Informatyka, studia dzienne, mgr II st. semestr I

Przetwarzanie obrazu i dźwięku 2011/2012

Prowadzący: dr inż. Arkadiusz Tomczyk środa, 8:15

Data oddania: ________________ Ocena: ________________

Paweł Musiał 178726

Łukasz Michalski 178724

Zadanie 2:
Filtracja w dziedzinie częstotliwości i segmentacja obrazu.*

Spis treści

1. Cel . . . 2

2. Wprowadzenie . . . 2

2.1. Filtracja w dziedzinie częstotliwości . . . 3

2.1.1. Algorytm szybkiej transformaty Fouriera . . . 3

2.1.2. Filtr dolnoprzepustowy . . . 4

2.1.3. Filtr górnoprzepustowy . . . 5

2.1.4. Filtr pasmowoprzepustowy . . . 5

2.1.5. Filtr pasmowozaporowy . . . 6

2.1.6. Filtr z detekcją krawędzi . . . 6

2.1.7. Filtr modyfikujący widmo . . . 7

2.2. Segmentacja . . . 7

3. Opis implementacji . . . 10

4. Materiały i metody . . . 10

5. Wyniki . . . 10

5.1. Filtracja w dziedzinie częstotliwości . . . 10

5.1.1. Filtr dolnoprzepustowy . . . 10

5.1.2. Filtr górnoprzepustowy . . . 11

5.1.3. Filtr pasmowoprzepustowy . . . 11

5.1.4. Filtr pasmowozaporowy . . . 12

5.1.5. Filtr z detekcją krawędzi . . . 12

5.1.6. Filtr modyfikujący widmo . . . 13

5.2. Segmentacja . . . 13

6. Dyskusja . . . 13

6.1. Filtracja w dziedzinie częstotliwości . . . 13

6.1.1. Filtr dolnoprzepustowy . . . 13

6.1.2. Filtr górnoprzepustowy . . . 13

6.1.3. Filtr pasmowoprzepustowy . . . 13

6.1.4. Filtr pasmowozaporowy . . . 13

6.1.5. Filtr z detekcją krawędzi . . . 14

6.1.6. Filtr modyfikujący widmo . . . 14

* SVN: `http://serce.ics.p.lodz.pl/svn/labs/poid/at_sr0800/lmpm`

6.2. Segmentacja 14
7. Wnioski 14
7.1. Filtracja w dziedzinie częstotliwości 14
7.2. Segmentacja 14
Literatura 14

1. Cel

Celem zadania było stworzenie aplikacji realizującej poniższe element :

- Zaimplementować proste i odwrotne szybkie przekształcenie Fouriera z decymacją w dziedzinie częstotliwości, a następnie zastosować je do obrazów. Powinna istnieć możliwość obejrzenia zarówno widma mocy, jak i widma fazy.
- Zaimplementować następujące metody filtracji:
  — (F1) Filtr dolnoprzepustowy (górnozaporowy).
  — (F2) Filtr górnoprzepustowy (dolnozaporowy).
  — (F3) Filtr pasmowoprzepustowy.
  — (F4) Filtr pasmozaporowy.
  — (F5) Filtr z detekcją krawędzi.
- Zaimplementować filtr modyfikujący fazę widma transformaty Fouriera. Modyfikacja ta polega, dla obrazu o wymiarach NxM , na pomnożeniu każdego elementu widma przez:

$$P(n,m) = \exp\left(j \cdot \left(\frac{-nk2\pi}{N} + \frac{-ml2\pi}{M}\left(k+l\right)\pi\right)\right)$$

- Zaimplementować metodę segmentacji - metoda podziału obszarów [2] w celu znalezienia spójnych obszarów o jednolitej barwie. Jako wynik należy wygenerować obrazy reprezentujące znaleziony obszary o jednolitej barwie (tyle masek ile znalezionych obszarów), na których kolorem białym oznaczone są te obszary, a czarnym pozostała część obrazu. Powinna istnieć możliwość nałożenia wybranych masek na obraz przy czym sposób wizualizacji tego nałożenia wybrany powinien zostać przez twórców aplikacji.

W sprawozdaniu zamieszczono wyniki działania poszczególnych algorytmów oraz porównanie ich pracy w różnych wariantach ustawień i dla różnych problemów. Badania przeprowadzono zarówno na obrazach kolorowych (24-bitowych), jak i w odcieniach szarości (8-bitowych), a część badań również na obrazach czarno-białych (1-bitowych).

2. Wprowadzenie

Obrazy przechowywane są w pamięci komputera w postaci bitowej, w której określa się ilość bitów przypadającą na każdy piksel obrazu. Kolorowe zdjęcia zapisywane są w formacie 24-bitowym co odpowiada 8-bitom na każdy z trzech składowych kanałów formatu RGB. Fotografie w odcieniach szarości wykorzystują tylko jeden kanał dlatego wystarczy tutaj przechowywać 8-bitów na każdy zapisany piksel. Taki zapis umożliwia uzyskanie 256 różnych wartość danego piksela czyli 256 różnych odcieni szarości poczynając od białego na kolorze czarnym kończąc. W przypadku obrazów kolorowych liczba kombinacji jest dużo większa i wynosi 16,777,216, co nie oznacza, że wszystkie uzyskane w ten sposób kolory są między sobą rozróżnialne. W tej pracy wszystkie przygotowane algorytmy operują na poszczególnych wartościach każdego kanału obrazu.

[2] ang. region splitting and merging

2.1. Filtracja w dziedzinie częstotliwości

Filtracja w dziedzinie częstotliwości analogicznie jak w przypadku analizowanego w poprzednim zadaniu, polega na przemnożeniu danych przez jakiś filtr $H(k)$. W większości przypadków udało się uzyskać filtrację idealną jak i filtrację z przejściem ciągłym. Co w pewnych przypadkach dawało bardziej zadowalające efekty.
Opisując metody filtracji będziemy posługiwać się poniższymi oznaczeniami :

— $D(u, v)$ odległość częstotliwości (u, v) od częstotliwości maksymalnej
— D_0 częstotliwość progowa
— D_L D_H - dolna, górna częstotliwość progowa
— n - parametr filtru Butterwortha.

Działanie filtrów demonstrowane będzie na przykładzie obrazu *lena.bmp* [3].

2.1.1. Algorytm szybkiej transformaty Fouriera

Postać ogólna dyskretnej transformaty Fouriera 1 oraz jej odwrotność 2

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \tag{1}$$

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(n) W_N^{-kn} \tag{2}$$

gdzie $W_N = e^{j2\pi/N}$.

Złożoność obliczeniowa tego przekształcenia w postaci powyżej przedstawionej jest wyższa od liniowej a zasadniczo kwadratowej (N^2), co nie jest zadowalającym wynikiem, gdybyśmy chcieli zastosować to przekształcenie na bardzo dużej ilości próbek zdyskretyzowanego sygnału. Jednakże dzięki właściwościom tego przekształcenia możliwa jest jego optymalizacja.

Poniżej przedstawiony algorytm szybkiej transformaty Fouriera(*FFT*) z decymacją w częstotliwości. W algorytmie tym rozdzielamy przekształcenie na dwa podrzędne operujące na połówkach danych podstawowych uzyskując 3. Następnie rozdzielamy operacje dla parzystych i nie parzystych próbek, uzyskując postacie 4, 5.

$$X(k) = \sum_{n=0}^{N/2-1} x(n) W_N^{kn} + \sum_{n=0}^{N/2-1} x(n) W_N^{kn} =$$
$$= \sum_{n=0}^{N/2-1} x(n) W_N^{kn} + W_N^{Nk/2} \sum_{n=0}^{N/2-1} x\left(n + \frac{N}{2}\right) W_N^{kn}$$
$$X(k) = \sum_{n=0}^{N/2-1} \left[x(n) + (-1)^k x(n + \frac{N}{2}) \right] W_N^{kn} \tag{3}$$

$$X(2k) = \sum_{n=0}^{N/2-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] \tag{4}$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} \left\{ \left[x(n) - x\left(n + \frac{N}{2}\right) \right] \right\} \tag{5}$$

gdzie użyto faku, iż $W_N^2 = W_{N/2}$.

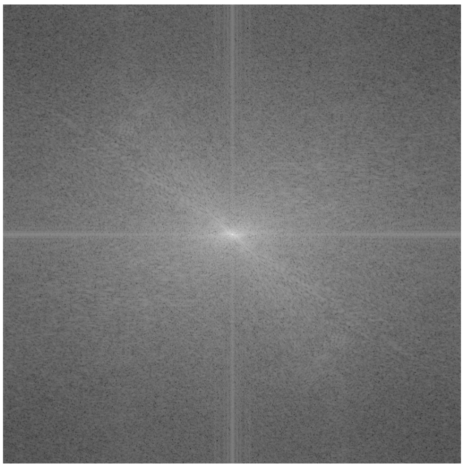
Algorytm ten opiera się na metodzie programowania „dziel i zwyciężaj” w której dzielimy zbiór danych w kolejnych krokach, uzyskując pod problemy łatwiejsze do rozwiązania a sumarycznie dające wynik przekształcenia na zbiorze

[3] `http://ics.p.lodz.pl/~tomczyk/available/po/images/lena.bmp`

sprzed podziału. Po zastosowaniu wyżej opisanej optymalizacji przekształcenia podstawowego 1, algorytm swoją złożoność $N \log_2 N$.Dokładniejszy opis algorytmu można znaleźć w [1].

Powyżej opisano jedynie algorytm prostego przekształcenia Fouriera, ponieważ algorytm dla przekształcenia odwrotnego będzie wyglądał analogicznie do opisanego tutaj, zmiany dotyczyć będą widocznych na pierwszy rzut oka różnic pomiędzy 1 i 2.

W naszym przypadku jednak mamy do czynienia z obrazem dwuwymiarowym, nie wektorem jednowymiarowym. Jednak dzięki własnościom transformaty Fouriera, możemy po prostu obliczyć transformate wektorów wierszowych a następnie kolumnowych, aby uzyskać pożądany wynik. Przekształcenie to transformuje dyskretne dane z przestrzeni $\mathcal{R} \rightarrow \mathcal{C}$, ważniejszym jednak będzie dla nas co przekształcone dane sobą prezentują, czyli przestrzeń częstotliwości danego sygnału wejściowego.



Rysunek 1: Widmo mocy obrazu *lena.bmp*.

2.1.2. Filtr dolnoprzepustowy

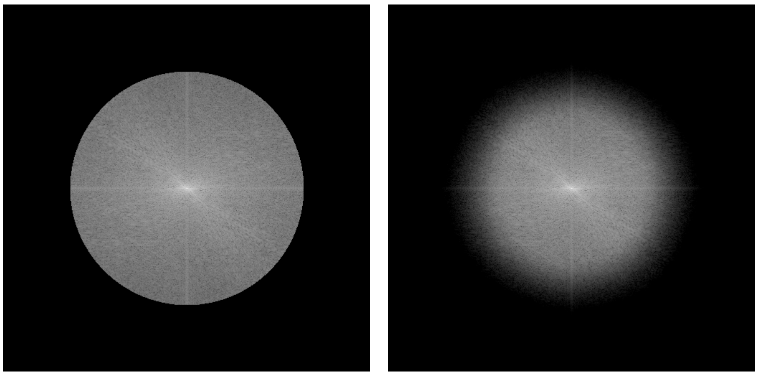
Filtr idealny.

$$H_{lp}(u,v) = \begin{cases} 1 & dla\ D(u,v) \leq D_0 \\ 0 & dla\ D(u,v) > D_0 \end{cases} \tag{6}$$

Filtr ciągły - Butterwortha [4].

$$H_{lp}(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}} \tag{7}$$

[4] Butterworth



(a) filtr idealny (b) filtr Butterwortha

Rysunek 2: Filtr dolnoprzepustowy, widmo mocy.

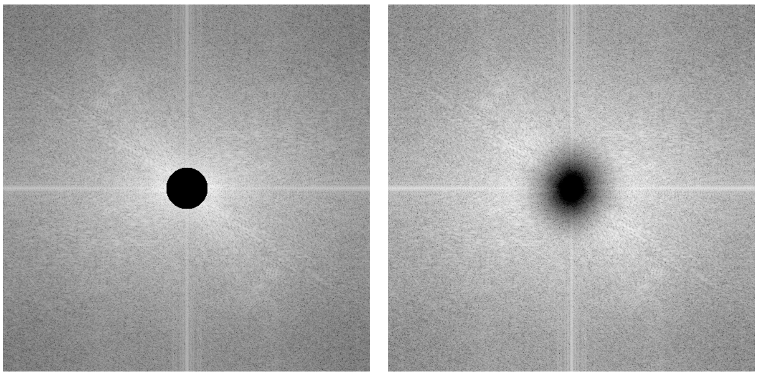
2.1.3. Filtr górnoprzepustowy

Filtr idealny.

$$H_{hp}(u,v) = \begin{cases} 1 & dla\ D(u,v) \geq D_0 \\ 0 & dla\ D(u,v) < D_0 \end{cases} \tag{8}$$

Filtr ciągły - Butterwortha.

$$H_{hp}(u,v) = 1 - H_{lp}(u,v) \tag{9}$$



(a) filtr idealny (b) filtr Butterwortha

Rysunek 3: Filtr dolnoprzepustowy, widmo mocy.

2.1.4. Filtr pasmowoprzepustowy

Filtr idealny.

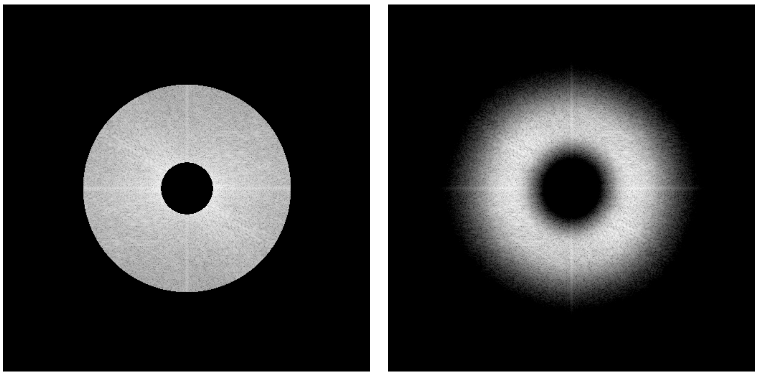
$$H_{hp}(u,v) = \begin{cases} 1 & dla\ D_L \leq D(u,v) \leq D_H \\ 0 & inny\ przypadek \end{cases} \tag{10}$$

Filtr ciągły - Butterwortha.

$$H_{lp}(u,v) = \frac{1}{1 + [D(u,v)/D_L]^{2n}} \tag{11}$$

$$H_{hp}(u,v) = 1 - \frac{1}{1 + [D(u,v)/D_H]^{2n}} \tag{12}$$

$$H_{bp}(u,v) = H_{lp}(u,v) \cdot H_{hp}(u,v) \tag{13}$$



(a) filtr idealny (b) filtr Butterwortha

Rysunek 4: Filtr dolnoprzepustowy, widmo mocy.

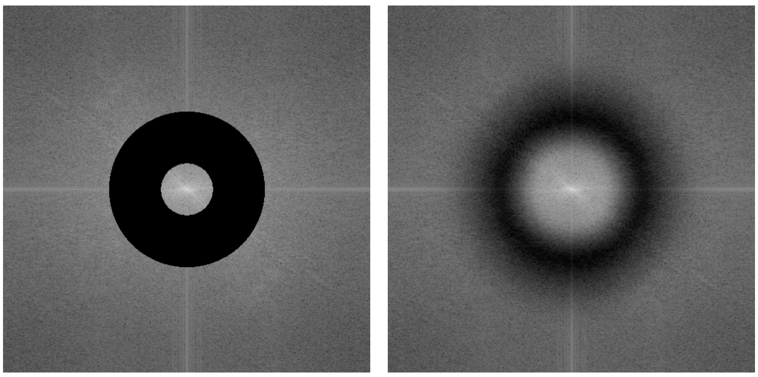
2.1.5. Filtr pasmowozaporowy

Filtr idealny.

$$H_{bs}(u,v) = \begin{cases} 0 & dla\ D_L \leq D(u,v) \leq D_H \\ 1 & inny\ przypadek \end{cases} \tag{14}$$

Filtr ciągły - Butterwortha.

$$H_{bs}(u,v) = 1 - H_{bp}(u,v) \tag{15}$$

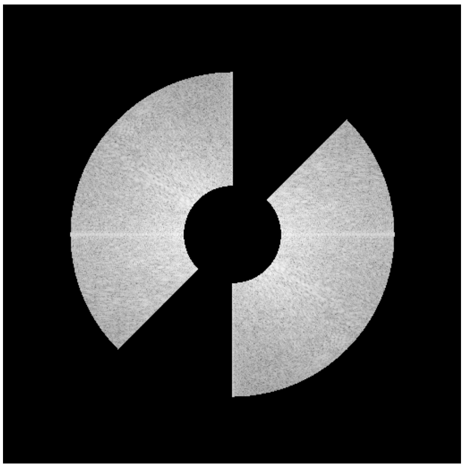


(a) filtr idealny (b) filtr Butterwortha

Rysunek 5: Filtr dolnoprzepustowy, widmo mocy.

2.1.6. Filtr z detekcją krawędzi

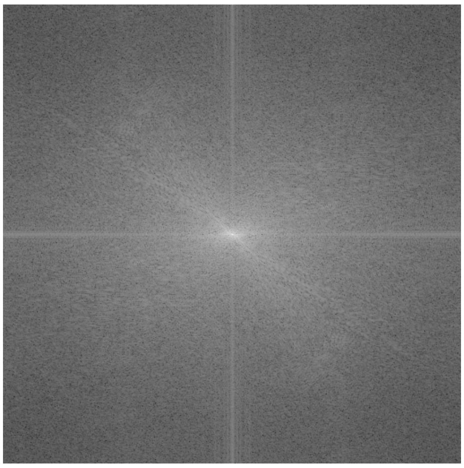
Filtracja ta jest jakby modyfikacją filtru pasmowoprzepustowego. Ponieważ z pasma wydzielonego przez filtr H_{bp} wycinamy symetrycznie względem maksymalnej częstotliwości łuki o pewnym kącie α. Dla jasności najlepiej zobrazować widmo mocy sygnału.



Rysunek 6: Filtr z detekcją krawędzi, widmo mocy.

2.1.7. Filtr modyfikujący widmo

$$H(u,v) = \exp\left(j \cdot \left(\frac{-uk2\pi}{N} + \frac{-vl2\pi}{M}\left(k+l\right)\pi\right)\right) \tag{16}$$



Rysunek 7: Filtr modyfikujący widmo, widmo mocy.

2.2. Segmentacja

Nasza grupa miała zadanie zrealizować segmentacje opartą o podział obrazu na rejony a następnie ich łączenie. Wykorzystaliśmy w tym celu drzewa czwórkowe do przechowywania danych. Sam algorytm segmentacji jest dość prosty i składa się z kilku kroków:

1. Należy sprawdzić czy podany obszar obrazu jest jednolity przy pomocy wcześniej zdefiniowanej fukcji porównującej wartości poszczególnych piksli.
2. Jeśli warunek z punktu pierwszego nie zachodzi należy ten obszar podzielić na cztery równe obszary i powtórzyć całą operację dla każdego z nich.
3. Jeżeli podany obszar jest jednolity jest on zapisywany do listy i jego przetwarzanie jest zakończone.
4. Dla tak uzyskanych obszarów wyliczany jest średni kolor, który reprezentuje każdy z nich.
5. Na podstawie średniego koloru następuje łączenie obszarów, które są podobne do siebie wykorzystując funkcję sprawdzającą spójność z punktu 1.

W punkcie pierwszym algorytmu wykorzystuje się do testowania spójności obszarów dowolnie zdefiniowaną funkcję porównującą wartości dwóch podanych pikseli. Dzięki dodaniu pewnej tolerancji tej różnicy można uzyskać obszary o podobnej charakterystyce. Sam algorytm rozpoczyna swoje działanie sprawdzając czy cały obszar obrazu jest spójny i kontynuując swoją pracę dzieli go na mniejsze części.

Po uzyskaniu już wszystkich jednolitych obszarów następuje ich łączenie w większe rejony zawierające w sobie obszary o podobnej charakterystyce. Dzięki temu zmniejsza się ilość poszczególnych pojedynczych komórek. Ten etap jest ważny z punktu widzenia tego algorytmu segmentacji. Jednak problem optymalnego znajdowania sąsiadów w drzewach czwórkowych nie jest trywialny. W tym celu wykorzystaliśmy algorytm QTLCLD [3] umożliwiający poprzez specjalną architekturę poszczególnych węzłów tego drzewa na znajdowanie sąsiadów w stałym czasie.

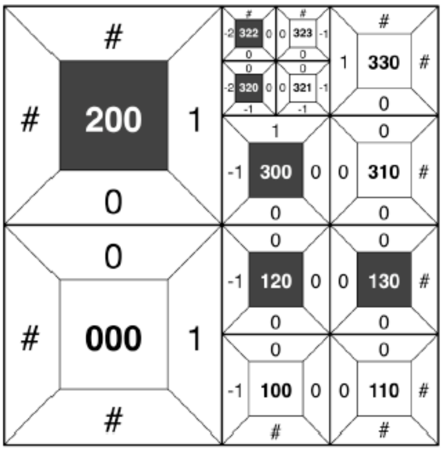
Algorytm QTLCLD opisany w [3] stanowi podobne podejście jak czwórkowe drzewa liniowe, dzięki temu czas znajdowania sąsiadów poszczególnych liści takiego drzewa jest znacznie krótszy niż w przypadku algorytmów przeglądając całą jego strukturę.

Główną zasadą jaką się tutaj stosuje jest przypisywanie odpowiednim komórką takiego drzewa specjalnych kodów dzięki którym można określić ich pozycję w całej strukturze. Zostało to przedstawione na rysunku 8.

Rysunek 8: Kody poszczególnych obszarów dla maksymalnego zagłębienia wynoszącego 3. Źródło [3].

Schemat przypisywania poszczególnych numerów z zakresu [0,3] został przedstawiony na pierwszej części rysunku 8 i jest on ustalony w trakcie podziału obszaru rodzica. Każde z dzieci dziedziczy po swoim rodzicu jego kod dodając na kolejnej pozycji określonej przez głębokość w drzewie na jakiej się znajduje swój własny indeks. Dzięki takie systemowi można łatwo zlokalizować położenie danego obszaru w całym drzewie oraz określić na jakiej głębokości się on znajduje.

Jednak by umożliwić sprawne i szybkie odnajdowanie sąsiadów danego obszaru należało dodać kolejny zestaw wartości tym razem określający różnice wielkości między danym obszarem a czterema jego sąsiadami. Wartości te są wyliczane w trakcie budowania drzewa, który to proces party jest o algorytm BFS. Na rysunku 9 przedstawiono powstałe w ten sposób drzewo reprezentujące cały obraz.



Rysunek 9: Kody poszczególnych obszarów wraz z różnicą wielkości do sąsiadów. Źródło [3].

Wykorzystując tak przygotowaną strukturę łatwo można uzyskać kod sąsiada danego obszaru w danym kierunku przy pomocy wzoru:

$$m_q = \begin{cases} ((n_q >> 2(r-l-dd)) << 2(r-l-dd)) \oplus_q (\Delta n_d << (2(r-l-dd))) & \text{dla} \quad dd < 0 \\ n_q \oplus_q (\Delta n_d << (2(r-l))) & \text{dla} \quad dd \geq 0 \end{cases} \tag{17}$$

gdzie:

— m_q - poszukiwany kod obszaru sąsiedniego
— n_q - kod obszaru bazowego
— r - maksymalna głębokość drzewa
— dd - różnica wielkości między obszarem bazowym a sąsiadem w zadanym kierunku
— l - poziom zagłębienia na jakim znajduje się bazowy obszar
— d - kierunek poszukiwań sąsiada z przedziału $[0, 3]$
— n_d - stała wartość dla odpowiedniego kierunku d i maksymalnego poziomu drzewa

Algorytm ten opiera się w dużej mierze na algorytmie Scharcka skąd wywodzi się użyty operator $\oplus_q$ zdefiniowany jak poniżej:

$$n_q \oplus_q k_q = (((n_q|t_y) + (k_q \wedge t_x)) \wedge t_x)|(((n_q|t_x) + (k_q \wedge t_y)) \wedge ty) \tag{18}$$

gdzie:

— $|$ - oznacza operację bitową OR
— $\wedge$ - oznacza operację bitową AND
— $+$ - jest normalny dodawaniem dwóch liczby

Dodatkowo we wzorze pojawiają się dwie stałe t_x i t_y,które są zdefiniowane następująco:

$$t_x = 01..0101 \quad \text{„01” powtórzone } r \text{ razy}$$
$$t_x = 10..1010 \quad \text{„10” powtórzone } r \text{ razy}$$

gdzie:

— r - maksymalna głębokość drzewa

Stałe n_d dla czterech kierunków poszukiwań użyte w równaniu 17 określone są następująco:

$$n_1 = 00..01 \quad \text{wschodni sąsiad}$$
$$n_1 = 00..10 \quad \text{północy sąsiad}$$
$$n_2 = t_y \quad \text{zachodni sąsiad}$$
$$n_3 = t_x \quad \text{południowy sąsiad}$$

Stałe n_1 i n_1 są uzupełniane zerami tak aby łączna ich długość bitów wynosiła $2r$.

3. Opis implementacji

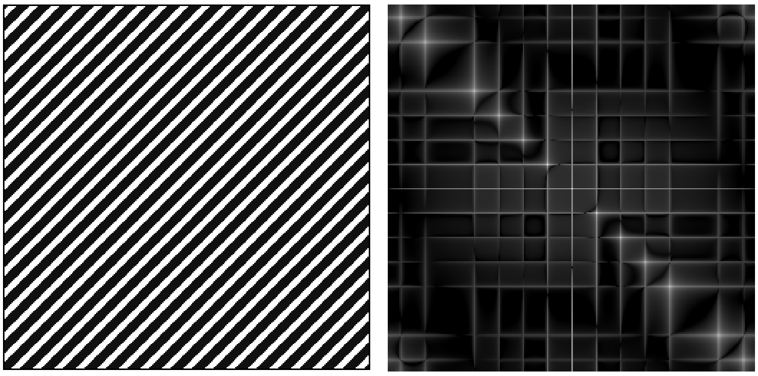
4. Materiały i metody

5. Wyniki

5.1. Filtracja w dziedzinie częstotliwości

Widma mocy jak i również fazy, nie będą prezentowane przy okazji prezentowania wyników filtracji, zostały one przedstawione przy okazji wyjaśnienia działania poszczególnych filtrów. Powodem tego jest fakt, iż widma obrazów naturalnych są trudne do interpretacji, natomiast dla obrazów sztucznych, np wzorców są one znacznie bardziej przejrzyste dla naszego oka.

Dla przykładu, widmo przypadkowego obrazu prezentującego pewien wzorzec 10, i obrazu naturalnego 1.



(a) obraz (b) widmo mocy

Rysunek 10: Widmo mocy obrazu sztucznego.

5.1.1. Filtr dolnoprzepustowy

(a) filtr idealny (b) filtr Butterwortha n=5

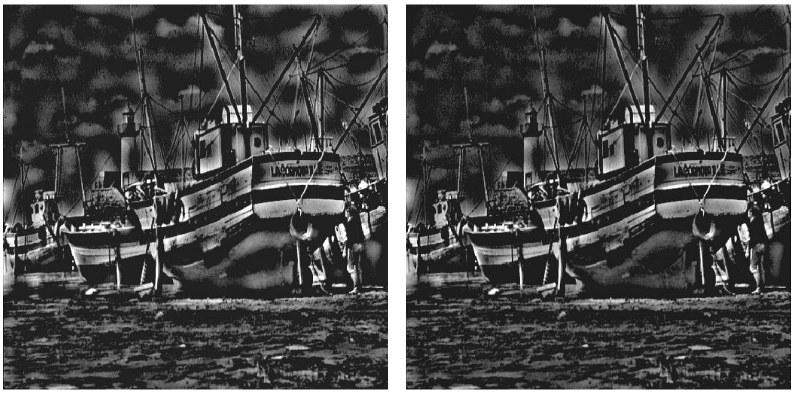
Rysunek 11: Filtr dolnoprzepustowy, próg $D_0 = 0.1$.

5.1.2. Filtr górnoprzepustowy

(a) $D_0 = 0.1$ (b) $D_0 = 0.02$, jasność +80

Rysunek 12: Filtr górnoprzepustowy.

5.1.3. Filtr pasmowoprzepustowy



(a) $D_L = 0.03$, $D_H = 0.6$, +100 jasność, filtr idealny (b) $D_L = 0.03$, $D_H = 0.6$, +100 jasność, filtr Butterwortha n=5

Rysunek 13: Filtr pasmowoprzepustowy.

(a) $D_L = 0.01$, $D_H = 0.6$, +80 jasność, filtr idealny (b) $D_L = 0.01$, $D_H = 0.6$, +80 jasność, filtr Butterwortha n=2

Rysunek 14: Filtr pasmowoprzepustowy.

5.1.4. Filtr pasmowozaporowy

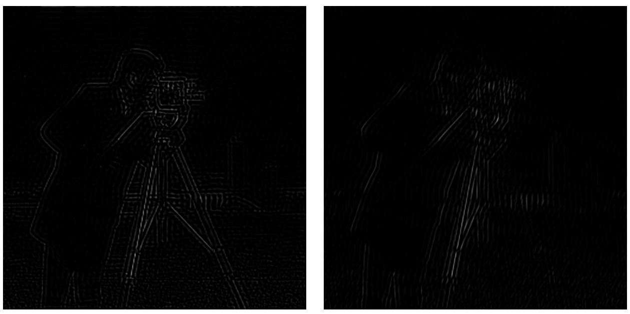
(a) $D_L = 0.02$, $D_H = 0.05$, filtr idealny (b) $D_L = 0.02$, $D_H = 0.05$, filtr Butterwortha n=2

Rysunek 15: Filtr pasmowozaporowy.

5.1.5. Filtr z detekcją krawędzi

(a) $D_L = 0.15$, $D_H = 0.6$, $\alpha = 90\,\mathrm{deg}$, $P(0;1)$ (b) $D_L = 0.15$, $D_H = 0.6$, $\alpha = 90\,\mathrm{deg}$, $P(1;0)$ (c) $D_L = 0.15$, $D_H = 0.6$, $\alpha = 90\,\mathrm{deg}$, $P(1;1)$

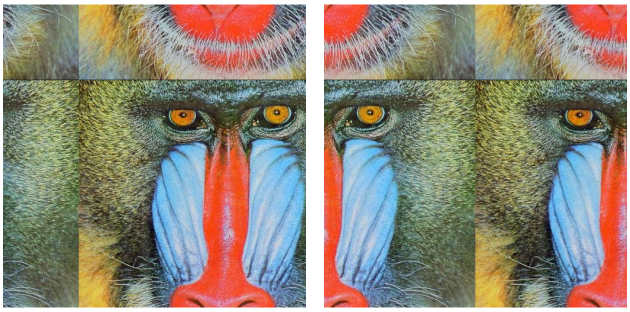
Rysunek 16: Filtr pasmowozaporowy.



(a) $D_L = 0.15$, $D_H = 0.6$, $\alpha = 30\,\mathrm{deg}$, $P(1;1)$ (b) $D_L = 0.15$, $D_H = 0.6$, $\alpha = 120\,\mathrm{deg}$, $P(1;1)$

Rysunek 17: Filtr pasmowozaporowy, różne kąty dla $P(1;1)$.

5.1.6. Filtr modyfikujący widmo



(a) $l = 0.25$, $k = 0.25$ (b) $l = 0.25$, $k = 0.5$

Rysunek 18: Filtr modyfikujący widmo.

5.2. Segmentacja

6. Dyskusja

6.1. Filtracja w dziedzinie częstotliwości

6.1.1. Filtr dolnoprzepustowy

Dzięki temu filtrowi uzyskujemy efekt rozmycia, jak widać na rysunku 11. Jednak im bardziej rozmyjemy obraz(im niższa wartość parametru D_0) tym bardziej widoczne stają się artefakty widoczne na rys 11a. Jednak stosując filtr Butterwortha, można te niepożądane efekty zniwelować(rys. 11b);

6.1.2. Filtr górnoprzepustowy

Filtr ten jest negacją filtru poprzedniego, więc jak się można spodziewać efekt będzie odwrotny jak w tamtym przypadku czyli wyostrzenie. Parametr D_0 w wyższych wartościach powoduje pozostawienie na obrazie jedynie najwyraźniejszych krawędzi obrazu (rys. 12a), a przy coraz mniejszych wartościach odwrotnie, czyli uwydatnianie krawędzi, obiektów wyróżniających się, kontrastujących z ich tłem (rys. 12b) .

6.1.3. Filtr pasmowoprzepustowy

Filtr ten daje podobny efekt jak w przypadku filtru górnoprzepustowego przy obraniu wysokich wartości parametrów D_L D_H.

Jednak w odróżnieniu od niego pozwala na wybranie interesującego nas pasma częstotliwości branego pod uwagę. Można uzyskać nim lepszy wynik przy wyostrzaniu obrazów rys. 13 rys. 14. Porównując wynik uzyskany za pomocą filtru idealnego i filtru Butterwortha widać, że obcinając „zero-jedynkowo" przedział częstotliwości, w znacznym stopniu zmieniamy obraz wynikowy w stosunku do wejściowego. Przejścia pomiędzy obszarami jasności a obszarami ciemnymi stają się bardziej skokowe, natomiast stosując filtr Butterwortha zachowują więcej odcieni szarości - zachowane jest ciągłe przejście.

6.1.4. Filtr pasmowozaporowy

Podobnie jak w przypadku filtru dolnoprzepustowego i górnoprzepustowego, filtr ten jest negacją filtru pasmowoprzepustowego. Ciekawy efekt jaki udało się uzyskać stosując ten filtr jest wygładzenie, rozmycie obrazu z jednoczesnym zachowaniem wyraźnych krawędzi (rys. 15). Podobnie jak w przypadku filtru dolnoprzepustowego, zastosowanie filtracji Butterwotha pozwala nam uniknąć artefaktów powstałych w przypadku filtracji idealnej.

6.1.5. Filtr z detekcją krawędzi

Analogicznie jak w przypadku wykrywania krawędzi zaprezentowanego w zadaniu pierwszym. Filtr ten pozwala na wykrycie linii oraz krzywych, jednak w odróżnieniu do tamtego rozwiązania filtr ten pozwala na zdefiniowane zakresu krzywizny identyfikowanych krzywych poprzez parametry α oraz P (punkt ramienia kąta o wierzchołku w (0,0)). Jak widać na rysunkach 16 gdzie dla ustalonego kąta zmieniamy ramie kąta, zmieniamy orientacje wykrywanych krzywych. Oraz zmieniając kąt (rys. 17) zmieniamy zakres krzywizny.

6.1.6. Filtr modyfikujący widmo

W odróżnieniu do filtrów opisanych powyżej filtr ten nie polega na odrzuceniu pewnego pasma częstotliwości, lecz przemnaża cały obraz przez filtr H 16. W wyniku otrzymujemy przesunięty cyklicznie obraz względem obu osi, przesunięcie to jest zdefiniowane za pomocą parametrów l, k.

6.2. Segmentacja

7. Wnioski

7.1. Filtracja w dziedzinie częstotliwości

Transformacja obrazu do dziedziny częstotliwości pozwala na wykonanie wielu filtracji obrazu w mniej zachłanny sposób. Najlepszym przykładem tutaj będzie porównanie filtru uśredniającego z filtrem dolnoprzepustowym do zadania rozmycia obrazu. Gdy chcemy uzyskać większe rozmycie w przypadku filtru uśredniającego należałoby zwiększyć maskę co z kolei zwiększałoby ilość potrzebnych obliczeń. Natomiast gdy operujemy filtrem dolnoprzepustowym, nie zwiększy to w żaden sposób ilości obliczeń, dzięki temu uzyskujemy dobrze skalującą się metodę. W zależności od zastosowania, dobrym pomysłem może być zastosowanie filtrów o charakterystykach innych od idealnej, na przykładzie zastosowanego w zadaniu filtru Butterwortha można zauważyć, że przy zastosowaniu filtru idealnego mogą wystąpić niepożądane efekty tj na rys. 11a , 13a oraz 15a.

7.2. Segmentacja

Literatura

[1] Fast Fourier Transform (FFT) `http://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/transform/fft.html`

[2] The Radix 2 Decimation In Frequency (DIF) Algorithm. `http://www.engineeringproductivitytools.com/stuff/T0001/PT03.HTM`

[3] Constant Time Neighbor Finding in Quadtrees: An Experimental Result, Kunio Aizawa, Koyo Motomura, Shintaro Kimura, Ryosuke Kadowaki, and Jia Fan