

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

На правах рукописи

Умнов Алексей Витальевич

**МЕТОДЫ РАЗРЕЖЕННЫХ ПРЕДСТАВЛЕНИЙ ДЛЯ
ОБНАРУЖЕНИЯ И ПОДАВЛЕНИЯ ЭФФЕКТА ЛОЖНОГО
ОКОНТУРИВАНИЯ НА ИЗОБРАЖЕНИЯХ**

Специальность 05.13.18 — математическое моделирование,
численные методы и комплексы программ

ДИССЕРТАЦИЯ

на соискание ученой степени

кандидата физико-математических наук

Научный руководитель:

доктор физико-математических наук

А. С. Крылов

Москва — 2018

Оглавление

Введение	4
1 Применение метода разреженных представлений для обнаружения эффекта ложного оконтуривания на изображениях	15
1.1 Эффект ложного оконтуривания	15
1.1.1 Причины возникновения эффекта ложного оконтуривания	15
1.1.2 Методы обнаружения эффекта ложного оконтуривания на изображениях .	22
1.2 Метод разреженных представлений	28
1.2.1 Задача разреженных представлений	28
1.2.2 Алгоритмы построения представлений	31
1.3 Обнаружение эффекта ложного оконтуривания на изображениях с помощью метода разреженных представлений	37
1.3.1 Связь эффекта ложного оконтуривания и размытия изображений	38
1.3.2 Обнаружение эффекта ложного оконтуривания на изображениях	41
2 Подавление эффекта ложного оконтуривания на изображениях	52
2.1 Методы подавления эффекта ложного оконтуривания на изображениях	52
2.2 Построение словарей в методе разреженных представлений	55
2.2.1 Задача построения словаря	55
2.2.2 Алгоритм построения словарей	57
2.2.3 Объединенные словари	59
2.2.4 Взаимная согласованность в методе разреженных представлений	62
2.3 Подавление эффекта ложного оконтуривания с помощью метода разреженных представлений	64

2.3.1	Подавление эффекта ложного оконтуривания на изображениях с помощью объединенных словарей	65
2.3.2	Влияние эффекта ложного оконтуривания и других искажений изображений на разреженность представлений	68
2.3.3	Подавление эффекта ложного оконтуривания на изображениях с помощью независимых словарей	73
2.3.4	Сравнение эффективности методов подавления эффекта ложного оконтуривания на изображениях	77
3	Программный комплекс	85
3.1	Вспомогательные модули	86
3.1.1	Модуль image.degrade	86
3.1.2	Модуль image.metrics	87
3.1.3	Модуль image.patches	90
3.1.4	Модуль sparse.algorithm	90
3.1.5	Модуль sparse.joint_coding	91
3.2	Реализация разработанных методов	92
3.2.1	Модуль ringing_detection	92
3.2.2	Модуль ringing_suppression	94
3.3	Графический интерфейс	96
	Заключение	100
	Список литературы	102

Введение

В связи с быстрым ростом производительности вычислительной техники цифровая обработка и анализ изображений в настоящее время играют большую роль во множестве приложений. Однако несовершенство формирующих и регистрирующих систем, а также некоторые алгоритмы обработки могут приводить к искажениям изображений, затрудняющим работу с ними [1, 2]. Для борьбы с подобными проблемами могут использоваться методы математического моделирования, позволяющие частично восстанавливать искаженные изображения. Актуальность проблемы подтверждается многочисленными исследованиями на эту тему в современной литературе [3–6].

Одним из подобных искажений является эффект ложного оконтуривания, связанный с эффектом Гиббса [7], и появляющийся при потере или искажении высокочастотной информации на изображении. Эффект ложного оконтуривания возникает в задачах обработки медицинских или аэрокосмических изображений и при сжатии изображений или видео [8–10]. Задача подавления этого эффекта важна в данных приложениях, так как часто в них отсутствует возможность получения изображения высокого качества, или же этот процесс является очень ресурсоемким. Также иногда требуется автоматическая оценка качества изображений, поэтому возникает задача обнаружения эффекта ложного оконтуривания.

Сложность данных задач обуславливается трудностью моделирования реальных изображений из-за наличия на них разнообразных деталей; при этом алгоритмы обработки должны сохранять содержимое и важные аспекты изображений, и не должны вносить новых искажений. Дополнительная трудность заключается в том, что эффект ложного оконтуривания может быть внешне похож на какие-то детали изображений, что усложняет как задачу обнаружения, так и задачу подавления этого эффекта. На данный момент задачи обнаружения и подавления эффекта ложного оконтуривания не являются решенными.

Целью диссертационной работы является разработка и программная реализация алгоритмов обнаружения и подавления эффекта ложного оконтуривания на изображениях на основе моделирования эффекта ложного оконтуривания с помощью эффекта Гиббса, и анализ

используемого в работе метода разреженных представлений [11] для изображений с эффектом ложного оконтуривания.

В **первой главе** рассматривается задача обнаружения эффекта ложного оконтуривания и разрабатывается алгоритм ее решения на основе моделирования эффекта ложного оконтуривания с помощью эффекта Гиббса. Для построения алгоритма используется метод разреженных представлений.

В первом параграфе описывается эффект ложного оконтуривания, который появляется при потере или искажении высокочастотной информации на изображении. Рассматривается связь этого эффекта с эффектом Гиббса, а также возможные источники его возникновения. Строится модель эффекта ложного оконтуривания на основе эффекта Гиббса. Приводится обзор методов для обнаружения эффекта ложного оконтуривания, в том числе рассматривается метод на основе вычисления полной вариации изображения, применимый к любому эффекту ложного оконтуривания вне зависимости от причины его возникновения [12]. Полной вариацией двумерной функции $I(x, y)$ (изображения) в прямоугольной области $x \in [a; b]$, $y \in [c; d]$ называется следующая величина:

$$\text{TV}(I) = \int_a^b \int_c^d \sqrt{I_x^2 + I_y^2} \, dx dy.$$

Также в целях повышения эффективности может использоваться полная вариация для одномерных функций, вычисляемая вдоль профилей контуров изображений. Пусть задана функция $f(x)$, тогда ее полной вариацией на отрезке $x \in [a; b]$ называется следующая величина:

$$\text{TV}(f) = \int_a^b |f'(x)| \, dx.$$

Основная идея методов на основе полной вариации заключается в том, что с ростом силы эффекта ложного оконтуривания на изображении растет и значение полной вариации этого изображения.

Во втором параграфе рассматривается метод разреженных представлений и его приложения для различных задач [11]. Основная идея метода заключается в моделировании обрабатываемого сигнала (в данном случае изображения или блока изображения) как линейной комбинации небольшого числа сигналов из заранее выбранного избыточного словаря. Обозначим сигнал как y , словарь как D , искомое представление как c и максимально допустимую ошибку представления как ε . Тогда, обозначая количество ненулевых элементов в векторе c как $\|c\|_0$, задача разреженного представления может формулироваться так:

$$\|c\|_0 \rightarrow \min \quad \text{при условии} \quad \|y - Dc\|_2^2 \leq \varepsilon.$$

Полученное представление c на практике обычно содержит всю наиболее важную информацию о сигнале. В общем случае эта задача является NP-полной, но существуют эффективные алгоритмы приближенного решения этой задачи. Одним из таких алгоритмов является «жадный» алгоритм, который последовательно добавляет наиболее подходящие коэффициенты в представление до тех пор, пока не будет достигнута требуемая точность. Избыточный словарь для представлений может выбираться различными способами в зависимости от задачи. Один распространенных подходов — конструировать словарь в виде объединения нескольких наборов базисных функций (например, базис дискретного преобразования Фурье и набор вейвлетов Хаара). Другой подход — построение словаря на наборе типичных изображений — подробно рассматривается во второй главе.

В третьем параграфе представлен алгоритм для обнаружения эффекта ложного оконтуривания на основе метода разреженных представлений [13]. Эффект ложного оконтуривания наиболее заметен вблизи резких контуров на изображении, поэтому основная идея алгоритма заключается в построении словаря, предназначенного для обработки блоков в этих областях. Показывается, что резкие контуры с эффектом ложного оконтуривания могут быть достаточно точно смоделированы в виде суммы идеального контура, размытого с гауссовским ядром, и чистого эффекта ложного оконтуривания, на рис. 1 изображено такое разложение для профилей контуров. При этом ширина осцилляций d чистого эффекта Гиббса и дисперсия гауссовского ядра σ связаны соотношением $\sigma \approx 0.336d$. Таким образом словарь составляется из двух частей. В первой части находятся блоки с идеальными контурами с различными степенями размытия, а во второй — блоки с чистым эффектом ложного оконтуривания с различной шириной осцилляции. Для каждой такой пары в словарь также добавляются их повернутые версии.

Алгоритм обнаружения эффекта ложного оконтуривания устроен следующим образом. Для каждого блока на изображении строится разложение этого блока по построенному словарю. Блоки, для которых ошибка разложения выше некоторого порога или у которых есть соседние с ними блоки со значительно меньшей ошибкой разложения, исключаются из рассмотрения. Это нужно для того, чтобы не рассматривать блоки без контуров и блоки, где контур не центрирован. Для оставшихся блоков рассматриваются их разложения по словарю и на основе них строится интегральная метрика силы эффекта ложного оконтуривания.

Предложенная метрика работает лучше метода на основе полной вариации на наборах изображений с произвольным эффектом ложного оконтуривания, и показывает хорошие результаты,

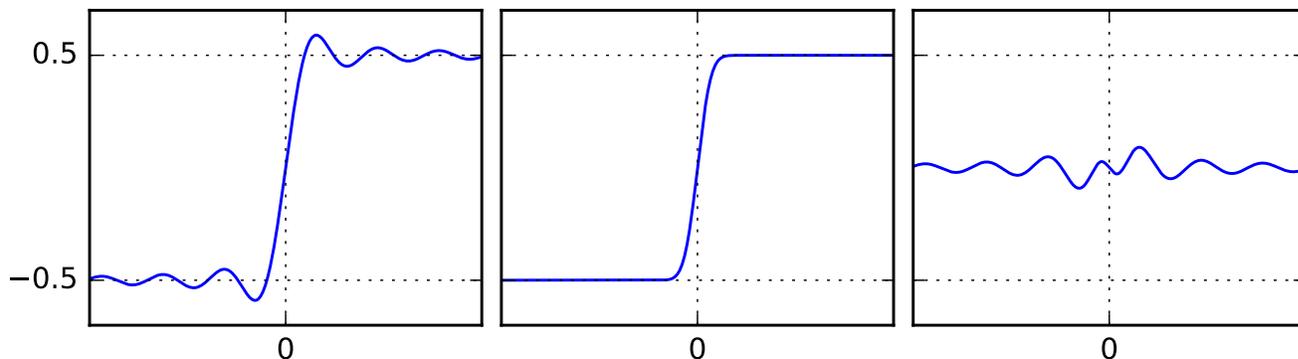


Рис. 1: Пример отделения чистого эффекта ложного оконтуривания от размытого контура: исходный контур с эффектом ложного оконтуривания (слева), ближайший размытый контур (в центре), чистый эффект ложного оконтуривания (справа).

сравнимые с другими методами, на наборах с эффектом ложного оконтуривания, получаемого в результате сжатия изображений алгоритмом сжатия JPEG2000.

Вторая глава посвящена задаче подавления эффекта ложного оконтуривания и построению алгоритмов ее решения на основе моделирования эффекта ложного оконтуривания с помощью эффекта Гиббса. Для построения алгоритмов используется метод разреженных представлений.

В первом параграфе приводится обзор методов подавления эффекта ложного оконтуривания. Существующие методы, работающие с чистым эффектом ложного оконтуривания, основаны на связи эффекта ложного оконтуривания и полной вариации [14]. Один из возможных методов подавления решает следующую задачу минимизации для восстановления изображения:

$$I^* = \arg \min_I (\|I - I_r\|_2^2 + \gamma \text{TV}(I)),$$

где I_r — входное изображение с эффектом ложного оконтуривания, I^* — восстановленное изображение, γ — регуляризирующий параметр.

Во втором параграфе рассматривается задача построения словаря по набору сигналов [11]. Эта задача расширяет метод разреженных представлений, позволяя строить словарь по набору типичных данных, что как правило позволяет получить более точную модель сигналов. Пусть матрица Y — набор сигналов (записанных в виде столбцов матрицы), матрица C — неизвестный набор представлений этих сигналов (дополнительно обозначим c_i — i -й столбец этой матрицы), матрица D — искомый словарь, и ε — максимально допустимая ошибка представления. Тогда задача построения словаря может формулироваться следующим образом:

$$\min_{D,C} \sum_i \|c_i\|_0 \quad \text{при условии} \quad \|Y - DC\|_2^2 \leq \varepsilon.$$

В общем случае эта задача является NP-полной, поэтому для ее решения используются приближенные алгоритмы. Один из них — алгоритм K-SVD — поочередно решает задачу относительно C с зафиксированным значением D (в этом случае задача превращается в набор независимых задач разреженного представления) и относительно отдельных столбцов D и соответствующих им элементов C с зафиксированными остальными элементами D и C (в этом случае задача решается с помощью сингулярного разложения матрицы ошибки). На практике такой подход дает хорошие результаты.

Построенные таким образом словари можно использовать для подавления искажений изображений. Для этого нужно построить словарь по набору естественных изображений без искажений, использовать этот словарь для построения представлений для изображений с искажениями, а потом восстановить изображение по представлению. Тогда разница между восстановленным изображением и исходным будет в большей мере состоять из чистых искажений (так как все остальные элементы хорошо представляется словарем). Такой подход позволяет удалять некоторые искажения (например, шум или отсутствующие пиксели), однако недостаточно хорошо работает для таких, для которых искажения соседних пикселей не являются независимыми (например, размытие или эффект ложного оконтуривания). Для решения подобных задач используется расширение описанного подхода — построение объединенных словарей, моделирующих произвольное преобразование изображений. По набору сигналов без искажений Y_1 и соответствующих им сигналов с искажениями Y_2 строятся два словаря D_1 и D_2 и общий набор представлений C , такие, что одно и то же представление c_i при умножении на словарь D_1 приближает i -й сигнал из Y_1 , а при умножении на словарь D_2 приближает i -й сигнал из Y_2 . Задачу построения объединенных словарей можно сформулировать так:

$$\min_{D_1, D_2, C} \sum_i \|c_i\|_0 \quad \text{при условии} \quad \|Y_1 - D_1 C\|_2^2 + \|Y_2 - D_2 C\|_2^2 \leq \varepsilon.$$

Известно, что данная задача с помощью преобразования сигналов сводится к исходной задаче обучения словарей и поэтому также может решаться с помощью алгоритма K-SVD. Для подавления рассматриваемого искажения на изображении y необходимо построить разреженное представление c для этого изображения по словарю D_2 и использовать это представление для получения приближения исходного изображения: $\hat{y} = D_1 c$.

Следующий раздел главы посвящен анализу разреженности представлений и взаимной согласованности для изображений [15]. Пусть x, y — действительные векторы, тогда их взаимной согласованностью $\mu(x, y)$ будем называть следующую величину:

$$\mu(x, y) \stackrel{\text{def}}{=} \frac{|(x, y)|}{\|x\|_2 \|y\|_2}.$$

Доказывается, что взаимная согласованность случайных векторов уменьшается с ростом их размерности n как функция $1/\sqrt{n}$. Исходя из этого показывается, что с ростом количества компонент в изображениях (например, с ростом их размера) уменьшается их средняя согласованность. На практике, как правило, уменьшение взаимной согласованности набора сигналов приводит к увеличению необходимого количества ненулевых коэффициентов в представлениях для этого сигнала.

В третьем параграфе предлагаются алгоритмы подавления эффекта ложного оконтуривания с помощью методов разреженных представлений.

Первый из методов основан на моделировании эффекта ложного оконтуривания с помощью объединенных словарей для обычных изображений и изображений с эффектом ложного оконтуривания [16]. Первым этапом алгоритма на основе объединенных словарей является преобразование изображения с эффектом ложного оконтуривания по описанной выше схеме с помощью объединенных словарей D_1 (для обычных изображений) и D_2 (для изображений с эффектом ложного оконтуривания). Такой алгоритм дает хорошие результаты, в частности, в отличие от методов, основанных на минимизации полной вариации, он хорошо сохраняет мелкие детали на изображении. Однако метод размывает резкие контуры на изображении, поэтому для улучшения результатов используется гибридный метод, объединяющий данный метод и метод на основе минимизации полной вариации. Для этого на изображении ищутся контуры, и для восстановления областей вблизи этих контуров используются результаты метода на основе полной вариации, а для всех остальных областей — результаты метода на основе объединенных словарей.

Далее производится анализ метода разреженных представлений при его применении к изображениям с эффектом ложного оконтуривания, шумом и размытием [17]. Доказывается, что моделирование чистого эффекта Гиббса для случайных векторов в среднем уменьшает их взаимную согласованность, то есть если X, Y — случайные нормально распределенные независимые векторы, а G — оператор, моделирующий эффект Гиббса, то

$$\mathbb{E}(\mu(G(X), G(Y))) > \mathbb{E}(\mu(X, Y)).$$

Исходя из этого показывается, что добавление размытия или эффекта ложного оконтуривания на изображении в среднем уменьшает количество коэффициентов, требуемых для представления изображения, а добавление шума на изображение увеличивает количество коэффициентов. На рис. 2 изображен график зависимости разреженности от допустимой ошибки представления для различных уровней эффекта ложного оконтуривания. Поэтому при использовании одинакового количества коэффициентов для изображений с эффектом ложного оконтуривания и без него ошибки восстановления неизбежны: либо для изображений с эффектом ложного оконтуривания будет использоваться больше коэффициентов, чем необходимо, и это приведет к неустойчивости представления, либо для изображений без эффекта ложного оконтуривания будет использоваться меньше коэффициентов, чем нужно, и это приведет к размытию изображения (восстанавливаются только крупные детали). Предполагается, что именно этот эффект приводит к размытию изображений в описанном выше методе.

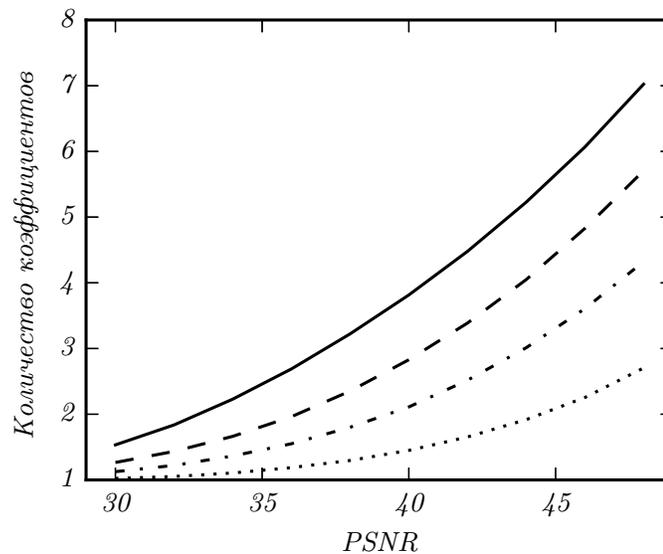


Рис. 2: Изменение разреженности при добавлении эффекта ложного оконтуривания; сплошная линия соответствует изображениям без искажения, штриховая, штрихпунктирная и точечная соответствуют изображениям с эффектом ложного оконтуривания с силой искажения $d = 1.5, 2, 3$ соответственно.

Второй метод подавления эффекта ложного оконтуривания основан на моделировании эффекта ложного оконтуривания с помощью словаря из двух независимых частей для устранения проблемы, описанной выше [17]. Первый словарь моделирует изображения без эффекта ложного оконтуривания и строится по набору обычных изображений. Второй словарь моделирует чистый

эффект ложного оконтуривания, который получается вычитанием изображения без эффекта ложного оконтуривания из соответствующего изображения с эффектом ложного оконтуривания. Для увеличения количества необходимых для представления коэффициентов перед обучением к изображениям применяется повышение резкости (можно показать, что оно, как и шум на изображении, увеличивает количество коэффициентов в представлении). Для удаления эффекта ложного оконтуривания с нового изображения необходимо применить к нему повышение резкости, построить его представление по объединенному словарю и откинуть коэффициенты, соответствующие чистому эффекту ложного оконтуривания.

Эффективность разработанных методов проверяется с помощью сравнения результатов с исходным изображением с помощью метрики близости SSIM на наборах изображений с различными уровнями эффектами ложного оконтуривания [18]. Показывается, что оба разработанных метода дают более качественные результаты, чем метод, основанный на полной вариации. При этом второй метод наиболее эффективен в областях контуров изображения, а первый метод хорошо работает во всех остальных областях изображения, в том числе для текстур с мелкими деталями. Также второй метод лучше работает при более высоких уровнях эффекта ложного оконтуривания, а первый метод — для более низких. Примеры обработки для всех методов изображены на рис. 3.

В **третьей главе** дается описание разработанного программного комплекса, включающего в себя реализацию разработанных алгоритмов обнаружения и подавления эффекта ложного оконтуривания, графический интерфейс для применения этих алгоритмов, а также набор вспомогательных модулей для работы с изображениями, эффектом ложного оконтуривания и методом разреженных представлений.

В **заключении** сформулированы основные результаты, полученные в диссертационной работе.

Научная новизна работы

1. Разработан алгоритм обнаружения эффекта ложного оконтуривания на изображениях на основе метода разреженных представлений.
2. Предложены два метода подавления эффекта ложного оконтуривания, применимые для различных параметров эффекта ложного оконтуривания.
3. Проведено исследование метода разреженных представлений для использования его при работе с эффектом ложного оконтуривания, шумом и размытием.

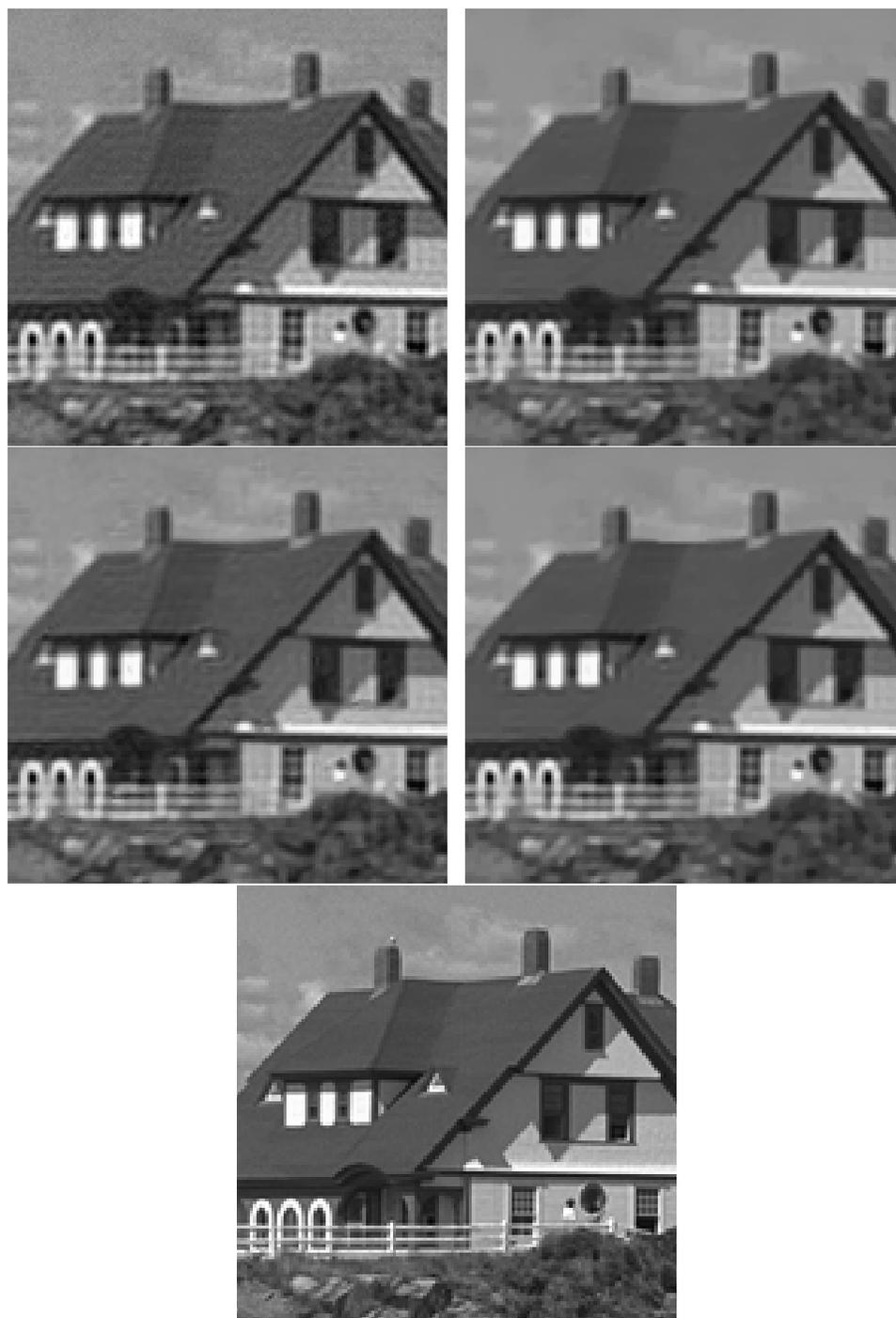


Рис. 3: Примеры обработки изображения с сильным эффектом ложного оконтуривания различными методами. В порядке слева-направо и сверху-вниз приведены: изображение с эффектом ложного оконтуривания ($d = 2.5$), результаты обработки методом на основе TV, результаты обработки методом на основе объединенных словарей, результаты обработки методом на основе независимых словарей, исходное изображение.

Теоретическая и практическая значимость работы

1. Создан программный комплекс для обнаружения и подавления эффекта ложного оконтуривания на изображениях.
2. Разработанные методы обнаружения и подавления эффекта ложного оконтуривания могут быть использованы как составная часть комплексных алгоритмов анализа и обработки изображений.

На защиту выносятся следующие результаты

1. Предложен и исследован метод математического моделирования эффекта ложного оконтуривания на изображениях с использованием метода разреженных представлений.
2. Разработаны и численно реализованы алгоритмы обнаружения и подавления эффекта ложного оконтуривания на изображениях.
3. Создан программный комплекс для обнаружения и подавления эффекта ложного оконтуривания на изображениях.

Степень достоверности результатов

Достоверность изложенных в работе математических утверждений обусловлена строгостью формулировок задач и математических доказательств. Проведен анализ работы предложенных методов для обнаружения и подавления эффекта ложного оконтуривания на изображениях, а также сравнение качества работы методов с другими методами решения поставленной задачи. Оценка качества производилась с помощью объективных метрик оценки качества изображений и общепринятых методов оценки качества алгоритмов обнаружения и подавления искажений на изображениях.

Апробация работы

Основные результаты докладывались и обсуждались на:

- 11-й международной конференции по анализу и распознаванию изображений («International Conference on Image Analysis and Recognition», ICIAR), Виламура, Португалия, 2014 [19];

- 12-й международной конференции по обработке сигналов («International Conference on Signal Processing», ICSP), Ханчжоу, Китай, 2014 [13];
- Международной конференции по продвинутым концепциям интеллектуальных систем технического зрения («Advanced Concepts of Intelligent Visual Systems», ACIVS), Катания, Италия, 2015 [16];
- Конференции «Тихоновские чтения 2015», ВМК МГУ имени М. В. Ломоносова (Москва, 2015);
- Заседание кафедры математической физики факультета ВМК МГУ имени М. В. Ломоносова, Москва, 2015 г.
- Международный научно-исследовательский семинар «Анализ и понимание изображений (Математические, когнитивные и прикладные проблемы анализа изображений и сигналов)» ФИЦ ИУ РАН, Москва, 2017.
- Международной конференции по применению разреженных представлений для приложений в мультимедиа, биомедицине и коммуникациях («Compressive sensing based trends applied to: multimedia, biomedicine and communications»), Будва, Черногория, 2017.

Глава 1

Применение метода разреженных представлений для обнаружения эффекта ложного оконтуривания на изображениях

В данной главе рассматривается задача обнаружения эффекта ложного оконтуривания на изображениях. В начале главы делается введение в эффект ложного оконтуривания, описываются механизмы и причины его возникновения. Рассматриваются другие работы, предлагающие методы обнаружения эффекта ложного оконтуривания. Далее разбирается разработанный метод обнаружения эффекта ложного оконтуривания, который основан на методе разреженных представлений. Сначала дается общее введение в метод разреженных представлений: описываются задачи, которые ставятся в этом подходе и алгоритмы их решения. Потом приводится исследование устройства контуров на изображениях с эффектом ложного оконтуривания, строится модель контуров с эффектом ложного оконтуривания, и на основе нее предлагается метод обнаружения эффекта ложного оконтуривания. Приводятся результаты экспериментов с этим методом и сравнение качества его работы с другими методами.

1.1 Эффект ложного оконтуривания

1.1.1 Причины возникновения эффекта ложного оконтуривания

Эффектом ложного оконтуривания на изображении называется искажение, которое возникает при искажении или потере высокочастотной информации. Этот эффект выражается в виде осцилляций вблизи резких контуров на изображении и может появляться в результате работы

алгоритмов обработки изображений и видео [8, 9, 20], а также на изображениях магнитно-резонансной томографии [10].

Алгоритмами обработки изображений, в результате которых может появиться эффект ложного оконтуривания, являются увеличение резкости изображения, подавление шума на изображении, повышение разрешения изображения. Наиболее часто встречающейся причиной возникновения эффекта ложного оконтуривания является сжатие изображений и видео [21–23].

На рис. 1.1 приведены изображения с эффектом ложного оконтуривания, возникающим на практике [24].

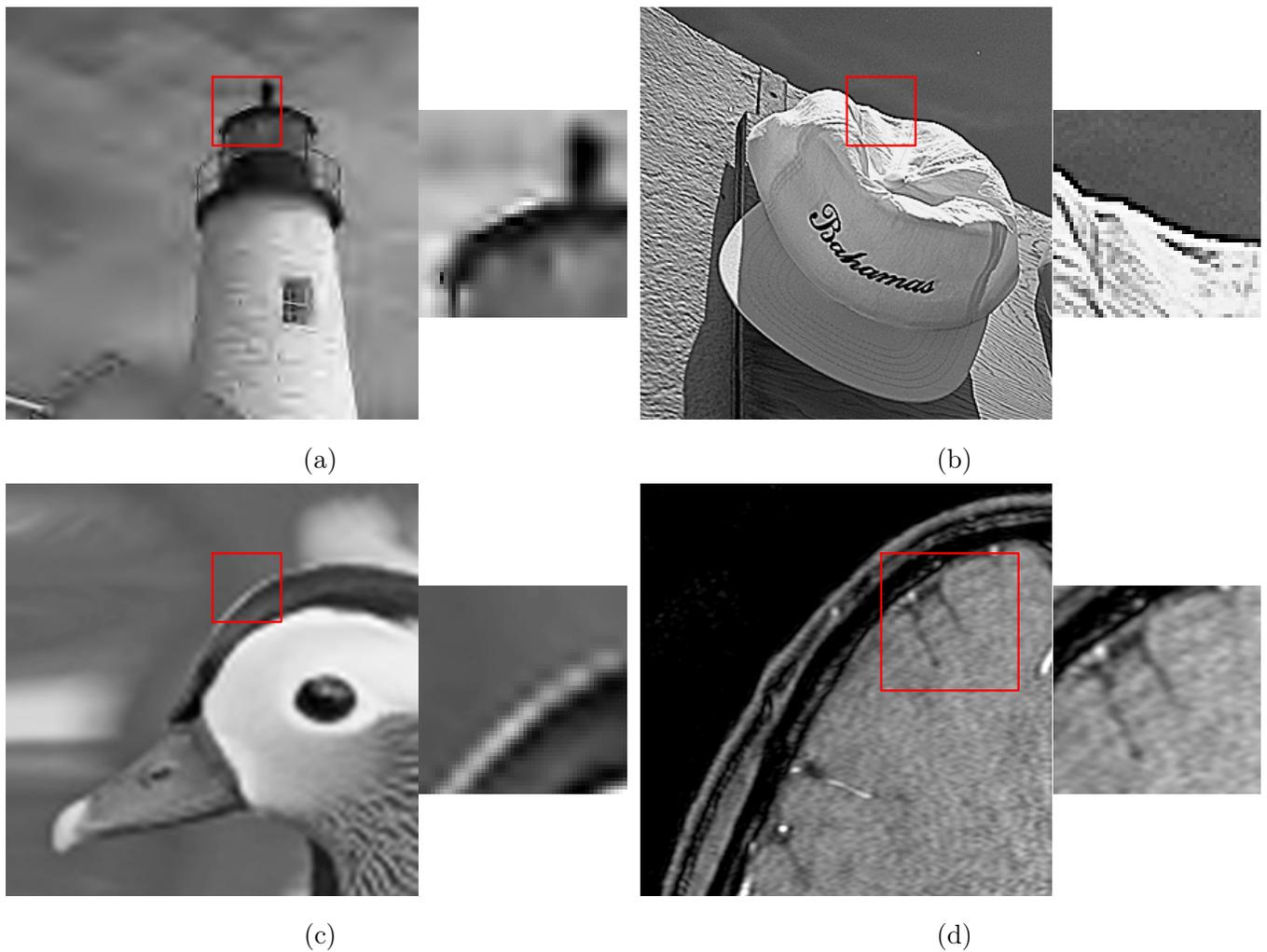


Рис. 1.1: Примеры эффекта ложного оконтуривания, возникающего а) в результате сжатия изображения, б) в результате повышения резкости изображения, с) в результате повышения разрешения изображения, д) на изображении магнитно-резонансной томографии.

В основе эффекта ложного оконтуривания лежит эффект Гиббса, отражающий поведение частичных сумм рядов Фурье при аппроксимации функции в точках разрыва [7, 25].

Обозначим операторы одномерного преобразования Фурье и обратного к нему через \mathcal{F}_1 и \mathcal{F}_1^{-1} соответственно:

$$\mathcal{F}_1(f)(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)e^{-ix\omega} dx,$$

$$\mathcal{F}_1^{-1}(f)(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)e^{ix\omega} dx.$$

Рассмотрим функцию с разрывом в точке $x = 0$:

$$f(x) = \begin{cases} \frac{1}{2}, & \text{если } x > 0; \\ -\frac{1}{2}, & \text{если } x \leq 0. \end{cases} \quad (1.1)$$

Эффект Гиббса можно моделировать с помощью частичных сумм ряда Фурье (что соответствует удалению части высоких частот в частотной области Фурье). Для этого введем параметр d (его смысл будет описан позже) и соответствующий ему частотный фильтр $l_d(\omega)$, удаляющий все частоты выше, чем $\omega_d = \frac{1}{2d}$:

$$l_d(\omega) = \begin{cases} 1, & \text{если } \omega \in [-\omega_d, \omega_d]; \\ 0, & \text{иначе.} \end{cases} \quad (1.2)$$

Далее, введем оператор

$$R_d(f) = \mathcal{F}_1^{-1}(l_d \cdot \mathcal{F}_1(f)),$$

который добавляет одномерный эффект ложного оконтуривания силы d к сигналу.

На рис. 1.2 изображен пример одномерного разрывной функции ($f(x)$) и ее приближения с использованием частичной суммы ряда Фурье ($R_d(f)$ для некоторого d), то есть с потерей высокочастотной информации. На приближенной функции наблюдается ярко выраженный эффект Гиббса.

Введем теперь понятие контура. Контурами как правило называют точки резкой смены значений сигнала. Для одномерных сигналов такие точки можно находить как точки с высоким значением производной, а для двумерных сигналов (например, изображений) — как точки с высоким значением модуля градиента. Также для лучшей локализации контуров обычно рассматривают только локальные максимумы производной или градиента. Одним из широко

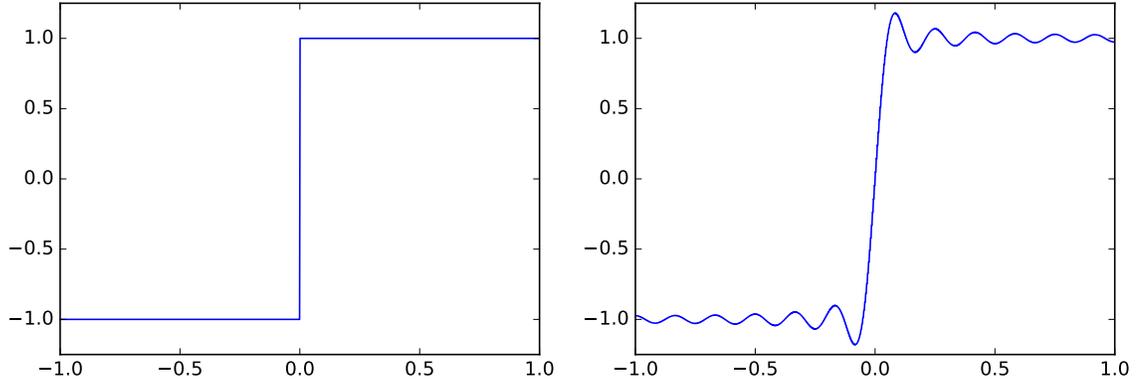


Рис. 1.2: Пример эффекта Гиббса для разрывной функции. Слева — исходная функция, справа — функция после удаления части высокочастотной информации.

используемых алгоритмов, использующим эти идеи, является алгоритм нахождения контуров Канни [26]. В данной работе в качестве определения контуров на изображении будем использовать результаты работы этого алгоритма.

На практике большинство профилей контуров не представляются как разрывы функции, вместо этого на них происходит быстрое изменение значения функции. Аппроксимация функции в точках быстрого изменения ее значений может приводить к аналогичному эффекту. Подобные контуры можно моделировать применением размытия к разрывной функции. Обозначим центрированную в нуле Гауссову функцию с дисперсией σ как $G_\sigma(x)$:

$$G_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}.$$

Тогда размытый контур моделируется следующим образом:

$$f_\sigma(x) = (f * G_\sigma)(x), \quad (1.3)$$

где $f * g$ — операция свертки. На рис. 1.3 изображен пример одномерного размытого контура ($f_\sigma(x)$ для некоторого σ) и его приближения с использованием частичной суммы ряда Фурье ($R_d(f_\sigma)$ для некоторого d), то есть с потерей высокочастотной информации. Видно, что на втором сигнале присутствуют ярко выраженные осцилляции, похожие по структуре на эффект Гиббса.

Эффект ложного оконтуривания для изображений моделируется аналогичным образом. Определим низкочастотный фильтр $l_d(\omega_x, \omega_y)$ как

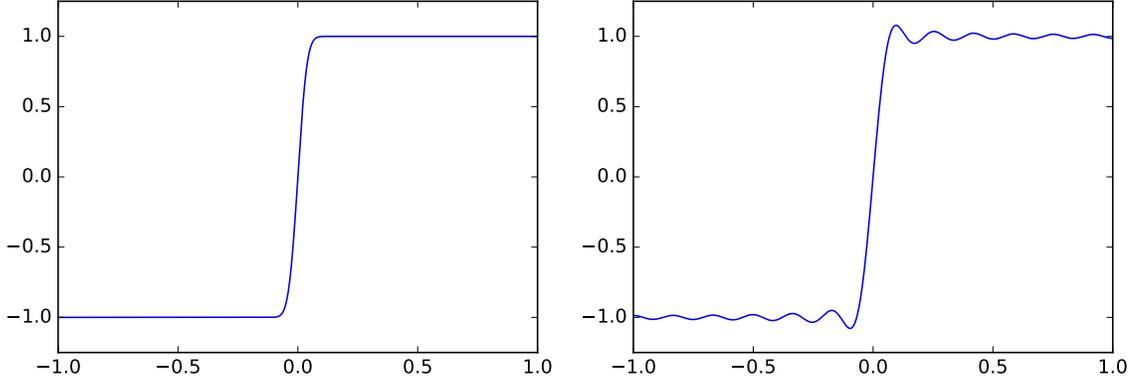


Рис. 1.3: Пример эффекта ложного оконтуривания для одномерного размытого контура. Слева — исходный контур, справа — контур после удаления части высокочастотной информации.

$$l_d(\omega_x, \omega_y) = \begin{cases} 1, & \text{если } \sqrt{\omega_x^2 + \omega_y^2} \leq \omega_d; \\ 0, & \text{иначе.} \end{cases}$$

Пусть $I(x, y)$ — некоторое изображение, \mathcal{F}_2 и \mathcal{F}_2^{-1} — операторы двумерного преобразования Фурье и обратного к нему соответственно:

$$\mathcal{F}_2(I)(\omega_x, \omega_y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y) e^{-i(x\omega_x + y\omega_y)} dx dy,$$

$$\mathcal{F}_2^{-1}(I)(\omega_x, \omega_y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y) e^{i(x\omega_x + y\omega_y)} dx dy.$$

Введем оператор

$$R_d(I) = \mathcal{F}_2^{-1}(l_d \cdot \mathcal{F}_2(I)), \quad (1.4)$$

который создает эффект ложного оконтуривания силы d на изображении.

На рис. 1.4 показан пример изображения $(I(x, y))$ и эффекта ложного оконтуривания для этого изображения $(R_d(I))$ для некоторого d . Видно, что на втором изображении присутствуют ярко выраженные осцилляции, аналогичные одномерному случаю.

Рассмотрим также альтернативную форму оператора R_d , дающую лучшее представление об источнике осцилляций в сигналах с эффектом ложного оконтуривания [7]. Для простоты рассмотрим одномерный случай, тогда оператор можно преобразовать следующим образом:



Рис. 1.4: Пример эффекта ложного оконтуривания для изображения. Слева — исходное изображение, справа — изображение после удаления части высокочастотной информации.

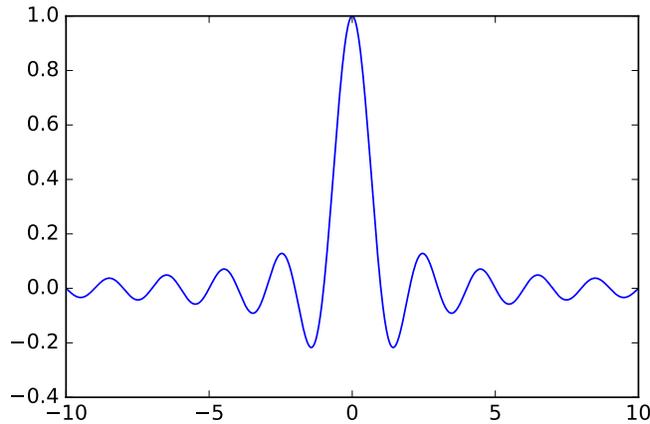
$$R_d(f) = \mathcal{F}_1^{-1}(l_d \cdot \mathcal{F}_1(f)) = \sqrt{2\pi} \mathcal{F}_1^{-1}(l_d) * \mathcal{F}_1^{-1}(\mathcal{F}_1(f)) = d \operatorname{sinc}\left(\frac{d}{2\pi}x\right) * f, \quad (1.5)$$

где

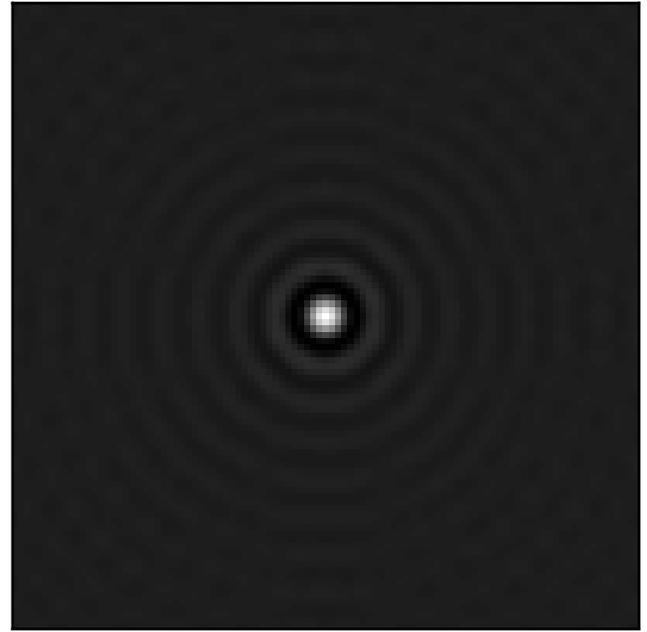
$$\operatorname{sinc}(x) = \frac{\sin \pi x}{\pi x}.$$

Мы получили, что оператор производит свертку исходного сигнала с осциллирующей функцией $\operatorname{sinc}(x)$. На рис. 1.5 приведен график функции $\operatorname{sinc}(x)$ в одномерном случае и изображение аналогичной функции в двухмерном случае. При свертке в областях плавных изменений сигнала эта функция сглаживается, и осцилляции не возникают, а точки резких изменений (контуры) вызывают эффект ложного оконтуривания. Параметр d задает долю удаляемых частот, и с его увеличением усиливается эффект ложного оконтуривания. Из формулы (1.5) также видно, что ширина осцилляций эффекта ложного оконтуривания и амплитуда осцилляций пропорциональны этому параметру.

Отметим, что на практике сигналы и изображения представляются в виде набора дискретных отсчетов, поэтому вместо обычного преобразования Фурье используется дискретное преобразование Фурье. Также вместо дискретного преобразования Фурье можно рассматривать дискретное



(a)



(b)

Рис. 1.5: График функции $\text{sinc}(x)$ в одномерном случае и изображение аналогичной функции в двухмерном случае.

косинусное преобразование, для которого удаление частот дает очень похожий эффект ложного оконтуривания.

Помимо удаления высокочастотной информации сигнала, может иметь место также ее искажение, которое приводит к похожим осцилляторным эффектам. В области цифровой обработки сигналов, как правило, все подобные искажения (в том числе эффект Гиббса) называют эффектом ложного оконтуривания.

Кратко рассмотрим некоторые алгоритмы обработки изображений, и опишем, почему в результате их применения может возникать эффект ложного оконтуривания.

В задаче повышения качества размытых изображений можно считать, что на вход поступает изображение на котором утеряна часть информации о высоких частотах, и задача состоит в их восстановлении. Из-за ошибок их восстановления возникают искажения высоких частот, которые могут приводить к эффекту ложного оконтуривания. Примером могут служить методы, основанные на обращении интегрального преобразования свертки изображения с оператором размытия (задача как правило является некорректной), в которых в частотной области производится деление частотных составляющих исходного изображения на заданную функцию.

Значения этой функции в области высоких частот могут быть низкими, поэтому при делении на них оценка значений для этих областей будет неустойчивой. Аналогичная ситуация происходит и в задаче повышения разрешения изображения, где исходное изображение фактически задает низкие частоты искомого изображения, а высокие частоты необходимо найти.

В задаче подавления шума на входном изображении искажена в основном высокочастотная информация. Тогда алгоритмы, восстанавливающие эту информацию, могут вносить в нее искажения за счет ошибок восстановления, что может приводить к эффекту ложного оконтуривания.

При сжатии изображений алгоритмом JPEG на одном из этапов производится дискретное косинусное преобразование изображения и его последующее квантование. В результате квантования происходит искажение информации, которое наиболее выражено для высоких частот (так как они имеют меньшие значения); это может приводить к эффекту ложного оконтуривания. В алгоритме JPEG2000 и других методах вейвлетной компрессии вместо косинусного преобразования используется вейвлет-преобразование, но в нем также возникают искажения высоких частот, в результате которых возникает эффект ложного оконтуривания. Осцилляции в нем могут иметь различную форму, в зависимости от выбора вейвлетов в алгоритме.

1.1.2 Методы обнаружения эффекта ложного оконтуривания на изображениях

Рассмотрим задачу обнаружения эффекта ложного оконтуривания для изображений. Для этой задачи возможны два варианта: обнаружение в случае, когда известно изображение без искажений (изображение-источник) и в случае, когда такое изображение не известно.

В задаче обнаружения эффекта ложного оконтуривания с изображением-источником на вход алгоритму подается изображение, на котором может присутствовать эффект ложного оконтуривания, а также изображение-источник, на котором заведомо нет этого искажения. На выходе необходимо определить, насколько ярко выраженным является эффект ложного оконтуривания на анализируемом изображении. Сложность данной задачи заключается в том, что на изображении могут присутствовать другие искажения, такие, как шум и размытие и их необходимо отличать от эффекта ложного оконтуривания.

В задаче обнаружения эффекта ложного оконтуривания без изображения-источника на вход алгоритму подается только анализируемое изображение, для которого, аналогично предыдущей задаче, необходимо определить, насколько ярко выраженным на нем является эффект ложного оконтуривания. Очевидно, что эта задача является более сложной, чем предыдущая, при этом

за счет меньших ограничений она более применима на практике. В данной работе предлагается решение именно для этой задачи.

Рассмотрим теперь, какие существуют известные алгоритмы для решения этих двух задач. Эти алгоритмы можно разделить по тому, на какие причины возникновения ложного оконтуривания они ориентированы.

Алгоритмов, работающих с произвольным эффектом ложного оконтуривания, мало [12]. В основном они основаны на вычислении полной вариации для изображения.

Многие алгоритмы ориентированы на работу с эффектом ложного оконтуривания, который получается в результате сжатия изображений с помощью алгоритмов сжатия JPEG [8, 27, 28] и JPEG 2000 [21, 29]. При этом в алгоритме JPEG эффект ложного оконтуривания обычно слабо выражен на фоне других искажений, поэтому он далее не рассматривается.

Также существуют методы, которые работают с эффектом ложного оконтуривания, получаемым в результате работы других алгоритмов: деконволюции [30, 31] и повышения разрешения [32]. Так как эти методы являются специфическими для данных преобразований, подробно они не рассматриваются.

Опишем теперь подробнее несколько наиболее релевантных алгоритмов: на основе вычисления полной вариации [12], на основе анализа ширины контура [21] и на основе анализа искажений контура [29]; с этими методами в дальнейшем будет производиться сравнение предложенного метода. Отметим сразу, что все рассмотренные методы обладают одним общим недостатком — они плохо отличают эффект ложного оконтуривания от таких искажений, как шум и размытие (это будет подробнее показано в разделе 1.3). Предложенный далее метод разрабатывался, в том числе, с целью устранить этот недостаток.

Метод на основе вычисления полной вариации

В работе [12] Насоновым А.В. и Крыловым А.С. рассматривается задача оценки меры эффекта ложного оконтуривания на изображениях без источника. Предлагается метрика, основанная на вычислении полной вариации для областей изображения.

Полная вариация непрерывно дифференцируемой функции $f(x)$ на отрезке $[a; b]$ определяется как:

$$\text{TV}(f) = \int_a^b |f'(x)| dx.$$

В дискретном одномерном случае для отсчетов $a = x_1 < \dots < x_n = b$ полная вариация вычисляется так (полагаем, что шаг сетки равен 1):

$$\text{TV}(f) = \sum_{i=2}^n |f(x_i) - f(x_{i-1})|.$$

Аналогичным образом определяется взвешенная полная вариация:

$$\text{TV}(f, w) = \int_a^b |f'(x)| w(x) dx,$$

и ее дискретный аналог:

$$\text{TV}(f, w) = \sum_{i=2}^n |f(x_i) - f(x_{i-1})| w \left(\frac{x_{i-1} + x_i}{2} \right),$$

где $w(x) \geq 0$ — некоторая весовая функция.

Отметим, что эффект ложного оконтуривания и полная вариация тесно связаны между собой (чем и обусловлен ее выбор в рассматриваемой работе). Более подробно эта связь и ее применения будут рассмотрены в разделе 2.1.

Так как эффект ложного оконтуривания наиболее выражен вблизи контуров, предложенная метрика работает именно с контурами. Размытый одномерный контур моделируется с помощью такой функции:

$$f_d(x) = \begin{cases} 0, & \text{если } x < -d/2, \\ x/d + 1/2, & \text{если } x - d/2 \leq x < d/2, \\ 1, & \text{если } x \geq d/2, \end{cases}$$

где d — ширина контура. Отметим, что эта модель контура отличается от используемой в данной работе модели (1.3), однако на практике эти отличия не являются существенными и не оказывают влияние на эффективность методов.

Для произвольного контура на изображении его ширину можно найти следующим образом. Для профиля контура находится такое значение d , для которого функция f_d будет близка к профилю. Этот выбор производится таким образом, чтобы результат был устойчив к влиянию эффекта ложного оконтуривания и шума.

Далее для данного контура с шириной d оценивается уровень эффекта ложного оконтуривания. Так как реальные контуры не являются бесконечными, необходимо рассматривать только первые несколько осцилляций эффекта ложного оконтуривания, и это делается с помощью введения весовой функции для вычисления полной вариации

$$w_{\alpha d} = e^{-\frac{x^2}{2(\alpha d)^2}},$$

где параметр α определяет количество рассматриваемых осцилляций, и обычно выбирается как $\alpha = 3$. В работе показывается, что хорошей метрикой для вычисления уровня эффекта ложного оконтуривания для контура E является следующая функция:

$$R_E(f, d) = \frac{\text{TV}(f * G_{m(\alpha, n_0) \cdot d}, w_{\alpha d})}{\text{TV}(f * G_d, w_{\alpha d})},$$

где параметр n_0 определяет оценку уровня шума на изображении, а функция $m(\alpha, n_0)$ определяется для каждой пары значений α, n_0 по результатам численных экспериментов.

Эффект ложного оконтуривания наиболее выражен вблизи наиболее резких контуров на изображении. Введем обозначения областей Basic Edge Neighborhood (BEN) и Basic Edge Points (BEP) [14, 33]. Для построения этих областей на изображении находятся ярко выраженные контуры, отстоящие достаточно далеко от других контуров (во избежание интерференции эффекта ложного оконтуривания). Будем называть такие контуры базовыми контурами (basic edges). Области, соответствующие точкам базовых контуров, обозначаются BEP, а области на небольшом расстоянии от них обозначаются BEN. На рис. 1.6 схематически изображены эти области. Для улучшения качества работы алгоритма уровень эффекта ложного оконтуривания оценивается только для базовых контуров.

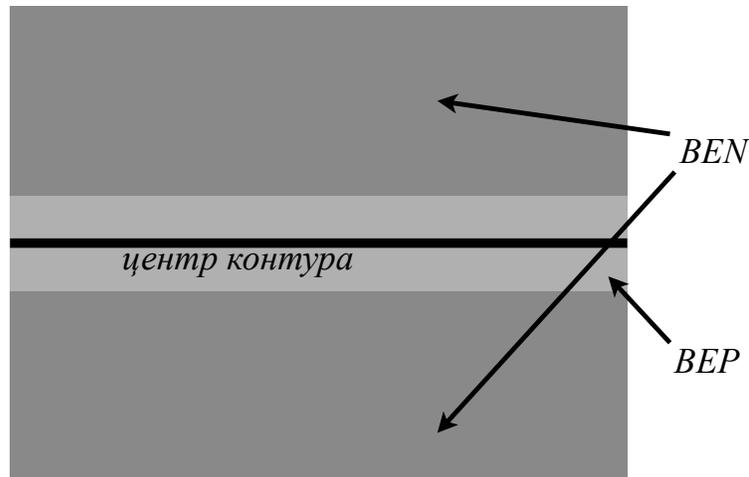


Рис. 1.6: Схематическое изображение областей BEN и BEP для базовых контуров.

Полный алгоритм определения уровня эффекта ложного оконтуривания представлен в 1.1. Основным преимуществом данного метода является то, что он применим к любому эффекту ложного оконтуривания вне зависимости от причины его возникновения.

Алгоритм 1.1 Метод обнаружения эффекта ложного оконтуривания на основе вычисления полной вариации.

Вход: Изображение I , оценка уровня шума на нем n_0 , значение параметра α .

Выход: Уровень эффекта ложного оконтуривания на изображении R_I .

- 1: Найти на изображении базовые контуры.
 - 2: Оценить полупериод осцилляций эффекта ложного оконтуривания D . Для этого рассматривается набор значений ширины контуров d_E для каждого контура E , полученного на предыдущем шаге. Этот набор трактуется как распределение вероятностей, и в качестве D выбирается значение соответствующее наибольшей плотности. Также для увеличения надежности метода после этого контуры с шириной, сильно отличающейся от D , исключаются из рассмотрения.
 - 3: Для каждого оставшегося контура E с профилем f вычислить значение уровня эффекта ложного оконтуривания $R_E(f, D)$ для этого контура. Вычислить R_I как среднее от этих значений.
-

Метод на основе анализа ширины контура

В [21] Марцилиано П. и др. рассматривается задача оценки уровня эффекта ложного оконтуривания с источником для изображений сжатых с помощью алгоритма сжатия JPEG2000. Метод основан на анализе ширины контуров на изображении.

На первом шаге в методе производится поиск всех ярко выраженных контуров на изображении. Для этого в каждой точке изображения вычисляется приближенное значение модуля градиента яркости, и все точки со значением выше некоторого порога полагаются точками контуров (отсечение по заданному порогу сделано для уменьшения влияния шума на метод и для того, чтобы метод не учитывал слабо выраженные контуры). Также в методе предлагается использовать только вертикальные контуры, так как на практике это не ухудшает эффективность метода (эффект ложного оконтуривания обычно в одинаковой степени выражен как по вертикали, так и по горизонтали), но позволяет уменьшить его ресурсоемкость.

Далее для каждого найденного контура строится его профиль и для него ищутся локальный минимум и локальный максимум в окрестности центра контура. Расстояние от левого из экстремумов до центра обозначается как ширина контура слева E_l , а расстояние правого из экстремумов до центра как ширина контура справа E_r . Далее по ним вычисляется ширина эффекта ложного оконтуривания слева R_l и справа R_r :

$$R_l = R - E_l,$$

$$R_r = R - E_r,$$

где R — фиксированная ширина эффекта ложного оконтуривания, вычисляемая на основе информации о вейвлетах, используемых при сжатии изображения.

Далее обозначим блок исходного изображения на расстоянии R от центра контура как I_1 и соответствующий блок оцениваемого изображения как I_2 . Тогда окончательная мера эффекта ложного оконтуривания M для данного контура вычисляется так:

$$M = (R_l + R_r) \cdot (\max(I_1 - I_2) - \min(I_1 - I_2)).$$

Мера эффекта ложного оконтуривания на изображении далее вычисляется как среднее значение величин M по всем контурам на изображении.

Основным достоинством данного метода является его низкая ресурсоемкость. Однако этот метод решает более простую задачу, чем рассматриваемая, а именно — для оценки уровня эффекта ложного оконтуривания в нем используется источник, и рассматривается только эффект ложного оконтуривания, полученный в результате сжатия изображения алгоритмом JPEG2000. Поиск локальных экстремумов на одном из этапов делает этот метод крайне неустойчивым к шуму.

Метод на основе анализа искажений контура

В [29] Лиангом Л. и др. рассматривается задача оценки качества изображения без источника для изображений сжатых с помощью алгоритма сжатия JPEG2000; одним из компонентов метрики для оценки качества является метрика для величины эффекта ложного оконтуривания на изображении. Метод основан на анализе искажений областей в районе контуров.

На первом шаге в алгоритме находятся контуры и рассматриваются регионы вокруг них (так как эффект ложного оконтуривания наиболее выражен именно в этих областях). Авторы выбирают регионы 5×5 пикселей с одной и с другой стороны от контура (обозначим их $R_0(x_0)$ и $R_1(x_0)$ соответственно) на расстоянии в 3 пикселя от центра контура x_0 .

Далее для каждого пикселя в этих регионах вычисляется величина

$$A(x) = \left(I(x) - \frac{1}{|R_N(x)|} \sum_{x_i \in R_N(x)} I(x_i) \right)^2,$$

где $I(x)$ — яркость пикселя x , и $R_N(x)$ — квадратный регион 3×3 пикселя вокруг пикселя x . Для уменьшения влияния шума значения $A(x)$ фильтруются по двум порогам T_l, T_h :

$$\tilde{A}(x) = \begin{cases} A(x), & \text{если } T_l < A(x) < T_h; \\ 0, & \text{иначе.} \end{cases}$$

Авторы выбирают следующие значения порогов (для регионов в точке контура x_0): $T_l = 0$, $T_h = 0.65 \cdot \max\{A(x) \mid x \in R_i(x_0), i = 0, 1\}$.

Далее для региона $R_i(x_0)$ вокруг пикселя контура x_0 вычисляется

$$A(R_i(x_0)) = \sqrt{\frac{1}{|R_i(x_0)|} \sum_{x \in R_i(x_0)} \tilde{A}(x)},$$

где $i = 0, 1$ и соответствует регионам с разных сторон от контура. Эта величина корректируется с учетом порога минимально заметных искажений $JND(x)$ для данного значения яркости x в соответствии с работой [34]:

$$AP(R_i(x_0)) = \frac{A(R_i(x_0))}{1 + JND(R_i(x_0))}.$$

Итоговая метрика эффекта ложного оконтуривания вычисляется как среднее от описанной выше меры по всем регионам вокруг контуров на изображении.

Данный метод хорошо оценивает степень искажения изображения эффектом ложного оконтуривания, полученным в результате сжатия алгоритмом JPEG2000, однако он неприменим к другим типам эффекта ложного оконтуривания.

1.2 Метод разреженных представлений

Метод разреженных представлений в последнее время получил широкое распространение в задачах связанных с обработкой сигналов [11, 35], и, в частности, изображений [36]. Основная идея этого метода заключается в представлении входного сигнала (в данном случае изображения или блока изображения) в виде линейной комбинации небольшого числа сигналов из заранее выбранного набора. Ограничение на количество сигналов регуляризирует представление и на практике приводит к устойчивости результата к шуму и искажениям.

1.2.1 Задача разреженных представлений

Сформулируем задачу разреженных представлений [11].

В методе разреженных представлений сигналы и изображения представляются в виде наборов дискретных отсчетов, поэтому обозначим $y \in \mathbb{R}^n$ — произвольный сигнал, где n — количество отсчетов в нем. При работе с изображениями все значения его пикселей обычно записываются в вектор подряд по строкам (порядок значений не важен для метода при условии, что он одинаков для всех изображений). Также из соображений эффективности обычно рассматриваются не целые изображения, а его небольшие блоки.

Далее, пусть дан набор элементарных сигналов $\{d_1, \dots, d_m\}$ (способы выбора этого набора будут описаны позже), причем $d_i \in \mathbb{R}^n$ для $i = 1, \dots, m$. Запишем эти сигналы по столбцам в матрицу $D = [d_1; \dots; d_m]$, которую далее будем называть словарем; тогда $D \in \mathbb{R}^{n \times m}$. Отметим, что, как правило, словарь выбирается так, что $m \geq n$.

Обозначим через $\|x\|_0$ количество ненулевых коэффициентов в векторе x :

$$\|x\|_0 = |\{i \mid x[i] \neq 0\}|.$$

Эту величину также принято называть l_0 -нормой (хотя на самом деле функция не является нормой) по аналогии с l_p -нормами:

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, \quad (1.6)$$

которые поточечно сходятся к l_0 -норме при $p \rightarrow 0$ (отметим, что $\|\cdot\|_p$ является нормой только при $p \geq 1$). Действительно, обозначим

$$[x_i = 0] = \begin{cases} 1, & \text{если } x_i = 0; \\ 0 & \text{иначе,} \end{cases}$$

тогда

$$\lim_{p \rightarrow 0} \|x\|_p^p = \lim_{p \rightarrow 0} \sum_{i=1}^n |x_i|^p = \sum_{i=1}^n \lim_{p \rightarrow 0} |x_i|^p = \sum_{i=1}^n [x_i = 0] = |\{i \mid x[i] \neq 0\}| = \|x\|_0.$$

Пусть теперь ε — максимально допустимая ошибка представления, а $c \in \mathbb{R}^m$ — искомое представление, тогда задачу разреженных представлений можно сформулировать следующим образом:

$$\|c\|_0 \rightarrow \min_c \quad \text{при условии} \quad \|y - Dc\|_2^2 \leq \varepsilon. \quad (1.7)$$

Для этой задачи возможна также альтернативная формулировка:

$$\|y - Dc\|_2^2 \rightarrow \min_c \quad \text{при условии} \quad \|c\|_0 \leq k, \quad (1.8)$$

где k — ограничение на число коэффициентов в представлении.

Обе задачи в общем случае являются NP-полными, поэтому поиск точного решения на практике невозможен. В разделе 1.2.2 рассматриваются приближенные методы решения этой задачи, на практике дающие хорошие результаты.

Большую роль для эффективности метода играет выбор словаря D . Существует два основных подхода к выбору словарей: конструирование словаря из фиксированных базисов для сигналов, и построение словаря на основе набора характерных для задачи сигналов. Ниже будет рассмотрен первый из них, второй рассматривается в разделе 2.2.

Для представления векторов в n -мерном пространстве можно использовать различные базисы. Любой сигнал можно представить в виде линейной комбинации элементов базиса, но на практике удобны такие базисы, в которых большая доля коэффициентов будет близка к нулю. Это позволяет делать разложение устойчивым к шуму и искажениям.

Примерами таких базисов могут служить, например, базис дискретного преобразования Фурье или базисы вейвлетов [35]. За счет межпиксельных корреляций интенсивности разложение естественных изображений по такому базису как правило содержит небольшое число больших коэффициентов и большое число маленьких коэффициентов. Выбор подходящего базиса сильно зависит от задачи и характерных для этой задачи изображений и искажений; в некоторых задачах подходящие базисы могут быть созданы на основе априорных знаний о входных изображениях (например, так делается в данной работе в разделе 1.3.2).

В задачах (1.7) и (1.8) для увеличения разнообразия словаря (и, как следствие, упрощения разреженного разложения по нему) обычно величину m выбирают в несколько раз больше величины n . Для этого можно собрать в один словарь несколько базисов, например, выбрав $m = 2n$ и взяв базис дискретного косинусного преобразования Фурье и базис вейвлетов Хаара.

Использование в качестве словаря наборов из нескольких базисов также позволяет интерпретировать разложение сигнала в зависимости от того, элементы какого базиса используются [11]. Например, можно предположить, что один класс вейвлетов хорошо представляет крупные контуры на изображении, а другой хорошо представляет текстуры. Тогда после разложения сигнала по словарю из объединения вейвлетов этих классов можно восстановить отдельно контуры и отдельно текстуры на изображении, используя для них элементы из разных классов.

Отметим также, что, несмотря на разнообразие различных доступных базисов, подбор подходящего словаря для задачи как правило является очень трудоемким и не позволяет очень точно

настраивать словарь под задачу. Из-за этого второй подход к построению словаря (основанный на обучении) как правило дает более качественные результаты. Однако он требует создания большого набора характерных сигналов для обучения, и не во всех задачах присутствует такая возможность.

1.2.2 Алгоритмы построения представлений

Рассмотрим существующие алгоритмы для приближенного решения задач (1.7) и (1.8). Эти алгоритмы можно разделить на две группы по способу решения этой задачи: жадные алгоритмы, добавляющие коэффициенты по одному, и ослабляющие алгоритмы, решающие вместо исходной задачи другую задачу, находящую разреженные решения, но не являющуюся NP-полной.

Жадные алгоритмы

Отметим для начала, что ключевой сложностью при решении задач (1.7) и (1.8) является нахождение набора коэффициентов, которые будут ненулевыми (носителя). Если же набор коэффициентов зафиксирован, то минимизация квадратичной ошибки сводится к задаче линейной регрессии.

Действительно, предположим, что в результате некоторого алгоритма получен носитель $S = (i_1, i_2, \dots, i_k)$ — набор индексов элементов словаря D , который будет использоваться при построении. Обозначим D_S — ограничение словаря на этот набор, то есть матрица составленная из столбцов с индексами из S : $D_S = D[i_1, i_2, \dots, i_k]$. Так как носитель зафиксирован, то и количество ненулевых коэффициентов в решении ($\|c\|_0$) также зафиксировано, поэтому задача (1.7) сводится к следующей задаче:

$$\text{Найти } c_S, \text{ т.ч. } \|y - D_S c_S\|_2^2 \leq \varepsilon.$$

Вектор \hat{c} , дающий минимальное значение отклонения $\|y - D_S c_S\|_2^2$ получается по формуле:

$$\hat{c} = (D_S^T D_S)^{-1} D_S^T y, \quad (1.9)$$

и, если отклонение для \hat{c}_S не больше ε , то это и будет решением задачи, а в противном случае решения нет.

Аналогично задача (1.8) сводится к задаче:

$$\|y - D_S c_S\|_2^2 \rightarrow \min_{c_S},$$

и вектор \hat{c}_S , получаемый по формуле (1.9), будет решением этой задачи.

После получения решения c_S на заданном носителе S решение c на всем словаре получается записыванием коэффициентов из c_S на соответствующие позиции в c : $c[i_j] = c_S[j]$ для всех $j = 1, \dots, k$ и $c[i] = 0$ для всех остальных позиций.

Самым простым алгоритмом построения носителя является **пороговый метод** [37]. Этот метод для каждого элемента словаря независимо оценивает, насколько этот элемент может быть полезен для представления целевого вектора y , а потом из наилучших элементов составляет носитель. Полезность каждого элемента оценивается следующим образом. Предположим, что i — индекс рассматриваемого элемента, и решим исходную задачу на носителе $S = (i)$. Тогда D_S будет состоять из одного столбца, и задача сводится к вычислению минимальной возможной ошибки $e(i)$ при приближении y вектором $\alpha_i d_i$, где α_i — параметр:

$$e(i) = \min_{\alpha_i} \|y - \alpha_i d_i\|_2^2.$$

По формуле (1.9) оптимальным решением $\hat{\alpha}_i$ будет такое:

$$\hat{\alpha}_i = \frac{(d_i, y)}{\|d_i\|_2^2}. \quad (1.10)$$

Пороговый метод для задачи (1.8) представлен в алгоритме 1.2.

Алгоритм 1.2 Пороговый метод для задачи (1.8)

Вход: словарь D , целевой вектор y , ограничение количества коэффициентов k .

Выход: вектор коэффициентов c .

- 1: Вычислить ошибки $e(i)$ для каждого столбца D ;
 - 2: Отсортировать элементы по увеличению этой ошибки;
 - 3: Найти k индексов с наименьшими ошибками $e(i)$: $S = (i_1, \dots, i_k)$;
 - 4: Найти оптимальное решение c_S на носителе S ;
 - 5: Построить решение c на всем словаре по c_S .
-

Для решения задачи (1.7) в общем случае нельзя использовать этот алгоритм, однако часто на практике вместо задачи (1.7) можно использовать задачу (1.8).

Можно рассмотреть частный случай задачи (1.7), когда пороговый алгоритм применим для ее решения, — если все элементы словаря D ортогональны (тогда обязательно $m \leq n$). В этом случае оптимальные коэффициенты при любом элементе не зависят от рассматриваемого носителя с этим элементом. Действительно, рассмотрим произвольный носитель $S = (i_1, \dots, i_k)$ и найдем

коэффициент при элементе с индексом i_1 для оптимального приближения целевого вектора на носителе S . Матрица D ортогональна, а значит и D_S тоже ортогональна. Поэтому

$$D_S^T D_S = \begin{pmatrix} \|d_{i_1}\|_2^2 & 0 & \dots & 0 \\ 0 & \|d_{i_2}\|_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \|d_{i_k}\|_2^2 \end{pmatrix}.$$

Тогда

$$\hat{c}_S = (D_S^T D_S)^{-1} D_S^T y = \left(\frac{(d_{i_1}, y)}{\|d_{i_1}\|_2^2}, \frac{(d_{i_2}, y)}{\|d_{i_2}\|_2^2}, \dots, \frac{(d_{i_k}, y)}{\|d_{i_k}\|_2^2} \right)^T.$$

Видим, что оптимальный коэффициент при i_1 получается точно такой же, как при использовании одного элемента в формуле (1.10).

Ошибка при оптимальном приближении целевого вектора y с помощью словаря D_S в такой задаче вычисляется по ошибкам каждого из элементов, входящих в носитель S . Действительно, пусть $S = (i_1, \dots, i_k)$, и обозначим вектор

$$z_j = \frac{(d_j, y)}{\|d_j\|_2^2} d_j.$$

Тогда, если \hat{x} — оптимальное решение на этом носителе по формуле (1.9), то очевидно, что

$$D_S \hat{c}_S = z_{i_1} + \dots + z_{i_k}.$$

Кроме того, заметим, что все z_i ортогональны между собой.

Далее заметим следующее

$$\|z_{i_j}\|_2^2 = (z_{i_j}, z_{i_j}) = \left(\frac{(d_{i_j}, y)}{\|d_{i_j}\|_2^2} \right)^2 (d_{i_j}, d_{i_j}) = \frac{(d_{i_j}, y)^2}{\|d_{i_j}\|_2^2} = (y, z_{i_j}).$$

Из этого получаем:

$$\begin{aligned} \|y - D_S \hat{c}_S\|_2^2 &= \|y - (z_{i_1} + \dots + z_{i_k})\|_2^2 = \\ &= (y - (z_{i_1} + \dots + z_{i_k}), y - (z_{i_1} + \dots + z_{i_k})) = \\ &= (y, y) - 2((y, z_{i_1}) + \dots + (y, z_{i_k})) + (z_{i_1}, z_{i_1}) + \dots + (z_{i_k}, z_{i_k}) = \\ &= \|y\|_2^2 - (\|z_{i_1}\|_2^2 + \dots + \|z_{i_k}\|_2^2). \end{aligned}$$

Из этой формулы видно, что для решения задачи (1.7) достаточно построить независимые решения для каждого из элементов, а потом, начиная с решения с наибольшей нормой, добавлять их по одному, пока сумма их норм не превзойдет $\|y\|_2^2 - \varepsilon$ (отсюда происходит название алгоритма, так как можно считать, что мы исключаем из рассмотрения элементы с наименьшей нормой по некоторому порогу). Также, заметим, что ошибка для одиночного элемента $e(i)$ вычисляется как $e(i) = \|y\|_2^2 - \|z_i\|_2^2$.

Пороговый метод для задачи (1.7) с ортогональным словарем представлен в псевдокоде 1.3. Отметим, что из приведенных выше рассуждений данный частный случай задачи (1.7) не является NP-полным, и алгоритм 1.3 находит точное решение этой задачи.

Алгоритм 1.3 Пороговый метод для задачи (1.7) с ортогональным словарем

Вход: ортогональный словарь D , целевой вектор y , ограничение ошибки ε .

Выход: вектор коэффициентов c .

- 1: Вычислить ошибки $e(i)$ для каждого столбца D ;
 - 2: Отсортировать элементы по увеличению этой ошибки (то есть по убыванию $\|z_i\|_2^2$);
 - 3: Положить $r := \|y\|_2^2$ и $S := \emptyset$;
 - 4: **Пока** $r > \varepsilon$ **выполнять**
 - 5: Взять очередной индекс j из отсортированного списка индексов;
 - 6: Добавить его в носитель, $S := S \cup \{j\}$;
 - 7: Обновить ошибку, $r := r - \|z_j\|_2^2$.
 - 8: **Конец цикла**
 - 9: Найти оптимальное решение c_S на носителе S ;
 - 10: Построить решение c на всем словаре по c_S .
-

Рассмотрим теперь более сложный жадный алгоритм — **ортогональный поиск соответствий** (orthogonal matching pursuit) [38]. В отличие от порогового метода этот алгоритм добавляет элементы в носитель по одному и после каждого добавления пересчитывает ошибку для текущего носителя и для всех остальных элементов. Алгоритм для задачи (1.7) описан в 1.4, а алгоритм для задачи (1.8) описан в 1.5. На практике этот алгоритм дает более качественные результаты, чем пороговый метод, но является более ресурсоемким.

Ослабляющие алгоритмы

Основная проблема при работе с l_0 -нормой заключается в том, что она не является дифференцируемой, и даже не является непрерывной. Вместо нее можно рассматривать l_p -норму, определенную

Алгоритм 1.4 Ортогональный поиск соответствий для задачи (1.7)

Вход: Словарь D , целевой вектор y , ограничение ошибки ε .

Выход: вектор коэффициентов c .

- 1: Положить $r := y$, $S := \emptyset$;
 - 2: **Пока** $\|r\|_2^2 > \varepsilon$ **выполнять**
 - 3: Для каждого элемента $j \notin S$ вычислить ошибку при приближении r этим элементом:
 $e(j) = \min_{\alpha} \|r - \alpha d_j\|_2^2$ используя формулу (1.10);
 - 4: Найти i , соответствующее минимальной ошибке $e(i)$;
 - 5: Добавить i в носитель, $S := S \cup \{i\}$;
 - 6: Найти коэффициенты c_S для приближения y текущим носителем S по формуле (1.9)
 - 7: Обновить ошибку, $r := y - D_S c_S$.
 - 8: **Конец цикла**
 - 9: Найти оптимальное решение c_S на носителе S ;
 - 10: Построить решение c на всем словаре по c_S .
-

Алгоритм 1.5 Ортогональный поиск соответствий для задачи (1.8)

Вход: Словарь D , целевой вектор y , ограничение на количество коэффициентов k .

Выход: вектор коэффициентов c .

- 1: Положить $r := y$, $S := \emptyset$;
 - 2: **Пока** $|S| < k$ **выполнять**
 - 3: Для каждого элемента $j \notin S$ вычислить ошибку при приближении r этим элементом:
 $e(j) = \min_{\alpha} \|r - \alpha d_j\|_2^2$ используя формулу (1.10);
 - 4: Найти i , соответствующее минимальной ошибке $e(i)$;
 - 5: Добавить i в носитель, $S := S \cup \{i\}$;
 - 6: Найти коэффициенты c_S для приближения y текущим носителем S по формуле (1.9);
 - 7: Обновить ошибку, $r := y - D_S c_S$.
 - 8: **Конец цикла**
 - 9: Найти оптимальное решение c_S на носителе S ;
 - 10: Построить решение c на всем словаре по c_S .
-

в (1.6). Использование этой нормы обуславливается тем, что она поточечно сходится к l_0 -норме при $p \rightarrow 0$, как было показано в разделе 1.2.1.

На практике l_p -нормы приводят к разреженным решениям только в случае, когда $0 < p \leq 1$, поэтому случай $p > 1$ обычно не рассматривается [11]. Задача для l_p -нормы, аналогичная задаче (1.7), тогда ставится так:

$$\|c\|_p \rightarrow \min_c \quad \text{при условии} \quad \|y - Dc\|_2^2 \leq \varepsilon.$$

Задача, аналогичная (1.8) ставится так:

$$\|y - Dc\|_2^2 \rightarrow \min_c \quad \text{при условии} \quad \|c\|_p \leq k.$$

Существует набор приближенных алгоритмов для решения этой задачи, например, алгоритм FOCUSS [39]. Отдельно рассмотрим случай $p = 1$, часто используемый на практике. Для него задачи формулируются так:

$$\|c\|_1 \rightarrow \min_c \quad \text{при условии} \quad \|y - Dc\|_2^2 \leq \varepsilon, \quad (P_1)$$

$$\|y - Dc\|_2^2 \rightarrow \min_c \quad \text{при условии} \quad \|c\|_1 \leq k. \quad (P'_1)$$

Также существует специальная постановка для случая $p = 1$, удобная на практике:

$$\|y - Dc\|_2^2 + \lambda \|c\|_1 \rightarrow \min_c. \quad (Q_1)$$

Все эти задачи являются задачами выпуклой оптимизации и для их решения можно использовать соответствующие методы (например, см. [40]). Однако эти методы рассчитаны на более общие постановки задач, поэтому имеют высокую алгоритмическую сложность. Поэтому существуют методы, непосредственно ориентированные на данные задачи, например, метод LARS [41] для задач (P_1) и (P'_1) . Отметим также, что задача (P'_1) фактически является задачей LASSO [42].

Рассмотрим также алгоритм, решающий задачу (Q_1) , и хорошо работающий на практике — алгоритм ISTA [43]. Обозначим $f(x) = \|y - Dx\|_2^2$. В методе предлагается рассмотреть проксимальную регуляризацию линеаризованной функции $f(x)$ и использовать итеративную схему оптимизации:

$$x_{k+1} = \arg \min_x \left\{ f(x_k + \langle x - x_k, \nabla f(x_k) \rangle) + \frac{1}{2t_k} \|x - x_k\|_2^2 + \lambda \|x\|_1 \right\},$$

где t_k — некоторая последовательность длин шагов для оптимизации. Отбросив в выражении выше члены не зависящие от x , и преобразовав его, получаем:

$$x_{k+1} = \arg \min_x \left\{ \frac{1}{2t_k} \|x - (x_k - t_k \nabla f(x_k))\|_2^2 + \lambda \|x\|_1 \right\}.$$

Далее можно видеть, что минимизацию можно производить покомпонентно, поэтому вводя покомпонентный оператор $\mathcal{T}_\alpha(x) = \text{sign}(x) \max(|x| - \alpha, 0)$, получаем:

$$x_{k+1} = \mathcal{T}_{\lambda t_k}(x_k - t_k \nabla f(x_k)),$$

или, подставляя выражение для $f(x)$:

$$x_{k+1} = \mathcal{T}_{\lambda t_k}(x_k - 2t_k D^T(Dx_k - y)).$$

Существует также ускоренный вариант данного алгоритма — FISTA [44].

Все рассмотренные выше задачи можно использовать как самостоятельные задачи разреженных представлений, или же как способы приближенного решения задач (1.7) и (1.8). Во втором случае после решения ослабленной задачи необходимо выделить носитель полученного решения, и далее найти наилучшее приближение целевого вектора с использованием этого носителя по формуле (1.9). Такой подход обычно называют алгоритмом поиска базиса (basis pursuit), и при определенных условиях его результат совпадает с решением задачи основанной на l_0 -норме [45].

1.3 Обнаружение эффекта ложного оконтуривания на изображениях с помощью метода разреженных представлений

Основная идея разработанного метода обнаружения эффекта ложного оконтуривания, основанного на методе разреженных представлений [13], заключается в анализе наиболее ярко выраженных контуров на изображении и построении метрики уровня эффекта ложного оконтуривания на основе них. Словарь для анализа контуров с помощью метода разреженных представлений создается заранее на основе модели эффекта ложного оконтуривания.

1.3.1 Связь эффекта ложного оконтуривания и размытия

изображений

Исследуем, как устроены контуры на изображениях с эффектом ложного оконтуривания и на изображениях без него.

Рассматриваются такие искажения изображений эффектом ложного оконтуривания, которые действуют единообразно на все области на изображении, поэтому для задачи обнаружения этих искажений не обязательно анализировать все области с контурами. Далее в методе анализируются только такие блоки изображения с контурами, для которых, во-первых, граница контура близка к прямой линии, и, во-вторых, центр контура расположен в центре блока. Такие блоки можно моделировать дублированием значений одномерного профиля контура перпендикулярно его оси. Технически для этого достаточно сделать это в случае, когда ось профиля параллельна одной из осей изображения, и потом повернуть изображение на необходимый угол. На рис. 1.7 изображены примеры одномерного профиля контура, а также двумерных контуров, полученных из него.



Рис. 1.7: Пример одномерного профиля (слева), двумерного профиля, полученного из него дублированием значений вдоль горизонтальной оси (в центре) и поворота этого двумерного контура (справа).

Контуры на изображениях без эффекта ложного оконтуривания можно моделировать с помощью размытого разрывного контура, профиль которого задается формулой (1.3).

Контуры на изображениях с эффектом ложного оконтуривания искажены, во-первых, осцилляциями вблизи центра контура, и во-вторых размытием (которое складывается из размытия исходного контура, а также из размытия, возникающего вследствие ослабления высоких частот). Для анализа осцилляций предлагается отделить размытие контура от осцилляций. Для этого

словарь для метода разреженных представлений строится из набора базисных элементов для построения размытых контуров и набора базисных элементов для построения чистых осцилляций.

Таким образом в разработанном методе словарь состоит из двух наборов базисных изображений. Первый набор изображений — размытые контуры — строится как всевозможные повороты профиля размытого контура, задаваемой формулой (1.3) при различных значениях величины размытия σ . Второй набор изображений — для построения чистых осцилляций — строится следующим образом. Рассматривается профиль идеального контура (1.1) и к нему применяется оператор R_d для создания эффекта ложного оконтуривания с различными значениям силы эффекта ложного оконтуривания d . Для каждого такого профиля ищется наиболее похожий на него профиль размытого контура (1.3). Похожесть профилей измеряется по их среднеквадратичному отклонению друг от друга. Так как размытые контуры не содержат быстрых изменений, после вычитания такого размытого контура из исходного осцилляции сохраняются, таким образом мы разделим исходный искаженный контур на размытый контур и чистый эффект ложного оконтуривания. На рис. 1.8 изображен весь описанный выше процесс отделения чистого эффекта ложного оконтуривания.

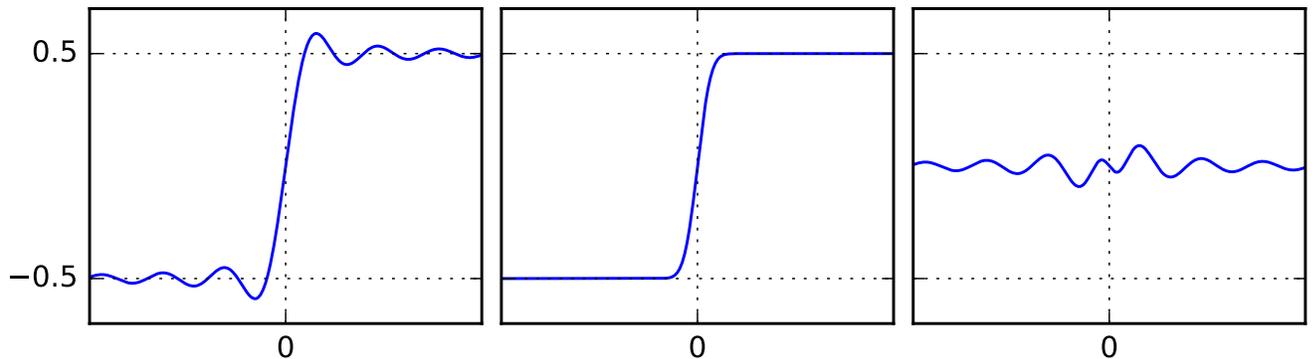


Рис. 1.8: Пример отделения чистого эффекта ложного оконтуривания от размытого контура: исходный контур с эффектом ложного оконтуривания (слева), ближайший размытый контур (в центре), чистый эффект ложного оконтуривания (справа).

Таким образом, проделывая этот процесс при различных значениях d и различных поворотах профиля мы получаем набор для построения чистых осцилляций. На рис. 1.9 изображены случайные примеры элементов полученного словаря. Отметим, что данные элементы были также домножены на весовую функцию, которая уменьшается при удалении от центра контура; более подробно это взвешивание будет описано в 1.3.2.

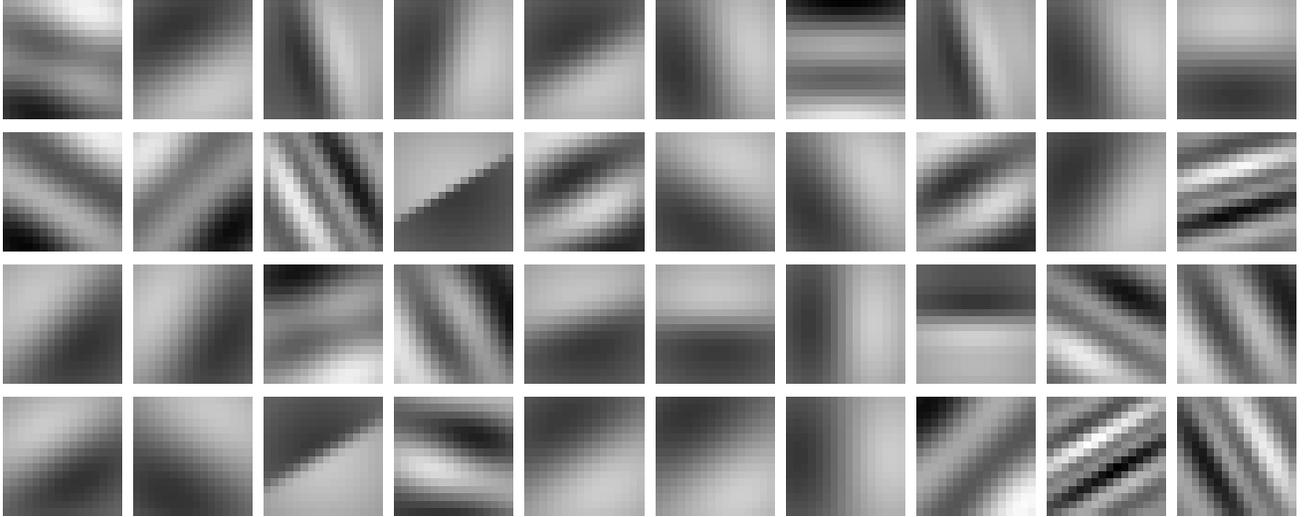


Рис. 1.9: Выборка случайных элементов из словаря для обнаружения эффекта ложного оконтуривания.

Исследуем, как зависят между собой сила эффекта ложного оконтуривания контура d и наиболее подходящая степень размытия σ . Для этого сформируем сетку значений d и для каждого значения на сетке найдем подходящее значение σ . В данной работе были взяты значения в диапазоне $d \in [2, 16]$, так как эффект ложного оконтуривания вне этого диапазона либо практически незаметен, либо настолько сильно искажает изображение, что на практике не встречается. Шаг сетки был выбран равным 0.05. На рис. 1.10 изображен график этой зависимости. Заметим, что ступенчатость графика связана с использованием дискретного преобразования Фурье, для которого близкие значения d дают одинаковые наборы обнуляемых коэффициентов.

Можно видеть, что график хорошо приближается линейной функцией. Коэффициенты наиболее подходящей функции можно найти методом наименьших квадратов, и она получается такой:

$$\sigma \approx 0.336 \cdot d.$$

1.3.2 Обнаружение эффекта ложного оконтуривания на

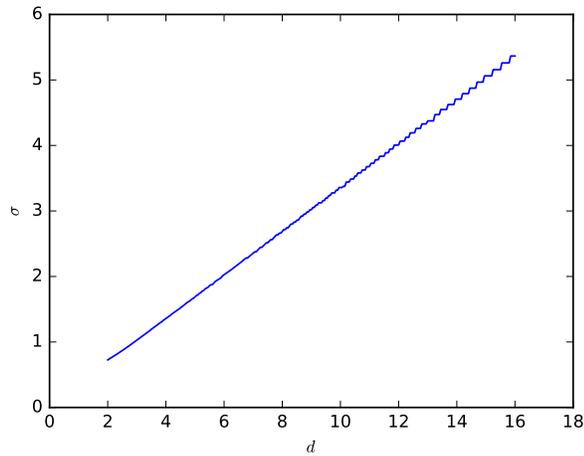


Рис. 1.10: Зависимость размытия σ оптимального приближения размытым контуром от силы эффекта ложного оконтуривания d .

изображениях

Описание метода

Имея словарь из двух частей, построенный в разделе 1.3.1, построим теперь алгоритм обнаружения эффекта ложного оконтуривания.

Пусть y — блок входного изображения. Найдем разреженное представление c для этого блока (наилучшие результаты получались при использовании задачи (1.8) с параметром $k = 5$). Словарь D для этого представления состоит из двух частей: части соответствующей размытым контурам D_1 и части, соответствующей чистым осцилляциями эффекта ложного оконтуривания D_2 . Разобьем по этому принципу представление на две части: c_1 соответствует атомам из подсловаря D_1 , а c_2 соответствует атомам из подсловаря D_2 . Тогда, обозначив за e ошибку представления, получаем:

$$y = Dc + e = D_1c_1 + D_2c_2 + e.$$

Отметим, что, хотя словарь строился как разложение одного контура на два элемента, в разложении выше одному подсловарю может соответствовать больше одного элемента. Это происходит из-за того, что реальные контуры не всегда являются идеально прямыми, и алгоритм приближает их несколькими контурами, повернутыми относительно друг друга.

Обозначим $b = D_1c_1$ — слагаемое блока, соответствующее его размытому контуру на нем, и $r = D_2c_2$ — слагаемое блока, соответствующее чистым осцилляциями эффекта ложного оконтуривания на нем. Итого получаем:

$$y = b + r + e. \quad (1.11)$$

Отметим также, что для улучшения качества алгоритма при разложении используется взвешенная l_2 -норма, в качестве весов используется функция Гаусса G_{σ_w} с центром в центре изображения (в результате экспериментов было получено, что наилучшие результаты получаются при $\sigma_w = 8$). Таким образом во-первых увеличивается вклад пикселей, которые находятся ближе к центру контура, и во-вторых достигается единообразность обработки всех поворотов контура. Для упрощения анализа результатов разложения и увеличения эффективности метода вместо умножения нормы на весовую функцию на нее домножаются элементы словаря и целевой вектор (взвешивание также присутствует на всех соответствующих примерах изображений).

На рис. 1.11 приведены примеры разложения блоков изображений с эффектом ложного оконтуривания, возникающим в результате различных причин.

Приведем краткое описание блоков и анализ их разложения, по порядку сверху вниз:

- Блок со смоделированным эффектом ложного оконтуривания. Эффект ложного оконтуривания отделен практически полностью.
- Блок с сильным шумом. Вклад эффекта ложного оконтуривания в представления близок к нулю.
- Блок с эффектом ложного оконтуривания, полученным в результате сжатия изображения алгоритмом JPEG2000. Эффект ложного оконтуривания отделен частично.
- Блок с эффектом ложного оконтуривания, полученным в результате изменения разрешения с помощью фильтра Ланцоша. Эффект ложного оконтуривания отделен частично.

Полученное разложение блока изображения позволяет независимо анализировать чистые осцилляции возникающие от эффекта ложного оконтуривания, и остальные детали изображения. Так, можно предположить, что на изображении с эффектом ложного оконтуривания чистые осцилляции (слагаемое r в разложении (1.11)) будут более выражены, чем шум и другие искажения (слагаемое e в разложении (1.11)). Тогда для измерения уровня эффекта ложного оконтуривания блока изображения можно предложить следующую метрику:

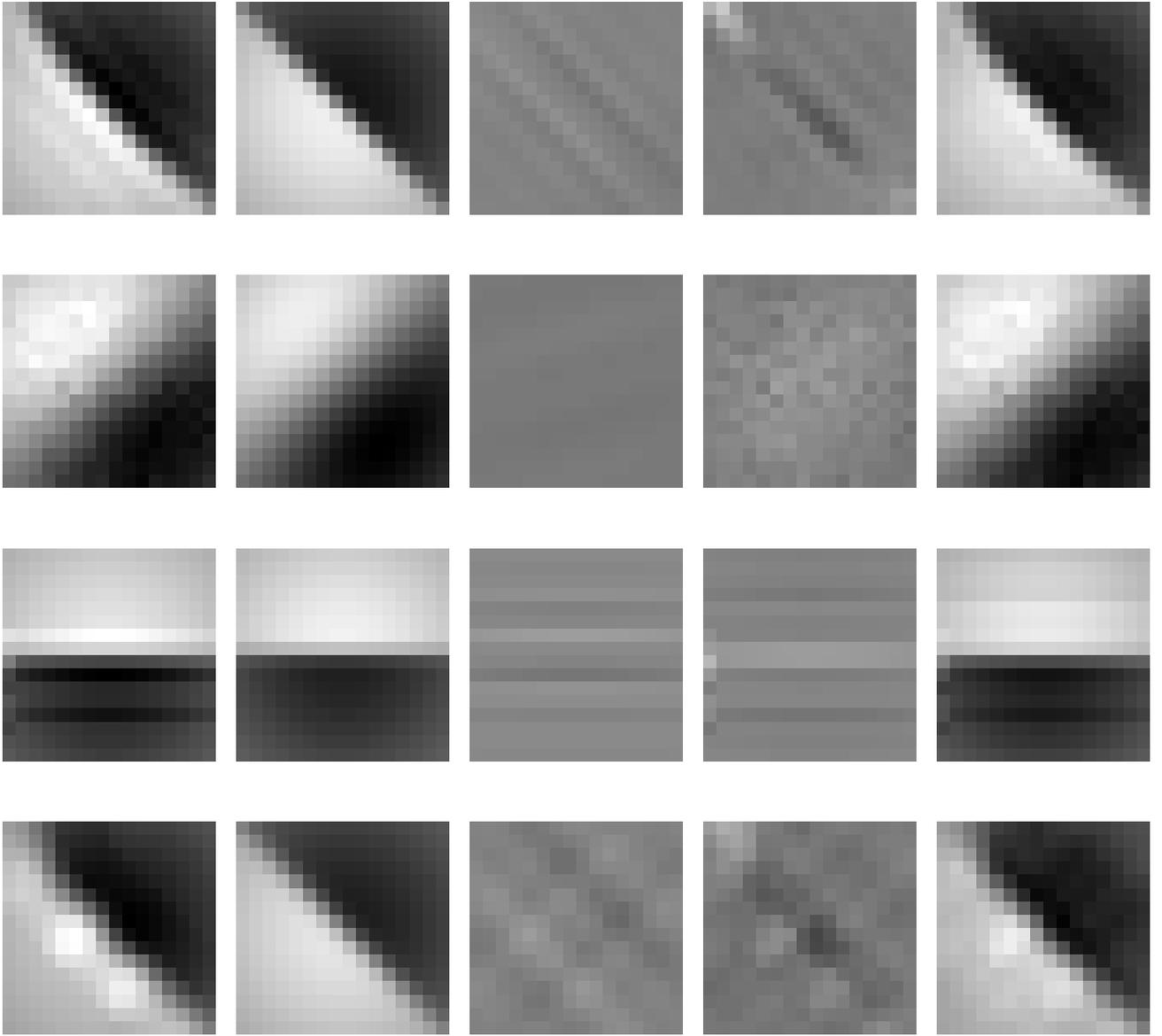


Рис. 1.11: Примеры разложения блоков изображений по формуле 1.11. Каждая строка соответствует одному блоку, по столбцам слева направо: исходный блок (y), вклад контура (b), вклад эффекта ложного оконтуривания (r), ошибка разложения (e), контур без эффекта ложного оконтуривания ($y - r$).

$$M_{ringing}(y) = \frac{\text{TV}(r)}{\text{TV}(e) + \delta}, \quad (1.12)$$

где δ — параметр сглаживания, а функция TV вычисляет полную вариацию соответствующего изображения следующим образом. Пусть заданы отсчеты $x_1 < \dots < x_n$ по горизонтальной оси, отсчеты $y_1 < \dots < y_m$ по вертикальной оси, и на них заданы значения яркости изображения

$I(x, y)$. Тогда полная вариация вычисляется по следующей приближенной формуле (более подробно полная вариация будет также обсуждаться в разделе 2.1):

$$\text{TV}(I) = \sum_{j=1}^m \sum_{i=2}^n |I(x_i, y_j) - I(x_{i-1}, y_j)| + \sum_{i=1}^n \sum_{j=2}^m |I(x_i, y_j) - I(x_i, y_{j-1})|.$$

Не все блоки на изображении можно использовать для вычисления уровня эффекта ложного оконтуривания. Если блок не содержит контур, или этот контур имеет форму сильно отличающуюся от прямой линии, данный метод не сможет его корректно обработать. В предложенном методе перед вычислением метрики выше используется фильтрация по алгоритму, приведенному в 1.6.

Алгоритм 1.6 Фильтрация блоков перед вычислением уровня эффекта ложного оконтуривания.

- 1: Для каждого блока строится его разложение по словарю с размытыми контурами D_1 и вычисляется величина ошибки при таком разложении.
 - 2: Если ошибка превышает некоторый порог, то блок отбрасывается.
 - 3: Рассматривается область в 3×3 пикселя вокруг блока (8 сдвигов блока на 1 пиксель). Если в этой окрестности присутствует хотя бы 2 блока с меньшей ошибкой разложения по словарю D_1 , то этот блок отбрасывается. Этот шаг нужен, чтобы не учитывать сдвинутые контуры, которые могут дать низкую ошибку, но не являются центрированными, поэтому не подходят для дальнейшей обработки. Использование порога в 2 блока помогает сделать метод более устойчивым.
 - 4: Все оставшиеся блоки считаются подходящими.
-

Далее сила эффекта ложного оконтуривания на изображении вычисляется как среднее значение метрики по всем подходящим для обработки блокам.

Анализ эффективности метода

Проведем анализ эффективности разработанного метода обнаружения эффекта ложного оконтуривания. Для анализа использовалась база изображений с различными типами искажений [24]. База содержит исходные изображения (без искажений), а также изображения с различными искажениями. Для тестирования разработанного метода эти искажения были поделены на две группы: искажения, не создающие эффект ложного оконтуривания, а также искажения, создающие эффект ложного оконтуривания. Это деление приведено в таблице 1.1. Для изображений с размытием используется размытие с функцией Гаусса в качестве ядра, а для изображений с шумом используется белый нормальный шум.

Изображения без эффекта ложного оконтуривания	Изображения с эффектом ложного оконтуривания
<ul style="list-style-type: none"> • Без искажений; • Размытие; • Шум. 	<ul style="list-style-type: none"> • Смоделированный эффект ложного оконтуривания R_d (эффект Гиббса); • Искажения в результате сжатия с помощью алгоритмов JPEG или JPEG 2000; • Искажения в результате изменения разрешения с помощью фильтра Ланцоша.

Таблица 1.1: Разбиение искажений для тестирования методов обнаружения эффекта ложного оконтуривания.

Для измерения качества работы произвольного классификатора используются три следующие характеристики: доля ложного принятия (false acceptance rate, FAR), доля верного принятия (true acceptance rate, TAR) и доля верных ответов (accuracy). Для введения этих характеристик обозначим N_{FA} — количество изображений, не содержащих эффект ложного оконтуривания, которые классификатор пометил как содержащие этот эффект, N_{TA} — количество изображений содержащих эффект ложного оконтуривания и помеченных верно, N_n — количество изображений без эффекта ложного оконтуривания, N_r — количество изображений с эффектом ложного оконтуривания, N_+ — количество верно классифицированных изображений ($N_+ = N_{TA} + (N_n - N_{FA})$), N — количество всех изображений ($N = N_n + N_r$). Тогда

$$FAR = \frac{N_{FA}}{N_n},$$

$$TAR = \frac{N_{TA}}{N_r},$$

$$Accuracy = \frac{N_+}{N}.$$

Метод разрабатывался для автоматического обнаружения изображений с эффектом ложного оконтуривания, поэтому высокие значения FAR для него нежелательны (иначе количество дополнительно необходимой ручной работы будет большим). Поэтому в качестве основной характеристики оценки алгоритма выбрана следующая: значение TAR при условии, что FAR не превышает 5%. Дополнительная характеристика измерения качества алгоритма — это его доля правильных ответов (accuracy).

Приведенный метод дает оценку уровня эффекта ложного оконтуривания, и для использования его в качестве классификатора необходимо выбрать некоторый порог значения этого уровня; изображения с уровнем эффекта ложного оконтуривания меньшим порога будут считаться не содержащими эффект ложного оконтуривания, а изображения с уровнем большим порога будут считаться содержащими этот эффект. Оценка качества работы метода с помощью описанных выше характеристик производится следующим образом. Для каждой из характеристик рассматриваются различные значения порога уровня эффекта ложного оконтуривания. В качестве итогового значения характеристики выбирается наилучшее из полученного набора значений.

Для начала проверим качество работы метода для произвольного эффекта ложного оконтуривания. Для этого было выбрано два набора изображений, составы этих наборов приведены в таблице 1.2.

Набор	Изображения без эффекта ложного оконтуривания	Изображения с эффектом ложного оконтуривания
Первый набор	<ul style="list-style-type: none"> • Без искажений. 	<ul style="list-style-type: none"> • Смоделированный эффект ложного оконтуривания R_d (эффект Гиббса).
Второй набор	<ul style="list-style-type: none"> • Без искажений; • Размытие; • Шум. 	<ul style="list-style-type: none"> • Смоделированный эффект ложного оконтуривания R_d (эффект Гиббса); • Сжатие с помощью JPEG или JPEG 2000; • Изменение разрешения с помощью фильтра Ланцоша.

Таблица 1.2: Наборы изображения для проверки качества метода для произвольного эффекта ложного оконтуривания.

Отметим, что второй набор является более сложным для классификации, так как алгоритму необходимо отличать эффект ложного оконтуривания от других искажений, таких как шум и размытие. В таблице 1.3 приведены результаты тестирования разработанного метода в сравнении с методом на основе вычисления полной вариации. Видно, что разработанный метод работает лучше метода, основанного на вычислении полной вариации, причем показывает высокую устойчивость к другим искажениям. Отметим, что метод, основанный на полной вариации [12] показал себя значительно хуже для второго набора. Это связано с тем, что в нем содержится

множество изображений с искажениями, которые данный метод плохо отличает от эффекта ложного оконтуривания (например, шум).

Метод	Первый набор		Второй набор	
	TAR (FAR 5%)	Accuracy	TAR (FAR 5%)	Accuracy
Разработанный	71.2%	92.3%	69.2%	85.5%
На основе TV [12]	59.6%	87.2%	5.8%	69.2%

Таблица 1.3: Результаты сравнения для смоделированного эффекта ложного оконтуривания, и эффекта ложного возникающего в результате различных алгоритмов обработки.

Далее проверим качество работы метода в применении к обнаружению эффекта ложного оконтуривания, полученного в результате сжатия изображения алгоритмом JPEG2000. Для этого было выбрано три набора изображений, составы этих наборов приведены в таблице 1.4.

Набор	Изображения без эффекта ложного оконтуривания	Изображения с эффектом ложного оконтуривания
Первый набор	<ul style="list-style-type: none"> Слабый шум и размытие. 	<ul style="list-style-type: none"> Сжатие с помощью JPEG 2000.
Второй набор	<ul style="list-style-type: none"> Сильный шум и размытие. 	<ul style="list-style-type: none"> Сжатие с помощью JPEG 2000.
Третий набор	<ul style="list-style-type: none"> Слабый шум и размытие; Сильный шум. 	<ul style="list-style-type: none"> Сжатие с помощью JPEG 2000.

Таблица 1.4: Наборы изображений для проверки качества метода для эффекта ложного оконтуривания, полученного в результате сжатия изображения алгоритмом JPEG2000.

Результаты сравнения разработанного метода и методов, основанных на анализе ширины контуров [21] и на анализе искажений контуров [29], приведены в таблице 1.5.

Из таблиц видно, что для наборов простых изображений (где необходимо отделять изображения с эффектом ложного оконтуривания от изображений без искажений) предложенный метод показывает хорошие результаты, сравнимые с результатами других методов. При этом он является гораздо более устойчивым к другим искажениям и успешно отличает их от эффекта ложного оконтуривания, а другие методы плохо справляются с этой задачей.

Метод	TAR (FAR 5%)	Accuracy
Первый набор		
Разработанный	63.5%	83.3%
Анализ ширины контуров [21]	21.2%	78.2%
Анализ искажений контуров [29]	73.1%	80.8%
Второй набор		
Разработанный	69.2%	85.6%
Анализ ширины контуров [21]	3.8%	62.5%
Анализ искажений контуров [29]	53.8%	76.9%
Третий набор		
Разработанный	75.0%	87.5%
Анализ ширины контуров [21]	13.5%	60.6%
Анализ искажений контуров [29]	1.9%	58.7%

Таблица 1.5: Результаты сравнения для эффекта ложного оконтуривания, полученного в результате сжатия изображения алгоритмом JPEG2000.

Приведем несколько примеров изображений и значений разработанной метрики для них. На рис. 1.12 приведены примеры изображений с смоделированным эффектом ложного оконтуривания. На рис 1.13 приведены примеры изображений с различными искажениями (с эффектом ложного оконтуривания и без него). На рис. 1.14 приведены примеры синтетических изображений с эффектом ложного оконтуривания (получаемого с помощью нерезкого маскирования), и без него.

Представленный метод можно использовать для контроля уровня эффекта ложного оконтуривания при использовании алгоритмов, которые могут его создавать. Рассмотрим, например, приложение, использованное в работе [19]. В нем для обработки дерматологических изображений необходимо проводить повышение резкости входных изображений. Одним из способов повышения резкости является метод нерезкого маскирования [46], и этот метод может создавать эффект ложного оконтуривания, поэтому его параметры необходимо выбирать так, чтобы уровень эффекта ложного оконтуривания был не слишком высоким. Так как изображение, к которому применяется повышение резкости, может иметь другие искажения (например, оно, как правило, является размытым), необходимо использовать метод оценки, который устойчив к подобным искажениям; например, таковым является представленный метод.



(a) Исходное изображение, $M = 0.671$.



(b) $d = 1.5$, $M = 0.766$.

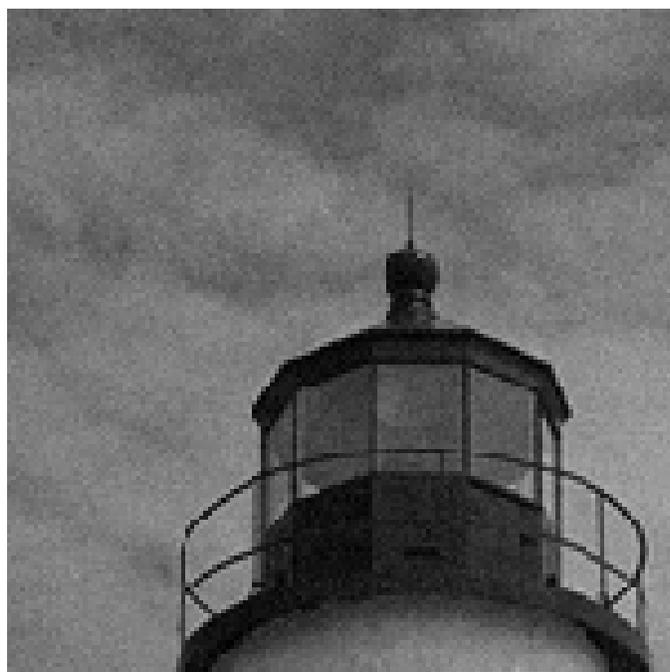


(c) $d = 2$, $M = 0.827$.



(d) $d = 3$, $M = 0.901$.

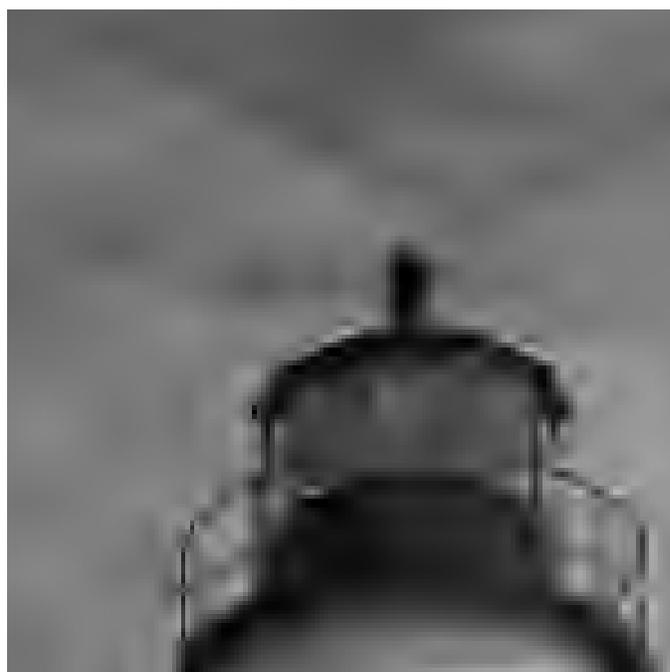
Рис. 1.12: Примеры изображений с смоделированным эффектом ложного оконтуривания и значений разработанной метрики для них.



(a) Шум, $M = 0.488$.



(b) Размытие и слабый шум, $M = 0.619$.



(c) JPEG2000, $M = 0.824$.



(d) Фильтр Ланцоша, $M = 0.925$.

Рис. 1.13: Примеры изображений с различными искажениями (с эффектом ложного оконтуривания и без него) и значений разработанной метрики для них.

(a) $M = 0.728$.(b) $M = 0.932$.(c) $M = 0.726$.(d) $M = 0.985$.

Рис. 1.14: Примеры синтетических изображений с применением нерезкого маскирования и значений разработанной метрики для них.

Глава 2

Подавление эффекта ложного оконтуривания на изображениях

В данной главе рассматривается задача подавления эффекта ложного оконтуривания. Сначала рассматриваются другие работы, предлагающие различные методы подавления эффекта ложного оконтуривания. Далее рассматривается задача построения словарей в методе разреженных представлений. Описывается один из часто используемых на практике алгоритмов построения словарей, а также основанный на нем алгоритм построения объединенных словарей. Также приводятся результаты исследования связи разреженности и взаимной согласованности для метода разреженных представлений. В конце приводятся описания разработанных алгоритмов подавления эффекта ложного оконтуривания. Первый алгоритм моделирует эффект ложного оконтуривания как преобразование изображений с помощью объединенных словарей. Вторым алгоритмом основан на влиянии эффекта ложного оконтуривания на разреженность представлений: сначала приводятся результаты исследования этой связи, а потом дается описание алгоритма, моделирующего эффект ложного оконтуривания с помощью независимых словарей. Далее приводятся результаты экспериментов с разработанными методами, и сравнение качества их работы между собой и с другими методами.

2.1 Методы подавления эффекта ложного оконтуривания на изображениях

При решении задачи подавления эффекта ложного оконтуривания на вход алгоритму подается изображение, про которое известно, что оно искажено эффектом ложного оконтуривания (с

известным уровнем эффекта ложного оконтуривания). Задача состоит в том, чтобы восстановить изображение так, чтобы оно было как можно ближе к исходному неискаженному изображению.

Подавлению эффекта ложного оконтуривания на изображениях, получаемого в результате их сжатия алгоритмами JPEG или JPEG2000, посвящено множество статей, см. например [47–49]. С другой стороны, существует не так много работ, которые рассматривают эффект ложного оконтуривания, получаемый из других источников [31, 50, 51], а также работ, рассматривающих чистый эффект ложного оконтуривания [14, 52]. Последние основаны на связи эффекта ложного оконтуривания и полной вариации [2], и минимизации полной вариации для улучшения изображения [53]. Рассмотрим этот подход более подробно.

Полная вариация произвольной дифференцируемой функции в области A может вычисляться по формуле:

$$\text{TV}(f) = \int_A \|\nabla f(x)\|_2 dx.$$

Для двумерной функций в прямоугольной области $x \in [a; b]$, $y \in [c; d]$ эту формулу можно переписать так:

$$\text{TV}(f) = \int_a^b \int_c^d \sqrt{f_x^2 + f_y^2} dx dy. \quad (2.1)$$

Для того, чтобы применить эту функцию к изображениям, необходимо рассмотреть ее приближение для дискретного набора отсчетов. Пусть заданы отсчеты $x_1 < \dots < x_n$ по горизонтальной оси, отсчеты $y_1 < \dots < y_m$ по вертикальной оси (шаг сетки равен 1), и на них заданы значения яркости изображения $I(x, y)$. Частные производные можно приблизить по следующим формулам:

$$\begin{aligned} \frac{\partial I}{\partial x}(x_i, y_j) &= I(x_i, y_j) - I(x_{i-1}, y_j), \\ \frac{\partial I}{\partial y}(x_i, y_j) &= I(x_i, y_j) - I(x_i, y_{j-1}). \end{aligned}$$

Тогда формула (2.1) переписывается следующим образом:

$$\text{TV}_i(I) = \sum_{i=2}^n \sum_{j=2}^m \sqrt{(I(x_i, y_j) - I(x_{i-1}, y_j))^2 + (I(x_i, y_j) - I(x_i, y_{j-1}))^2}.$$

Мы получили формулу для вычисления изотропной (инвариантной по всем направлениям) полной вариации. На практике эта формула не всегда удобна, так как ее производная определена не во всех точках (что необходимо для градиентных методов оптимизации). Поэтому вместо нее часто используют анизотропную полную вариацию:

$$\mathrm{TV}_a(f) = \int_a^b \int_c^d \sqrt{f_x^2} + \sqrt{f_y^2} \, dx dy = \int_a^b \int_c^d |f_x|^2 + |f_y|^2 \, dx dy. \quad (2.2)$$

В дискретном случае анизотропная полная вариация может записываться так:

$$\mathrm{TV}_a(I) = \sum_{j=1}^m \sum_{i=2}^n |I(x_i, y_j) - I(x_{i-1}, y_j)| + \sum_{i=1}^n \sum_{j=2}^m |I(x_i, y_j) - I(x_i, y_{j-1})|.$$

Полная вариация широко используется для восстановления изображений [53–55], например, в задаче подавления шума, впервые этот подход был представлен в [53]. Пусть $I_0(x, y)$ — исходное изображение, а его зашумленная версия $I_n(x, y)$ вычисляется по формуле

$$I_n(x, y) = I_0(x, y) + \varepsilon,$$

где ε — белый гауссовский шум с нулевым средним и стандартным отклонением σ (известным в задаче). Тогда восстановленное изображение $I^*(x, y)$ предлагается искать как решение следующей задачи:

$$I^* = \min_I \mathrm{TV}_i(I) \quad \text{при условии}$$

$$\sum_{x,y} I(x, y) = \sum_{x,y} I_n(x, y), \quad \sum_{x,y} \frac{1}{2} (I(x, y) - I_n(x, y))^2 = \sigma.$$

В работе [53] эта система приводится к параболическому уравнению, для которого предлагается приближенный численный метод решения. Показывается, что метод позволяет эффективно удалять шум на изображениях.

Полная вариация также тесно связана с эффектом Гиббса, и, соответственно, с эффектом ложного оконтуривания. Осцилляции эффекта Гиббса значительно усиливают полную вариацию сигнала. Для одномерных аналоговых сигналов можно показать, что фильтрация сигнала с конечной полной вариацией с помощью фильтра высоких частот (1.2) позволяет сделать полную вариацию отфильтрованного сигнала бесконечной [2]. На этой связи основаны некоторые методы подавления эффекта ложного оконтуривания [14, 52, 56], рассмотрим их подробнее.

В этом подходе для изображения I значение полной вариации этого изображения $\mathrm{TV}(I)$ используется как мера выраженности эффекта ложного оконтуривания. Обозначим исходное изображение как I_0 , а его искаженную эффектом ложного оконтуривания версию как I_r . Тогда для восстановления изображения необходимо найти изображение I^* , которое с одной стороны мало отличается от исходного, но с другой стороны имеет не слишком большое значение полной вариации. Исходя из этих соображений задача поиска I^* может быть сформулирована так:

$$I^* = \arg \min_I (\|I - I_r\|_2^2 + \gamma \text{TV}_a(I)), \quad (2.3)$$

где параметр γ регулирует силу подавления эффекта ложного оконтуривания. Метод минимизации этой функции описан в разделе 3.2.2.

Основная проблема данного подхода заключается в том, что функционал полной вариации плохо отличает эффект ложного оконтуривания от текстур. Таким образом, часть текстур на изображении сильно искажается в результате подавления (см. примеры в разделе 2.3.4).

2.2 Построение словарей в методе разреженных представлений

2.2.1 Задача построения словаря

В разделе 1.2 была описана общая задача разреженных представлений и способы ее решения для заданного словаря. Рассмотрим теперь задачу построения словаря по набору данных [11]. Эта процедура также часто называется обучением словаря.

Предположим, что мы рассматриваем различные сигналы y , которые принадлежат некоторому семейству $\mathcal{M} \subset \mathbb{R}^n$ (все сигналы содержат n отсчетов), и задача состоит в том, чтобы построить словарь для представления любых сигналов этого семейства. Например, такими семействами могут быть фотореалистичные изображения или МРТ-изображения; наборы элементарных сигналов в словаре для различных семейств будут различны [11, 35, 57].

Будем говорить, что \mathcal{M} является разреженным семейством, если любой сигнал $y \in \mathcal{M}$ выражается как

$$y = D_0 c + e, \quad (2.4)$$

где

- $D_0 \in \mathbb{R}^{n \times m}$ — неизвестный словарь (параметр семейства), m — размер словаря;
- c — неизвестное представление сигнала, для которого выполняется $\|c\|_0 \leq k$ (k — параметр семейства);
- e — аддитивный шум, для которого выполняется $\|e\|_2^2 \leq \varepsilon$ (ε — параметр семейства).

Отметим, что такие семейства можно рассматривать также как модели для создания синтетических сигналов. Действительно, если задать некоторые распределения для представлений c и шума ε , то для синтеза нового сигнала достаточно сгенерировать случайные вектора c и ε , и построить сигнал по формуле (2.4).

На практике обычно нельзя определить, является ли данное семейство \mathcal{M} разреженным семейством. Однако эффективность методов разреженных представлений показывает, что такие модели достаточно хорошо приближают реальные данные. Поэтому в дальнейшем будем предполагать, что рассматриваемые семейства являются разреженными семействами. Будем также считать, что нам известна одна из величин k или ε . На практике они обычно неизвестны и являются параметрами алгоритмов, которые необходимо выбирать исходя из априорных знаний о модели, а также по результатам экспериментов.

Задача построения словаря для семейства \mathcal{M} заключается в нахождении такого словаря D , чтобы с его помощью можно было как можно более точно представить сигналы этого семейства. Для этого необходимо иметь набор сигналов $\{y_i\}_{i=1}^N$, $y_i \in \mathcal{M}$, и словарь строится так, чтобы он наиболее точно приближал эти сигналы. Для удобства постановки задачи вместе со словарем ищется также и набор представлений $\{c_i\}_{i=1}^N$ для всех обучающих сигналов. Размер словаря m обычно заранее фиксируют, причем так, чтобы было выполнено $m \gg n$.

Тогда при известном уровне шума ε мы получаем задачу

$$\min_{D, \{c_i\}_{i=1}^N} \sum_{i=1}^N \|c_i\|_0, \quad \text{при условии} \quad \|y_i - Dc_i\|_2^2 \leq \varepsilon, \quad i = 1, \dots, N. \quad (2.5)$$

В случае же, если известно ограничение на количество элементов в сигнале k мы получаем задачу

$$\min_{D, \{c_i\}_{i=1}^N} \sum_{i=1}^N \|y_i - Dc_i\|_2^2, \quad \text{при условии} \quad \|c_i\|_0 \leq k, \quad i = 1, \dots, N. \quad (2.6)$$

Для удобства приведенные задачи обычно записывают в матричной форме. Обозначим Y — матрица, в которой по столбцам расположены сигналы для обучения:

$$Y = [y_1, y_2, \dots, y_N].$$

Через C обозначим матрицу, в которой по столбцам расположены представления этих сигналов:

$$C = [c_1, c_2, \dots, c_N].$$

Также введем псевдо-норму $\|\cdot\|_0$ для матриц:

$$\|C\|_0 = \max_j \|c_j\|_0,$$

где c_j — j -й столбец матрицы C .

Также через $\|\cdot\|_F$ будем обозначать норму Фробениуса:

$$\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}.$$

Тогда задачу (2.5) можно переписать так:

$$\min_{D,C} \|C\|_0, \quad \text{при условии} \quad \|Y - DC\|_F^2 \leq \varepsilon. \quad (2.7)$$

А задачу (2.6) можно переписать так:

$$\min_{D,C} \|Y - DC\|_F^2, \quad \text{при условии} \quad \|C\|_0 \leq k. \quad (2.8)$$

2.2.2 Алгоритм построения словарей

Задачи (2.7) и (2.8) являются NP-полными, аналогично задачам разреженных представлений (1.7) и (1.8). Поэтому для их решения используются приближенные алгоритмы.

В данной работе для построения словарей используется алгоритм K-SVD [58]. Основная идея этого алгоритма заключается в поочередном решении задачи относительно разных групп переменных так, чтобы при фиксированных остальных переменных для задачи существовал алгоритм решения. Отметим, что при этом эти задачи не обязательно решать точно, достаточно того, чтобы на каждой итерации текущее решение улучшалось.

Рассмотрим задачу (2.8), в которой параметрами минимизации являются словарь D и набор представлений C . Зафиксируем словарь D и рассмотрим задачу относительно представлений C :

$$\min_{\{c_i\}_{i=1}^N} \sum_{i=1}^N \|y_i - Dc_i\|_F^2, \quad \text{при условии} \quad \|c_i\|_0 \leq k, \quad i = 1, \dots, N.$$

Эту задачу можно переписать следующим образом:

$$\sum_{i=1}^N \min_{c_i} \|y_i - Dc_i\|_F^2, \quad \text{при условии} \quad \|c_i\|_0 \leq k, \quad i = 1, \dots, N.$$

Таким образом задача распадается на N независимых подзадач разреженных представлений вида (1.8), которые можно решать методами, описанными в разделе 1.2.2.

На следующем шаге можно было бы зафиксировать представления X , и тогда задача обновления словаря свелась бы к задаче наименьших квадратов. В этом случае мы получим алгоритм MOD [59], однако другой подход, описанный ниже, на практике дает более качественные результаты.

Рассмотрим один столбец словаря D с индексом j_0 как параметр и зафиксируем все остальные столбцы. Обозначим столбец словаря с индексом j как d_j и строку матрицы представлений с индексом j как x_j , тогда ошибку представления можно переписать следующим образом:

$$\|Y - DC\|_F^2 = \left\| Y - \sum_{j=1}^m d_j c_j \right\|_F^2 = \left\| \left(Y - \sum_{j \neq j_0} d_j c_j \right) - d_{j_0} c_{j_0} \right\|_F^2.$$

Обозначим матрицу ошибки без использования j_0 -того столбца как E_{j_0} :

$$E_{j_0} = Y - \sum_{j \neq j_0} d_j c_j.$$

Тогда задача относительно d_{j_0} переписывается так:

$$\min_{d_{j_0}, C} \|E_{j_0} - d_{j_0} c_{j_0}\|_2^2, \quad \text{при условии} \quad \|C\|_0 \leq k.$$

Оптимальные значения d_{j_0} и c_{j_0} для минимизации ошибки могут быть найдены с помощью сингулярного разложения матрицы E_{j_0} и использованием только наибольшего собственного значения. Однако такое представление может нарушить условие разреженности матрицы C . Поэтому при построении этого разложения необходимо учитывать только те элементы C , которые уже являются ненулевыми после предыдущей итерации. Обозначим через $c_{j_0}^R$ ограничение вектора c_{j_0} на его ненулевые элементы, и $E_{j_0}^R$ ограничение матрицы E_{j_0} на эти же элементы. Тогда задача переписывается следующим образом:

$$\min_{d_{j_0}, c_{j_0}^R} \|E_{j_0}^R - d_{j_0} c_{j_0}^R\|_2^2.$$

Эта задача уже решается с помощью сингулярного разложения матрицы $E_{j_0}^R$, и с его помощью обновляются соответствующие ячейки словаря D и матрицы представлений C . Отметим, что для ускорения алгоритма можно использовать приближение сингулярного разложения, более подробно описанное в [60].

Алгоритм K-SVD для решения задачи (2.8) представлен в псевдокоде 2.1. Для задачи (2.7) алгоритм аналогичен и здесь не приводится.

Алгоритм 2.1 Алгоритм K-SVD для решения задачи (2.8).

Вход: Матрица сигналов Y , ограничение на количество элементов в представлении k , количество элементов словаря m , количество итераций L ;

Выход: Словарь D и представления C ;

- 1: Инициализировать словарь D случайными значениями или случайными столбцами матрицы Y .
 - 2: **Цикл** по итерациям $i = 1, \dots, L$ **выполнять**
 - 3: Зафиксировать словарь D и решить задачу относительно C как задачу разреженных представлений для каждого сигнала;
 - 4: **Цикл** по столбцам $j_0 = 1, \dots, m$ **выполнять**
 - 5: Найти представления, использующие этот столбец, и вычислить $E_{j_0}^R$;
 - 6: Найти сингулярное разложение $E_{j_0}^R$ и обновить с его помощью d_{j_0} и c_{j_0} .
 - 7: **Конец цикла**
 - 8: **Конец цикла**
-

2.2.3 Объединенные словари

В задачах, где необходимо производить преобразование сигналов (в том числе в задаче подавления эффекта ложного оконтуривания), можно использовать подход, называемый обучением объединенных словарей. Одно из популярных применений данного подхода — повышение разрешения изображений [61]. Опишем его подробнее.

В данной модели рассматривается два разреженных семейства сигналов \mathcal{M}_1 и \mathcal{M}_2 , и предполагается наличие взаимно однозначного соответствия между сигналами из них. Например, для задачи подавления эффекта ложного оконтуривания это будет семейство обычных изображений и семейство изображений, искаженных эффектом ложного оконтуривания. Каждому обычному изображению будет соответствовать его искаженная версия.

Отметим, что для произвольных сигналов удаление или искажение высоких частот не может давать взаимно однозначное соответствие. Однако если каждое из семейств \mathcal{M}_1 и \mathcal{M}_2 содержит достаточно малую долю всех сигналов (например, фотореалистичных изображений значительно меньше, чем всех возможных изображений), то предположение о взаимной однозначности является допустимым.

Обозначим для семейства \mathcal{M}_1 его параметры как $D_{01}, k_1, \varepsilon_1$ и для семейства \mathcal{M}_2 как $D_{02}, k_2, \varepsilon_2$ (словарь, разреженность и уровень шума соответственно). Будем говорить, что эти семейства имеют объединенное представление, если у них совпадают величины шума и разреженности ($k_1 = k_2 = k, \varepsilon_1 = \varepsilon_2 = \varepsilon$), а также между сигналами существует взаимно однозначное соответствие $F : \mathcal{M}_1 \rightarrow \mathcal{M}_2$, причем для парных сигналов $y_1 \in \mathcal{M}_1$ и $y_2 \in \mathcal{M}_2, y_2 = F(y_1)$, выполнено:

$$y_1 = D_{01}c + e_1, \quad y_2 = D_{02}c + e_2, \quad (2.9)$$

где c — объединенное представление, e_1, e_2 — шум, причем $\|c\|_0 \leq k$ и $\|e_1\|_2^2 \leq \varepsilon, \|e_2\|_2^2 \leq \varepsilon$.

Далее, предположим, что рассматриваемые семейства имеют объединенное представление, и даны два набора сигналов для обучения $\{y_{1i}\}_{i=1}^N \subset \mathcal{M}_1, \{y_{2i}\}_{i=1}^N \subset \mathcal{M}_2$, причем известно, что сигналы выбраны так, что $y_{2i} = F(y_{1i})$. Также предположим, что нам известен параметр k или параметр ε (аналогично обычному обучению словарей на практике они обычно неизвестны и являются параметрами алгоритма). Тогда задача обучения объединенных словарей заключается в нахождении словарей D_1, D_2 и представлений $\{c_i\}_{i=1}^N$ таких, чтобы они описывали обучающий набор сигналов как можно лучше. Аналогично обозначениям в разделе 2.2.1 введем матрицы с сигналами Y_1, Y_2 и матрицу с представлениями C :

$$Y_1 = [y_{11}, y_{12}, \dots, y_{1N}], \quad Y_2 = [y_{21}, y_{22}, \dots, y_{2N}], \quad C = [c_1, c_2, \dots, c_N].$$

Тогда при известном ограничении на количество элементов в сигнале k можно рассматривать задачу

$$\min_{D_1, D_2, C} \left(\|Y_1 - D_1 C\|_F^2 + \|Y_2 - D_2 C\|_F^2 \right), \quad \text{при условии} \quad \|C\|_0 \leq k. \quad (2.10)$$

В случае же, если известен уровень шума ε , мы получаем задачу

$$\min_{D_1, D_2, C} \|C\|_0, \quad \text{при условии} \quad \|Y_1 - D_1 C\|_F^2 \leq \varepsilon, \quad \|Y_2 - D_2 C\|_F^2 \leq \varepsilon. \quad (2.11)$$

Для решения этих задач необходимо произвести дополнительные преобразования. Введем матрицу Y_J , в которой запишем конкатенированные вектора сигналов из матриц Y_1, Y_2 :

$$Y_J = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}.$$

Таким образом получим матрицу из такого же количества векторов, но большей размерности.

Введем словарь D_J , в котором также запишем конкатенированные столбцы исходных словарей:

$$D_J = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix}.$$

Тогда задачу (2.10) можно переписать так:

$$\min_{D_J, C} \|Y_J - D_J C\|_F^2 \quad \text{при условии} \quad \|C\|_0 \leq k.$$

Задачу (2.11) можно переписать так:

$$\min_{D_J, C} \|C\|_0, \quad \text{при условии} \quad \|Y - D_J C\|_F^2 \leq \varepsilon.$$

Можно видеть, что полученные задачи эквивалентны обычным задачам обучения словарей, которые можно решать алгоритмом из раздела 2.2.2. После выполнения обучения для получения требуемых словарей D_1, D_2 достаточно просто извлечь соответствующие компоненты из словаря D_J .

Объединенные словари позволяют проводить преобразование сигнала из одного семейства в соответствующий ему сигнал из другого семейства. Для этого воспользуемся условием (2.9). Пусть нам дан сигнал y_1 , и мы хотим получить соответствующий ему сигнал y_2 . Построим по сигналу y_1 его разреженное представление с помощью задачи (1.7) или (1.8), $y_1 \approx D_1 c$. По этому представлению далее можно приближенно восстановить y_2 , так как $y_2 \approx D_2 c$. Полный алгоритм преобразования в случае известного ограничения на количество элементов k приведен в 2.2, для известного ограничения ε на уровень шума алгоритм аналогичен и не приводится.

Алгоритм 2.2 Алгоритм преобразования сигналов для объединенных словарей.

Вход: Объединенные словари D_1, D_2 , ограничение на количество элементов в представлении k , входной сигнал y_1 из семейства \mathcal{M}_1 .

Выход: Приближение преобразованного сигнала y_2^* из семейства \mathcal{M}_2 , $y_2^* \approx y_2 = F(y_1)$.

- 1: Найти представление c для y_1 с помощью словаря D_1 решив задачу (1.8).
 - 2: Восстановить y_2^* с помощью этого представления: $y_2^* = D_2 c$.
-

2.2.4 Взаимная согласованность в методе разреженных представлений

При работе с разреженными представлениями большую роль играет степень разреженности используемых представлений [35], которая тесно связана со взаимной согласованностью сигналов в словаре. Введем сначала понятие взаимной согласованности сигналов. Пусть x, y — действительные векторы, тогда их взаимной согласованностью $\mu(x, y)$ будем называть следующую величину:

$$\mu(x, y) \stackrel{\text{def}}{=} \frac{|(x, y)|}{\|x\|_2 \|y\|_2}.$$

Далее, пусть D — словарь, состоящий из столбцов d_1, \dots, d_m , тогда его взаимная согласованность определяется так [62]:

$$\mu(D) \stackrel{\text{def}}{=} \max_{i \neq j} \frac{|(d_i, d_j)|}{\|d_i\|_2 \|d_j\|_2} = \max_{i \neq j} \mu(d_i, d_j).$$

На практике, как правило, рост взаимной согласованности набора сигналов приводит к уменьшению оптимального количества ненулевых коэффициентов в представлениях для этого сигнала. Также наличие такой связи подтверждается тем, что в ряде теорем, связанных с разреженными представлениями, количество коэффициентов уменьшается с ростом взаимной согласованности словаря [62]. Рассмотрим, например, задачу (1.7) и предположим, что в ней существует решение c , удовлетворяющее неравенству

$$\|c\|_0 < \frac{1}{2} \left(1 + \frac{1}{\mu(D)} \right).$$

Тогда такое решение будет единственным. В этом случае при отсутствии шума ($\varepsilon = 0$) алгоритм ортогонального поиска соответствий (1.4) найдет точное решение. Аналогичные результаты имеют место и при наличии шума ($\varepsilon \neq 0$), их полные формулировки приведены в [62].

Таким образом, мы в дальнейшем будем считать, что между этими параметрами есть обратная зависимость (и это будет подтверждено экспериментами). Исследуем теперь, как ведет себя взаимная согласованность для различных сигналов [15].

Лемма. Пусть X, Y — случайные независимые гауссовские векторы размерности n с независимыми компонентами, $X, Y \sim \mathcal{N}(0, E)$. Тогда

$$\mathbb{E}(\mu(X, Y)) = \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n+1}{2})}.$$

Доказательство. Обозначим $X' = X/\|X\|_2, Y' = Y/\|Y\|_2$. Тогда

$$\mu(X, Y) = \frac{|(X, Y)|}{\|X\|_2\|Y\|_2} = \left| \left(\frac{X}{\|X\|_2}, \frac{Y}{\|Y\|_2} \right) \right| = |(X', Y')|.$$

Обозначим через S^{n-1} единичную сферу в n -мерном пространстве. Тогда распределение X' — это равномерное распределение на S^{n-1} , т.е. $X' \sim \mathcal{U}(S^{n-1})$ [63]. Аналогично $Y' \sim \mathcal{U}(S^{n-1})$. Тогда, обозначив через $p_{X'}(x)$ плотность X' , а через $p_{Y'}(y)$ — плотность Y' , получаем

$$\begin{aligned} \mathbb{E}(|(X', Y')|) &= \int_{S^{n-1}} \int_{S^{n-1}} |(x, y)| p_{X'}(x) p_{Y'}(y) dx dy \stackrel{1}{=} \frac{1}{A_n^2} \int_{S^{n-1}} \int_{S^{n-1}} |(x, y)| dx dy \stackrel{2}{=} \\ &= \frac{1}{A_n^2} \int_{S^{n-1}} \left(\int_0^{2\pi} \int_0^\pi \dots \int_0^\pi |\cos \phi_1| \sin \phi_{n-2} \sin^2 \phi_{n-3} \dots \sin^{n-2} \phi_1 d\phi_1 \dots d\phi_{n-2} d\theta \right) dy = \\ &= \frac{1}{A_n^2} \int_{S^{n-1}} \left(\frac{2\pi^{\frac{n-1}{2}}}{\Gamma(\frac{n-1}{2})} \int_0^\pi |\cos \phi_1| \sin^{n-2} \phi_1 d\phi_1 \right) dy = \frac{1}{A_n^2} \int_{S^{n-1}} \left(\frac{2\pi^{\frac{n-1}{2}}}{\Gamma(\frac{n-1}{2})} \cdot \frac{2}{n-1} \right) dy = \\ &= \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n+1}{2})}. \end{aligned}$$

1. $X', Y' \sim \mathcal{U}(S^{n-1})$, поэтому $p_{X'}(x) = p_{Y'}(y) = 1/A_n$, где $A_n = n\pi^{\frac{n}{2}}/\Gamma(\frac{n}{2} + 1)$ — площадь поверхности единичной сферы в n -мерном пространстве, $\Gamma(n)$ — гамма-функция.

2. Перейдем в интеграле по dx к сферическим координатам $\phi_1, \dots, \phi_{n-2}, \theta$, таким, что $0 \leq \phi_i \leq \pi, 0 \leq \theta \leq 2\pi$, причем выберем их таким образом, чтобы $(x, y) = \cos \phi_1$. \square

Исследуем, как ведет себя полученная функция при росте значений n . Обозначим:

$$M(n) = \frac{\Gamma(\frac{n}{2})}{\sqrt{\pi}\Gamma(\frac{n+1}{2})}.$$

Воспользовавшись тем, что $\Gamma(n + \alpha)/\Gamma(n + \beta) \stackrel{n \rightarrow \infty}{\sim} n^{\alpha-\beta}$ [64], получаем

$$M(n) \stackrel{n \rightarrow \infty}{\sim} \sqrt{\frac{2}{\pi}} \cdot \frac{1}{\sqrt{n}}.$$

Также покажем, что эта функция является строго убывающей. Для этого используем неравенство из [65]:

$$\frac{\Gamma(x+1)}{\Gamma(x+s)} > \left(x + \frac{s}{2}\right)^{1-s}, \quad x > 0, 0 < s < 1.$$

Легко показать, что из него следует неравенство

$$\frac{\Gamma(x + \frac{1}{2})}{\Gamma(x)} < \sqrt{x}, \quad x > 0.$$

Далее имеем

$$\frac{M(n+1)}{M(n)} = \frac{\Gamma(\frac{n+1}{2})\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n+2}{2})\Gamma(\frac{n}{2})} = \frac{\Gamma(\frac{n+1}{2})\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})\Gamma(\frac{n}{2})\frac{n}{2}} = \left(\frac{\Gamma(\frac{n}{2} + \frac{1}{2})}{\Gamma(\frac{n}{2})}\right)^2 \left(\frac{n}{2}\right)^{-1} < \left(\sqrt{\frac{n}{2}}\right)^2 \left(\frac{n}{2}\right)^{-1} = 1,$$

откуда следует, что $M(n+1) < M(n)$ для $n > 0$.

Покажем теперь, что на практике с ростом размера изображений их средняя согласованность уменьшается. На рис. 2.1 изображены графики зависимости средней взаимной согласованности для блоков реальных изображений (изображения из базы) и случайных изображений (белый шум). Видно, что средняя взаимная согласованность для блоков реальных изображений и случайных векторов ведет себя похожим образом и убывает с ростом размерности. Средняя согласованность для реальных изображений убывает медленнее, и это может быть связано с тем, что вероятностная масса для них в основном сосредоточена в меньшей области, чем вся n -мерная гиперсфера. Такой эффект могут давать различные ограничения на изображения, например, высокая корреляция между соседними пикселями.

2.3 Подавление эффекта ложного оконтуривания с помощью метода разреженных представлений

Представленный в предыдущей главе метод обнаружения эффекта ложного оконтуривания нельзя напрямую превратить в метод подавления эффекта ложного оконтуривания. Это связано с тем, что в задаче подавления необходимо обрабатывать все блоки изображения, и недостаточно рассмотреть только те, для которых строится надежное разложение по словарю. Используемый синтетический словарь не позволяет устойчиво работать с остальными блоками.

Поэтому предлагается использовать методы, основанные на обучении словаря. Они позволят подобрать более подходящий словарь, хорошо работающий для всех блоков. Ниже представлены два разработанных метода для подавления эффекта ложного оконтуривания, далее производится анализ их качества и областей применимости.

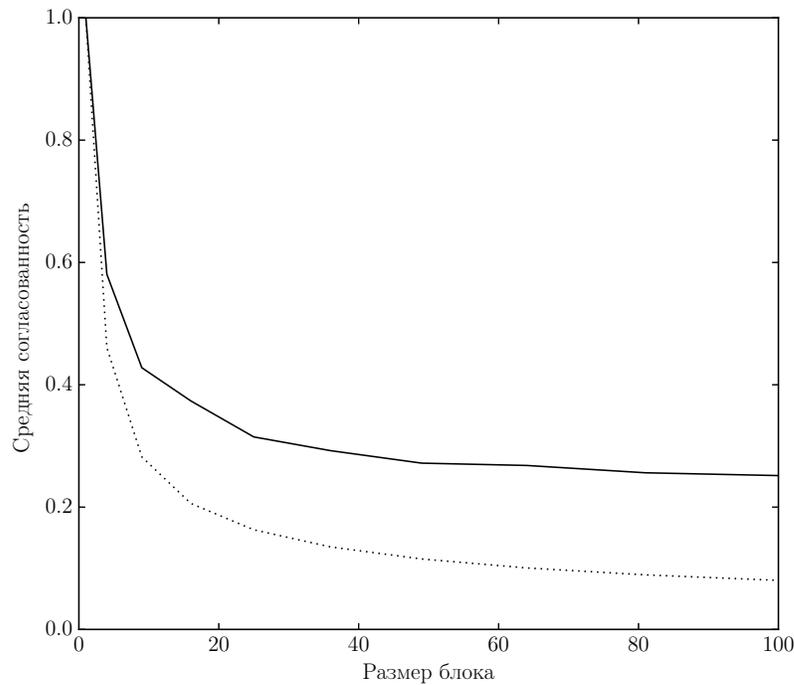


Рис. 2.1: Зависимость согласованности от размера блока для реальных (сплошная линия) и случайных (пунктирная линия) изображений.

2.3.1 Подавление эффекта ложного оконтуривания на изображениях с помощью объединенных словарей

Один из возможных способов подавления эффекта ложного оконтуривания основан на методе обучения объединенных словарей, описанном в разделе 2.2.3 [16]. Сначала необходимо создать набор изображений для обучения, аналогичных тем, которые в дальнейшем будут подаваться на вход алгоритму; для каждого изображения также строится соответствующее ему изображение с эффектом ложного оконтуривания. В целях эффективности изображения разбиваются на небольшие блоки (в данной работе размер в 6×6 пикселей давал наилучшие результаты), и каждый блок в обучении и тестировании обрабатывается независимо. Таким образом мы получаем набор пар блоков, и на этих блоках обучается объединенный словарь.

Далее обученный словарь используется для подавления эффекта ложного оконтуривания. Входное изображение разбивается на блоки, и для каждого блока с эффектом ложного оконтуривания восстанавливается соответствующий блок без эффекта ложного оконтуривания, например,

по алгоритму 2.2 для преобразования сигналов с помощью объединенных словарей. Полученные блоки усредняются для получения восстановленного изображения.

Такой подход дает достаточно хорошие результаты, но также был разработан набор модификаций, позволяющий заметно улучшить качество алгоритма.

Разреженность

На практике как правило используется одинаковое количество ненулевых коэффициентов при обучении словаря и при построении разреженных представлений. Однако для данного метода было обнаружено, что наилучшие результаты получаются при значительно отличающихся степенях разреженности (для обучения использовалось 10 коэффициентов, для применения — 2 коэффициента). Такой эффект связан с тем, что блоки с эффектом ложного оконтуривания содержат значительно меньше информации, чем блоки без него; более подробно этот эффект обсуждается в разделе 2.3.2.

Расширение словаря с помощью поворотов

При работе с эффектом ложного оконтуривания большую роль играют контуры на изображении, так как этот эффект наиболее заметен вблизи них. Было обнаружено, что некоторые контуры плохо представляются с помощью обученного словаря, так как в словаре отсутствуют блоки с контурами, имеющими угол наклона к горизонтали похожий на данный. При этом контуры на изображениях как правило инвариантны к поворотам (если данная форма контура может встретиться на естественном изображении, то и его повернутая версия — тоже), поэтому для улучшения качества работы алгоритма можно расширить словарь его повернутыми версиями. Обозначим за Rot_a оператор, который осуществляет поворот всех элементов словаря на угол a° , и за D_s и D_r обученные словари для обычных и искаженных блоков соответственно. Тогда расширенные словари строятся следующим образом:

$$D_s^{ext} = [D_s; Rot_{90}(D_s); Rot_{180}(D_s); Rot_{270}(D_s)],$$

$$D_r^{ext} = [D_r; Rot_{90}(D_r); Rot_{180}(D_r); Rot_{270}(D_r)].$$

Для расширения использовались только три данных поворота (90° , 180° , 270°), так как они могут быть применены к квадратным блокам без внесения ошибки интерполяции. Были

проведены эксперименты и с другими поворотами, но их использование не дает увеличения качества, вероятно как раз из-за ошибок интерполяции.

Отметим, что эту проблему также можно было бы попробовать решить путем расширения обучающей выборки с помощью тех же самых поворотов (как это сделано в разделе 2.3.3). Однако, так как в данной задаче обучающая выборка изначально достаточно большая, такой подход значительно увеличивает время обучения, поэтому он не использовался в данном методе.

Постобработка

При сравнении данного метода с методом, основанным на полной вариации, описанным в разделе 2.1, было обнаружено, что метод, основанный на полной вариации, дает более качественные результаты в окрестностях центров контуров. Для улучшения эффективности предлагается использовать гибридный метод, который разделяет области изображений на точки контуров и все остальные, и использовать разные алгоритмы для их обработки. Обозначим изображение, получаемое в результате обработки с помощью метода объединенных словарей, за $I_{jc}(x, y)$ (от «joint coding» — кодирование с объединенными словарями), изображение, полученное в результате обработки методом на основе полной вариации за $I_{tv}(x, y)$, и расстояние до центра ближайшего контура за $\rho_e(x, y)$ (см. описание ниже). Тогда гибридный результат получается по следующей формуле:

$$I(x, y) = \begin{cases} I_{tv}(x, y), & \text{если } \rho_e(x, y) \leq 3; \\ I_{jc}(x, y), & \text{иначе.} \end{cases}$$

Набор пикселей, лежащих в центрах контуров, строится с помощью применения алгоритма поиска контуров Канни [26] (см. также раздел 1.1.1). Далее для отделения осцилляций эффекта ложного оконтуривания и текстур от реальных контуров производится маскирование контуров. В данном методе использовался подход из [66, 67], в котором оставлялись только контуры, удовлетворяющие условию

$$g(x_0, y_0) > \max_{x, y} g(x, y) \phi((x - x_0)^2 + (y - y_0)^2),$$

где $g(x, y)$ — это модуль градиента в пикселе с координатами (x, y) (градиент вычисляется при поиске контуров в алгоритме Канни), $\phi(d^2) = 1/2 \cdot \exp(-d^2/2\sigma^2)$, σ — параметр алгоритма детектирования контуров (здесь использовалось значение $\sigma = 2$). Расстояние до ближайшего контура вычисляется с помощью евклидова преобразования расстояния [68].

2.3.2 Влияние эффекта ложного оконтуривания и других искажений изображений на разреженность представлений

В этом разделе будет рассмотрено, как эффект ложного оконтуривания, а также такие искажения, как размытие изображения и добавления шума к изображению, влияют на среднюю согласованность. Как обсуждалось в разделе 2.2.4, согласованность тесно связана с разреженностью представлений.

Исследуем, как эффект ложного оконтуривания влияет на среднюю согласованность.

Пусть $x \in \mathbb{R}^n$ — действительный вектор. Обозначим через $\mathcal{C}(x) \in \mathbb{R}^n$ дискретное косинусное преобразование от этого вектора и через $\mathcal{C}^{-1}(x)$ — обратное дискретное косинусное преобразование.

Оператор добавления эффекта ложного оконтуривания $G_k(x)$ для $0 < k < n$ определяется как оператор, который для вектора размерности n удаляет высокие частоты, оставляя только k координат, соответствующих нижним частотам:

$$G_k(x) = \mathcal{C}^{-1}(I_k \cdot \mathcal{C}(x)),$$

где $u \cdot v$ — поэлементное умножение векторов, и

$$(I_k)_i = \begin{cases} 1, & \text{если } i \leq k, \\ 0, & \text{если } i > k. \end{cases}$$

Отметим, что в этих обозначениях намеренно не учитываются пространственные координаты (как в обозначениях для изображений в разделе 1.1.1). Это сделано для упрощения преобразований, и не влияет на результат.

Утверждение. Пусть X, Y — случайные нормально распределенные независимые векторы размерности n : $X, Y \sim \mathcal{N}(0, E)$, где E — единичная матрица. Тогда

$$\mathbb{E}(\mu(G_k(X), G_k(Y))) > \mathbb{E}(\mu(X, Y)).$$

Доказательство. Из леммы 2.2.4 значение правой части искомого выражения равно $M(n)$. Теперь рассмотрим левую часть:

$$\begin{aligned}
\mathbb{E}(\mu(G_k(X), G_k(Y))) &= \mathbb{E}\left(\frac{|(G_k(X), G_k(Y))|}{\|G_k(X)\|_2 \cdot \|G_k(Y)\|_2}\right) = \\
&= \mathbb{E}\left(\frac{|(\mathcal{C}^{-1}(I_k \cdot \mathcal{C}(X)), \mathcal{C}^{-1}(I_k \cdot \mathcal{C}(Y)))|}{\|\mathcal{C}^{-1}(I_k \cdot \mathcal{C}(X))\|_2 \cdot \|\mathcal{C}^{-1}(I_k \cdot \mathcal{C}(Y))\|_2}\right) = \mathbb{E}\left(\frac{|(I_k \cdot \mathcal{C}(X), I_k \cdot \mathcal{C}(Y))|}{\|I_k \cdot \mathcal{C}(X)\|_2 \cdot \|I_k \cdot \mathcal{C}(Y)\|_2}\right) = \\
&= \mathbb{E}\left(\frac{|([\mathcal{C}(X)]_{1,\dots,k}, [\mathcal{C}(Y)]_{1,\dots,k})|}{\|[\mathcal{C}(X)]_{1,\dots,k}\|_2 \cdot \|[\mathcal{C}(Y)]_{1,\dots,k}\|_2}\right),
\end{aligned}$$

где $[X]_{1,\dots,k}$ — ограничение вектора на первые k координат. Здесь используется ортогональность оператора \mathcal{C}^{-1} .

Исследуем теперь, как распределены случайные величины $\mathcal{C}(X), \mathcal{C}(Y)$. Пусть T — матрица преобразования \mathcal{C} . Эта матрица является ортогональной. Запишем случайные векторы X, Y как один случайный вектор $Z = [X; Y] = (x_1, \dots, x_n, y_1, \dots, y_n)$. Этот вектор является гауссовским вектором размерности $2n$ с нулевым средним и единичной матрицей ковариации: $Z \sim \mathcal{N}(0, E)$. Пусть теперь Z_C — аналогичное соединение векторов $\mathcal{C}(X), \mathcal{C}(Y)$, т.е. $Z_C = [\mathcal{C}(X); \mathcal{C}(Y)]$. Тогда $Z_C = UZ$, где

$$U = \begin{pmatrix} T & 0 \\ 0 & T \end{pmatrix},$$

так как $\mathcal{C}(X) = TX$ и $\mathcal{C}(Y) = TY$. При этом

$$Z_C \sim \mathcal{N}(U0, UEU^T) = \mathcal{N}(0, UU^T) = \mathcal{N}(0, E).$$

Таким образом, Z_C — гауссовский вектор с независимыми координатами, а значит $\mathcal{C}(X), \mathcal{C}(Y)$ — независимые гауссовские векторы размерности n с независимыми компонентами, $\mathcal{C}(X), \mathcal{C}(Y) \sim \mathcal{N}(0, E)$. Тогда $[\mathcal{C}(X)]_{1,\dots,k}, [\mathcal{C}(Y)]_{1,\dots,k}$ — также независимые гауссовские векторы размерности k с независимыми компонентами. Значит, по лемме 2.2.4 получаем

$$\mathbb{E}(\mu(G_k(X), G_k(Y))) = \mathbb{E}\left(\frac{|([\mathcal{C}(X)]_{1,\dots,k}, [\mathcal{C}(Y)]_{1,\dots,k})|}{\|[\mathcal{C}(X)]_{1,\dots,k}\|_2 \cdot \|[\mathcal{C}(Y)]_{1,\dots,k}\|_2}\right) = M(k).$$

Так как $k < n$, из строгого убывания функции M получаем искомое неравенство. \square

Отметим, что при работе с изображениями обычно используется двумерное косинусное преобразование. Оно также является ортогональным (только это свойство преобразования использовалось при доказательстве), поэтому доказанное выше утверждение верно и для изображений. Таким образом, мы получаем, что при росте силы эффекта ложного оконтуривания

на изображении должна увеличиваться средняя согласованность его блоков, и, следовательно должна увеличиваться и разреженность.

Эффект размытия также может быть представлен как умножение преобразования Фурье изображения на некоторую маску. Например, в случае гауссовского размытия, это будет функция Гаусса с центром в точке, соответствующей нулевой частоте. Эта функция значительно уменьшает коэффициенты для одной части частот и полностью обнуляет коэффициенты для другой части частот (с учетом квантования). То есть количество компонент уменьшается и, следовательно, взаимная согласованность и разреженность должны увеличиваться аналогично эффекту ложного оконтуривания. Отметим также, что тогда в результате повышения резкости изображения взаимная согласованность и разреженность должны наоборот уменьшаться (это соображение необходимо для построения метода подавления эффекта ложного оконтуривания в разделе 2.3.3). При этом эффект ложного оконтуривания помимо осцилляций вносит небольшое размытие на изображение, поэтому к нему также можно применять повышение резкости.

Рассмотрим теперь эффект добавления белого шума на примере гауссовского белого шума. Как было сказано ранее, реальные изображения не совсем соответствуют модели независимых случайных компонент из-за связи между пикселями, и из-за этого взаимная согласованность для них в среднем выше, чем для случайных векторов. Однако при добавлении шума (который как раз является набором независимых компонент) связь между пикселями уменьшается и изображение становится более соответствующим модели с независимыми компонентами. Следовательно, взаимная согласованность должна уменьшаться и количество коэффициентов должно расти.

Продемонстрируем теперь приведенные выше соображения с помощью численных экспериментов. Эксперименты производились на изображениях из базы [24].

Для демонстрации влияния различных искажений на взаимную согласованность изображений были рассмотрены наборы случайных блоков изображений, и была посчитана средняя взаимная согласованность для пар блоков из этого набора. Для каждого искажения были рассмотрены различные значения уровня искажения и был построен график зависимости средней взаимной согласованности от уровня искажения. Под уровнем искажения подразумевается доля удаляемых частот в случае эффекта ложного оконтуривания (параметр d , см. раздел 1.1.1), степень размытия в случае размытия и сила шума в случае белого шума. Эти графики изображены на рис. 2.2, 2.3, 2.4 для эффекта ложного оконтуривания, размытия и шума соответственно. Видно, что для эффекта ложного оконтуривания и размытия взаимная согласованность растет с ростом уровня искажения, а для шума наоборот убывает, что соответствует соображениям, описанным выше.

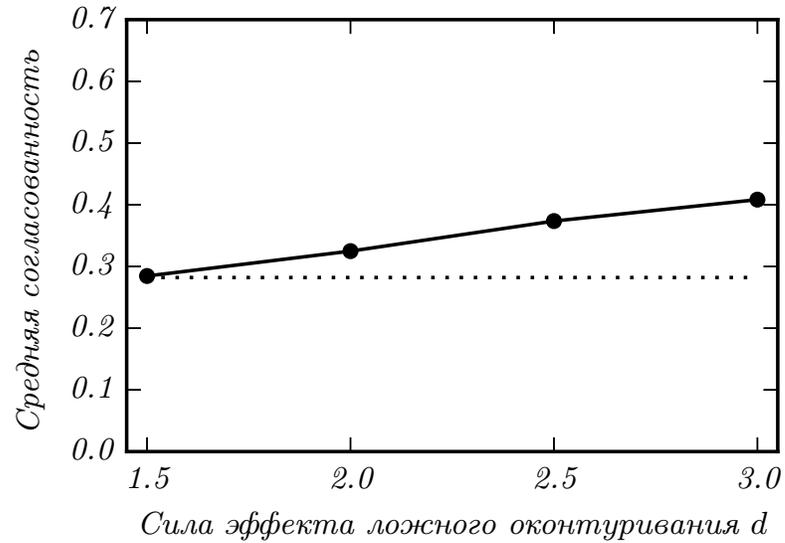


Рис. 2.2: Изменение взаимной согласованности при добавлении эффекта ложного оконтуривания; пунктиром обозначено значение взаимной согласованности для обычных изображений, а сплошной линией — для изображений с различной силой эффекта ложного оконтуривания d .

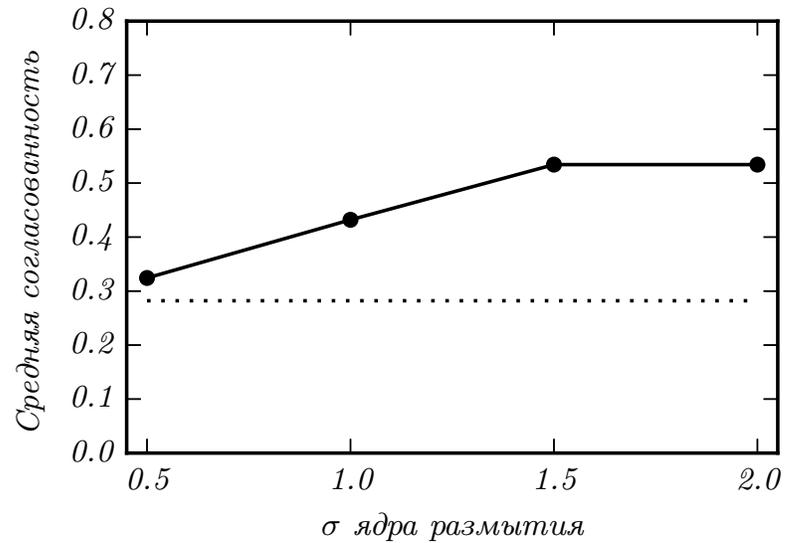


Рис. 2.3: Изменение взаимной согласованности при добавлении размытия; пунктиром обозначено значение взаимной согласованности для обычных изображений, а сплошной линией — для изображений с различной силой размытия.

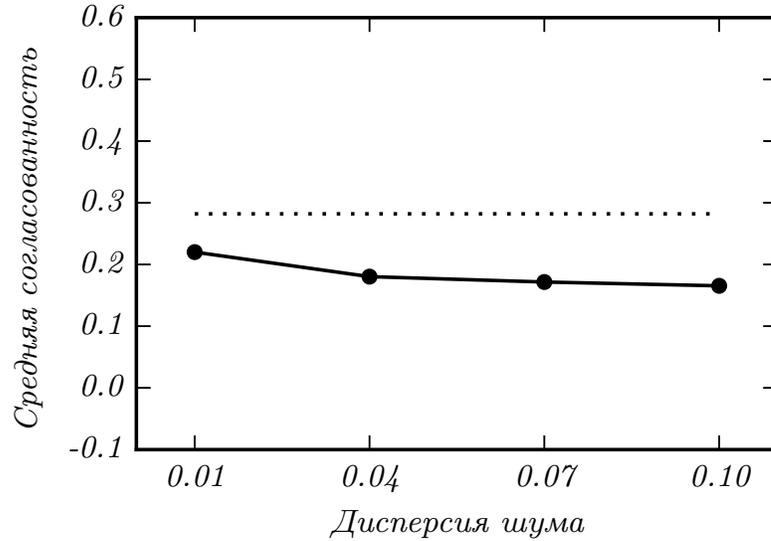


Рис. 2.4: Изменение взаимной согласованности при добавлении шума; пунктиром обозначено значение взаимной согласованности для обычных изображений, а сплошной линией — для изображений с различной дисперсией шума (считая, что яркость принимает значения от 0 до 1).

Для оценки подходящей степени разреженности для естественных изображений или изображений с одним из рассматриваемых искажений использовалось обучение словарей на выборке блоков изображений рассматриваемого типа, и вычислялась средняя разреженность по всем представлениям блоков обучающей выборки. Данная величина зависит не только от типа сигнала, но и от величины допустимой ошибки ε , поэтому обучение проводилось при различных значениях ошибки, и строилась зависимость разреженности от нее. Таким образом, характеристикой разреженности для данного типа и уровня искажения является график зависимости разреженности от ошибки ε . Ошибка ε приводится в терминах функции PSNR (см. определение в разделе 3.1.2).

Графики зависимости разреженности от ошибки изображены на рис. 2.5, 2.6, 2.7. Видно, что для эффекта ложного оконтуривания и размытия количество коэффициентов растет с ростом уровня искажения, а для шума наоборот убывает, что соответствует соображениям, описанным выше.

На рис. 2.8 изображены графики разреженности представлений для блоков обычных изображений, изображений с эффектом ложного оконтуривания, а также изображений с эффектом ложного оконтуривания после повышения резкости. Повышение резкости выполняется методом деконволюции Винера-Ханта [69] с Гауссовским ядром с различным значением параметра дис-

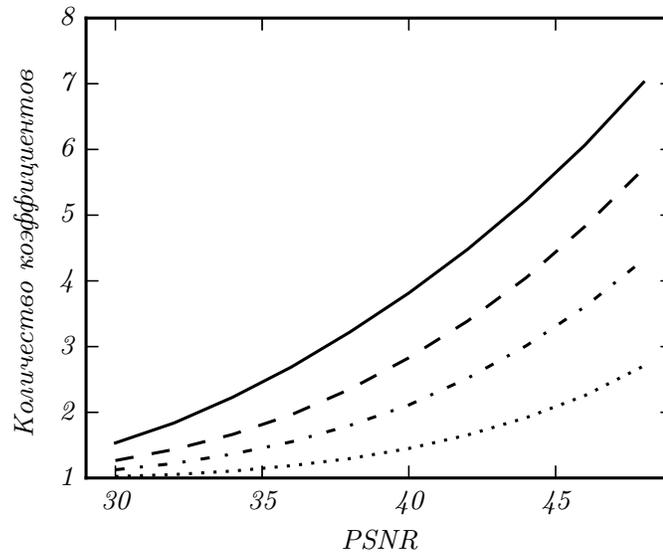


Рис. 2.5: Изменение разреженности при добавлении эффекта ложного оконтуривания; сплошная линия соответствует изображениям без искажения, штриховая, штрихпунктирная и точечная соответствуют изображениям с эффектом ложного оконтуривания с силой искажения $d = 1.5, 2, 3$ соответственно.

персии σ . Видно, что повышение резкости повышает количество необходимых коэффициентов, и этот эффект усиливается при более высоких значениях σ .

2.3.3 Подавление эффекта ложного оконтуривания на изображениях с помощью независимых словарей

В разделе 2.3.1 был предложен метод подавления эффекта ложного оконтуривания на основе разреженных представлений, и в этом методе на этапе обучения строился словарь одновременно для изображений с эффектом ложного оконтуривания и изображений без него. На этапе применения алгоритма он работал только с изображениями с эффектом ложного оконтуривания, и было показано, что оптимальные параметры разреженности при обучении и применении метода сильно отличались. Такой эффект предположительно был связан с тем, что для применения метода разреженных представлений к обычным изображениям и к изображениям с эффектом ложного оконтуривания необходимо использовать различную степень разреженности (как было показано в разделе 2.3.2).

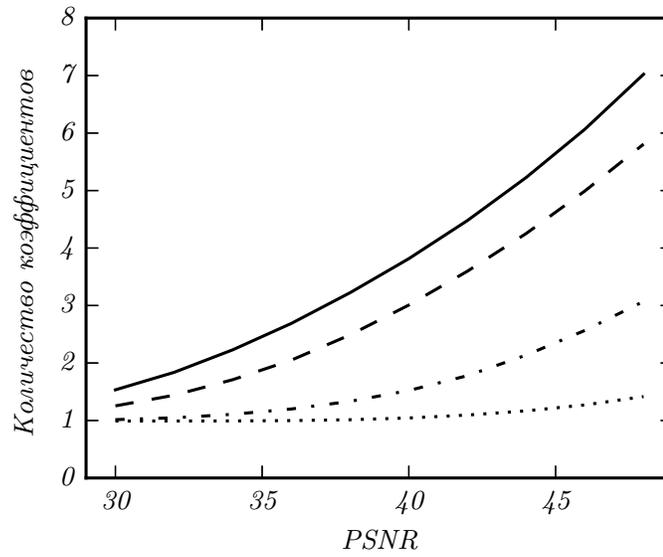


Рис. 2.6: Изменение разреженности при добавлении размытия; сплошная линия соответствует изображениям без искажения, штриховая, штрихпунктирная и точечная соответствуют изображениям с размытием с ядром с $\sigma = 0.5, 1, 2$ соответственно.

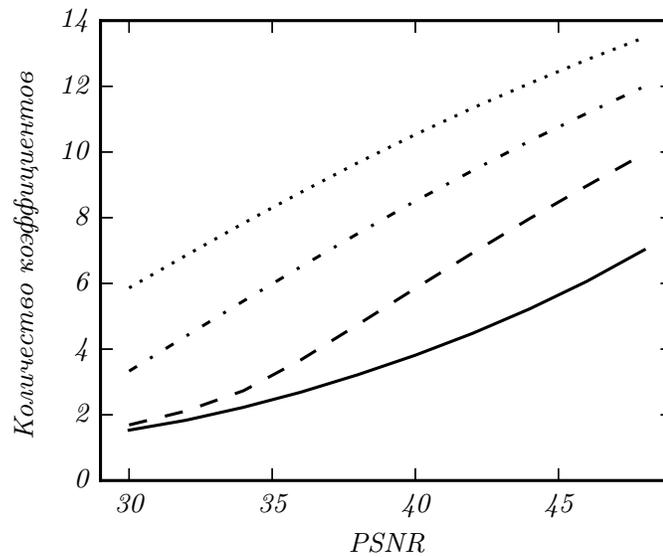


Рис. 2.7: Изменение разреженности при добавлении шума; сплошная линия соответствует изображениям без искажения, штриховая, штрихпунктирная и точечная соответствуют изображениям с шумом дисперсией 0.02, 0.05, 0.1 соответственно.

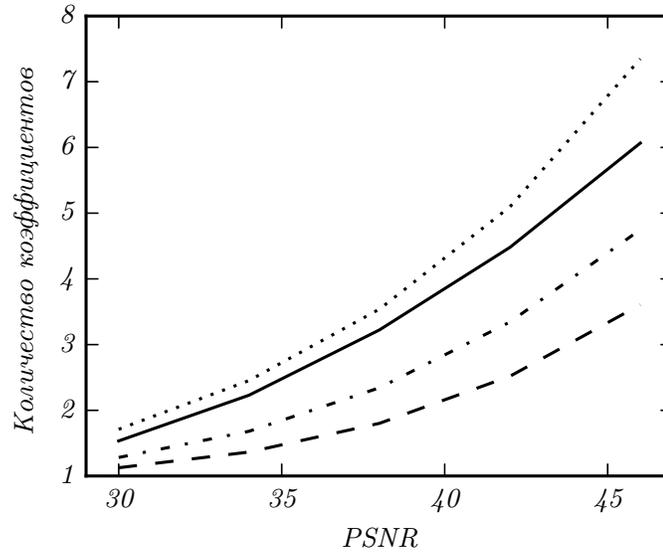


Рис. 2.8: Изменение разреженности при повышении резкости; сплошная линия соответствует изображениям без искажения, штриховая соответствует изображениям с эффектом ложного оконтуривания с силой искажения равной 2, штрихпунктирная и точечная соответствуют предыдущим изображениям с дополнительным повышением резкости с гауссовским ядром с σ , равным 1 и 1.5 соответственно.

Далее представлен другой алгоритм подавления эффекта ложного оконтуривания, также основанный на методе разреженных представлений [17]. В нем словари для обычных изображений и изображений с эффектом ложного оконтуривания обучаются отдельно, таким образом решая указанную выше проблему.

Опишем алгоритм, в котором для обработки каждого блока используются два заранее обученных независимых словаря.

Первый словарь D_1 моделирует изображения без эффекта ложного оконтуривания и обучается на любом наборе блоков естественных изображений.

Второй словарь D_2 моделирует чистый эффект ложного оконтуривания. Для его обучения используется разница между изображением с искусственно созданным эффектом ложного оконтуривания и соответствующим исходным изображением.

Для улучшения качества обучения словаря D_2 он обучается только на блоках, взятых из областей BEN (области вблизи наиболее ярко выраженных контуров, см. раздел 1.1.2), так как в этих областях находятся наиболее выраженные осцилляции. Такое ограничение значительно уменьшает размер обучающей выборки. Однако эффект ложного оконтуривания инвариантен

к поворотам, поэтому вместе с каждым изображением в выборке можно рассматривать и его повернутые версии. Это позволяет искусственно расширять размер обучающей выборки. Рассматривались только повороты на 90° , 180° и 270° во избежание ошибок интерполяции.

Из двух описанных выше словарей D_1 и D_2 составляется словарь $D = [D_1; D_2]$, получаемый объединением множества элементарных сигналов этих словарей (столбцы соответствуют сигналам). Далее для входного изображения y строится разреженное представление $c = [c_1; c_2]$ по объединенному словарю с помощью задачи (1.7):

$$y \approx Dc = D_1c_1 + D_2c_2.$$

Обозначим y_1 — искомое изображение без эффекта ложного оконтуривания, тогда $y = y_1 + y_2$, где y_2 — изображение с чистым эффектом ложного оконтуривания. При построении представления c , алгоритм использует элементарные сигналы из обоих словарей, причем требование на минимальность количества использованных сигналов приводит к тому, что эти сигналы используются наиболее эффективно. Поэтому для восстановления y_1 как правило используются элементарные сигналы словаря D_1 , а для восстановления y_2 — элементарные сигналы словаря D_2 . Исходя из этого положим $y_1 \approx D_1c_1$.

Для улучшения работы алгоритма к нему применяется несколько модификаций.

Во-первых, было обнаружено, что повышение резкости изображений с эффектом ложного оконтуривания при обучении словаря D_2 позволяет увеличить эффективность метода. Для повышения резкости используется алгоритм деконволюции Винера-Ханта [69]. Это предположительно связано с тем, что повышение резкости позволяет увеличить количество коэффициентов в представлениях для этого словаря, и, таким образом, сбалансировать количество коэффициентов на этапе тестирования (это приводит к тому, что для словарей D_1 и D_2 нужна примерно одинаковая разреженность). Кроме этого, такой эффект может также достигаться за счет того, что повышение резкости усиливает эффект ложного оконтуривания на изображении, и на этапе обучения это позволяет алгоритму лучше настроиться на отделение эффекта ложного оконтуривания от элементов обычных изображений.

Во-вторых, было обнаружено, что качество контуров может быть дополнительно улучшено при помощи добавления варпинга контуров [70]. Этот алгоритм находит на изображении контуры, и производит искажение исходной равномерной сетки, сдвигая точки вблизи контуров к центру ближайшего контура (величина сдвига определяется с помощью параметра силы варпинга A). Далее производится интерполяция обратно на равномерную сетку. Для каждого значения уровня эффекта ложного оконтуривания d были протестированы различные значения параметра силы

варпинга A , и было выбрано значение, дающее наилучшее качество. Найденное соответствие между этими параметрами приведено в таблице 2.1.

Значение d	Значение A
1.5	0.25
2	1.5
2.5	1.75
3	2

Таблица 2.1: Соответствие между силой эффекта ложного оконтуривания d и значением параметра силы варпинга A .

В-третьих, было обнаружено, что алгоритм дает хорошее качество в областях вблизи контуров, однако на всем изображении проигрывает методу, основанному на полной вариации. Наилучшее качество алгоритма достигается в областях наиболее ярко выраженных контуров на изображении (вероятно потому, что словарь D_2 обучался в этих областях). Исходя из этого, качество обработки можно улучшить, объединив эти два метода. Обозначим изображение, получаемое в результате обработки с помощью данного метода разреженных представлений за $I_{sd}(x, y)$ (от «separate dictionaries»), изображение, полученное в результате обработки методом на основе полной вариации за $I_{tv}(x, y)$, и множество точек на изображении, лежащих в объединении областей ВЕН и ВЕР за B (области базовых контуров, см. раздел 1.1.2). Тогда гибридный результат получается по следующей формуле:

$$I(x, y) = \begin{cases} I_{sd}(x, y), & \text{если } x \in B; \\ I_{tv}(x, y), & \text{иначе.} \end{cases}$$

2.3.4 Сравнение эффективности методов подавления эффекта ложного оконтуривания на изображениях

В данном разделе производится сравнение эффективности разработанных методов подавления ложного оконтуривания, а также производится анализ параметров их применимости [18]. Для сравнения была использована база изображений [24]. Изображения были поделены на две группы — для обучения и для тестирования.

Основной метрикой для оценки качества подавления эффекта ложного оконтуривания была выбрана величина SSIM [71] (см. определение в разделе 3.1.2). Дополнительно также приводятся

значения для метрики PSNR, однако отметим, что она не очень хорошо подходит для измерения качества в задачах обработки изображений [71]. Так как эффект ложного оконтуривания наиболее выражен в областях вблизи контуров, для более полного анализа метрики вычислялись в трех вариантах: на всем изображении, в областях BEN и в областях BEP.

На изображениях из базы был смоделирован эффект ложного оконтуривания R_d (эффект Гиббса) в соответствии с описанием в разделе 1.1.1, также для соответствия реальному эффекту ложного оконтуривания на них был добавлен небольшой шум. Эта процедура была проделана для разных значений d . Для краткости введем обозначения для методов: TV — метод на основе полной вариации (от «total variation», раздел 2.1), JC — метод на основе объединенных словарей (от «joint coding», раздел 2.3.1), SD — метод на основе независимых словарей (от «separate dictionaries», раздел 2.3.3). Результаты тестирования приведены в таблице 2.2 в терминах SSIM и в таблице 2.3 в терминах PSNR (указаны средние значения по тестовому набору). На рис. 2.9 приведены примеры обработки изображения различными методами.

Разработанные методы подавления эффекта ложного применимы для различных параметров входных изображений.

Метод JC показывает падение качества обработки изображений для более высоких уровней эффекта ложного оконтуривания. Это ухудшение связано с тем, что для блоков с эффектом ложного оконтуривания и блоков без него необходима разная степень разреженности представлений (как было показано в разделе 2.3.2). Метод SD не страдает от этой проблемы и дает хорошее качество обработки для высоких уровней эффекта ложного оконтуривания. На рис. 2.10 приведен пример сравнения качества обработки изображения с сильным эффектом ложного оконтуривания. Разница в качестве обработки особенно заметна вблизи ярко выраженных контуров.

Однако метод SD использует приближительную модель для эффекта ложного оконтуривания как суммы исходного изображения и чистого эффекта ложного оконтуривания, поэтому метод JC (использующий более общую модель) дает более качественные результаты для более низких уровней эффекта ложного оконтуривания. На рис. 2.11 приведен пример сравнения качества обработки изображения с более слабым эффектом ложного оконтуривания. Разница в качестве обработки особенно заметна в областях, содержащих мелкие детали.

Отметим также, что метод JC работает одинаково на всем изображении, а метод SD показывает наиболее высокое качество в областях вблизи контуров. Это связано с тем, что эффект ложного оконтуривания (а следовательно и эффект разницы разреженности представлений) наиболее выражен в областях вблизи контуров.

Область	Метод TV	Метод JC	Метод SD
$d = 1.5$			
Все изображение	0.9588	0.9592	0.9587
Области BEN	0.9880	0.9872	0.9875
Области ВЕР	0.9887	0.9885	0.9856
$d = 2$			
Все изображение	0.9343	0.9357	0.9344
Области BEN	0.9841	0.9812	0.9844
Области ВЕР	0.9754	0.9753	0.9771
$d = 2.5$			
Все изображение	0.9140	0.9231	0.9143
Области BEN	0.9803	0.9756	0.9805
Области ВЕР	0.9578	0.9578	0.9649
$d = 3$			
Все изображение	0.8915	0.8792	0.8922
Области BEN	0.9749	0.9658	0.9764
Области ВЕР	0.9436	0.9429	0.9584

Таблица 2.2: Эффективность подавления эффекта ложного оконтуривания для различных методов, указано среднее значение SSIM для тестового набора. Жирным шрифтом выделен наилучший результат в строке.

Во всех рассмотренных выше методах было исследовано два подхода к решению задачи разреженных представлений: с помощью алгоритма ортогонального поиска соответствий для постановки задачи с l_0 -нормой и с помощью алгоритма ISTA для постановки задачи с l_1 -нормой (алгоритмы описаны в разделе 1.2.2). Во всех случаях постановка задачи с использованием l_0 -нормы показала наилучшие результаты. На рис. 2.12 приведены примеры обработки одного и того же изображения с помощью обоих алгоритмов в методе JC без постобработки.

Область	Метод TV	Метод JC	Метод SD
$d = 1.5$			
Все изображение	32.91	33.14	32.89
Области BEN	35.79	36.16	35.54
Области BEP	32.29	32.27	31.43
$d = 2$			
Все изображение	30.43	30.45	30.44
Области BEN	34.50	34.24	34.55
Области BEP	29.12	29.04	29.75
$d = 2.5$			
Все изображение	28.96	28.87	28.99
Области BEN	33.62	33.05	33.63
Области BEP	26.92	26.85	28.37
$d = 3$			
Все изображение	27.78	27.57	27.84
Области BEN	32.64	31.36	32.82
Области BEP	25.36	25.20	27.53

Таблица 2.3: Эффективность подавления эффекта ложного оконтуривания для различных методов, указано среднее значение PSNR для тестового набора. Жирным шрифтом выделен наилучший результат в строке.



Рис. 2.9: Примеры обработки изображения различными методами. В порядке слева-направо и сверху-вниз приведены: исходное изображение, изображение с эффектом ложного оконтуривания ($d = 2$), результаты обработки методом TV, результаты обработки методом JC, результаты обработки методом SD.

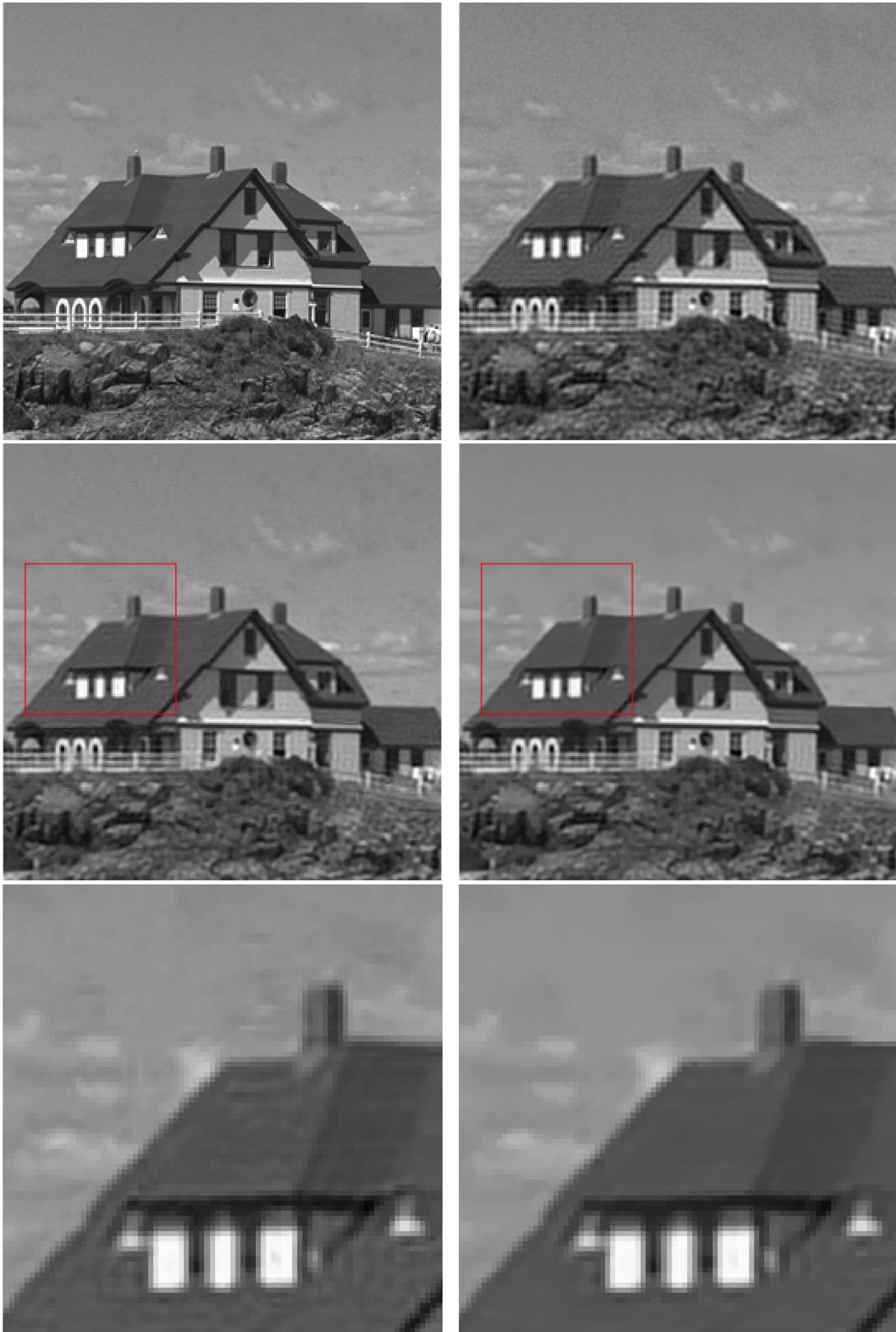


Рис. 2.10: Примеры обработки изображения с сильным эффектом ложного оконтуривания ($d = 2.5$). В порядке слева-направо и сверху-вниз приведены: исходное изображение, изображение с эффектом ложного оконтуривания, результаты обработки методом JC, результаты обработки методом SD, увеличенный фрагмент для метода JC, увеличенный фрагмент для метода SD.

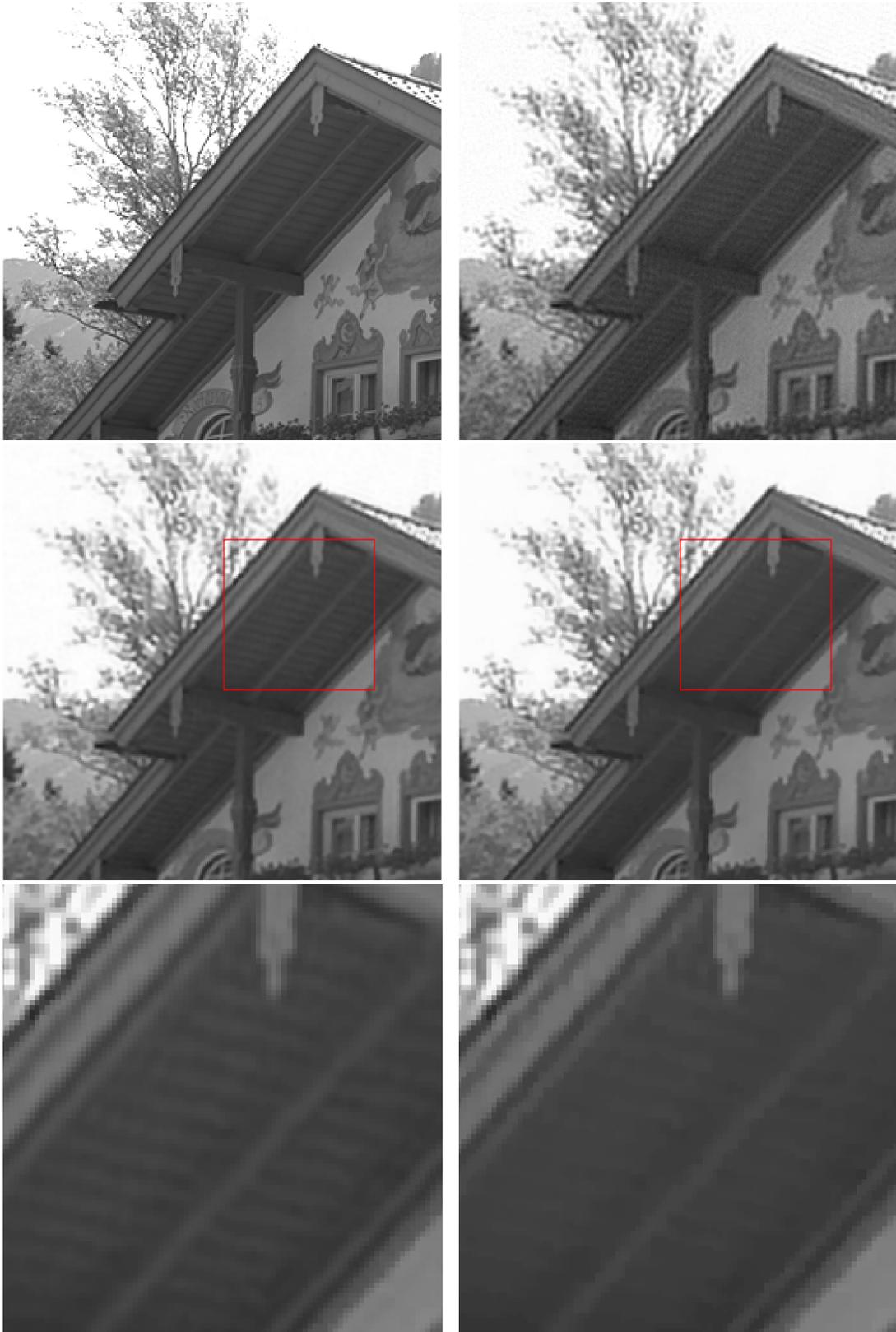


Рис. 2.11: Примеры обработки изображения со слабым эффектом ложного оконтуривания ($d = 2$). В порядке слева-направо и сверху-вниз приведены: исходное изображение, изображение с эффектом ложного оконтуривания, результаты обработки методом JC, результаты обработки методом SD, увеличенный фрагмент для метода JC, увеличенный фрагмент для метода SD.



(a) l_0 -норма, SSIM = 0.97.

(b) l_1 -норма, SSIM = 0.95.

Рис. 2.12: Примеры результатов обработки с использованием алгоритмов для l_0 -нормы и для l_1 -нормы. Используется метод JC без постобработки.

Глава 3

Программный комплекс

В работе были предложены алгоритмы, позволяющие производить обнаружение и подавление эффекта ложного оконтуривания на изображениях. Был разработан программный комплекс, реализующий эти алгоритмы. В данной главе приводится описание программной реализации указанных методов, а также описывается структура программного комплекса.

Программы были разработаны на языке программирования Python для операционных систем Linux, Windows и Mac OS. При разработке использовалась система научных библиотек SciPy [72], библиотека для построения графиков Matplotlib [73], библиотека методов машинного обучения Scikit-learn [74] и библиотека обработки изображений Scikit-image [75]. В экспериментах также использовалась система Jupyter notebook из пакета IPython [76]. Для разработки графического интерфейса использовалась библиотека Kivy [77].

Разработанные алгоритмы работают с изображениями в градациях серого. Для преобразования цветного изображения, поступающего в виде цветовых компонент R , G , B (интенсивности красного, зеленого и синего каналов соответственно) в изображение в градациях серого Y используется следующая формула [78]:

$$Y = 0.213R + 0.715G + 0.072B.$$

Алгоритмы могут работать с изображениями в форматах BMP, JPEG, JPEG 2000, PNG, TIFF.

Программный комплекс состоит из следующих частей:

- Набор вспомогательных модулей для работы с изображениями, эффектом ложного оконтуривания, и методом разреженных представлений.

- Программная реализация разработанных методов обнаружения и подавления эффекта ложного оконтуривания на изображениях.
- Графический интерфейс для методов обнаружения и подавления эффекта ложного оконтуривания на изображениях.

3.1 Вспомогательные модули

Вспомогательные модули имеют следующую структуру:

- Пакет `image`
 - Модуль `image.degrade`, содержит набор функций для моделирования различных искажений изображений.
 - Модуль `image.metrics`, содержит набор функций для вычисления различных характеристик изображений.
 - Модуль `image.patches`, содержит набор функций для разложения изображений на блоки, и комбинирования блоков для восстановления изображений.
- Пакет `sparse`
 - Модуль `sparse.algorithm`, содержит набор функций, реализующих алгоритмы для работы с разреженными представлениями.
 - Модуль `sparse.joint_coding`, содержит набор функций и классов для работы с объединенными словарями.

Для всех вспомогательных модулей также был создан набор юнит-тестов с помощью библиотеки `pytest`.

Опишем теперь функции в этих модулях.

3.1.1 Модуль `image.degrade`

Модуль `image.degrade` имеет следующую структуру:

```
image.degrade
    ringing(image, ringing_rate, reflect_borders)
    psf_camera_downsample(image, downsample_rate, base_sigma)
```

Функция **ringing** получает на вход изображение `image`, уровень эффекта ложного оконтуривания `ringing_rate` (равен значению параметра d в соответствии с обозначениями в разделе 1.1.1) и флаг использования отражений `reflect_borders` (см. ниже). Функция моделирует эффект ложного оконтуривания выбранного уровня (в соответствии с процессом описанным в разделе 1.1.1) и возвращает полученное изображение. Для выполнения дискретного преобразования Фурье используется алгоритм быстрого преобразования Фурье [1].

Разберем одну важную деталь для моделирования эффекта ложного оконтуривания. Использование дискретного преобразования Фурье для изображения конечного размера приводит к тому, что границы изображения фактически становятся контурами (так как изображение конечного размера продолжается на бесконечную плоскость по периодичности). Тогда при моделировании эффекта ложного оконтуривания границы также порождают осцилляции (см. пример на рис. 3.1, а), что является нежелательным эффектом, так как на практике такое как правило не происходит. Во избежание этого эффекта можно произвести отражение изображения вдоль всех его границ (см. пример на рис. 3.1, б), и производить моделирование эффекта ложного оконтуривания для увеличенного изображения. После добавления на изображение эффекта ложного оконтуривания дополнительные отраженные части удаляются. На рис. 3.1, с, изображен пример полученного изображения. За использование этой модификации в функции отвечает флаг использования отражений `reflect_borders`.

Также отметим, что хотя результат применения оператора R_d (см. раздел 1.1.1) не должен содержать мнимых компонент, так как фильтр l_d является эрмитовой функцией, т.е. $l_d(-x) = \overline{l_d(x)}$, на практике в результате погрешностей вычисления они все-таки возникают, и их необходимо удалять.

Функция **psf_camera_downsample** получает на вход изображение `image`, коэффициент понижения разрешения `downsample_rate` (обозначим его за q) и параметр `base_sigma` (обозначим его за σ_0) и производит понижение разрешения изображения моделируя процесс понижения разрешения для реальных фотокамер [79]. Сначала производится размытие изображения с Гауссовой функцией G_σ в качестве ядра, где $\sigma = \sigma_0 \sqrt{q^2 - 1}$ (как правило используется $\sigma_0 \in [0.4, 0.5]$). Далее производится уменьшение разрешения путем исключения из изображения пикселей с шагом q .

3.1.2 Модуль `image.metrics`

Модуль `image.metrics` имеет следующую структуру:

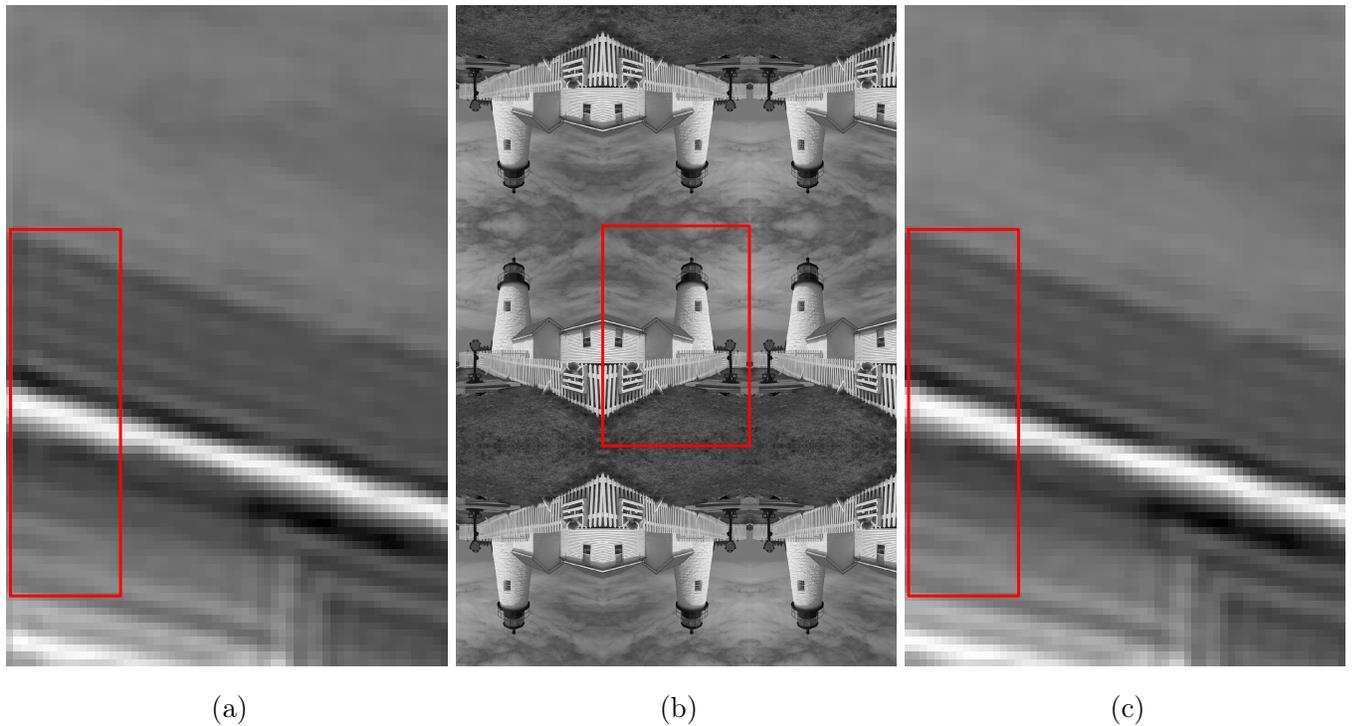


Рис. 3.1: Использование отражений для моделирования эффекта ложного оконтуривания. (a) Моделирование без использования отражений, красным выделена область с осцилляциями от границ изображения; (b) отражение изображения вдоль границ, красным выделены исходные границы изображения; (c) Моделирование с использованием отражений, красным выделена область аналогичная (a).

```

image.metrics
    mse(source_image, test_image, area)
    psnr(source_image, test_image, area)
    ssim(source_image, test_image, area)
    total_variation(image, area)

```

Функция `mse` получает на вход два изображения одинакового размера `source_image` (обозначим его как s) и `test_image` (обозначим его как t), а также ограничение области вычисления `area` (обозначим его как a), и вычисляет среднеквадратичное отклонение между ними: $MSE(s, t) = \|s - t\|_2^2 / WH$, где W, H — ширина и высота изображений соответственно. Также есть возможность ограничить вычисление на некоторую область, заданную бинарной маской a .

В таком случае среднеквадратичное отклонение вычисляется так: $\text{MSE}(s, t, a) = \|a \cdot (s - t)\|_2^2 / N$, где N — количество единиц в бинарной маске a .

Функция **psnr** получает на вход два изображения одинакового размера **source_image** (обозначим его как s) и **test_image** (обозначим его как t), а также ограничение области вычисления **area** (обозначим его как a), и вычисляет значение функции схожести PSNR (peak signal-to-noise ratio) для этих изображений:

$$\text{PSNR}(s, t, a) = 10 \log_{10} \left(\frac{M}{\text{MSE}(s, t, a)} \right),$$

где M — максимальное значение яркости, принимаемое пикселями на изображении, и функция MSE — среднеквадратичная ошибка, вычисляемая по формуле выше. Более высокие значения PSNR соответствуют большей схожести изображений.

Функция **ssim** получает на вход два изображения одинакового размера **source_image** (обозначим его как s) и **test_image** (обозначим его как t), а также ограничение области вычисления **area** (обозначим его как a) вычисляет значение меры схожести SSIM (structural similarity) для этих изображений [71]. Для вычисления этой функции оба изображения разбиваются на блоки (как правило размера 8×8 пикселей), и для каждой пары соответствующих блоков x, y вычисляются следующие величины: μ_x, μ_y — средние значения блоков x, y соответственно, σ_x^2, σ_y^2 — дисперсии блоков x, y соответственно, σ_{xy} — ковариация блоков x, y . Исходя из этих величин вычисляется SSIM-индекс для пары блоков x, y :

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

где c_1, c_2 — сглаживающие коэффициенты (см. [71]). Мера схожести SSIM полных изображений далее вычисляется как среднее значение SSIM-индекса всех пар их блоков. Также можно вычислять значение схожести SSIM только на некоторой области, усредняя значения только по набору блоков, заданных бинарной маской a .

Данная мера схожести была разработана как развитие для меры схожести PSNR, и в большей степени отражает особенности человеческого восприятия изображения [71].

Функция **total_variation** получает на вход изображение **image** и ограничение области **area** и вычисляет значение полной вариации изображения в данной области. Используется анизотропная формула полной вариации (2.2).

3.1.3 Модуль `image.patches`

Модуль `image.patches` имеет следующую структуру:

```
image.patches
    image_patches(image, patch_shape, step)
    transform_patchwise(image, patch_shape, patch_step, transform)
```

Функция `image_patches` получает на вход изображение `image`, размеры блока `patch_shape` и шаг для выбора блока `step` и извлекает из изображения все блоки заданного размера и с заданным шагом.

Функция `transform_patchwise` получает на вход изображение `image`, размеры блока `patch_shape`, шаг для выбора блока `step` и функцию преобразования `transform`, и производит поблочное преобразование изображения. Изображение разбивается на блоки, к каждому блоку применяется преобразование, и далее из этих блоков составляется новое изображение. Для пикселей, относящихся к нескольким накладывающимся блокам, используется среднее значение, полученное из всех накладывающихся блоков.

3.1.4 Модуль `sparse.algorithm`

Модуль `sparse.algorithm` имеет следующую структуру:

```
sparse.algorithm
    omp(dictionary, target, n_nonzero_coefs, tol)
    ista(dictionary, target, regularization, step_size, max_iters)
    learn_dictionary_ksvd(Y, n_nonzero_coefs, tol, n_components,
                          iteration_count, init_dictionary)
```

Функция `omp` получает на вход словарь `dictionary`, целевой вектор `target` и параметры `n_nonzero_coefs` и `tol`, и находит приближенное решение задачи разреженных представлений с помощью алгоритма `orthogonal matching pursuit` (см. раздел 1.2.2). Только один из параметров `n_nonzero_coefs` и `tol` может присутствовать (другой должен быть равен `None`), и он определяет, какая задача решается. Если присутствует параметр `n_nonzero_coefs`, то решается задача (1.8), и этот параметр задает максимальное количество ненулевых коэффициентов k в этой задаче.

Алгоритмы из данного модуля подробно описаны в разделе [2.2.3](#). Опишем интерфейс его функций.

Функция `learn_joint_dictionaries` получает на вход два набора сигналов `signals_first` и `signals_second`, а также метод для обучения словарей `dict_learning_method` (в виде функции) и набор параметров для обучения `dict_learning_parameters`, и производит на этих сигналах обучение объединенных словарей. Функция возвращает пару словарей, а также набор представлений для входных сигналов с помощью этих словарей.

Функция `joint_dictionaries_transform` получает на вход сигнал для преобразования `source_vector`, пару объединенных словарей `dictionary_first` и `dictionary_second` (обученных с помощью функции `learn_joint_dictionaries`), а также метод для построения разреженных представлений `sparse_coding_method` (в виде функции) и набор параметров для этого метода, и производит преобразование с помощью этих словарей в соответствии с алгоритмом [2.2](#).

3.2 Реализация разработанных методов

Реализация предложенных методов имеет следующую структуру:

- Модуль `ringing_detection`, содержит набор классов и функций для обнаружения эффекта ложного оконтуривания на изображениях.
- Модуль `ringing_suppression`, содержит набор классов и функций для подавления эффекта ложного оконтуривания на изображениях с использованием объединенных словарей и независимых словарей.

Опишем теперь функции и классы в этих модулях.

3.2.1 Модуль `ringing_detection`

Модуль `ringing_detection` имеет следующую структуру:

```
ringing_detection
    ringing_phantoms_dict(patch_shape, angles, ringing_rates)
    dictionary_error(patch, dictionary, l0_value)
    apply_to_patches(image, patch_shape, function)
    split_to_edge_and_ringing(image, dictionary, omp_parameters)
    build_image_decompositions(image, edges, dictionary, omp_parameters,
```

```

                                patch_shape)
aggregate_estimations(decomposition_data)
class RingingDetector
    __init__(dictionary, patch_shape)
    ringing_level(image)

```

Все алгоритмы, используемые в данном модуле, описаны в разделе 1.3.2. Опишем интерфейс функций и классов.

Функция **ringing_phantoms_dict** конструирует словарь из синтетических контуров с эффектом ложного оконтуривания и без него для дальнейшего использования в алгоритме обнаружения эффекта ложного оконтуривания. Функция получает на вход размер блока изображения **patch_shape**, список углов поворотов контуров **angles**, список уровней эффекта ложного оконтуривания **ringing_rates**. Для уменьшения влияния ошибок интерполяции контуры изначально создаются в более высоком разрешении, а потом их разрешение понижается с помощью функции **image.degrade.psf_camera_downsample**.

Функция **dictionary_error** получает на вход блок изображения **patch**, словарь **dictionary** и количество ненулевых коэффициентов **l0_value**, и находит ошибку представления для этого блока с помощью данного словаря (после решения задачи (1.8)).

Функция **apply_to_patches** получает на вход изображение **image**, размер блока **patch_shape** и функцию **function**, и разбивает данное изображение на блоки, а потом применяет к ним данную функцию. Используется вместе с функцией **dictionary_error** для нахождения карты ошибок представления для изображения.

Функция **split_to_edge_and_ringing** получает на вход изображение **image**, словарь **dictionary**, а также набор параметров алгоритма ОМР **omp_parameters**, и строит разложение данного блока на составляющую контура, эффекта ложного оконтуривания, и ошибку в соответствии с формулой (1.11).

Функция **build_image_decompositions** получает на вход изображение **image**, карту подходящих для использования контуров **edges**, словарь **dictionary**, набор параметров алгоритма ОМР **omp_parameters**, а также размер блока **patch_shape**, и получает список разложений всех подходящих для использования контуров на составляющие контуров, эффекта ложного оконтуривания, и ошибок.

Функция `aggregate_estimations` получает на вход список разложений блоков изображения `decomposition_data` и строит для каждого блока оценку уровня эффекта ложного оконтуривания по формуле (1.12), а потом агрегирует результаты.

Класс `RinginDetector` выполняет весь алгоритм обнаружения эффекта ложного оконтуривания. Он имеет следующие методы:

- Конструктор `__init__` получает на вход размер блока `patch_shape` и словарь `dictionary`;
- Метод `ringing_level` получает на вход изображение `image` и вычисляет уровень эффекта ложного оконтуривания для него.

3.2.2 Модуль `ringing_suppression`

Модуль `ringing_suppression` имеет следующую структуру:

```
ringing_suppression
    deringing_tv(image, strength)
    learn_deringing_joint(normal_images, ringing_images, patch_shape,
                          dict_learning_method, dict_learning_parameters)
    class DeringingJoint
        __init__(normal_dictionary, ringing_dictionary,
                 patch_shape, patch_step,
                 sparse_coding_method, sparse_coding_parameters)
        process(image)
    merge_by_edges(image_first, image_second, edge_distance)
    learn_pure_ringing(images, patch_shape, sharp_strength,
                      dict_learning_method, dict_learning_parameters)
    class DeringingSeparate
        __init__(normal_dictionary, ringing_dictionary,
                 patch_shape, patch_step,
                 sparse_coding_method, sparse_coding_parameters)
        process(image)
    merge_by_basic_edges(image_first, image_second)
```

Все алгоритмы, используемые в данном модуле, описаны в разделе 2.3. Опишем интерфейс функций и классов.

Функция `deringing_tv` получает на вход изображение для обработки `image` и силу подавления эффекта ложного оконтуривания `strength`, и выполняет подавление эффекта ложного оконтуривания с помощью метода основанного на полной вариации. Для этого решается задача минимизации по формуле (2.3), где параметр регуляризации γ равен силе подавления эффекта ложного оконтуривания `strength`. Минимизация выполняется с помощью субградиентного метода [80]. Опишем этот метод.

Обозначим за z переменную минимизации и за $F(z)$ — минимизируемую функцию. Метод итеративно обновляет текущее решение $z^{(k)}$ по формуле

$$z^{(k+1)} = z^{(k)} - \alpha_k g^{(k)},$$

где $g^{(k)} \in \partial F(z^{(k)})$ — любой субградиент функции $F(z)$ в точке $z^{(k)}$, и α_k — величина шага. Вектор g называется субградиентом выпуклой функции $F(z)$ в точке z' , если выполняется следующее условие: $F(z) \geq F(z') + \langle g, z - z' \rangle$ для всех z . Для изменения величины шага используется следующая формула: $\alpha_k = \alpha_0 \gamma^k$.

Функция `learn_deringing_joint` получает на вход набор обычных изображений `normal_images` и соответствующих изображений с эффектом ложного оконтуривания `ringing_images`, размер блока `patch_shape`, а также метод для обучения словарей `dict_learning_method` (в виде функции) и набор параметров для обучения `dict_learning_parameters`, и строит объединенные словари для подавления эффекта ложного оконтуривания (с учетом их расширения поворотами).

Класс `DeringingJoint` выполняет подавление эффекта ложного оконтуривания с помощью объединенных словарей (2.3.1). Он имеет следующие методы:

- Конструктор `__init__` получает пару объединенных словарей `normal_dictionary` и `ringing_dictionary`, размер блока `patch_shape` и шаг для выбора блока `patch_step`, а также метод для построения разреженных представлений `sparse_coding_method` (в виде функции) и набор параметров для этого метода.
- Метод `process` получает на вход изображение `image` и выполняет подавление эффекта ложного оконтуривания для него.

Функция `merge_by_edges` получает на вход два изображения, полученные разными методами обработки `image_first` и `image_second`, а также порог расстояния `edge_distance`, и составляет комбинированное изображение: для точек на расстоянии от контуров не больше

`edge_distance` используются значения изображения `image_first`, а для всех остальных — значения изображения `image_second`.

Функция `learn_pure_ringing` получает на вход набор обычных изображений `images`, размер блока `patch_shape`, уровень повышения резкости `sharp_strength`, а также метод для обучения словарей `dict_learning_method` (в виде функции) и набор параметров для обучения `dict_learning_parameters`, и строит словарь для чистого эффекта ложного оконтуривания (с учетом расширения поворотами и повышения резкости).

Класс `DeringingSeparate` выполняет подавление эффекта ложного оконтуривания с помощью независимых словарей (2.3.3). Он имеет следующие методы:

- Конструктор `__init__` получает пару независимых словарей `normal_dictionary` и `ringing_dictionary`, размер блока `patch_shape` и шаг для выбора блока `patch_step`, а также метод для построения разреженных представлений `sparse_coding_method` (в виде функции) и набор параметров для этого метода.
- Метод `process` получает на вход изображение `image` и выполняет подавление эффекта ложного оконтуривания для него.

Функция `merge_by_basic_edges` получает на вход два изображения, полученные разными методами обработки `image_first` и `image_second` и составляет комбинированное изображение: для точек из объединения областей BEN и BEP используются значения изображения `image_first`, а для всех остальных — значения изображения `image_second`.

3.3 Графический интерфейс

Графический интерфейс состоит из трех вкладок: для обнаружения эффекта ложного оконтуривания, для подавления эффекта ложного оконтуривания с помощью объединенных словарей и для подавления эффекта ложного оконтуривания с помощью независимых словарей.

Вкладка для обнаружения эффекта ложного оконтуривания называется **Ring detection**, ее вид изображен на рис. 3.2. Вкладка состоит из следующих элементов:

- **Path to dictionary** — путь к предварительно сконструированному словарю. Изменить путь можно с помощью кнопки **Load**.
- **Patch size** — поле для изменения используемого в алгоритме размера блока. Выбранный словарь должен соответствовать этому размеру.

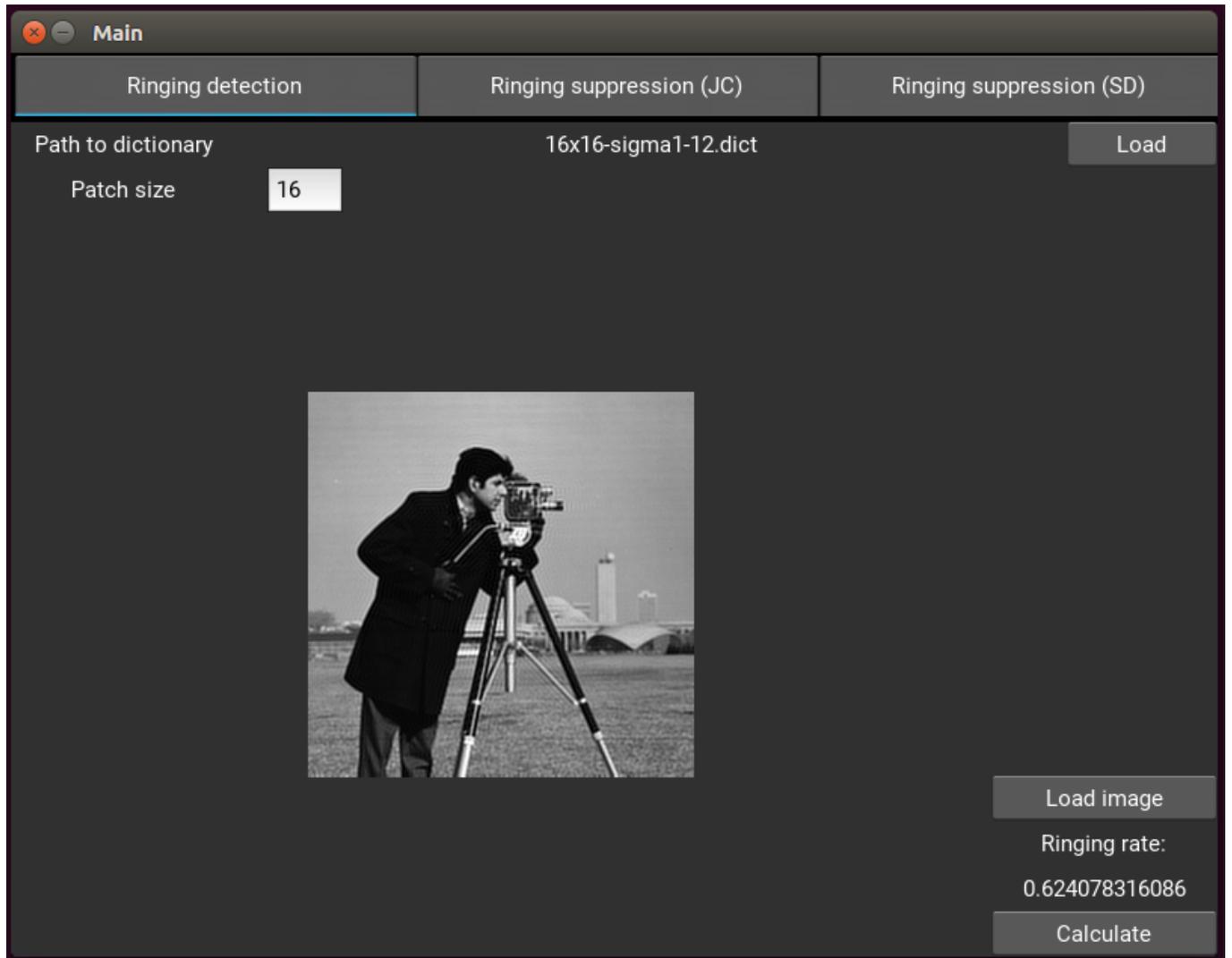


Рис. 3.2: Графический интерфейс, обнаружение эффекта ложного оконтуривания.

- Поле, в котором отображается входное изображение. Загрузить изображение можно с помощью кнопки **Load image**.
- **Ringing rate** — в этом поле отображается оцененный уровень эффекта ложного оконтуривания для входного изображения. Вычисление этого уровня производится с помощью кнопки **Calculate**.

Вкладка для подавления эффекта ложного оконтуривания с помощью объединенных словарей называется **Ringing suppression (JC)**, ее вид изображен на рис. 3.3. Эта вкладка состоит из следующих элементов:



Рис. 3.3: Графический интерфейс, подавление эффекта ложного оконтуривания с помощью объединенных словарей.

- **Path to dictionary** — путь к паре предварительно обученных объединенных словарей. Изменить путь можно с помощью кнопки **Load**.
- **Patch size** — поле для изменения используемого в алгоритме размера блока. Выбранный словарь должен соответствовать этому размеру.
- **Patch step** — поле для изменения используемого в алгоритме шага выбора блока.
- **TV deringing strength** — поле для изменения используемой в алгоритме силы подавление эффекта ложного оконтуривания с помощью метода основанного на полной вариации (используется при объединении результатов).

- Поле, в котором отображается входное изображение. Загрузить изображение можно с помощью кнопки **Load image**.
- Поле, в котором отображается обработанное изображение. Получить это изображение можно с помощью кнопки **Suppress ringing**, сохранить его можно с помощью кнопки **Save result**.

Вкладка для подавления эффекта ложного оконтуривания с помощью объединенных словарей называется **Ringing suppression (SD)**, ее вид изображен на рис. 3.4. Эта вкладка состоит из таких же элементов, как и вкладка **Ringing suppression (JC)**.



Рис. 3.4: Графический интерфейс, подавление эффекта ложного оконтуривания с помощью независимых словарей.

Заключение

Основные результаты:

1. Предложен и исследован метод математического моделирования эффекта ложного оконтуривания на изображениях с использованием метода разреженных представлений.
2. Разработаны и численно реализованы алгоритмы обнаружения и подавления эффекта ложного оконтуривания на изображениях.
3. Создан программный комплекс для обнаружения и подавления эффекта ложного оконтуривания на изображениях.

Основные результаты докладывались и обсуждались на:

- 11-й международной конференции по анализу и распознаванию изображений («International Conference on Image Analysis and Recognition», ICIAR), Виламюра, Португалия, 2014 [19];
- 12-й международной конференции по обработке сигналов («International Conference on Signal Processing», ICSP), Ханчжоу, Китай, 2014 [13];
- Международной конференции по продвинутым концепциям интеллектуальных систем технического зрения («Advanced Concepts of Intelligent Visual Systems», ACIVS), Катания, Италия, 2015 [16];
- Конференции «Тихоновские чтения 2015», ВМК МГУ имени М. В. Ломоносова (Москва, 2015);
- Заседании кафедры математической физики факультета ВМК МГУ имени М. В. Ломоносова, г. Москва, 16 сентября 2015 г.
- Международном научно-исследовательском семинаре «Анализ и понимание изображений (Математические, когнитивные и прикладные проблемы анализа изображений и сигналов)» ФИЦ ИУ РАН, Москва, 2017.

- Международной конференции по применению разреженных представлений для приложений в мультимедиа, биомедицине и коммуникациях («Compressive sensing based trends applied to: multimedia, biomedicine and communications»), Будва, Черногория, 2017.

Приведенные в диссертационной работе исследования показали теоретическую и практическую перспективность исследования и разработки методов обнаружения и подавления эффекта ложного оконтуривания на изображениях с помощью разреженных представлений.

Результаты диссертации опубликованы в 6 научных работах, из них:

- Работы [15–18] опубликованы в журналах Scopus, WoS, RSCI.
- Работы [13, 19] опубликованы в трудах конференций и индексируются в системе Scopus.

Список литературы

1. *Гонсалес Р., Вудс Р.* Цифровая обработка изображений. — М.: Техносфера, 2005. — 1104 с.
2. *Малла С.* Вэйвлеты в обработке сигналов. — М.: Мир, 2005. — 671 с.
3. *Park S. C., Park M. K., Kang M. G.* Super-resolution image reconstruction: a technical overview // IEEE signal processing magazine. — 2003. — Vol. 20, no. 3. — Pp. 21–36.
4. Survey of image denoising techniques / M. C. Motwani, M. C. Gadiya, R. C. Motwani, F. C. Harris // Proceedings of GSPX. — 2004. — Pp. 27–30.
5. *Rajni R., Anutam A.* Image denoising techniques - An overview // International Journal of Computer Applications. — 2014. — Vol. 86, no. 16. — Pp. 13–17.
6. *Guillemot C., Le Meur O.* Image inpainting: Overview and recent advances // IEEE signal processing magazine. — 2014. — Vol. 31, no. 1. — Pp. 127–144.
7. *Jerri A. J.* Mathematics and Its Applications. Vol. 446. The Gibbs phenomenon in Fourier analysis, splines and wavelet approximations. — Springer Science & Business Media, 2013. — 340 pp.
8. *Liu H., Klomp N., Heynderickx I.* A no-reference metric for perceived ringing // Proceedings of International Workshop on Video Processing and Quality Metrics for Consumer Electronics. — 2009.
9. *Punchihewa A., Bailey D. G.* Artefacts in image and video systems; classification and mitigation // Proceedings of image and vision computing New Zealand. — 2002. — Pp. 197–202.
10. A short overview of MRI artefacts / L. Erasmus, D. Hurter, M. Naudé, H. Kritzinger, S. Acho // SA Journal of Radiology. — 2004. — Vol. 8, no. 2. — Pp. 13–17.
11. *Elad M.* Sparse and Redundant Representations. — Springer, 2010. — 397 pp.
12. *Nasonov A. V., Krylov A. S.* Scale-space method of image ringing estimation // Proceedings of International Conference on Image Processing (ICIP'09). — 2009. — Pp. 2793–2796.

13. Sparse method for ringing artifact detection / A. Umnov, A. Nasonov, A. Krylov, D. Yong // Proceedings of International Conference on Signal Processing (ICSP'2014). — 2014. — Pp. 662–667.
14. *Nasonov A. V., Krylov A. S.* Adaptive image deringing // Proceedings of GraphiCon2009. — 2009. — Pp. 151–154.
15. *Крылов А., Умнов А.* Влияние эффекта Гиббса на взаимную согласованность в методе разреженных представлений для изображений // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. — 2016. — № 4. — с. 10–15.
16. *Umnov A., Krylov A., Nasonov A.* Ringing artifact suppression using sparse representation // Lecture Notes in Computer Science. — 2015. — Vol. 9386. — Pp. 35–45.
17. *Умнов А., Крылов А.* Исследование метода разреженных представлений для подавления эффекта ложного оконтуривания // Компьютерная оптика. — 2016. — т. 40, № 6. — с. 895–903.
18. *Umnov A., Krylov A.* Sparse Approach to Image Ringing Detection and Suppression // Pattern Recognition and Image Analysis. — 2017. — Vol. 27, no. 4. — Pp. 754–762.
19. Image warping in dermatological image hair removal / A. Nasonova, A. Nasonov, A. Krylov, I. Pechenko, A. Umnov, N. Makhneva // Proceedings of International Conference on Image Analysis and Recognition (ICIAR'2014). — 2014. — Pp. 159–166.
20. *Ahmed F., Mahmud M. S.* Filtering Effect in Holographic Image Processing // International Journal of Scientific and Research Publications. — 2015. — P. 432.
21. Perceptual blur and ringing metrics: application to JPEG2000 / P. Marziliano, F. Dufaux, S. Winkler, T. Ebrahimi // Signal Processing: Image Communication. — 2004. — Vol. 19, no. 2. — Pp. 163–172.
22. Annoyance of individual artifacts in MPEG-2 compressed video and their relation to overall annoyance / C. C. Koh, S. K. Mitra, J. M. Foley, I. E. Heynderickx // Proceedings of International Society for Optics and Photonics, Electronic Imaging. Vol. 5666. — 2005. — Pp. 595–606.
23. *Yuen M., Wu H.* A survey of hybrid MC/DPCM/DCT video coding distortions // Signal processing. — 1998. — Vol. 70, no. 3. — Pp. 247–278.
24. База данных изображений с эффектом Гиббса и ложным оконтуриванием [Электронный ресурс]. — Режим доступа: <http://imaging.cs.msu.ru/ru/research/ringing/database>.

25. *Фихтенгольц Г. М.* Курс дифференциального и интегрального исчисления. в 3 т. Т. 3. — М.: Физматлит, 2002. — 728 с.
26. *Canny J.* A Computational Approach To Edge Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1986. — Vol. 8, no. 6. — Pp. 679–714.
27. *Feng X., Allebach J. P.* Measurement of ringing artifacts in JPEG images // Proceedings of International Society for Optics and Photonics, Electronic Imaging. Vol. 6076. — 2006. — Pp. 74–83.
28. *Hu S., Pizlo Z., Allebach J. P.* JPEG ringing artifact visibility evaluation // Proceedings of International Society for Optics and Photonics, Electronic Imaging. Vol. 9016. — 2014. — Pp. 3–5.
29. No-reference perceptual image quality metric using gradient profiles for JPEG2000 / L. Liang, S. Wang, J. Chen, S. Ma, D. Zhao, W. Gao // Signal Processing: Image Communication. — 2010. — Vol. 25, no. 7. — Pp. 502–516.
30. *Zuo B.-X., Ming D.-L., Tian J.-W.* Perceptual ringing metric to evaluate the quality of images restored using blind deconvolution algorithms // Optical Engineering. — 2009. — Vol. 48, no. 3.
31. *Mosleh A., Langlois J. P., Green P.* Image deconvolution ringing artifact detection and removal via psf frequency analysis // Proceedings of European Conference on Computer Vision (ECCV'2014). — 2014. — Pp. 247–262.
32. Adaptive directional sharpening with overshoot control / A. Bruna, A. Buemi, M. Guarnera, G. Santoro // Proceedings of International Society for Optics and Photonics, Electronic Imaging. Vol. 6. — 2008.
33. *Nasonov A., Krylov A.* Edge quality metrics for image enhancement // Pattern Recognition and Image Analysis. — 2012. — Vol. 22, no. 2. — Pp. 346–353.
34. Just noticeable distortion model and its applications in video coding / X. Yang, W. Ling, Z. Lu, E. P. Ong, S. Yao // Signal Processing: Image Communication. — 2005. — Vol. 20, no. 7. — Pp. 662–680.
35. *Mallat S.* A wavelet tour of signal processing: the sparse way. — Academic press, 2008. — 832 pp.
36. *Elad M., Figueiredo M. A., Ma Y.* On the role of sparse and redundant representations in image processing // Proceedings of the IEEE. — 2010. — Vol. 98, no. 6. — Pp. 972–982.

37. *Donoho D. L.* Nonlinear solution of linear inverse problems by wavelet–vaguelette decomposition // Applied and computational harmonic analysis. — 1995. — Vol. 2, no. 2. — Pp. 101–126.
38. *Mallat S. G., Zhang Z.* Matching pursuits with time-frequency dictionaries // IEEE Transactions on signal processing. — 1993. — Vol. 41, no. 12. — Pp. 3397–3415.
39. *Gorodnitsky I. F., Rao B. D.* Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm // IEEE Transactions on signal processing. — 1997. — Vol. 45, no. 3. — Pp. 600–616.
40. *Boyd S., Vandenberghe L.* Convex Optimization. — Cambridge university press, 2004. — 716 pp.
41. Least angle regression / B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, [et al.] // The Annals of statistics. — 2004. — Vol. 32, no. 2. — Pp. 407–499.
42. *Tibshirani R.* Regression shrinkage and selection via the lasso // Journal of the Royal Statistical Society. Series B. — 1996. — Pp. 267–288.
43. *Daubechies I., Defrise M., De Mol C.* An iterative thresholding algorithm for linear inverse problems with a sparsity constraint // Communications on pure and applied mathematics. — 2004. — Vol. 57, no. 11. — Pp. 1413–1457.
44. *Beck A., Teboulle M.* A fast iterative shrinkage-thresholding algorithm for linear inverse problems // SIAM journal on imaging sciences. — 2009. — Vol. 2, no. 1. — Pp. 183–202.
45. *Donoho D. L., Elad M.* On the stability of the basis pursuit in the presence of noise // Signal Processing. — 2006. — Vol. 86, no. 3. — Pp. 511–532.
46. Nonlinear unsharp masking methods for image contrast enhancement / G. Ramponi, N. Strobel, S. K. Mitra, T.-H. Yu // Journal of Electronic Imaging. — 1996. — Vol. 5, no. 3. — Pp. 353–366.
47. *Nosratinia A.* Postprocessing of JPEG-2000 images to remove compression artifacts // IEEE Signal Processing Letters. — 2003. — Vol. 10, no. 10. — Pp. 296–299.
48. Efficient quadtree based block-shift filtering for deblocking and deringing / G. Zhai, W. Lin, J. Cai, X. Yang, W. Zhang // Journal of Visual Communication and Image Representation. — 2009. — Vol. 20, no. 8. — Pp. 595–607.
49. Maximum-likelihood parameter estimation for image ringing-artifact removal / S. Yang, Y.-H. Hu, T. Q. Nguyen, D. L. Tull // IEEE transactions on circuits and systems for video technology. — 2001. — Vol. 11, no. 8. — Pp. 963–973.

50. Dual range deringing for non-blind image deconvolution / L. Zou, H. Zhou, S. Cheng, C. He // Proceedings of International Conference on Image Processing (ICIP'2010). — 2010. — Pp. 1701–1704.
51. *Cao Y., Eggermont P. P., Terebey S.* Cross-entropy maximization and its application to ringing suppression in image reconstruction // IEEE Transactions on Image Processing. — 1999. — Vol. 8, no. 2. — Pp. 286–292.
52. *Sitdikov I. T., Krylov A. S.* Variational image deringing using varying regularization parameter // Pattern Recognition and Image Analysis. — 2015. — Vol. 25, no. 1. — Pp. 96–100.
53. *Rudin L. I., Osher S., Fatemi E.* Nonlinear total variation based noise removal algorithms // Physica D: Nonlinear Phenomena. — 1992. — Vol. 60, no. 1–4. — Pp. 259–268.
54. *Rudin L. I., Osher S.* Total variation based image restoration with free local constraints // Proceedings of International Conference on Image Processing (ICIP'94). Vol. 1. — IEEE. 1994. — Pp. 31–35.
55. A new alternating minimization algorithm for total variation image reconstruction / Y. Wang, J. Yang, W. Yin, Y. Zhang // SIAM Journal on Imaging Sciences. — 2008. — Vol. 1, no. 3. — Pp. 248–272.
56. *Krylov A., Nasonov A.* Adaptive total variation deringing method for image interpolation // Proceedings of International Conference on Image Processing (ICIP'08). — 2008. — Pp. 2608–2611.
57. *Peyré G.* Manifold models for signals and images // Computer Vision and Image Understanding. — 2009. — Vol. 113, no. 2. — Pp. 249–260.
58. *Aharon M., Elad M., Bruckstein A.* K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation // IEEE Transactions on signal processing. — 2006. — Vol. 54, no. 11. — Pp. 4311–4322.
59. *Engan K., Aase S. O., Husoy J. H.* Method of optimal directions for frame design // Proceedings of International Conference on Acoustics, Speech, and Signal Processing. Vol. 5. — 1999. — Pp. 2443–2446.
60. *Rubinstein R., Zibulevsky M., Elad M.* Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit // Cs Technion. — 2008. — Vol. 40, no. 8. — Pp. 1–15.
61. Coupled dictionary training for image super-resolution / J. Yang, Z. Wang, Z. Lin, S. Cohen, T. Huang // IEEE transactions on image processing. — 2012. — Vol. 21, no. 8. — Pp. 3467–3478.

62. *Donoho D. L., Elad M., Temlyakov V. N.* Stable recovery of sparse overcomplete representations in the presence of noise // *IEEE Transactions on information theory.* — 2006. — Vol. 52, no. 1. — Pp. 6–18.
63. *Muller M. E.* A note on a method for generating points uniformly on n-dimensional spheres // *Communications of the ACM.* — 1959. — Vol. 2, no. 4. — Pp. 19–20.
64. *Абрамовиц М., Стиган И.* Справочник по специальным функциям. — М.: Наука, 1979. — 832 с.
65. *Kershaw D.* Some extensions of W. Gautschi's inequalities for the gamma function // *Mathematics of Computation.* — 1983. — Pp. 607–611.
66. *Nasonov A. V., Krylov A. S.* Finding Areas of Typical Artifacts of Image Enhancement Methods // *Pattern Recognition and Image Analysis.* — 2011. — Vol. 21, no. 2. — Pp. 316–318.
67. *Nasonov A. V., Krylov A. S.* Image enhancement quality metrics // *Proceedings of GraphiCon'2011.* — 2011. — Pp. 128–131.
68. 2D Euclidean distance transforms: A comparative survey / R. Fabbri, L. da F. Costa, J. C. Torelli, O. M. Bruno // *ACM Computing Surveys.* — 2008. — Vol. 40, no. 1. — 2:1–2:44.
69. *Orieux F., Giovannelli J.-F., Rodet T.* Bayesian estimation of regularization and point spread function parameters for Wiener–Hunt deconvolution // *Journal of the Optical Society of America A.* — 2010. — Vol. 27, no. 7. — Pp. 1593–1607.
70. *Krylov A. S., Nasonova A. A., Nasonov A. V.* Image enhancement by non-iterative grid warping // *Pattern Recognition and Image Analysis.* — 2016. — Vol. 26, no. 1. — Pp. 161–164.
71. Image quality assessment: from error visibility to structural similarity / Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli // *IEEE transactions on image processing.* — 2004. — Vol. 13, no. 4. — Pp. 600–612.
72. SciPy: Open source scientific tools for Python [Electronic resource] / E. Jones, T. Oliphant, P. Peterson, [et al.]. — Access mode: <http://www.scipy.org/>.
73. *Hunter J. D.* Matplotlib: A 2D graphics environment // *Computing in Science & Engineering.* — 2007. — Vol. 9, no. 3. — Pp. 90–95.

74. Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay // Journal of Machine Learning Research. — 2011. — Vol. 12. — Pp. 2825–2830.
75. scikit-image: image processing in Python / S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, the scikit-image contributors // PeerJ. — 2014. — Vol. 2. — e453.
76. *Pérez F., Granger B. E.* IPython: a System for Interactive Scientific Computing // Computing in Science and Engineering. — 2007. — Vol. 9, no. 3. — Pp. 21–29.
77. Kivy: cross-platform Python framework for NUI development [Electronic resource]. — Access mode: <http://kivy.org>.
78. Poynton's Color FAQ [Electronic resource]. — Access mode: <http://www.poynton.com/PDFs/ColorFAQ.pdf>.
79. *Krylov A. S., Lukin A. S., Nasonov A. V.* Edge-preserving nonlinear iterative image resampling method // Proceedings of International Conference on Image Processing (ICIP'09). — 2009. — Pp. 385–388.
80. *Boyd S., Xiao L., Mutapcic A.* Subgradient methods // Lecture notes of EE392o, Stanford University, Autumn Quarter. — 2003. — Vol. 2004.

Список иллюстраций

- | | | |
|-----|--|----|
| 1 | Пример отделения чистого эффекта ложного оконтуривания от размытого контура: исходный контур с эффектом ложного оконтуривания (слева), ближайший размытый контур (в центре), чистый эффект ложного оконтуривания (справа). | 7 |
| 2 | Изменение разреженности при добавлении эффекта ложного оконтуривания; сплошная линия соответствует изображениям без искажения, штриховая, штрихпунктирная и точечная соответствуют изображениям с эффектом ложного оконтуривания с силой искажения $d = 1.5, 2, 3$ соответственно. | 10 |
| 3 | Примеры обработки изображения с сильным эффектом ложного оконтуривания различными методами. В порядке слева-направо и сверху-вниз приведены: изображение с эффектом ложного оконтуривания ($d = 2.5$), результаты обработки методом на основе TV, результаты обработки методом на основе объединенных словарей, результаты обработки методом на основе независимых словарей, исходное изображение. | 12 |
| 1.1 | Примеры эффекта ложного оконтуривания, возникающего а) в результате сжатия изображения, б) в результате повышения резкости изображения, в) в результате повышения разрешения изображения, г) на изображении магнитно-резонансной томографии. | 16 |
| 1.2 | Пример эффекта Гиббса для разрывной функции. Слева — исходная функция, справа — функция после удаления части высокочастотной информации. | 18 |
| 1.3 | Пример эффекта ложного оконтуривания для одномерного размытого контура. Слева — исходный контур, справа — контур после удаления части высокочастотной информации. | 19 |
| 1.4 | Пример эффекта ложного оконтуривания для изображения. Слева — исходное изображение, справа — изображение после удаления части высокочастотной информации. | 20 |

1.5	График функции $\text{sinc}(x)$ в одномерном случае и изображение аналогичной функции в двухмерном случае.	21
1.6	Схематическое изображение областей VEN и ВЕР для базовых контуров.	25
1.7	Пример одномерного профиля (слева), двумерного профиля, полученного из него дублированием значений вдоль горизонтальной оси (в центре) и поворота этого двумерного контура (справа).	38
1.8	Пример отделения чистого эффекта ложного оконтуривания от размытого контура: исходный контур с эффектом ложного оконтуривания (слева), ближайший размытый контур (в центре), чистый эффект ложного оконтуривания (справа).	39
1.9	Выборка случайных элементов из словаря для обнаружения эффекта ложного оконтуривания.	40
1.10	Зависимость размытия σ оптимального приближения размытым контуром от силы эффекта ложного оконтуривания d	41
1.11	Примеры разложения блоков изображений по формуле 1.11. Каждая строка соответствует одному блоку, по столбцам слева направо: исходный блок (y), вклад контура (b), вклад эффекта ложного оконтуривания (r), ошибка разложения (e), контур без эффекта ложного оконтуривания ($y - r$).	43
1.12	Примеры изображений с смоделированным эффектом ложного оконтуривания и значений разработанной метрики для них.	49
1.13	Примеры изображений с различными искажениями (с эффектом ложного оконтуривания и без него) и значений разработанной метрики для них.	50
1.14	Примеры синтетических изображений с применением нерезкого маскирования и значений разработанной метрики для них.	51
2.1	Зависимость согласованности от размера блока для реальных (сплошная линия) и случайных (пунктирная линия) изображений.	65
2.2	Изменение взаимной согласованности при добавлении эффекта ложного оконтуривания; пунктиром обозначено значение взаимной согласованности для обычных изображений, а сплошной линией — для изображений с различной силой эффекта ложного оконтуривания d	71
2.3	Изменение взаимной согласованности при добавлении размытия; пунктиром обозначено значение взаимной согласованности для обычных изображений, а сплошной линией — для изображений с различной силой размытия.	71

- 2.4 Изменение взаимной согласованности при добавлении шума; пунктиром обозначено значение взаимной согласованности для обычных изображений, а сплошной линией — для изображений с различной дисперсией шума (считая, что яркость принимает значения от 0 до 1). 72
- 2.5 Изменение разреженности при добавлении эффекта ложного оконтуривания; сплошная линия соответствует изображениям без искажения, штриховая, штрихпунктирная и точечная соответствуют изображениям с эффектом ложного оконтуривания с силой искажения $d = 1.5, 2, 3$ соответственно. 73
- 2.6 Изменение разреженности при добавлении размытия; сплошная линия соответствует изображениям без искажения, штриховая, штрихпунктирная и точечная соответствуют изображениям с размытием с ядром с $\sigma = 0.5, 1, 2$ соответственно. 74
- 2.7 Изменение разреженности при добавлении шума; сплошная линия соответствует изображениям без искажения, штриховая, штрихпунктирная и точечная соответствуют изображениям с шумом дисперсией 0.02, 0.05, 0.1 соответственно. 74
- 2.8 Изменение разреженности при повышении резкости; сплошная линия соответствует изображениям без искажения, штриховая соответствует изображениям с эффектом ложного оконтуривания с силой искажения равной 2, штрихпунктирная и точечная соответствуют предыдущим изображениям с дополнительным повышением резкости с гауссовским ядром с σ , равным 1 и 1.5 соответственно. 75
- 2.9 Примеры обработки изображения различными методами. В порядке слева-направо и сверху-вниз приведены: исходное изображение, изображение с эффектом ложного оконтуривания ($d = 2$), результаты обработки методом TV, результаты обработки методом JC, результаты обработки методом SD. 81
- 2.10 Примеры обработки изображения с сильным эффектом ложного оконтуривания ($d = 2.5$). В порядке слева-направо и сверху-вниз приведены: исходное изображение, изображение с эффектом ложного оконтуривания, результаты обработки методом JC, результаты обработки методом SD, увеличенный фрагмент для метода JC, увеличенный фрагмент для метода SD. 82
- 2.11 Примеры обработки изображения со слабым эффектом ложного оконтуривания ($d = 2$). В порядке слева-направо и сверху-вниз приведены: исходное изображение, изображение с эффектом ложного оконтуривания, результаты обработки методом JC, результаты обработки методом SD, увеличенный фрагмент для метода JC, увеличенный фрагмент для метода SD. 83

2.12	Примеры результатов обработки с использованием алгоритмов для l_0 -нормы и для l_1 -нормы. Используется метод JS без постобработки.	84
3.1	Использование отражений для моделирования эффекта ложного оконтуривания. (а) Моделирование без использования отражений, красным выделена область с осцилляциями от границ изображения; (b) отражение изображения вдоль границ, красным выделены исходные границы изображения; (с) Моделирование с использованием отражений, красным выделена область аналогичная (а).	88
3.2	Графический интерфейс, обнаружение эффекта ложного оконтуривания.	97
3.3	Графический интерфейс, подавление эффекта ложного оконтуривания с помощью объединенных словарей.	98
3.4	Графический интерфейс, подавление эффекта ложного оконтуривания с помощью независимых словарей.	99

Список таблиц

1.1	Разбиение искажений для тестирования методов обнаружения эффекта ложного оконтуривания.	45
1.2	Наборы изображения для проверки качества метода для произвольного эффекта ложного оконтуривания.	46
1.3	Результаты сравнения для смоделированного эффекта ложного оконтуривания, и эффекта ложного возникающего в результате различных алгоритмов обработки.	47
1.4	Наборы изображений для проверки качества метода для эффекта ложного оконтуривания, полученного в результате сжатия изображения алгоритмом JPEG2000.	47
1.5	Результаты сравнения для эффекта ложного оконтуривания, полученного в результате сжатия изображения алгоритмом JPEG2000.	48
2.1	Соответствие между силой эффекта ложного оконтуривания d и значением параметра силы варпинга A	77
2.2	Эффективность подавления эффекта ложного оконтуривания для различных методов, указано среднее значение SSIM для тестового набора. Жирным шрифтом выделен наилучший результат в строке.	79
2.3	Эффективность подавления эффекта ложного оконтуривания для различных методов, указано среднее значение PSNR для тестового набора. Жирным шрифтом выделен наилучший результат в строке.	80