Линейни филтри за обработка на полутонови изображения. Представяне в обектното и в честотното пространство. Диференциални филтри. Нелинейни филтри

Иван Иванов, Изкуствен Интелект, 23610

I. Линейни филтри за обработка на полутонови изображения

Едно полутоново изображение може да бъде разгледано като функция:

$$I : Row \times Column \rightarrow Intensity$$

която съпоставя на всеки пиксел (определен от неговия ред и колона) определен интензитет.

Филтър в контекста на полутоновите изображения представлява функция:

$$\Phi : I \rightarrow I$$

която трансформира входно изображение в изходно изображение. Процесът на прилагането на филтър ϕ върху входно изображение се нарича филтриране.

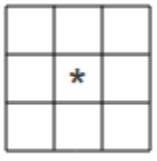
Един филтър се нарича линеен тогава и само тогава когато:

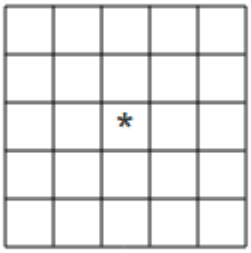
$$\Phi(\alpha f + \beta g) = \alpha\Phi(f) + \beta\Phi(g) \quad (1)$$

където f и g са изображения, операцията от вида αf е скалиране на пикселите на изображение с коефициент α, а операцията + е почленно (пиксел по пиксел) събиране на интензитетите на две изображения.

II. Конволюция и корелация

Разпространен клас филтри са тези, които представят всеки пиксел от изходното изображение (изображението на изхода на функцията) като линейна комбинация от пикселите на входното изображение. Вместо да бъдат изчислявани, обаче, линейни комбинации от всички пиксели на входното изображение, често се използва само локална околност около пиксела чиито интензитет се изчислява. Практически обикновено се избира правоъгълна околност с нечетен брой редове и колони. Това позволява лесно да бъде определен нейния център. Често използвани са околности с размер *3*3* или *5*5*.





Тъй като се изчисляват линейни комбинации от интензитетите на пикселите, попадащи в околността на даден пиксел, можем да дефинираме матрица *h* от коефициенти (тегла) от интересуващата ни околност. Тази матрица *h* еднозначно определя операцията:

$$g(x,y) = \sum_{i=x-n}^{i=x+n} \sum_{j=y-m}^{j=y+m} f(i,j)h(x-i,y-j) \quad (2)$$

Формула *(2)* определя как пиксел *(x, y)* на изходното изображение *g* се представя като линейна комбинация от пиксели от правоъгълна околност с размер *(2n + 1) * (2m + 1)* с коефициенти (тегла) зададени от матрицата *h*. Централният пиксел на *h* се приема, че е с координати *(0, 0)*. Тази операция се нарича дискретна конволюция на изображението *f* с матрицата *h*. За всяка матрица *h* формула *(2)* може да се разглежда като филтър (функция трансформираща входно изображение в изходно изображение). Такъв филтър се нарича конволуционен филтър.

Корелацията е операция дефинирана с формулата:

$$g(x,y) = \sum_{i=x-n}^{i=x+n} \sum_{j=y-m}^{j=y+m} f(i,j)h(i-x,j-y) \quad (3)$$

Тя е сходна на конволюция, но приложена с матрица *h*, която е обърната хоризонтално и вертикално.

Операциите конволюция и корелация са еквивалентни ако матрицата *h* е симетрична *(h(x, y) = h(-x, -y))*.

III. Филтри за изглаждане на изображения

Изглаждането (smoothing) цели да намали шума и други малки флуктуации в изображението (замъгляват се и дребните детайли). Филтрите, които които ще разгледаме, работят в обектното пространство (spatial domain), но ефекта им е сходен с премахването на високо честотните съставки в честотното пространство (*low-pass* филтър в честотното пространство). Характерно за тях е, че елементите на матрицата *h* (маската) са не-негативни и сумата им е единица. Често се използва конволюционен филтър с матрица на коефициентите *h*:

$$h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4)$$

При тази операция всеки пиксел от изображението на изхода има стойност равна на средното аритметично от неговия и тези на 8-те околни пиксела интензитети във входното изображение. Вместо всеки пиксел от околността да има еднакъв принос при усредняването, може пикселите разположени по-близо до централния да имат по-голямо относително влияние върху крайния резултат от по-периферните пиксели. За целта коефициентите в матрицата *h* се подбират като апроксимации на двумерно Гаусово разпределение. Възможни, например, са следните матрици:

$$h = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (5)$$

$$h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (6)$$

Разбира се могат да се използват и значително по-големи от *3 * 3* матрици.

IV. Диференциални филтри

Диференциалните филтри целят да открият и подчертаят областите в изображението където настъпват резки промени в интензитета (контури). За целта те намират локални производни в изображението, като търсят местата, където тези производни имат най-големи абсолютни стойности. Прилагането на диференциалните филтри в обектната област има сходен ефект с премахването на ниските честоти в честотното пространство (прилагането на *high-pass* филтър). Това ги прави много чувствителни към шума в изображението и за това прилагането им често се предхожда от операция на изглаждане.

За намирането на контурите в изображение е полезно да разполагаме с градиента на интензитета във всяка негова точка. Този градиент е вектор, съставен от двете частни производни, отразяващи промяната в интензитета в хоризонтална и вертикална посока.

$$\nabla g = \left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right) \quad (7)$$

Посоката на градиента задава посоката на най-голям растеж на функцията *g,* а неговата големина задава скоростта на растеж в тази посока. Градиента за всеки пиксел на изображението може да бъде изчислен чрез прилагането на филтрите на *Sobel*. Частните производни по *x* могат да бъдат апроксимирани чрез конволюция с матрицата h_x:

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (8)$$

Частните производни по y могат да бъдат апроксимирани чрез конволюция с матрицата h_y:

$$h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (9)$$

Нека:

$$G_x = h_x * g \quad (10)$$

$$G_y = h_y * g \quad (11)$$

Тогава посоката на градиента в пиксел *(x, y)* се задава от формулата:

$$\Theta = atan2(G_y, G_x) \quad (12)$$

А големината от формулата:

$$G = \sqrt{(G_x^2 + G_y^2)} \quad (13)$$

Така дефинираният филтър на *Sobel*, обаче, вече не е линеен.

Алтернативен подход е използването на филтъра на *Laplace*. Той е линеен, апроксимира големината на градиента, но не дава информация относно неговата посока. Една негова разновидност представлява конволюция с матрицата *h*:

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (14)$$

V. Представяне на изображенията в честотното пространство – теорема за конволюцията

От изчислителна гледна точка операцията конволюцията става значително по-скъпа с увеличаването на матрицата *h*. Ако входното изображение има n пиксела, то цената за извършването на конволюцията варира от *O(n)* за малка матрица *h*, до $O(n^2)$ когато броят на елементите в матрицата *h* доближава *n*. Оказва се, обаче, че алгоритмичната сложност при големи матрици *h* може да бъде редуцирана до *O(n*log(n))*. Това е възможно поради теоремата за конволюцията, която ни позволява да редуцираме операцията конволюция в обектното пространство до операцията поточково умножение в честотното пространство.

Теорема за конволюцията: Нека *f* и *g* са функции и нека тяхната конволюция да бъде означена като *f * g*. Нека *F* да бъде оператор за трансформация на *Fourier* и нека *F(f)* и *F(g)* да бъдат трансформациите на *Fourier* за *f* и *g*. Тогава:

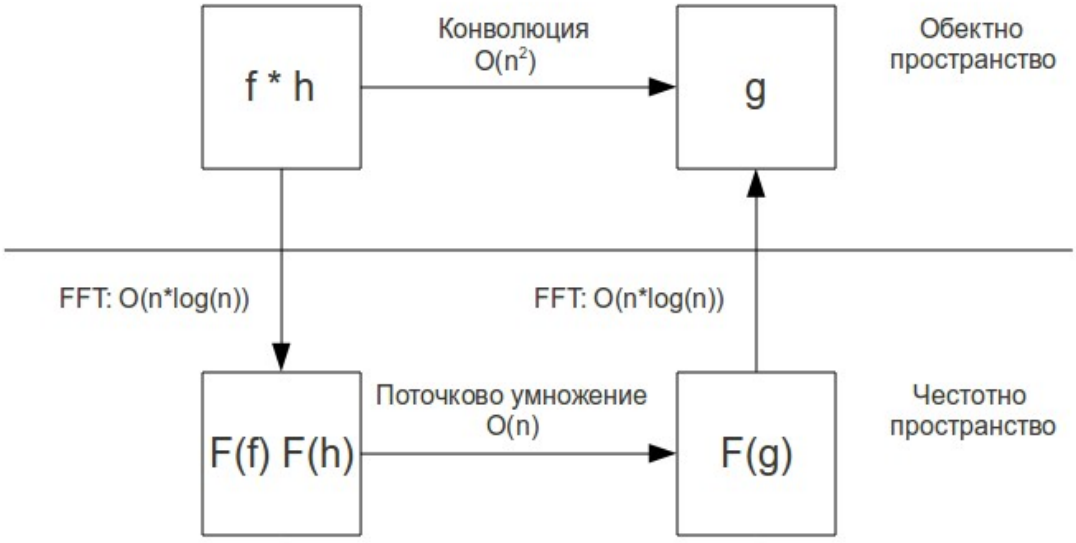
$$F(f * g) = F(f) \cdot F(g) \quad (15)$$

като приложим обратната трансформация на *Fourier* към двете страни на уравнението получаваме:

$$f * g = F^{-1}(F(f) \cdot F(g)) \quad (16)$$

Трансформацията на *Fourier* и обратната трансформация на *Fourier* могат да бъдат реализирани

в *O(n*log(n))* чрез използването на *FFT*. Операцията поточково умножение и се реализира в *O(n)*.



VI. Нелинейни филтри

Всеки филтър, който не отговаря на условие *(1)* е нелинеен. Вече разгледаме пример за такъв филтърът на *Sobel*. Друг често използван нелинеен филтър е медианния филтър. Тоя избира медианата на стойностите от околността вместо да взима линейна комбинация от тях. Той съчетава добро редуциране на шума със запазване на контурите.