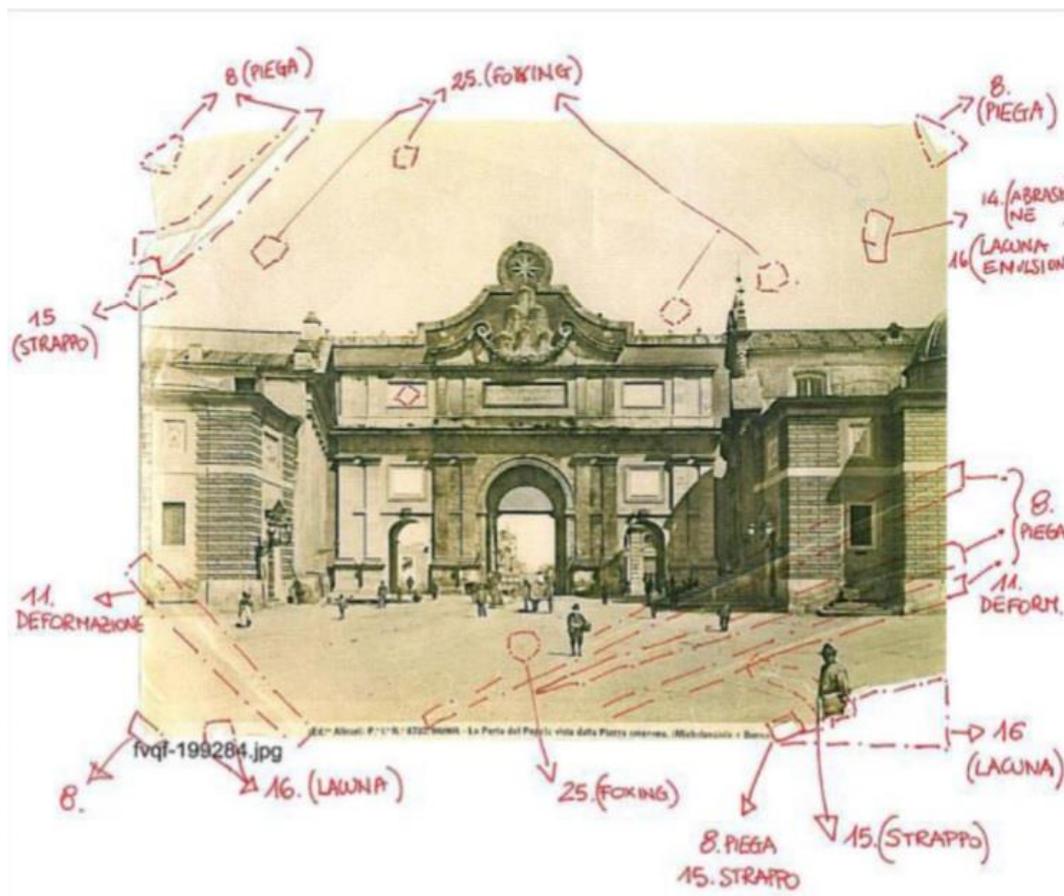


Рис. А1: Размеченная фотография для ручного восстановления



Рис. А2: Различные дефекты: часть А: пятна; часть В: загиб; часть С: выбеливание; часть D: фоксинг; часть Е: потертости; часть F: порча человеком; часть G: эффект муара; часть Н: смазанное изображение.



Рис. А3: часть А: исходное изображение; часть В: изображение, восстановленное предложенным способом; часть С: изображение, восстановленное вручную.

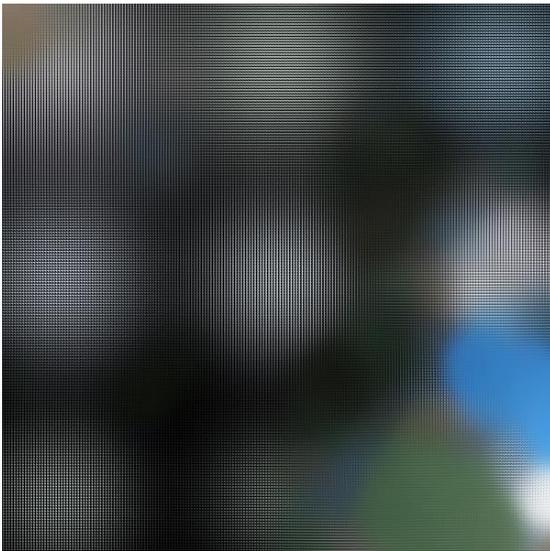


Рис. А4: Пример обученной карты Кохонена

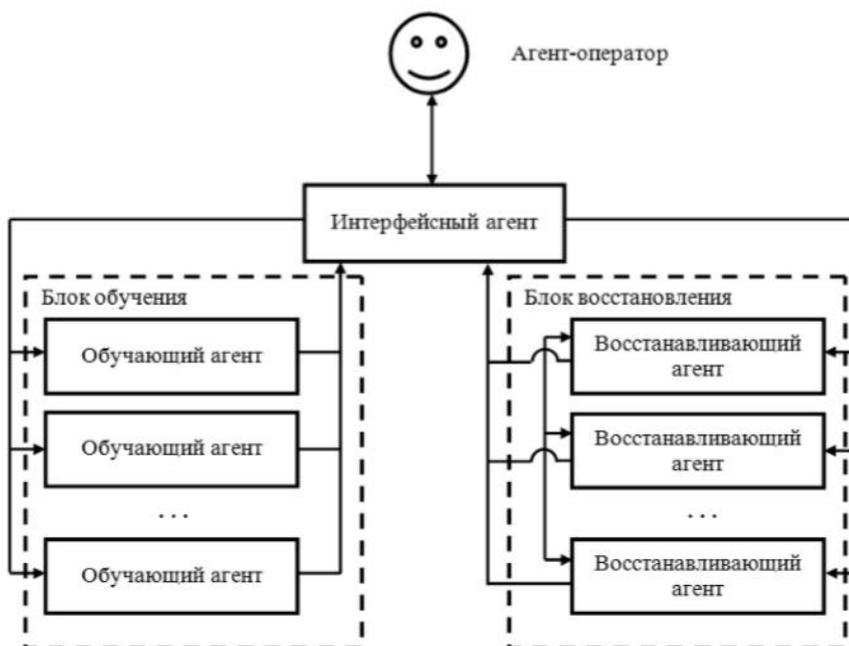


Рис. А5: Мультиагентный подход: Агент-оператор (Agent-Operator) – единственный агент-человек в системе. Интерфейсный агент (Interface Agent) – программный агент, обеспечивающий взаимодействие всех других агентов системы. Группа обучающих агентов (Learning Agent) выполняет процесс обучения карт Кохонена. Группа агентов восстановления (Inpainting Agent) производит реконструкцию текстур по обученным картам.

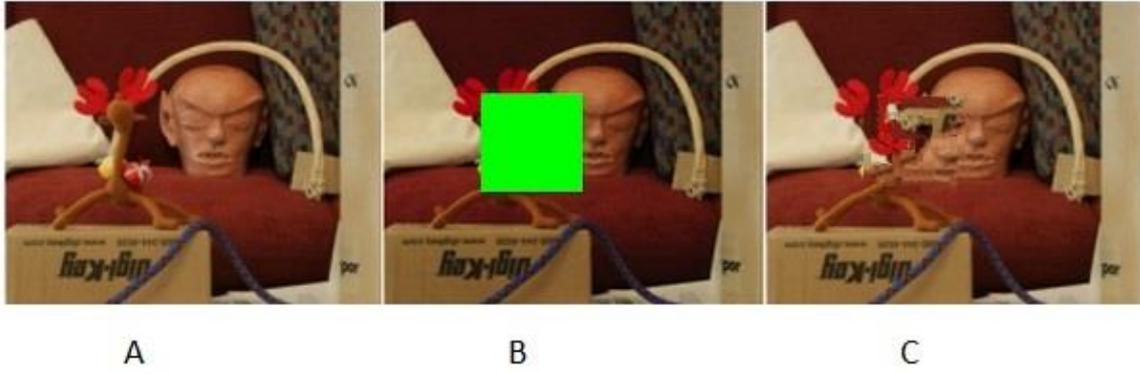
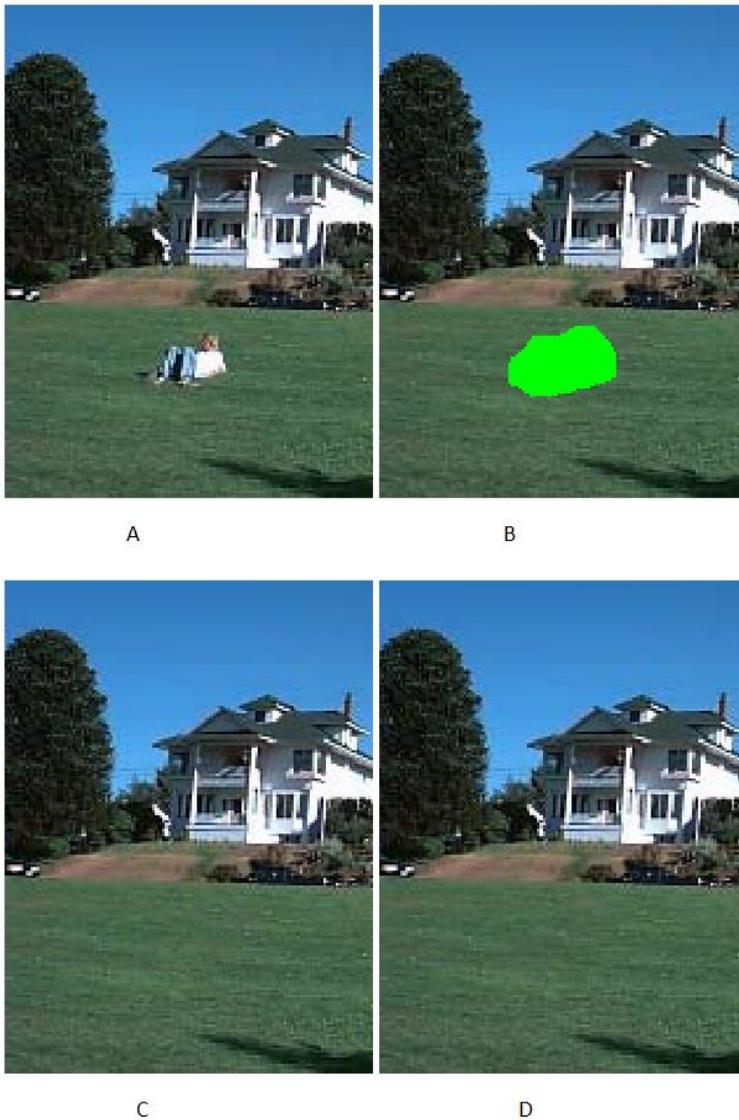


Рис. А6: Дефект метода Криминиси: часть А: исходное изображение; часть В: объект для удаления отмечен не полностью; часть С: возникшие деформации.





Е

Рис. А7: Результаты работы алгоритмов: часть А: исходное изображение; часть В: область, отмеченная для удаления; часть С: результат работы метода Криминиси; часть D: результат работы метода карты Кохонена; часть Е: Результат работы метода, основанного на решении дифференциальных уравнений в частных производных.



А



В



С



Д

Рис. А8: Результаты работы алгоритмов: часть А: изображение, зашумленное на 25%; часть В: восстановление по картам Кохонена; часть С : восстановление медианным

фильтром, окно  $8 \times 8$ ; часть D: восстановление медианным фильтром после 10 итераций, окно  $3 \times 3$ .



Рис. А9: Результаты работы алгоритмов: часть А: Изображение, зашумленное на 25%; часть В: восстановление по картам Кохонена; часть С : восстановление медианным фильтром, окно  $6 \times 6$ .