

Пространственная фильтрация

Содержание

1	Базовые методы обработки изображений	2
2	Пространственная фильтрация	3
2.1	Базовые преобразования уровней серого	3
2.2	Гистограмма	4
2.3	Пороговая обработка	8
2.4	Корреляция и свертка	10
2.5	Фильтры и сглаживание	12

1 Базовые методы обработки изображений

В этой лекции мы рассмотрим базовые методы обработки изображений. Для начала давайте вспомним, что такое обработка изображений. Это методы и задачи, где входными и выходными данными являются изображения. Зачем нужна обработка изображений? Для того, чтобы изображение стало лучше для восприятия человеком или для восприятия компьютером – с эстетической точки зрения или для дальнейшего анализа, визуального человеком или автоматизированного компьютером.

Приведём несколько примеров таких методов. К простейшим методам обработки изображений можно отнести методы шумоподавления, изменения яркости, повышения контрастности, повышения резкости, цветовой коррекции. На этом слайде приведено несколько примеров изображения Лены, которые можно "исправить" с эстетической точки зрения при помощи простых методов обработки изображений. Кстати, эта фотография не случайно попала на этот слайд. Это фрагмент снимка из Playboy, выбранный одним из специалистов по обработке изображений для иллюстрации нового метода в своей статье в 70-е годы. После публикации статьи, эта картинка стала де-факто стандартным тестовым изображением для всевозможных методов коррекции и обработки изображений, получившая своё имя по имени модели, изображённой на снимке – Лена.

Ещё один пример обработки изображений с эстетическими целями является построение HDR-фотографий (High Dynamic Range – высокий динамический диапазон). Такие картинки обычно получают из нескольких кадров, снятых с разной выдержкой с одной и той же точки в одно и то же время.

Также к методам обработки изображений с целью улучшения с эстетической точки зрения можно отнести применение различных художественных спецэффектов.

Ещё несколько примеров методов обработки изображений, широко используемых для упрощения дальнейшего анализа изображений: Бинаризация (отделение фона от объектов), выделение компонент связности (сегментация), выделение контуров или краёв на изображении – типичные примеры предварительной обработки изображений, облегчающие задачу последующего семантического анализа.

2 Пространственная фильтрация

2.1 Базовые преобразования уровней серого

Перед тем как углубляться в методы обработки изображений, давайте вспомним, как обычно представляется изображение. Изображение – это по сути двумерная функция, которая пространственным координатам (x, y) сопоставляет вектор, представляющий цвет точки в данной позиции. В случае полутонового (черно-белого) изображения, цвет одной точки изображения можно выразить одномерной величиной, являющейся значением интенсивности (или яркости) заданной точки.

В случае цветного изображения, как мы обсуждали ранее, цвет чаще всего задаётся трёхмерным вектором.

Дискретное изображение представляется в виде дискретной функции. Пространственная плоскость изображения представляется в виде координатной сетки. Одна ячейка сетки соответствует одному пикселю.

Множество методов обработки изображений распадается на две большие категории: обработка в пространственной области (пространственные методы) и обработка в частотной области (частотные методы). Термин пространственная область относится к плоскости изображения как таковой, и пространственные методы обработки изображений основаны на прямом манипулировании пикселями. Методы обработки в частотной области основываются на модификации сигнала, получаемого путём применения к исходному изображению преобразования Фурье. Мы начнём с пространственных методов, а потом разберёмся с частотным представлением изображений, преобразованием Фурье и частотными методами.

Многие методы обработки изображений оперируют с окрестностями пикселей. На экране приведён пример окрестности точки (x, y) размером 3×3 . Все методы пространственной обработки описываются уравнением $g(x, y) = T[f(x, y)]$, где $f(x, y)$ – исходное изображение, $g(x, y)$ – обработанное изображение, а T – оператор над f , определённый в некоторой окрестности точки (x, y) . T может также оперировать над последовательностью входных изображений, например, выполняя поэлементное сложение нескольких изображений.

Оператор T выполняется в каждой точке (x, y) , используя пиксели внутри окрестности данной точки, давая в результате выходное значение g для данной точки. Окрестности чаще всего являются квадратными или прямоугольными. В простейшем случае, окрестность может быть размером 1×1 (т.е. состоит из одного пикселя). В таком случае g зависит только от значения f в точке (x, y) . Такие преобразования называют поэлементными, или градационными. Оператор T в этом случае становится функцией отображения

$s = T(r)$, где в случае полутоновых изображений для простоты обозначения r и s – переменные, обозначающие значения яркостей изображений $f(x, y)$ и $g(x, y)$ в каждой точке (x, y) .

Например, если $T(r)$ имеет вид, показанный на левом графике, то эффект от такого преобразования выразится в получении изображения более высокого контраста по сравнению с оригиналом. Пиксели со значениями яркости меньше t станут темнее, а пиксели со значениями яркости больше t станут светлее. В предельном случае, показанном на правом графике, преобразование $T(r)$ даёт бинарное (двух-градационное) изображение – все пиксели на изображении темнее порогового значения t станут чёрными, а все пиксели светлее t станут белыми. Отображение такой формы называют пороговой функцией.

На данном графике приведены наиболее распространённые градационные преобразования, используемые для улучшения изображений. Преобразование изображения в негатив с яркостями в диапазоне $[0, L - 1]$ осуществляется при помощи функции $s = L - 1 - r$. Такой переворот уровней яркости создаёт эквивалент фотографического негатива.

Наиболее часто используемыми являются линейное растяжение, логарифмическое и степенное преобразования. Логарифмическое преобразование отображает узкий диапазон малых значений яркости на исходном изображении в более широкий диапазон выходных значений. Это позволяет сделать более различимыми детали в затемнённых областях изображения, но при этом яркие области изображения потеряют детализацию. Степенное преобразование действует очень похожим образом, но в отличие от логарифмических функций здесь возникает целое семейство кривых возможного преобразования, получаемых простым изменением параметра γ .

Ещё одним распространённым преобразованием является использование кусочно-линейных функций. Главное преимущество кусочно-линейных функций заключается в том, что их форма может быть сколь угодно сложной. Основным недостаток в том, что для описания нужно задавать большое количество параметров. Одним из простейших случаев использования кусочно-линейных функций является преобразование, усиливающее контрасты. На экране показано типичное преобразование, используемое для усиления контраста, результат его применения на малоконтрастном изображении, и результат порогового преобразования.

2.2 Гистограмма

Гистограмма цифрового изображения с количеством уровней серого $[0, L - 1]$ – это дискретная функция $h(r_k) = n_k$, где r_k – это k -тый уровень серого и n_k – это количество пикселей в изображении с количеством уровней серого равным r_k . Обычно гистограмму нормализуют путем деления каждо-

го из ее значений на общее количество пикселей в изображении, обозначенное n . Таким образом, нормализованная гистограмма задается как $p(r_k) = n_k/n$, for $k = 0, 1, \dots, L - 1$. Грубо говоря, $p(r_k)$ дает оценку вероятности появления серого уровня r_k . Обратите внимание, что сумма всех компонентов нормализованной гистограммы равна 1.

Гистограммы являются основой многих методов обработки пространственных областей. Изменение гистограмм эффективно используется для улучшения изображения, из гистограммы можно извлечь статистические данные об изображении, которые применяются для сжатия изображения, его сегментации, а также используются во многих других приложениях для обработки изображений. Гистограммы легко вычислить, что делает их популярным инструментом для обработки изображений в реальном времени.

Давайте посмотрим на несколько примеров изображений и их гистограмм. На экране мы видим два варианта одного и того же изображения: темный и светлый, а также соответствующие гистограммы. Горизонтальная ось каждой гистограммы соответствует значениям уровня серого – r_k . Вертикальная ось соответствует значениям $h(r_k) = n_k$. На гистограммах даны ненормализованные значения, показывающие график зависимости $h(r_k) = n_k$ от r_k .

Обратите внимание, что компоненты гистограммы, соответствующие темному изображению, сосредоточены на нижней (темной) части шкалы серого. Точно так же компоненты гистограммы яркого изображения смещены к верхней части шкалы серого.

На следующем слайде есть еще два примера – изображение с низкой контрастностью и изображение с высокой контрастностью и соответствующие гистограммы. Изображение с низкой контрастностью имеет узкую гистограмму с центром в середине шкалы серого. Для монохромного изображения это подразумевает тусклый размытый серый цвет. Однако, мы видим, что компоненты гистограммы в высококонтрастном изображении покрывают широкий диапазон серой шкалы. Таким образом, можно сделать интуитивно понятный вывод, что изображение, пиксели которого имеют тенденцию занимать весь диапазон возможных уровней серого и, кроме того, имеют тенденцию к равномерному распределению, будет обладать высокой контрастностью и будет демонстрировать большое разнообразие серых тонов. В результате получается изображение с большим динамическим диапазоном и большим количеством деталей на уровне серого. Далее мы покажем, что можно разработать функцию преобразования, которая может автоматически достичь этого эффекта, только на основе информации, имеющейся в гистограмме входного изображения.

Видоизменение гистограмм Видоизменение гистограмм, т.е. операции, изменяющие изображение путем преобразования его гистограммы, являются простыми и эффективными методами регулировки контрастности. Целью преобразования гистограммы является улучшение или подавление пикселей изображения в определенном диапазоне уровней серого. Преобразования гистограмм могут быть линейными и нелинейными. Распространенными примерами линейных преобразований являются линейное растяжение и сжатие гистограммы. При линейном преобразовании, например, можно «растянуть» гистограммы слишком темных и слишком светлых изображений, просто умножив уровень серого каждого пикселя в исходном изображении на некоторое значение. Это «растяжение» гистограммы дает высококонтрастное изображение.

Но линейное преобразование редко приводит к равномерному распределению и, следовательно, мы не получим в итоге действительно высококонтрастное изображение. В таком случае можно воспользоваться нелинейным преобразованием, например, эквализацией гистограммы. Это типичный пример нелинейного преобразования гистограммы.

Задача эквализации гистограммы – преобразовать гистограмму произвольной формы в гистограмму, имеющую одинаковую частоту во всем диапазоне возможных значений серого, или получить равномерное распределение. Поэтому мы хотим, чтобы полученная гистограмма была «плоской», как показано на правом графике на экране.

Это достигается путем равного деления совокупной гистограммы исходного изображения на такое количество отрезков, которое равно количеству дискретных уровней серого. Каждому отрезку будет соответствовать одно число на выровненном изображении. График слева, который вы видите на экране, показывает такую совокупную гистограмму. Давайте теперь определим это формально.

Пусть переменная r представляет уровни серого изображения, которое нужно улучшить. Предположим, что r принимает значения $[0, L - 1]$, где $r = 0$ соответствует черному цвету и $r = L - 1$ – соответствует белому цвету. Вероятность появления уровня серого r_k в изображении задается следующим выражением:

$$p_r(r_k) = \frac{n_k}{n}, \quad k = 0, 1, 2, \dots, L - 1 \quad (1)$$

где n – общее количество пикселей в изображении, n_k – количество пикселей уровня серого равного r_k , и L – общее количество возможных уровней серого в изображении. Эквализация гистограммы (или линеаризация гистограммы – это преобразование вида $s = T(r)$, в результате которого мы получаем

уровень s для каждого значения пикселя r из исходного изображения.

$$s_k = T(r_k) = \sum_{i=0}^k p_r(r_i) = \sum_{i=0}^k \frac{n_i}{n} \quad (2)$$

Итак, обработанное (выходное) изображение получается путем преобразования каждого пикселя с уровнем r_k входного изображения в соответствующий пиксель с уровнем s_k в выходном изображении через уравнение 2.

Если мы заменим дискретные гистограммы непрерывными функциями, вероятности – функциями плотности вероятности, а суммы – интегралами, можно доказать, что такое преобразование приведет к однородной функции плотности вероятности. В дискретном случае это невозможно доказать, но такое преобразование обычно расширяет гистограмму входного изображения, поэтому уровни изображения, полученного после эквализации гистограммы, будут охватывать более полный диапазон оттенков серого.

Давайте рассмотрим несколько примеров. На экране показан результат выполнения эквализации гистограммы для слишком темных и слишком светлых изображений. Можно отметить значительное улучшение контрастности этих изображений. Справа показаны гистограммы исходного изображения и итогового изображения после эквализации гистограммы. Можно заметить, что после эквализации гистограмма охватывает весь спектр серой шкалы и не имеет каких-либо значительных всплесков, таким образом, она очень близка к однородной.

На следующем слайде показан результат выполнения эквализации гистограммы на изображениях с низкой и высокой контрастностью. Обратите внимание, что эквализация гистограммы не дала значительных визуальных различий для изображения с высокой контрастностью, потому что гистограмма этого изображения уже охватывает полный спектр серой шкалы. Интересно отметить, что, хотя все эти гистограммы различны, гистограммы выровненных изображений визуально очень похожи. Это неудивительно, потому что разница между изображениями заключается только в контрасте, а не в содержании.

Эти примеры иллюстрируют возможности эквализации гистограмм как адаптивного инструмента улучшения изображений. Мы уже обсуждали ранее преимущества изображения, в котором значения оттенков серого охватывают всю шкалу серого. Помимо получения на выходе изображения с расширенной шкалой оттенков серого, еще одним преимуществом метода эквализации гистограммы является то, что он полностью «автоматический». Другими словами, для некоторого изображения процесс эквализации гистограммы состоит просто из выполнения уравнения 2, основанного на информации, извлекаемой непосредственно из исходного изображения, без необходимости уточнения дополнительных параметров.

2.3 Пороговая обработка

Одна из задач обработки изображений основана на бинаризации гистограмм с помощью пороговой обработки. Бинаризация – это процесс разделения значений пикселей на две группы: объекты и фон. Результатом бинаризации является двоичное изображение. Бинаризация с установлением пороговых значений – самый простой метод сегментации изображений.

Предположим, что гистограммы интенсивности, показанные на экране, соответствуют некоторым изображениям. Гистограмма слева соответствует изображению, составленному из светлых объектов на темном фоне, а гистограмма справа – изображению двух типов объектов (один светлее другого) на темном фоне. Один из очевидных способов извлечения объектов из фона – выбрать точку T , разделяющую режимы на гистограмме (или пару пороговых значений T_1 и T_2 для второго изображения, чтобы разделить все режимы).

Порог называется глобальным, если значение порога T зависит только от значений уровней серого на изображении (т.е. он является одинаковым для всех позиций пикселей). Если значение T зависит как от значений уровней серого, так и от координат данного пикселя, то порог называется локальным. Если же T дополнительно зависит и от некоторого локального свойства точки (например, среднего уровня серого в области с центром в данной точке), то порог называется адаптивным.

Один из самых простых способов выбрать порог – вручную, визуально осмотрев гистограмму изображения. Есть также много методов, которые могут выбирать порог автоматически. Ниже пример одного из самых простых:

1. Выберите случайным образом начальной значение порога T .
2. Сегментируйте изображение по порогу T . Это даст два региона пикселей: G_1 , состоящий из всех пикселей со значениями уровней серого $< T$ и G_2 , состоящий из пикселей со значениями $\geq T$.
3. Вычислите среднюю интенсивность значений μ_1 и μ_2 для пикселей для регионов G_1 и G_2 .
4. Вычислите пороговое значение:

$$T = \frac{1}{2}(\mu_1 + \mu_2) \quad (3)$$

5. Повторите шаги 2-4 до тех пор, пока разница T между двумя последовательными операциями будет меньше, чем заранее определенный порог.

На экране представлены примеры пороговой обработки в применении к изображению в оттенках серого. Исходное изображение находится в верхнем левом углу. Результат применения глобального порога находится в правом верхнем углу, а результат применения адаптивного порога – в правом нижнем углу.

Основы пространственной фильтрации До сих пор мы в основном рассматривали точечную обработку, когда улучшение в любой точке изображения зависит только от уровня серого в этой точке. Однако, анализ больших областей дает значительно больше гибкости. Общий подход заключается в использовании функции значений исходного изображения f в предопределенной области (x, y) для определения значения выходного изображения g в (x, y) . В соответствие с этим, один из основных подходов основан на использовании так называемых масок (также называемых фильтрами, ядрами, шаблонами или окнами). По сути, маска представляет собой небольшой (скажем, 3×3) двумерный массив, в котором значения коэффициентов маски определяют характер процесса, например, повышение резкости изображения. Методы улучшения изображения, основанные на таком подходе, часто называют обработкой по маске или фильтрацией.

Название этого фильтра пришло из области частотной обработки изображения, о которой мы поговорим чуть позже. В частотной области «фильтрация» означает пропуск, изменение или отклонение определенных частотных компонентов изображения. Например, фильтр, пропускающий низкие частоты, называется фильтром нижних частот. Общий эффект, производимый фильтром нижних частот, заключается в сглаживании изображения за счет его размытия. Мы можем добиться аналогичного сглаживания непосредственно на самом изображении, используя пространственные фильтры.

Теперь давайте посмотрим на операции фильтрации, которые выполняются непосредственно над пикселями изображения. Мы используем термин пространственная фильтрация, чтобы отличить этот тип процесса от более традиционной фильтрации в частотной области.

Пространственная фильтрация изменяет изображение, модифицируя значение каждого пикселя при помощи функции, определяемой значениями пикселя и его соседей. Если операция, выполняемая над пикселями изображения, является линейной, то фильтр называется линейным пространственным фильтром. В противном случае фильтр представляет собой нелинейный пространственный фильтр.

На экране проиллюстрирована механика пространственной фильтрации. Процесс состоит в простом перемещении маски фильтра от точки к точке изображения. В каждой точке (x, y) отклик фильтра в этой точке вычисляется с использованием заранее определенного отношения. Для линейной

пространственной фильтрации ответ определяется суммой произведений коэффициентов фильтра и пикселей изображения, заключенных в ядро.

В общем случае, линейная фильтрация изображения f размером $M \times N$ с фильтрующей маской размером $m \times n$ формально задается выражением, показанным на экране. Координаты x и y изменяются таким образом, что каждый пиксель изображения f попадает в центр (начало координат) ядра один раз. Для фиксированного значения (x, y) мы вычисляем сумму произведений для ядра произвольного нечетного размера. Это уравнение является центральным инструментом линейной фильтрации.

В математике эта операция над двумя функциями (f – изображение и w – фильтр) называется *пространственной корреляцией*. Корреляция заключается в перемещении центра ядра по изображению и вычислении суммы произведений в каждой точке. Еще одна очень важная операция – это *пространственная свертка*. Механизм пространственной свертки такой же, за исключением того, что ядро корреляции поворачивается на 180°. Таким образом, когда значения ядра симметричны относительно его центра, корреляция и свертка дают один и тот же результат. По этой причине линейную пространственную фильтрацию часто называют «сверткой маски с изображением». Точно так же маски фильтров иногда называют *масками свертки*. Термин *ядро свертки* также широко используется.

Причину поворота ядра в свертке мы обсудим в дальнейшем.

2.4 Корреляция и свертка

Лучше всего можно понять разницу между корреляцией и сверткой на примере. Давайте посмотрим на эту одномерную иллюстрацию.

В верхней части экрана показаны функция $1 - D$, f , и ядро, w . Ядро имеет размер 1×5 . Единственная разница между корреляцией и сверткой в том, что для свертки ядро предварительно повернуто на 180°.

Чуть ниже на экране вы видите начальную позицию, используемую для выполнения корреляции и свертки, в которой w расположен так, что его центральный коэффициент совпадает с началом координат f .

Первое, что можно заметить, это то, что часть w лежит за пределами f , поэтому суммирование в этой области не определено. Решение этой проблемы – дополнить функцию f достаточным количеством 0 с обеих сторон. В общем случае, если размер ядра $1 \times m$, нам нужны $(m-1)/2$ нулей по обе стороны от f , чтобы обрабатывать начальную и конечную конфигурации w при помощи f . Сейчас вы видите функцию с правильным заполнением. В этой начальной конфигурации все коэффициенты ядра покрывают допустимые значения.

Первые значения корреляции и свертки представляют собой сумму произведений с соответствующими ядрами в этой начальной позиции. Чтобы получить остальные значения корреляции и свертки, мы продолжаем сдви-

гать относительные положения w и f на один шаг за раз до тех пор, пока не дойдем до конца f . Внизу экрана показывались результаты корреляции и свертки.

Обратите внимание, что потребовалось 8 шагов, чтобы полностью сдвинуть w по f так, чтобы центральный коэффициент из w попал в каждый пиксель f . Иногда бывает полезно, чтобы каждый элемент w попадал в каждый пиксель f . Для этого мы должны начать с крайнего правого элемента w , совпадающего с началом координат f , и закончить крайним левым элементом w , совпадающим с последним элементом f (тут потребуется дополнительное заполнение). Внизу экрана показан результат этой расширенной или полной корреляции и свертки. Как видите, мы можем получить «стандартную» корреляцию и свертку, обрезая полные расширенные версии.

Следует отметить два важных момента. Во-первых, как корреляция, так и свертка являются функциями смещения ядра фильтра относительно изображения. Другими словами, первые значения корреляции и свертки соответствуют нулевому смещению ядра, вторые – смещению на одну единицу и так далее. Второе, на что следует обратить внимание, – корреляция ядра w с функцией, содержащей все нули и единственную единицу, дает копию w , но повернутую на 180° . Функция, которая содержит одну единицу, и остальные – нули, называется дискретным единичным импульсом. Корреляция ядра с дискретным единичным импульсом дает повернутую версию ядра в месте расположения импульса.

В случае свертки результат предварительного вращения ядра состоит в том, что теперь у нас есть точная копия ядра в месте расположения единичного импульса. Таким образом, утверждение, что свертка функции с импульсом дает копию функции в месте расположения импульса является основой теории линейных систем.

Эта одномерная концепция легко распространяется на изображения.

Корреляция и свертка двумерного ядра Для ядра размером $m \times n$ мы заполняем изображение как минимум $(m - 1)/2$ строками нулей сверху и снизу, а также $(n - 1)/2$ столбцами нулей слева и справа. В примере на экране m и n равны 3, поэтому мы дополняем f одной строкой, состоящей из нулей, сверху и снизу, и одним столбцом из нулей слева и справа.

На экране также показано начальное положение ядер для выполнения корреляции и свертки, а также конечный результат после того, как центр w попадает в каждый пиксель f , и мы вычисляем сумму произведений в каждой позиции. Как и раньше, результатом корреляции является копия ядра, повернутая на 180° . Для свертки мы предварительно вращаем ядро, как и раньше, и снова вычисляем скользящую сумму произведений. В результате мы получим, что свертка функции с импульсом копирует функцию в месте

расположения импульса. Как отмечалось ранее, корреляция и свертка дают одинаковый результат, если значения ядра симметричны относительно центра.

Обобщая все это в виде уравнения, получим, что корреляция ядра w размера $m \times n$ с изображением $f(x, y)$ задается верхним уравнением на экране:

$$(w \star f)(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (4)$$

Как упоминалось ранее, $a = (m-1)/2$, $b = (n-1)/2$, и мы предполагаем, что f заполнено соответствующим образом.

Аналогично, свертку ядра w размера $m \times n$ с изображением $f(x, y)$ можно определить как

$$(w \star f)(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t) \quad (5)$$

В этом уравнении знаки минус выравнивают координаты f и w при повороте одной из функций на 180° . Это уравнение задает процесс суммирования произведений, который мы будем называть линейной пространственной фильтрацией. То есть линейная пространственная фильтрация и пространственная свертка синонимичны.

Свойства Операция свертки удовлетворяет следующим алгебраическим свойствам:

- Коммутативность
- Ассоциативность
- Распределенность
- Ассоциативность со скалярным умножением

Корреляция не является ни коммутативной, ни ассоциативной.

Поскольку свертка коммутативна, неважно, что поворачивается – w или f , вращение ядра определяется заранее.

2.5 Фильтры и сглаживание

Сглаживание – линейные фильтры Сглаживающие фильтры используются для размытия и уменьшения шума. Размытие используется на этапах предварительной обработки, таких как удаление мелких деталей из изображения перед извлечением (большого) объекта, а также для заполнения

небольших промежутков в линиях или кривых. Снижение шума может быть достигнуто путем размытия с помощью линейного фильтра, а также с помощью нелинейной фильтрации.

Выходной сигнал (отклик) сглаживающего линейного пространственного фильтра – это просто среднее значение пикселей, содержащихся в области маски фильтра. Эти фильтры иногда называют фильтрами усреднения.

Идея сглаживающих фильтров проста. Если мы заменяем значения каждого пикселя в изображении на среднее значение уровней серого в области, определенной маской фильтра, мы получим изображению с уменьшенными «резкими» переходами в уровнях серого. Поскольку случайный шум обычно состоит именно из таких резких переходов уровней серого, наиболее очевидным применением сглаживания является уменьшение шума. Однако края изображения (которые почти всегда являются теми характеристиками, которые желательно сохранить) также характеризуются резкими переходами в уровнях серого, поэтому усредняющие фильтры имеют нежелательный побочный эффект – они размывают края. В основном фильтры усреднения используются для уменьшения «несущественных» деталей изображения. Под «несущественными» мы подразумеваем области пикселей, которые малы по сравнению с размером маски фильтра.

На экране показаны два сглаживающих фильтра размером 3×3 . Использование первого фильтра дает стандартное среднее значение пикселей под маской. Такой фильтр пространственного усреднения, в котором все коэффициенты равны, иногда называют блочным фильтром.

Вторая маска, показанная на экране, немного интереснее. Эта маска дает так называемое средневзвешенное значение, что означает, что пиксели умножаются на разные коэффициенты, что придает большее значение (вес) одним пикселям за счет других. В маске, показанной справа, пиксель в центре маски умножается на большее значение, чем любой другой, что придает пикселю в центре маски большее значение при вычислении среднего значения. Вес остальных пикселей находится в обратной зависимости от их расстояния от центра маски. Эта стратегия максимального взвешивания центральной точки является попыткой уменьшить размытие в процессе сглаживания. Мы могли бы выбрать другие веса для достижения той же общей цели. На практике, как правило, трудно увидеть различия между изображениями, сглаженными с помощью любой из масок, показанных на экране, или аналогичных масок, потому что область, охватываемая этими масками в любом месте изображения, очень мала.

Усредняющий фильтр Давайте посмотрим, как усредняющий фильтр преобразует изображение. Обозначим исходное изображение – F , а изображение, полученное в результате – G . Начнем применять усредняющий в левом

верхнем углу. Итоговое значение для позиции, выделенной на изображении G , представляет собой среднее значение всех пикселей под маской в изображении F . Поскольку все значения под маской в этой позиции равны 0, итоговое значение для этой позиции в G также равно 0.

Затем мы перемещаем маску на один пиксель вправо и вычисляем новое среднее значение для этой позиции. В этом случае у нас есть один пиксель со значением 90, поэтому в результате получается среднее значение 10.

Мы продолжаем перемещать это скользящее окно и вычислять значения для всех внутренних позиций для изображения G .

Обработка краев Вы, наверное, заметили, что мы не можем вычислять значения по краям изображения G . Эта проблема характерна для всех пространственных фильтров, анализирующих некую область, и существует несколько способов справиться с ней. Самый простой – не вычислять результирующие значения для позиций вдоль края. Полученное отфильтрованное изображение будет меньше исходного, но все пиксели отфильтрованного изображения будут обработаны с использованием полной маски. Если требуется, чтобы результат был того же размера, что и оригинал, традиционный подход состоит в том, чтобы «заполнить» исходное изображение по краям либо нулевыми значениями (или другим постоянным уровнем серого), либо выполнить заполнение путем репликации или зеркального отражения столбцов и строк краев. Количество заполненных пикселей зависит от размера фильтра. Для фильтра 3×3 , как в этом примере, нам нужно заполнить одну строку и один столбец со всех сторон исходного изображения. Часто для описания заполнения используется следующая терминология.

- Допустимое заполнение (valid padding) – заполнение не выполняется; итоговое изображение меньше исходного.
- То же заполнение (same padding) – исходное изображение дополняется по краям, чтобы фильтрация не изменяла размер изображения.

Теперь рассмотрим пример сглаживания при помощи усредняющего фильтра. Слева – исходное изображение, справа – результат сглаживания при помощи усредняющего фильтра. Обратите внимание, что полученное изображение имеет некоторые краевые эффекты. Это связано с тем, что все пиксели в области одинаково влияют на результирующее значение. Чтобы устранить краевые эффекты, мы можем выбрать другой фильтр, например, такой, в котором взвешенный вклад соседних пикселей зависит от их близости к центру.

Сглаживание – фильтр Гаусса Одним из фильтров, обладающих этим свойством, является фильтр Гаусса, также часто называемый размытием по

Гауссу. Это еще один из часто используемых сглаживающих фильтров. В нем используется ядро, имеющее гауссовскую "колоколообразную" форму. Идея сглаживания по Гауссу состоит в том, чтобы использовать это двумерное распределение как функцию "рассеяния точки". Коэффициенты ядра уменьшаются с увеличением расстояния от центра ядра. Центральные пиксели имеют больший вес, чем периферийные.

Математически применение фильтра Гаусса к изображению аналогично свертке изображения с помощью функции Гаусса. Оно также называется двумерным преобразованием Вейерштрасса.

Ядро Гаусса непрерывно. Чтобы применить этот фильтр к цифровому изображению, нам нужен его дискретный эквивалент. Дискретное приближение к функции Гаусса теоретически требует бесконечно большого ядра свертки, так как гауссово распределение нигде не равно нулю. К счастью, распределение Гаусса очень близко к нулю примерно при трех стандартных отклонениях от среднего. 99% распределения попадает в 3 стандартных отклонения. Это означает, что мы обычно можем ограничить размер ядра, чтобы оно содержало только значения в пределах трех стандартных отклонений от среднего. Чаще всего дискретным эквивалентом является дискретное ядро Гаусса, которое создается точками дискретизации из непрерывной двумерной функции Гаусса.

Стандартное отклонение фильтра можно интерпретировать как оценку его размера, и оно напрямую влияет на степень "размытия". Большие значения σ дают более широкий пик (большее размытие). Важно отметить, что размер ядра должен увеличиваться с увеличением σ , чтобы сохранить гауссовский характер фильтра. На краю маски коэффициенты должны быть близки к 0.

На экране показан пример размытия изображения с помощью фильтров Гаусса. Фильтр с большим значением σ дает большее размытие, а также требует большего размера маски.

Сейчас на экране мы видим сравнение результатов фильтрации с применением усредняющего фильтра и фильтром Гаусса.

Сглаживание – медианный фильтр Другие часто используемые пространственные фильтры – это нелинейные фильтры, основанные на порядковых статистиках, т.е. производится упорядочивание (ранжирование) пикселей, содержащихся в области изображения, охватываемой фильтром, и последующая замена значения центрального пикселя значением, полученным после ранжирования. Наиболее известным примером в этой категории фильтров является медианный фильтр, который, как следует из названия, заменяет значение пикселя на медианное значение уровней серого в окрестности этого пикселя (исходное значение пикселя включается в вычисление медианы).

Медианные фильтры довольно популярны, потому что для определенных типов случайного шума они обеспечивают отличные возможности шумоподавления со значительно меньшим размытием, чем линейные сглаживающие фильтры аналогичного размера. Медианные фильтры особенно эффективны при наличии импульсного шума, также называемого шумом соли и перца, поскольку этот шум на изображении выглядит как белые и черные точки, наложенные на изображение.

Давайте посмотрим на крайнее левое изображение на экране. Это рентгеновское изображение печатной платы, сильно искаженной биполярным импульсным шумом (также известным как шум соли и перца (salt and pepper noise)). Изображение в центре иллюстрирует преимущество медианной фильтрации в подобных случаях над усредняющим фильтром. Мы видим результат обработки зашумленного изображения с маской усреднения размером 3×3 , а самое правое изображение является результатом использования медианного фильтра 3×3 . Изображение обработанное усредняющим фильтром имеет меньше видимых шумов, но в результате мы имеем значительное размытие. Превосходство медианной фильтрации над фильтрацией при помощи усреднения в этом случае вполне очевидно. В целом, медианная фильтрация намного лучше подходит для удаления аддитивного шума соли и перца, чем усреднение.

Повышение резкости Теперь, когда мы знаем, как размыть изображение, легко добиться повышения резкости. Повышение резкости широко используется на практике – от цифровой печати и медицинских снимков до систем автоматического контроля в производстве и автономного управления в военных системах. Основная идея повышения резкости состоит в том, чтобы выделить мелкие детали на изображении или улучшить те детали, которые были размыты. Что мы теряем при размытии изображения? Мы теряем мелкие детали. Если мы добавим эти детали к исходному изображению, мы получим изображение с повышенной резкостью. Эта идея используется в простейшей технике повышения резкости, называемой «нерезким фильтром» или «нерезким маскированием». В этой технике используется размытое или «нерезкое» негативное изображение для создания маски по исходному изображению, что и объясняет название фильтра. Затем маска объединяется с исходным изображением, создавая изображение, которое менее размыто, чем исходное.

Мы вернемся к повышению резкости изображения позже, когда будем обсуждать обнаружение неоднородностей – отдельных точек, линий и краев.