Tomasz Kryjak **02.12.2013 v1.4**

PRZETWARZANIE OBRAZÓW CYFROWYCH

Detekcja krawędzi
Transformata Hougha

Cel:

- zapoznanie z metodami detekcji krawędzi:
  - Sobel, Prewitt, Roberts - przypomnienie
  - Laplasjan z Gaussa (LoG - *Laplacian of Gaussian*)
  - Canny
- zapoznanie z transformatą Hougha
  - dla pojedynczego punktu
  - dla kilku punktów
  - dla prostych figur
- wykorzystanie transformaty Hougha do detekcji linii prostych na rzeczywistym obrazie
- transformata Hougha w przestrzeni ab - zadanie dodatkowe

I. Detekcja krawędzi

Detekcja krawędzi przez wiele lat była podstawą algorytmów segmentacji. Krawędzie wykrywane są najczęściej z wykorzystaniem pierwszej (**gradient**) i drugiej (**Laplasjan**) pochodnej. Wykorzystanie obu metod zaprezentowane zostało w ćwiczeniu "Przetwarzanie wstępne. Filtracja kontekstowa".

W obecnym ćwiczeniu poznane detektory krawędzi: **Sobela**, **Prewitta** i **Robertsa** zostaną porównane z bardziej zaawansowanymi: **Laplasjan z funkcji Gaussa (LoG)**, **Zero Crossing** i **Canny**.

a) Laplasjan z Gaussa (LoG)

Funkcja Gaussa: $h(r)=-e^{(\frac{-r^2}{2\cdot\sigma^2})}$ gdzie $r^2=x^2+y^2$ i σ – odchylenie standardowe.

Działanie filtracji Gaussowskiej zostało przedstawione w ćwiczeniu "Przetwarzanie wstępne" - w jej wyniku następuje rozmazanie obrazu.

Laplasjan z tej funkcji:

$$\nabla^2 h(r)=-[\frac{r^2-\sigma^2}{\sigma^4}]e^{(\frac{-r^2}{2\cdot\sigma^2})}$$

Funkcję (z oczywistych powodów) nazywamy Laplasjan z Gaussa (LoG). Ponieważ druga pochodna jest operacją liniową, konwolucja obrazu z $\nabla^2 h(r)$ daje taki sam efekt jak zastosowanie filtracji Gaussa na obrazie, a następnie obliczenie Laplasjanu z wyniku.

Lokalizacja krawędzi polega na znalezieniu miejsca, gdzie na obrazie po filtracji LoG następuje zmiana znaku.

1. Otwórz program **Matlab**. Ustal ścieżkę **Current Directory** na swój własny katalog na dysku D. Utwórz nowy m-plik (**New Script**) lub (**New->Script**). Na początku wykonaj polecenia `close all; clearvars; clc;`

2. Wczytaj obraz "dom.png" (wcześniej ściągnij archiwum ze strony www i rozpakuj w odpowiednim katalogu).

3. Przeprowadź detekcję krawędzi - wykorzystaj funkcję `edge` z parametrem 'log'. Zwróć uwagę na składnię:
   - thresh - próg binaryzacji (jeżeli się go nie poda to zostanie dobrany automatycznie),
   - sigma - odchylenie standardowe Gaussa.

   Rezultat detekcji krawędzi wyświetl.

Detekcja przejścia przez zero (*Zero-Crossing Detector*) - detektor opiera się na tej samej koncepcji co LoG - jedyna różnica to ręczne podawanie funkcji filtra (nie musi być Gaussa).

b) **algorytm Canny'ego**

Algorytm Canny'ego to często wykorzystywana metoda detekcji krawędzi zaproponowana w 1986r. przez Johna F. Cannego. W pierwszym kroku obraz rozmywany jest filtrem Gaussa. Następnie na podstawie gradientów G_x i G_y (uzyskiwanych np. za pomocą maski Sobela) wyliczana jest amplituda gradientu: $g(x,y)=\sqrt{(G_x^2+G_y^2)}$ oraz kierunek $\alpha(x,y)=\arctan(G_y/G_x)$. W kolejnym kroku występuje procedura, która "śledzi" grzbiety na obrazie amplitudy - wyeliminowane zostają piksele, które nie mają wartości maksymalnej (*nonmaximal suppresion*). Ostatnią fazą, jest binaryzacja z dwoma progami T1 i T2 (T1 < T2). Zasada jest następująca: za piksel krawędziowy uznaje się piksel o jasności >T2 lub piksel o jasności > T1, w którego bezpośrednim otoczeniu znajduje się co najmniej jeden piksel o jasności > T2.

1. Dla obrazu "dom.png" wykonaj detekcję krawędzi metodą Canny. Wykorzystaj funkcję `edge`, z parametrem 'canny'. Parametry:
   - thresh - wektor progów T1 i T2,
   - sigma - odchylenie standardowe dla Gaussa.

d) porównanie metod detekcji krawędzi:

1. Rozważmy metody Sobel, LoG i Canny. Dla obrazu "dom.png" wykonaj detekcję krawędzi z automatycznym doborem parametrów, a następnie dla każdej z metod dobierz parametry "ręcznie" - tak aby jak najlepiej wykryć krawędzie (chodzi o wydobycie bryły domu, z pominięciem faktury dachówek i cegieł)

2. [P] Wyniki zaprezentuj prowadzącemu.

II. Transformacja Hough'a

Transformacja Hougha dla prostych jest metodą detekcji współliniowych punktów. Każda prosta może być jednoznacznie przedstawiona za pomocą dwóch parametrów. Przestrzeń tych parametrów to przestrzeń Hougha. Najczęściej wykorzystywanymi parametrami są współczynniki σ, θ opisujące równanie prostej w postaci normalnej:

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (1)$$

gdzie: ρ - promień wodzący, θ - kąt tworzony przez z ρ osią OX.

Własności transformaty Hougha:

- **prostej** w przestrzeni kartezjańskiej odpowiada **punkt** w przestrzeni Hougha,
- **punktowi** w przestrzeni kartezjańskiej odpowiada **krzywa sinusoidalna** w przestrzeni Hougha,
- **punkty leżące na tej samej prostej** (w przestrzeni kartezjańskiej) korespondują z sinusoidami przechodzącymi przez **wspólny punkt** w przestrzeni Hougha.

Metoda wyliczania transformaty Hougha składa się z następujących kroków:

- przez każdy badany (różny od zera) punkt obrazu prowadzony jest pęk prostych, przechodzących przez ten punkt,
- każda z tych prostych transformowana jest do przestrzeni Hougha i tworzy tam punkt o współrzędnych ρ, θ ,
- w ten sposób, każdy punkt obrazu pierwotnego (pęk prostych) jest odwzorowany w sinusoidalną krzywą w przestrzeni Hougha.

Przestrzeń Hougha jest przestrzenią akumulacyjną tzn. punkty sinusoidalnych krzywych, wygenerowanych dla różnych punktów obrazu pierwotnego dodają się w miejscach, w których krzywe te przecinają się. Powstałe w ten sposób (w przestrzeni Hougha) maksima odpowiadają zbiorom punktów, należących do jednej prostej. Wartość maksimum odpowiada liczbie współliniowych punktów. Współrzędne ρ, θ tego maksimum jednoznacznie określają położenie prostej na obrazie pierwotnym.

A. Transformata Hougha dla małej liczby punktów.

1. Utwórz nowy m-plik (**New Script**) lub (**New->Script**). Na początku wykonaj polecenia `close all; clearvars; clc;`

2. Stwórz "obraz" - macierz samych zer o rozmiarze 11x11 (funkcja `zeros`).

3. Wartość jednego, wybranego punktu z obrazu ustal na 1.

4. Wykonaj transformatę Hougha "obrazu":
   - wykorzystaj funkcję hough
   - ustal parametry: `'RhoResolution',0.1,` - rozdzielczość promienia wodzącego `'ThetaResolution',0.5` - rozdzielczość kąta (w stopniach)
   - funkcja zwraca macierz **H** (przestrzeń Hougha) oraz dwa wektory **theta i rho**

5. Wyświetl przestrzeń Hougha - za pomocą funkcji `imshow`. Warto wykorzystać skalowanie [].

Jak "wygląda" pojedynczy punkt w przestrzeni Hougha ?

6. Dodaj kolejny punkt do "obrazu". Jak zmienia się przestrzeń Hougha ?

7. Do "obrazu" dodaj jeszcze dwa punkty. Zaobserwuj przestrzeń Hougha. Spróbuj odtworzyć jedną z prostych na obrazie. W tym celu na wykresie za pomocą narzędzia "Data Cursor" (Tools->Data Cursor) odczytaj współrzędne wybranego maksimum w przestrzeni Hougha. Współrzędna "X" to θ (pozioma), a "Y" to ρ (pionowa).

8. W celu odczytania wartości ρ, θ należy wykorzystać wektory *rho* i *theta* zwracane przez funkcję Hough tj. np. ρ = *rho(odczytana_wsp_y);*

9. Wykorzystując równanie (1) odtwarzamy prostą. Na początku generujemy wektor x z zakresu 0-10 z krokiem 0.1 (przypomnienie: `x=0:0.1:10;`). Następnie przekształcamy równanie (1) do postaci y = ... i wstawiając parametry ρ, θ generujemy wektor y (`y = ((ρ-x*cos(θ))/sin(θ));`). Proszę zwrócić uwagę, jakie argumenty przyjmują funkcje `sin` i `cos` w Matlabie, a co otrzymujemy w wektorze *theta*.

   Uwaga !. Możliwe są dwa wyjścia: a) manualna konwersja ze stopni do radianów, b) użycie funkcji *sind* i *cosd*, które akceptują argumenty w stopniach.

10. Otrzymaną prostą wyświetlamy na "obrazie" - najpierw wyświetlamy "obraz" (nie przestrzeń Hough'a) - `imshow`, a następnie wykorzystując polecenie `hold on` oraz polecenie `plot` wyrysowujemy prostą. Uwagi:
    - wyświetlanie: `plot(x+1,y+1)` dodanie '1' ma związek z innymi układami współrzędnych - dla przestrzeni Hougha od punktu (0,0), a dla obrazka od punktu (1,1) – numerowanie w Matlabie jest od '1'.

11. [P] Zaprezentuj wyniki prowadzącemu. Zastanów się nad następującym problemem - czy maksimum w przestrzeni Hougha zawsze istnieje - np. w przypadku gdy na obrazie wejściowym znajdują się tylko dwa punkty.

B. Transformata Hougha dla pojedynczego obiektu

W tym podpunkcie pokazane zostanie praktycznie wykorzystanie transformaty Hougha - do detekcji prostych na sztucznym rysunku.

1. Utwórz nowy m-plik (**New Script**) lub (**New->Script**). Na początku wykonaj polecenia `close all; clearvars; clc;` Wczytaj obraz "kwadraty.png". Wyświetl go.

2. Wykonaj detekcję krawędzi - wykorzystaj jedną z metod z I części ćwiczenia. Ważne aby obraz krawędzi był jak najlepszej jakości - co oznacza cienkie, ciągłe krawędzie (dla tego przypadku nie powinno być trudne do uzyskania). Wyświetl obraz po detekcji krawędzi.

3. Wykonaj transformatę Hougha obrazu krawędziowego. Wykorzystaj funkcję `hough` z parametrami domyślnymi (rozdzielczość kątowa 1 stopień, a przestrzenna 1 piksel).

4. Wyświetl macierz H (koniecznie ze skalowaniem []). Czy widoczna jest taka liczba maksimów jakiej się spodziewamy?

5. W *Image Processing Toolbox* dostępna jest funkcja do automatycznej analizy przestrzeni Hougha - wyszukiwania maksimów - `houghpeaks` . Jako parametry przyjmuje ona macierz H oraz liczbę poszukiwanych maksimów. Dodatkowo można podać próg powyżej którego punkt uznawany jest za maksimum oraz rozmiar otoczenia jakie zostanie wyzerowane po detekcji maksimum (szczegóły w help'ie). Funkcja zwraca współrzędne maksimów.

6. Wykorzystaj funkcję `houghpeaks` - poszukujemy 8 maksimów.

7. Uzyskany wynik trzeba skontrolować. Najlepiej poprzez wyrysowanie na przestrzeni Hough'a odnalezionych punktów – funkcja plot. Warto użyć symbolu w postaci okręgu – parametr `'o'` .

8. W przypadku problemów należy odpowiednio spróbować wykorzystać parametry funkcji `houghpeaks`.

9. Kolejną użyteczną funkcją z *IPT* jest `houghlines`. Funkcja na podstawie wektorów *rho* i *theta* oraz rezultatu działania funkcji `houghpeaks` wyznacza linie obecne na obrazie. Funkcja zwraca wektor struktur z opisem wykrytych linii.

10. Wyznacz linie obecne na obrazie - `houghlines`. Do wyświetlania linii wykorzystaj przykładowy kod umieszczony w pomocy do funkcji `houghlines` (odpowiedni fragment).

11. [P] Zaprezentuj wyniki prowadzącemu.

C. Transformata Hougha dla obrazu rzeczywistego.

Bazując na kodzie stworzonym w punkcie B wyszukamy linie na obrazie rzeczywistym.

1. Wczytaj obraz "lab112.png". Wyświetl go.

2. Wykorzystując wszystkie poznane techniki przetwarzania obrazów (filtracja, przekształcenia morfologiczne, binaryzację, detekcję krawędzi) wyodrębnij krawędzie kwadratów - tak aby były jak najlepszej jakości (cienkie, ciągłe) - jednocześnie eliminując z obrazu pozostałe zakłócenia.

3. Wykorzystaj kod z podpunktu B i przeprowadź detekcję linii na obrazie.

4. Wypróbuj działanie transformacji Hougha na obrazie "dom.png". Zobacz jak wygląda przestrzeń Hougha. Wybierz odpowiednią liczbę maksimów.

5. [P] Wyniki zaprezentuj prowadzącemu.

D. * Transformata Hougha w przestrzeni ab**

Przestrzeń σ, θ nie jest jedyną przestrzenią w której punkt odpowiada parametrom prostej. Np. można spróbować wykorzystać tradycyjne równanie prostej:

$$y=ax+b \quad (2)$$

W tej przestrzeni reprezentacją **pęku prostych** jest **prosta**.

Zadanie: napisać funkcję, która jako argument przyjmuje obraz (binarny) oraz parametry:

- aMin - minimalna wartość parametru a
- aMax - maksymalna wartość parametru a
- aSkok - skok parametru a
- bMin - minimalna wartość parametru b
- bMax - maksymalna wartość parametru b
- bSkok - skok parametru b

Jako wynik ma zwrócić macierz przestrzeni Hougha **ab.**

Uwagi:

- zadanie może wyglądać na skomplikowane ale tak na prawdę wymaga tylko starannego przemyślenia,
- najważniejszy jest problem "adresowania" macierzy H. Macierz tę tworzymy na początku i wypełniamy wartościami 0. Rozmiar zależy od parametrów.
- trzeba wygenerować dwa wektory **a** i **b** ze wszystkimi możliwymi wartościami jakie te parametry mogę przyjąć,
- dla każdej z wartości 'a' obliczamy odpowiednią wartość 'b' i dokonujemy jej kwantyzacji (np. poprzez taką składnię: `[v bb] = min( abs(b_v - b));`. Wynik to parametr bb,
- wartość punktu w przestrzeni **ab** inkrementujemy: `H(aa,bb) = H(aa,bb) + 1;`,
- działanie funkcji należy przetestować na prostym przykładzie - jak w punkcie A.

Zastanów się w przypadku jakich prostych reprezentacja **ab** nie sprawdzi się.

Uwaga !

Podejście oparte o transformatę Hough'a nie jest tylko ograniczone do prostych. W ten sam sposób można wykrywać na obrazie wszystkie krzywe analityczne (tj. opisane równaniem). Problemem bywa tylko duża złożoność obliczeniowa i pamięciowa. Proszę przemyśleć jak w jaki sposób za pomocą transformaty Hough'a zrealizować wykrywanie okręgów i elips.