

1. Изображения как структура данных. Базовые операции над изображением. Свертка изображения

2. Морфологические операции. Эрозия, дилатация, замыкание и размыкание: что это и для чего могут быть использованы.

3. Фильтр границ Канни, для чего используется, какие параметры за что отвечают.

Фильтр границ Канни (Кэнни) – алгоритм, который среди всех пикселей изображения в градациях серого выделяет множество пикселей, которые образуют границы между объектами. Содержит следующие шаги:

1. Сглаживание изображения с целью устранения шума. Сглаживание выполняется путем сворачивания изображения с гауссовым ядром фиксированного размера: $I = H(\sigma) * I_0$. Слишком маленькие значения σ не смогут убрать шум, что приведет к множеству ложноположительных срабатываний, а слишком большие превратят все изображение в слабо меняющийся градиент и уничтожат все границы.
2. Вычисление градиента, то есть полей частных производных яркости по двум координатам. Как правило, используется разностная схема размера 3×3 , то есть свертка с ядром оператора Собеля:

$$D_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad D_y = D_x^T;$$
$$G_x = D_x * I, \quad G_y = D_y * I.$$

Однако этот шаг может быть объединен с предыдущим: для этого нужно сворачивать исходное изображение с дифференцированным гауссовым ядром, то есть с дискретными приближениями $\partial_x h(0, \sigma), \partial_y h(0, \sigma)$.

3. По полученным значениям вычисляется массив абсолютных значений градиента ($G = \sqrt{G_x^2 + G_y^2}$, поэлементно) и массив направлений. Все направления округляются до одного из основных: вертикаль, горизонталь или одна из двух диагоналей, при этом сторона (влево или вправо) значения не имеет.
4. Все пиксели помечаются как границы или как не-границы по следующим правилам в указанном порядке:
 - (a) Если элемент не является локальным максимумом в направлении *своего* градиента, то элемент отмечается как не-граница. Например, если округленное направление градиента в I_{ij} – вертикаль, то $G_{ij} \leq G_{i-1,j} \wedge G_{ij} \leq G_{i+1,j} \implies B_{ij} = 0$, где B – выходной булев массив того же размера, что и изображение.
 - (b) Если модуль градиента $G_{ij} < \theta_{\text{low}}$, то $B_{ij} = 0$.
 - (c) Если $G_{ij} > \theta_{\text{high}}$, то $B_{ij} = 1$.
 - (d) В противном случае, если пиксель является локальным максимумом, но его абсолютное значение лежит между двумя пороговыми значениями, то он считается границей, если хотя бы один из 8 соседних был определен как граница с использованием предыдущего правила.

Параметры θ_{low} , θ_{high} регулируют количество ошибок обоих родов, но придать им какой-либо физический смысл довольно сложно.

4. Преобразование Радона. Дискретное преобразование Радона. Оценка сложности.

5. Виды параметризации прямых на изображении и их свойства. Повторное вычисление преобразования Хафа и связь этой процедуры с поиском точки схода.

6. Преобразование Хафа и быстрое преобразование Хафа. Описание работы алгоритмов и их вычислительных характеристик.

7. Трехмерное быстрое преобразование Хафа для плоскостей. Параметризация, описание работы, вычислительная сложность.

8. Трехмерное быстрое преобразование Хафа для прямых. Параметризация, описание работы, вычислительная сложность.

9. История развития томографии. Строение томографа.

10. Взаимодействие рентгеновского излучения с веществом. Сведение зарегистрированных данных к виду преобразования Радона.

11. Преобразование Радона. Синограмма.

Опр. 1. Преобразование Радона Пусть $l_{\theta,s}$ – прямая, направляющий вектор

которой направлен под углом θ ($\theta = 0$ соответствует горизонтальной прямой) и удаленная от начала координат на расстояние s . Тогда преобразование Радона функции $f(x, y)$ называется интеграл этой функции по параметризованной прямой:

$$\begin{aligned} [\mathcal{R}f](\theta, s) &= \int_{l_{\theta, s}} f(x, y) dl = \int_{\mathbb{R}^2} f(x, y) \delta(x \cos \theta + y \sin \theta - s) dx dy = \\ &= \int_{\mathbb{R}} f(s \cos \theta + z \sin \theta, s \sin \theta - z \cos \theta) dz. \end{aligned}$$

Каждая точка Радон-образа функции представляет собой “сумму” по прямой с определенными параметрами. Как правило, по обоим параметрам рассматривается равномерная дискретная сетка значений, где θ меняется от 0 до π , s – от 0 до некоторого максимального значения, соответствующего размеру сцены. Полученный массив значений называется синограммой, так как Радон-образом точечной функции является синусоида.

12. Теорема о центральном сечении.

Преобразование Фурье функций от одной и от двух переменных задаются следующими формулами:

$$\hat{f}(\omega) = \int_{\mathbb{R}} f(t) e^{-2\pi i(\omega t)} dt, \quad (1)$$

$$\hat{F}(u, v) = \int_{\mathbb{R}^2} f(x, y) e^{-2\pi i(xu + yv)} du dv. \quad (2)$$

Также введем сокращенное обозначение для преобразования Радона, считая θ фиксированным параметром, а s – переменной:

$$p_{\theta}(s) := [\mathcal{R}f](\theta, s).$$

Теор. 1. *О центральном сечении. Преобразование Фурье от $p_{\theta}(s)$ совпадает со значениями двумерного преобразования Фурье от $f(x, y)$ на некоторой прямой:*

$$\hat{p}_{\theta}(\omega) = F(\omega \cos \theta, \omega \sin \theta)$$

□. Преобразуем определение преобразований Фурье и Радона, используя основное свойство дельта-функции (а также $\delta(t) = \delta(-t)$):

$$\begin{aligned}\hat{p}_\theta(\omega) &= \int_{\mathbb{R}} p_\theta(s) e^{-2\pi i \omega s} ds = \\ &= \int_{\mathbb{R}^3} f(x, y) \cdot \delta(x \cos \theta + y \sin \theta - s) e^{-2\pi i \omega s} dx dy ds = \int_{\mathbb{R}^2} f(x, y) e^{-2\pi i \omega (x \cos \theta + y \sin \theta)} dx dy; \\ F(u = \omega \cos \theta, v = \omega \sin \theta) &= \int_{\mathbb{R}^2} f(x, y) e^{-2\pi i (xu + yv)} \Big|_{\substack{u=\omega \cos \theta \\ v=\omega \sin \theta}} = \hat{p}_\theta(s).\end{aligned}$$

■

Таким образом, прямая, вырезанная из двумерного Фурье-образа исходной функции, проходящая через начало координат, фактически описывает интегралы этой функции вдоль всех прямых, параллельных вырезанной. Получить их можно при помощи обратного к (1) преобразования.

13. Алгоритм обратного проецирования (ВР).

Рассмотрим задачу восстановления исходной функции по ее Радон-образу. Если рассматривать модель непрерывного мира, то $f(x, y)$ может быть восстановлена при помощи **обратного проецирования** (**Back-projection**):

$$[\mathcal{B}(p_\theta(s))](x, y) = \int_0^\pi p_\theta(x \cos \theta + y \sin \theta) d\theta. \quad (3)$$

Выражение (3) не совпадает с $f(x, y)$, однако часто используется как его приближение.

$$f(x, y) = \mathcal{F}_2^{-1}[F(u, v)](x, y) = \int_{\mathbb{R}^2} F(u, v) e^{2\pi i (xu + yv)} du dv = \dots$$

Перейдем в последнем равенстве к полярным координатам по переменным интегрирования: $u = \omega \cos \theta, v = \omega \sin \theta, du dv = |\omega| d\omega d\theta$:

$$\dots = \int_0^\pi \int_{-\infty}^\infty F(\omega \cos \theta, \omega \sin \theta) e^{2\pi i (x \cos \theta + y \sin \theta) \omega} |\omega| d\omega d\theta = \dots$$

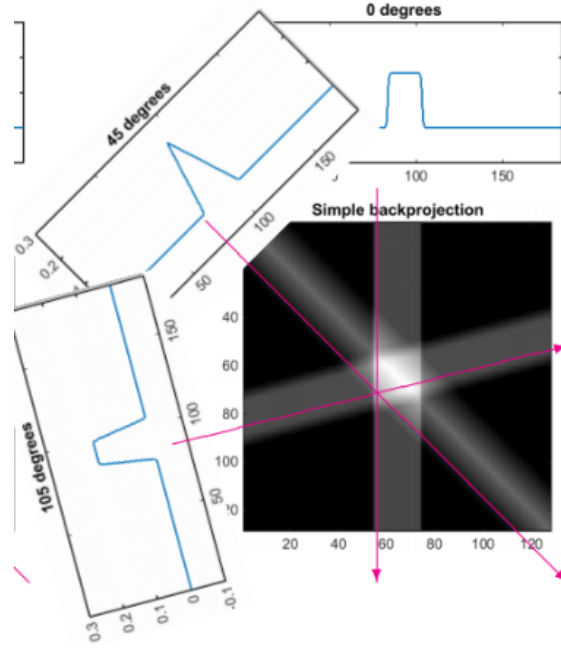


Рис. 1: Суть backprojection в одной картинке

Используем теорему 1 для замены $F(\omega \cos \theta, \omega \sin \theta) = \hat{p}_\theta(\omega)$. Также введем для краткости обозначение $s = x \cos \theta + y \sin \theta$.

$$\int_0^\pi \int_{\mathbb{R}} |\omega| \hat{p}_\theta(\omega) e^{2\pi i s \omega} d\omega d\theta = \int_0^\pi [\mathcal{F}^{-1}(|\omega| \hat{p}_\theta(\omega))](s) d\theta. \quad (4)$$

Если удалить из последнего выражения якобиан $|\omega|$, оставив под интегралом вместо выражения $\mathcal{F}^{-1}[\hat{p}_\theta(\omega)](s) = p_\theta(s) = p_\theta(x \cos \theta + y \sin \theta)$ то получится в точности выражение (3).

14. Алгоритм FBR.

Как было показано ранее, алгоритм обратного проецирования легко описать и реализовать, но он не является вполне точным с математической точки зрения, поскольку опускает якобиан $|\omega|$. В результате этого, как правило, восстановленное изображение получается размытым, значения, близкие к началу координат, завышены, а далекие от начала координат, наоборот, занижены. Причина этого в том, что после дискретизации с равномерной сеткой по s и

θ через пиксели, близкие к началу координат, проходит большее количество дискретных прямых.

Алгоритм **filtered backprojection** состоит в использовании вместо (3) более правильной формулы, выведенной в (4):

$$[\mathcal{B}_f(p_\theta(s))](x, y) = \mathcal{F}^{-1} [|\omega| \hat{p}_\theta(\omega)] (x \cos \theta + y \sin \theta) d\theta. \quad (5)$$

С точки зрения реализации следует добавить в алгоритм обратного проецирования дополнительный шаг: для каждого угла θ_i заменить «сырые» значения синограммы $p_{\theta_i}(s)$ на $\mathcal{F}^{-1} [|\omega| \hat{p}_{\theta_i}(\omega)](s)$. Множитель $|\omega|$ называется **Ramp filter**.

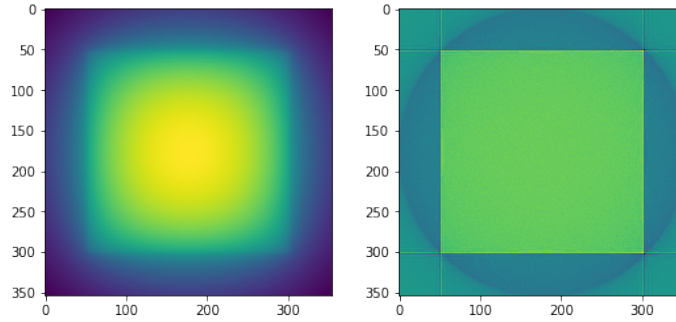


Рис. 2: Восстановление белого квадрата из синограммы при помощи backprojection (слева) и filtered backprojection (справа).

15. Способ использования БПХ для определения наклона шрифта.
16. Способ использования БПХ для слепой компенсации радиальной дисторсии.
17. Способ использования БПХ для определения степени сбития камеры. Эпиполярная геометрия.
18. Быстрое вычисление суммы по любому отрезку и четырехвершиннику на изображении с помощью БПХ.
19. Сочетание БПХ и принципа четырех русских для случаев прямых в трехмерном пространстве.
20. Быстрая линейная бинарная кластеризация с помощью БПХ.
21. Робастное решение задачи линейной регрессии путем вычисления М-оценок с помощью БПХ.