

Filtracja bilateralna

Tomasz Pięciak
AGH Akademia Górniczo-Hutnicza w Krakowie
<http://home.agh.edu.pl/pieciak/>

1 Splot obrazu i współczynników filtra

Splot obrazu wejściowego I ze współczynnikami filtra ψ dla ustalonego punktu obrazu \mathbf{x} można przedstawić następująco:

$$\hat{I}(\mathbf{x}) = \frac{1}{W_N} \sum_{\mathbf{p} \in \eta(\mathbf{x})} \psi(\|\mathbf{p} - \mathbf{x}\|) I(\mathbf{p}), \quad (1)$$

gdzie \hat{I} jest obrazem wynikowym (przefiltrowanym), $W_N = \sum_y \psi(y)$ jest parametrem normalizującym współczynniki filtra ψ , $\|\cdot\|$ jest odległością między punktami obrazu \mathbf{x} i \mathbf{p} według ustalonej metryki (np. norma ℓ^2), natomiast $\eta(\mathbf{x})$ – otoczeniem punktu \mathbf{x} . Funkcję ψ we wzorze (1) można dobrać na szereg sposobów, m.in.

- $\psi(y) = \frac{1}{y}$,
- $\psi(y) = G_{\sigma_s}(y)$.

gdzie $G_{\sigma_s}(y)$ jest funkcją Gaussa z parametrem skali σ_s .

2 Filtracja bilateralna

Zasadniczą wadą klasycznego splotu jest brak adaptacji współczynników filtra do lokalnego otoczenia $\eta(\mathbf{x})$ filtrowanego punktu \mathbf{x} . Oznacza to wykorzystanie tych samych współczynników filtra ψ niezależnie od filtracji obszaru jednorodnego, czy zawierającego krawędzie obiektów. Filtracja bilateralna, zaproponowana przez Tomasi & Manduchi [1], a następnie rozwinięta przez Elada [2] uwzględnia lokalne otoczenie filtrowanego punktu za pomocą zmieniających się parametrów filtra (Rys. 1).



Rysunek 1: (a) Rzeczywista krawędź obrazu, (b) współczynniki filtra ψ (brak adaptacji), (c) odległość γ w przeciwdziedzinie obrazu oraz (d) współczynniki filtra przy uwzględnieniu jednocześnie funkcji ψ i γ

1

Współczynniki filtra obliczane są na podstawie odległości filtrowanego punktu \mathbf{x} od każdego punktu otoczenia \mathbf{p} w dziedzinie przestrzennej obrazu oraz odległości punktów w przeciwdziedzinie obrazu (np. różnica w poziomie jasności punktów dla obrazu monochromatycznego):

$$\hat{I}(\mathbf{x}) = \frac{1}{W_N} \sum_{\mathbf{p} \in \eta(\mathbf{x})} \underbrace{\psi(\|\mathbf{p} - \mathbf{x}\|)}_{\psi} \underbrace{\gamma(|I(\mathbf{p}) - I(\mathbf{x})|)}_{\gamma} I(\mathbf{p}), \quad (2)$$

gdzie W_N jest współczynnikiem normalizującym filtr, γ odległością w przeciwdziedzinie obrazu, np. $\gamma(y) = \exp\left(-\frac{y^2}{2\sigma_s^2}\right)$, gdzie parameter σ_s jest utożsamiany z poziomem szumu w obrazie i należy go dobrać w sposób empiryczny¹.

3 Ćwiczenie laboratoryjne

3.1 Cel ćwiczenia

Celem ćwiczenia jest własna implementacja filtracji bilateralnej. W tym celu użyta zostanie wbudowana funkcja środowiska MathWorks MATLAB `colfilt`, która umożliwia obsługę dowolnego przekształcenia obrazu w przesuwym bloku oraz ustalonym oknie `local_window`².

3.2 Ćwiczenie

1. Ściągnij archiwum `MR_data.mat` i wczytaj go poleceniem `load`.
2. Zapoznaj się z dokumentacją funkcji `colfilt`, rozdziałem w dokumentacji środowiska MATLAB *Using Columnwise Processing to Speed Up Sliding Neighborhood or Distinct Block Operations* oraz *function_handle* (@).
3. W pierwszej kolejności zaimplementuj jednak klasyczny splot z wykorzystaniem funkcji `colfilt`. W tym celu zdefiniuj dwie funkcje: `convolution` oraz `convolution_local`. Funkcja `convolution` powinna przyjmować jako argumenty obraz wejściowy `data` oraz rozmiar okna `local_window` w którym nastąpi filtracja. W tej funkcji należy zadeklarować uchwyt do funkcji `convolution_local` i przekazać go do funkcji `colfilt`:

```
% uchwyt do funkcji 'convolution_local'
FUNCTION = @(data_, local_window_) convolution_local(data_, local_window_);

% przekazanie uchwytu FUNCTION do mechanizmu 'colfilt'
data_filtered = colfilt(data, local_window, 'sliding', FUNCTION, local_window);
```

¹W rzeczywistości poziom szumu jest estymowany w zależności od typu szumu (addytywny, multiplikatywny, zależny od sygnału). Estymacja parametrów szumu znacznie wykracza poza zakres tego laboratorium. Więcej na ten temat można poczytać w pracach:

- Pieciak T., *The Maximum Spacing Noise Estimation in Single-coil Background MRI Data*, In: 21st IEEE International Conference on Image Processing (ICIP), Paris, France, 2014. p. 1743–1747
- Aja-Fernández, S., Pieciak, T., Vegas-Sánchez-Ferrero, G., *Spatially variant noise estimation in MRI: A homomorphic approach*, Medical Image Analysis, 20.1 (2015): 184–197

²Sugerowane jest okno o nieparzystych rozmiarach, np. [9, 9]

2

Funkcja `convolution_local` będzie wywoływana dla każdego bloku obrazu przez wbudowany mechanizm `colfilt`. Pozwala to na definicję dowolnego przekształcenia blokowego – operacja wykonywana w przesuwym bloku obrazu powinna zostać zapisana w funkcji `convolution_local`.

4. W rzeczywistości, w celu przyspieszenia obliczeń, funkcja `colfilt` wywołuje przekazaną przez uchwyt funkcję dla wielu bloków jednocześnie. Wszystkie bloki zostają (automatycznie) przekształcone na wektory kolumnowe i połączone ze sobą. Dzięki temu wewnątrz funkcji `convolution_local` mamy do czynienia z tablicą o rozmiarze $(\text{local_window}(1) \cdot \text{local_window}(2)) \times N_x$ gdzie N_x jest liczbą przekształconych bloków.

Warto zwrócić uwagę na pewien szczegół wywołania mechanizmu `colfilt`: funkcja `colfilt` przyjmuje jako parametry obraz `data`, wielkość okna `local_window` oraz ponownie wielkość okna `local_window`. Ostatni parametr niezbędny jest to prawidłowego (ręcznego) przekształcenia każdego wektora `data(:, i)` w blok o rozmiarze `local_window`.

5. Szkielet funkcji `convolution_local` powinien wyglądać następująco:

```
for i=1:Nx
% lokalny blok
patch = reshape(data(:, i), local_window);

% przemnożenie bloku 'patch' przez współczynniki filtra 'h'
data_filtered(i) = sum(sum(patch .* h))
end
```

Funkcja `convolution_local` powinna zwrócić wektor poziomy o rozmiarze $1 \times N_x$. Element i -ty zwracanego wektora `data_filtered` przechowuje wynik operacji na i -tym bloku (wektorze kolumnowym).

6. Korzystając z przedstawionego postępowania zaimplementuj klasyczny splot oraz filtrację bilateralną za pomocą następujących funkcji:

- `convolution`,
- `convolution_local`,
- `bilateral`,
- `bilateral_local`.

4 Uwagi techniczne do implementacji

- Podczas implementacji klasycznego splotu oblicznie współczynników filtra h uwzględnij poza pętlą `for i=1:Nx ... end`. Współczynniki filtra możesz wygenerować funkcją `fspecial('gaussian', local_window, 25)`.
- W filtrze bilateralnym niezbędne jest wyznaczenie w wektorze `data(:, i)` pozycji N_c na której znajduje się centralny punkt okna `local_window`.
- W filtrze bilateralnym współczynniki filtra należy obliczać wewnątrz pętli `for i=1:Nx ... end`.

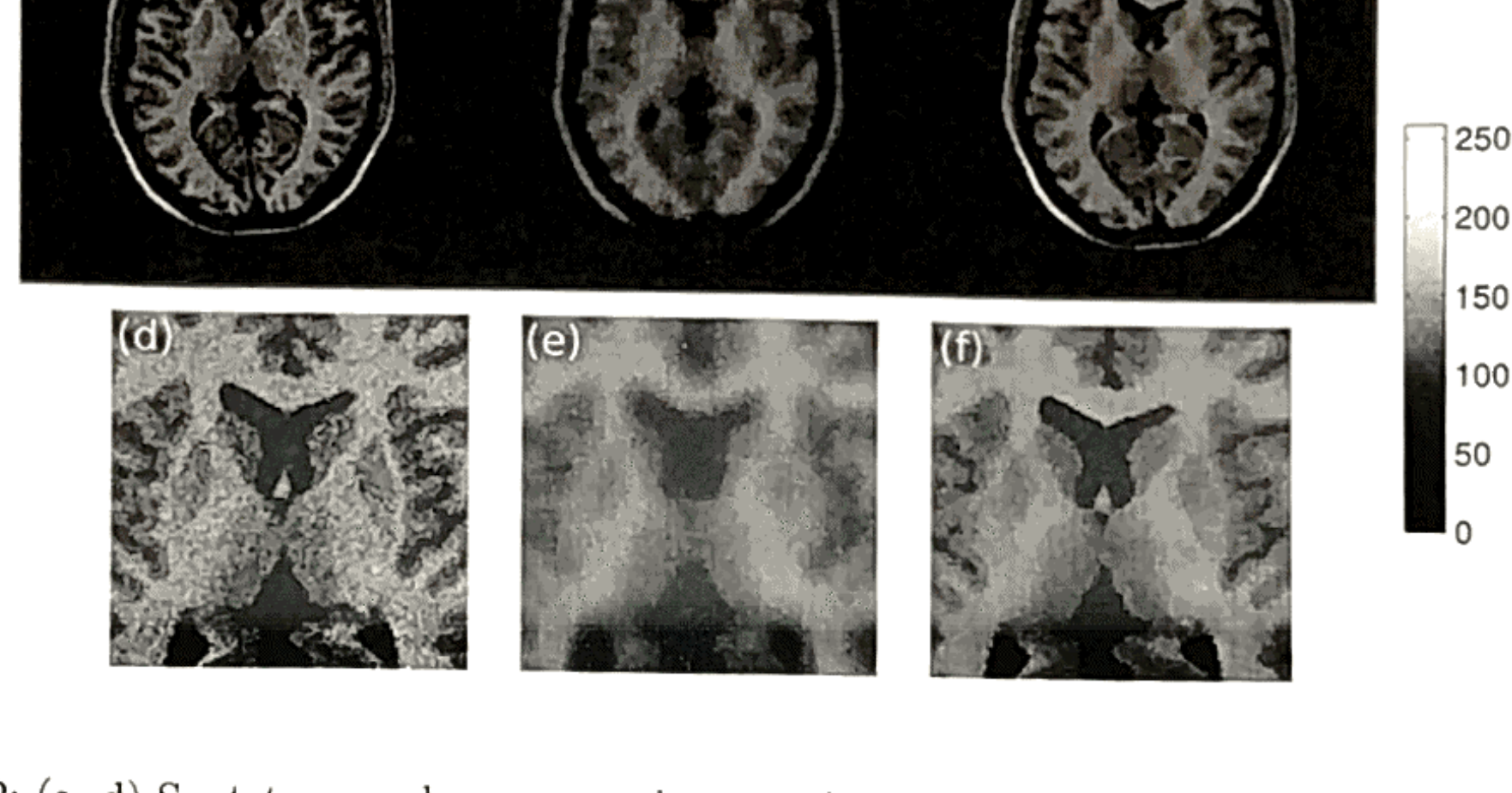
3

- W funkcjach `bilateral` i `bilateral_local` należy dodatkowo uwzględnić parametr σ_r .
- Obliczenie współczynników odległości w dziedzinie przestrzennej obrazu można zrealizować następująco:

```
[rows, cols] = find(patch == patch);
rows = reshape(rows, local_window);
cols = reshape(cols, local_window);
Ncy = ceil(local_window(1)/2);
Ncx = ceil(local_window(2)/2);
dist = sqrt((Ncy - rows).^2 + (Ncx - cols).^2);
```

5 Testy algorytmu

Dokonaj testów splotu i filtra bilateralnego na obrazach dostarczonych w archiwum `MR_data.mat`. Dobierz parametr σ_r dla każdego obrazu w sposób empiryczny, a następnie wyświetl obok siebie obraz zaszumiony oraz przefiltrowane za pomocą splotu i filtra bilateralnego. Czy dobierając stałą wartość parametru σ_r można przefiltrować każdy obraz?



Rysunek 2: (a, d) Syntetyczny obraz zaszumiony stacjonarnym szumem Rice'a, (b, e) wynik filtracji splotowej oraz (c, f) wynik filtracji bilateralnej

Literatura

- [1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 839–846.
- [2] M. Elad, "On the origin of the bilateral filter and ways to improve it," *Image Processing, IEEE Transactions on*, vol. 11, no. 10, pp. 1141–1151, 2002.

4