8 Фильтрация изображений

8.1 Общие принципы

Кроме простейших алгоритмов обработки фотоизображения, рассмотренных выше, существует великое множество полезных манипуляций с цифровыми фотоизображениями. А если изображение снято не в фокусе? В расплывчатых изображениях можно увеличить резкость, и, наоборот, четкие, контрастные изображения можно размыть, имитируя эффект смягчающих фотофильтров.

Фильтрация изображений является одной из самых фундаментальных операций технического зрения, распознавания образов и обработки изображений. Фактически, с той или иной фильтрации исходных изображений начинается работа подавляющего большинства методов. Рассматриваемые в данном модуле фильтры имеют, таким образом, чрезвычайную важность с точки зрения их применения в различных приложениях.

Основная стратегия, используемая при фильтрации, состоит в том, чтобы применять взвешенные суммы соседних пикселей, используя различные весовые коэффициенты. Несмотря на свою простоту такой процесс очень полезен. Он позволяет сглаживать шумы на изображении, определять края и получать интересные визуальные эффекты.

Для начала введём специальный термин: *маска фильтра (или скользящее окно, или апертура)* представляет собой матрицу размера $n \times m$ в общем случае. Она накладывается на изображение и осуществляется умножением элементов *маски фильтра* и соответствующих элементов изображения с последующей обработкой результата. В большинстве случаев апертура фильтра квадратная и имеет нечетные размеры. Обычно работают не с полными размерами апертуры, а с половинными размерами *r=[n/2], q=[m/2]*, где квадратные скобки это операция взятия целой части.

Заметим также, что если сумма коэффициентов ядра больше 1, то яркость изображения будет увеличиваться; если меньше 1 - яркость уменьшится.

Иногда операцию наложения фильтра называют операцией свертки, а маску фильтра – *ядром* свертки. Записывают операцию свертки функции f по ядру h следующим образом

$$g_{i,j} \equiv f_{i,j} \otimes h_{i,j} \quad \text{где } i,j \in Z$$

Или с точки зрения фильтрации изображения

$$\widetilde{f}_{i,j} == \sum_{k=-r}^{r} \sum_{l=-q}^{q} h_{k,l} f_{i-k,j-l}, \qquad r=[n/2],\ q=[m/2]$$

Когда маска передвигается к границе изображения, возникает так называемое *явление краевого эффекта.* Во избежание этого нежелательного эффекта необходимо, когда маска вышла за пределы исходного изображения, дополнить его ненулевыми элементами, например, элементами изображения, симметричными относительно его краев.

Апертура фильтра постепенно передвигается по изображению слева направо и сверху вниз на один пиксель (то есть на следующем шаге фильтр работает с окном, состоящим не только из элементов исходного изображения, но и из элементов, ранее подвергнувшихся преобразованию, – своего рода «принцип снежного кома»). Кроме того, заметим, что если речь идёт об апертуре, представляющей собой строку элементов изображения (A_1,A_2,A_3), то такое преобразование называется одномерным; соответственно, когда маска – матрица - это двумерное преобразование. Обычно

используется апертуры с нечетными размерами, чтобы совместить центральный элемент матрицы с тем пикселем изображения, для которого и вычисляется свертка в данный момент (иногда говорят, что пиксель находится под центром ядра)
Речь пойдёт, в основном, о фильтрах шумоподавления, но для того, чтобы можно было быстрее и нагляднее оценивать их работу, требуется реализовать ещё и возможность искусственного зашумления изображения. Одним из способов является получение изображения с равномерным шумом

$$I_{i,j} = g_{i,j} + Err(i,j),$$

где *Err(i,j)* – нормально распределенная случайная величина.

8.2 Фильтры шумоподавления

Одним из их возможных применений сглаживающих фильтров является шумоподавление, т.е. задача восстановления исходного изображения, к пикселям которого добавлен случайный шум. Шум меняется независимо от пикселя к пикселю и, при условии, что математическое ожидание значения шума равно нулю, шумы соседних пикселей будут компенсировать друг друга. Чем больше окно фильтрации, тем меньше будет усредненная интенсивность шума, однако при этом будет происходить и существенное размытие значащих деталей изображения.

При размывании перераспределяются цвета в изображении и смягчаются резкие границы, в то время как при увеличении резкости подчеркиваются различия между цветами смежных пикселей и выделяются незаметные детали. Такие фильтры называют сглаживающими, они позволяют подавить высокочастотные (мелкие) шумы, снижая локальную контрастность изображения.

Ядро **равномерного фильтра**, состоит из совокупности коэффициентов, каждый из которых меньше 1, а их сумма составляет 1. В простейшем случае каждый элемент ядра размером $n \times m$ равен *1/mn*. Это означает, что каждый пиксель поглотит что-то из цветов соседей, но полная яркость изображения останется неизменной.
Итоговое изображение получилось размытым по сравнению с оригиналом потому, что цвет каждого пикселя распространился среди соседей. Степень размывания можно увеличить либо используя ядро большего размера, чтобы распределить цвета среди большего числа соседей, либо, подбирая коэффициенты ядра и уменьшая влияние центрального коэффициента, либо фильтруя изображение еще раз с ядром размывания. Характерной чертой этого фильтра, отличающей его, к примеру, от эффекта расфокусировки линз в реальной жизни, является то, что образом белой точки на черном фоне будет равномерно серый квадрат.

Естественным предположением об исходном незашумленном изображении будет схожесть значений интенсивности пикселей, находящихся рядом. Причем чем меньше расстояние между пикселями, тем больше вероятность их похожести. Заметим, что в зашумленных изображениях схожесть пикселей никак не зависит от расстояния между ними. Исходя из вышесказанного можно предположить, что шумоподавление при помощи прямоугольного фильтра имеет существенный недостаток: пиксели, находящиеся на значительном расстоянии от обрабатываемого, оказывают на результат тот же эффект, что и ближние.

Более эффективное шумоподавление можно, таким образом, осуществить, если влияние пикселей на обрабатываемый будет уменьшаться с расстоянием. Этим свойством обладает **гауссовский фильтр**. Размытие с помощью Гауссова фильтра (Gaussian blur) это фактически свертка по функции:

$$\tilde{I}(i,j) = \sum_{l=-r}^{r}\sum_{k=-q}^{q} I(i-l)(j-k)\cdot A\cdot e^{-\frac{(l^2+k^2)}{2\sigma^2}},$$

где *i,j* – обрабатываемая точка изображения; $A_{i,j}$ –нормирующая величина, она выбирается так, чтобы сумма коэффициентов ядра фильтра была равно 1; σ - среднеквадратичное отклонение, задает степень размытия. При высоких σ влияние удаленных пикселей увеличивается и степень размытия повышается.

Значение *A* обычно выбирается следующим образом

$$h_{l,k} = e^{-\frac{(l^2+k^2)}{2\sigma^2}}$$, где $l=[-r \div r]$, $k=[-q \div q]$

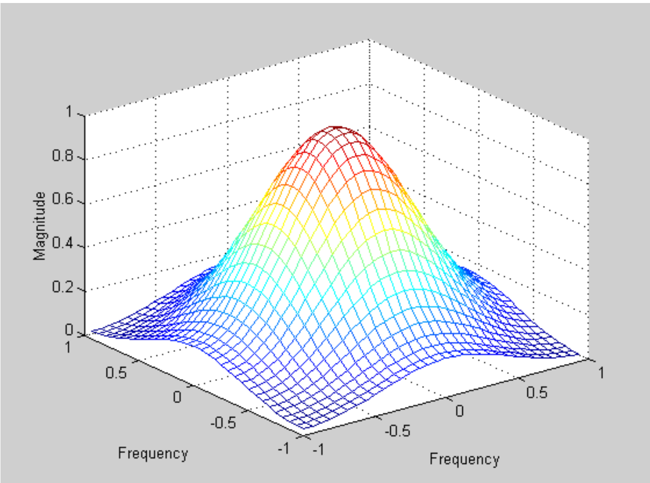
$$A = \frac{1}{\sum_{l=-r}^{r}\sum_{k=-q}^{q} h_{l,k}}$$

Пример маски для фильтра Гаусса размера 5×5 и с σ=0,5:

0.0000	0.0000	0.0002	0.0000	0.0000
0.0000	0.0113	0.0837	0.0113	0.0000
0.0002	0.0837	0.6187	0.0837	0.0002
0.0000	0.0113	0.0837	0.0113	0.0000
0.0000	0.0000	0.0002	0.0000	0.0000

Пример маски для фильтра Гаусса размера 5×5 и с σ= 0,7:

0.0001	0.0020	0.0055	0.0020	0.0001
0.0020	0.0422	0.1171	0.0422	0.0020
0.0055	0.1171	0.3248	0.1171	0.0055
0.0020	0.0422	0.1171	0.0422	0.0020
0.0001	0.0020	0.0055	0.0020	0.0001



Все линейные алгоритмы фильтрации приводят к сглаживанию резких перепадов яркости изображений, прошедших обработку. Этот недостаток, особенно существенный, если потребителем информации является человек, принципиально не может быть исключен в рамках линейной обработки.

Вторая особенность линейной фильтрации - ее оптимальность при гауссовском характере помех. Обычно этому условию отвечают шумовые помехи на изображениях, поэтому при их подавлении линейные алгоритмы имеют высокие показатели. Однако, часто приходится иметь дело с изображениями, искаженными помехами других типов. Одной из них является импульсная помеха. При ее воздействии на изображении наблюдаются белые или (и) черные точки, хаотически разбросанные по кадру. Применение линейной фильтрации в этом случае неэффективно - каждый из входных импульсов (по сути - дельта-функция) дает отклик в виде импульсной характеристики фильтра, а их совокупность способствует распространению помехи на всю площадь кадра.

Удачным решением перечисленных проблем является применение **медианной фильтрации**, предложенной Дж. Тьюки в 1971 г.

При медианной фильтрации используется окно, обычно имеющее центральную симметрию. Наиболее часто применяемые варианты окон - крест или квадрата. Размеры апертуры принадлежат к числу параметров, оптимизируемых в процессе анализа эффективности алгоритма. Отсчеты изображения, оказавшиеся в пределах окна, образуют *рабочую выборку* текущего шага.

Двумерный характер окна позволяет выполнять, по существу, двумерную фильтрацию, поскольку для образования оценки привлекаются данные как из текущих строки и столбца, так и из соседних. Если упорядочить последовательность $\{y_i,\ i = \overline{1,n}\}$ по возрастанию, то ее *медианой* будет тот элемент выборки, который занимает центральное положение в этой упорядоченной последовательности. Полученное таким образом число и является продуктом фильтрации для текущей точки кадра.

Если импульсная помеха не является точечной, а покрывает некоторую локальную область, то она также может быть подавлена, если размер этой локальной области будет меньше, чем половина размера апертуры медианного фильтра. Поэтому для подавления импульсных помех, поражающих локальные участки изображения, следует увеличивать размеры апертуры МФ.

Импульсная помеха. При ее наложении использовался датчик случайных чисел с равномерным на интервале [0, 1] законом распределения, вырабатывающий во всех точках кадра независимые случайные числа. Интенсивность помехи задавалась вероятностью p ее возникновения в каждой точке. Если для случайного числа n_{i_1,i_2}, сформированного в точке (i_1,i_2), выполнялось условие $n_{i_1,i_2} < p$, то яркость изображения x_{i_1,i_2} в этой точке *замещалась* числом 255, соответствующим максимальной яркости (уровню белого).

Вместе с тем, как говорилось выше, медианная фильтрация в меньшей степени сглаживает границы изображения, чем любая линейная фильтрация. Механизм этого явления очень прост и заключается в следующем. Предположим, что апертура фильтра находится вблизи границы, разделяющей светлый и темный участки изображения, при этом ее центр располагается в области темного участка. Тогда, вероятнее всего, рабочая выборка будет содержать большее количество элементов с малыми значениями яркости, и, следовательно, медиана будет находиться среди тех элементов рабочей выборки, которые соответствуют этой области изображения. Ситуация меняется на противоположную, если центр апертуры смещен в область более высокой яркости. Но это и означает наличие чувствительности у МФ к перепадам яркости.

8.3 Фильтры увеличения резкости

В ядре резкости центральный коэффициент больше 1, а окружен он отрицательными числами, сумма которых на единицу меньше центрального коэффициента. Таким образом, увеличивается любой существующий контраст между цветом пикселя и цветами его соседей. Увеличение резкости достигается точно так же, как и размывание, за исключением того, что используются другое ядро с иной целью увеличить, а не уменьшить четкость изображения.
Общий вид ядра 3×3 для повышения резкости выглядит следующим образом:

$$h = \begin{pmatrix} -k/8 & -k/8 & -k/8 \\ -k/8 & k+1 & -k/8 \\ -k/8 & -k/8 & -k/8 \end{pmatrix},$$

где параметр k определяет степень увеличения резкости. Обычно используется $k=2$.

Эффект повышения резкости достигается за счет того, что фильтр подчеркивает разницу между интенсивностями соседних пикселей, удаляя эти интенсивности друг от друга. Этот эффект будет тем сильней, чем больше значение центрального члена ядра. Характерным артефактом линейной контрастоповышающей фильтрации являются заметные светлые и менее заметные темные ореолы вокруг границ.

8.4 Оконтуривание (нахождение границ)

Все методы выделения границ работают с *яркостью* точки, то есть со значением, полученным из значений трёх цветовых каналов по известной формуле (см. п. 7.X). Однако для этого вовсе необязательно предварительно преобразовывать всё изображение к оттенкам серого, достаточно лишь получать значение яркости в тот момент и для той точки, с которой идёт работа, а полученное в результате преобразований значение повторять по трём каналам.

8.4.1 Поиск границ на основе градиента

Существует несколько подходов к проблеме выделения границ [Форсайт Понс]. Первое семейство методов основано на приближенном вычисление градиента, анализе его направления и абсолютной величины.

В точках большого перепада яркости градиент имеет большее значение. Это неудивительно, поскольку (частные) производные по определению соответствуют скоростям изменения функции (яркости) по вертикали и горизонтали. Взяв пиксели с соответствующей длиной градиента, большей порога α, мы получим некий алгоритм нахождения границ.

Недостаток алгоритма: Если же мы имеем дело с зашумленным изображением, то карту граничных точек будут загрязнять не только реально существующие, но и несущественные детали и просто шум. Так произойдет, поскольку в нашем алгоритме мы не учли, что граничные точки соответствуют не просто перепадам яркости, а перепадам яркости между относительно монотонными областями.

Как отделить подобные перепады яркости от перепадов яркости, вызванных шумами и несущественными деталями? Для этого изображение подвергают сглаживающей гауссовской фильтрации. Такое решение проблемы, на первый взгляд, парадоксально - для нахождения границ мы их с начала размываем. Данный прием основывается на том, что при сглаживающей фильтрации мелкие несущественные детали будут размываться существенно быстрее перепадов между областями.

Однако мы видим, что сильнее проявил себя другой недостаток алгоритма: четко выраженные границы проявились как жирные линии в несколько пикселей толщиной. К тому же неопределенность с выбором порога α не преодолена:

Результат необходимо получать при помощи подбора значения α вручную, и, очевидно, такое значение оптимально не для всех частей изображения

Простейшим дифференциальным оператором является взятие производной по x-координате. Существует множество способов определить аналогичный оператор для дискретных изображений при помощи линейного фильтра [Иванов и др.]. В частности, распространенными вариантами являются фильтры Превита (Prewitt) и Собеля (Sobel).

Маска фильтра Собеля для выделения горизонтальных границ:

$$h = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Маска фильтра Превита для выделения горизонтальных границ:

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}.$$

Фильтры, приближающие оператор производной по y-координате, получаются путем транспонирования матриц.

В силу того, что данные фильтры не меняют среднюю интенсивность изображения (сумма элементов ядра равна единице), в результате их применения получается, как правило, изображение со средним значением пикселя близким к нулю (сумма элементов ядра равна нулю). Вертикальным перепадам (границам) исходного изображения соответствуют пиксели с большими по модулю значениями на результирующем изображении.

Неплохие результаты дает фильтрация с ядром Робертса:

$$h = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$h = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

8.4.2 Поиск границ на основе лапласиана

Как известно из математического анализа, необходимым и достаточным условием экстремального значения первой производной функции в некой точке является равенство нулю второй производной в этой точке, причем по разные стороны от точки вторая производная должна иметь разные знаки. Про такую точку говорят, что вторая производная в ней пересекает ноль.

В двумерном случае, аналогом второй производной является скалярный оператор, называемый лапласианом:

$$\Delta f = \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)$$

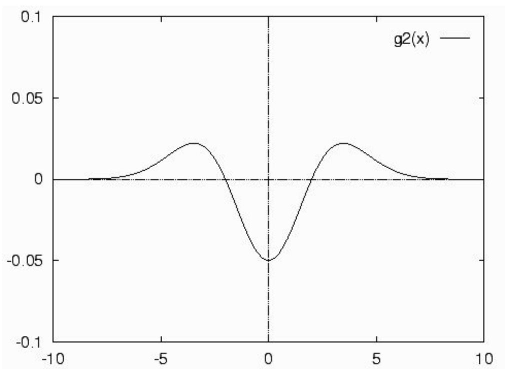
Нахождение границ на изображении может, таким образом, производиться по аналогии с одномерным случаем: граничными признаются точки, в которых лапласиан равен нулю и вокруг которых он имеет разные знаки. Кроме того, оценка лапласиана при помощи линейной фильтрации предваряется гауссовской сглаживающей фильтрацией, чтобы снизить чувствительность алгоритма к шуму (аналогично тому, как это описано выше).

Композиция линейных фильтров есть линейный фильтр. Поэтому гауссовское сглаживание и поиск лапласиана можно осуществить единовременно при помощи фильтра, который называется **лапласиан-гауссиана**. Поиск пересечений нуля, так же, как и линейная фильтрация, является сравнительно быстрой операцией, поэтому, описанный выше алгоритм применяется в системах, где принципиально и качество результата и быстродействие.

$$\Delta f(x, y) = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y}; \quad g(x, y) = Ae^{-\frac{x^2+y^2}{\sigma}};$$

$$\Delta g(x, y) = A' \left(2 - \frac{x^2 + y^2}{\sigma} \right) e^{-\frac{x^2+y^2}{\sigma}}$$

График второй производной одномерной функции Гаусса с $\sigma = 8$



Простейший вариант дискретного приближения ядра лапласиана:

$$h = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Пример матрицы размера 11×11

$$h=\begin{bmatrix} 0 & 0 & 0 & -1 & -1 & -2 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -2 & -4 & -8 & -9 & -8 & -4 & -2 & 0 & 0 \\ 0 & -2 & -7 & -15 & -22 & -23 & -22 & -15 & -7 & -2 & 0 \\ -1 & -4 & -15 & -24 & -14 & -1 & -14 & -24 & -15 & -4 & -1 \\ -1 & -8 & -22 & -14 & 52 & 103 & 52 & -14 & -22 & -8 & -1 \\ -2 & -9 & -23 & -1 & 103 & 178 & 103 & -1 & -23 & -9 & -2 \\ -1 & -8 & -22 & -14 & 52 & 103 & 52 & -14 & -22 & -8 & -1 \\ -1 & -4 & -15 & -24 & -14 & -1 & -14 & -24 & -15 & -4 & -1 \\ 0 & -2 & -7 & -15 & -22 & -23 & -22 & -15 & -7 & -2 & 0 \\ 0 & 0 & -2 & -4 & -8 & -9 & -8 & -4 & -2 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -2 & -1 & -1 & 0 & 0 & 0 \end{bmatrix}$$

В результате применения дискретного лапласиана большие по модулю значения соответствуют как вертикальным, так и горизонтальным перепадам яркости. h_Δ является, таким образом, фильтром, находящим границы любой ориентации. Нахождение границ на изображении может производиться путем применения этого фильтра и взятия всех пикселей, модуль значения которых превосходит некоторый порог. Однако такой алгоритм имеет существенные недостатки. Главный из них - неопределенность в выборе величины порога. Для разных частей изображения приемлемый результат обычно получается при существенно разных пороговых значениях. Кроме того, разностные фильтры очень чувствительны к шумам изображения.

Чтобы еще уменьшить чувствительность алгоритма к несущественным деталям, из числа граничных точек можно исключить те, длина градиента в которых меньше порога. Для одномерного случая это будет соответствовать разумному требованию того, чтобы в точке перепада величина первой производной была не слишком маленькой.

8.5 Примеры декоративных фильтров и спецэффектов

Акварелизация. Первый шаг в применении акварельного фильтра - сглаживание цветов в изображении. Одним из способов сглаживания является процесс медианнного осреднения цвета в каждой точке – так называемый *медианный фильтр*.

Медиана – средний элемент последовательности в результате её упорядочения по возрастанию/убыванию и присваиванию найденного значения только среднему элементу (речь снова о нечётной апертуре). Например, для той же апертуры 3 и двумерного фильтра мы должны упорядочить 9 точек (например, по возрастанию), после чего значение 5-ой точки упорядоченной последовательности отправить в центр окна

После сглаживания цветов медианным фильтром (иногда это делают несколько раз), необходимо обработать изображение ядром резкости, чтобы подчеркнуть границы переходов цветов.

Результирующее изображение напоминает акварельную живопись. Это лишь один пример, который показывает, как можно объединять различные методы обработки изображений и добиваться необычных визуальных эффектов.

Тиснение преобразует изображение так, что фигуры внутри изображения смотрятся так, как будто они выдавлены на поверхности.

Предварительно изображение обрабатывается каким либо сглаживающим фильтром, после чего применяется ядро тиснения.

В отличие от ядер размывания и резкости, в которых сумма коэффициентов равна 1, сумма весов в ядре тиснения равна 0. Это означает, что "фоновым" пикселам (пикселам, которые не находятся на границах перехода от одного цвета к другому) присваиваются нулевые значения, а нефоновым пикселам - значения, отличные от нуля.

Примеры ядер тиснения

$$h = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad h = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Есть и другие способы получить ядро для тиснения. Один из общих случаев выглядит так:

$$h = \begin{pmatrix} c & c & c \\ c & c & c \\ -2c & -2c & -2c \end{pmatrix},$$

где *c* - некоторый коэффициент, обычно от 0.5 до 1.

После того, как значение пиксела обработано ядром тиснения, выполняется нормировка - к полученному значению прибавляется 128. Таким образом, значением фоновых пикселов станет средний серый цвет. Суммы, превышающие 255, можно округлить до 255 или взять остаток по модулю 255, чтобы значение оказалось между 0 и 255.

Заметим, что направление подсветки изображения можно изменять, меняя позиции 1 и -1 в ядре. Если, например, поменять местами значения 1 и -1, то реверсируется направление подсветки.