

Przegląd algorytmów i narzędzi do analizy obrazów w Matlabie

Rafał GRABIAŃSKI i Zbigniew KRÓLIKOWSKI

29.05.2015

Struktura danych dla obrazu

Reprezentacja obrazu

Naturalną strukturą do przechowywania obrazów jest macierz.

Mamy cztery typy reprezentacji, które wpływają na to jak wartości w tej macierzy są przez program interpretowane.

- ▶ Binary (bilevel image) - tablica wartości logicznych 1 i 0. Interpretowanych jako kolor czarny i biały.
- ▶ Indexed (pseudocolor image) - tablica wartości całkowitych, które są indeksami w mapie kolorów $m \times 3$.
- ▶ Grayscale (intensity image) - tablica wartości stanowiących kontrast danego piksela. Najczęściej do obrazów w skali szarości, chociaż można też mapować z colormapą.
- ▶ Truecolor (RGB image) - macierz $m \times n \times 3$, w której w trzecim wymiarze przechowujemy dokładną informację o kolorach czerwonym, zielonym i niebieskim.

Struktura danych dla obrazu

Example

```
RGB = reshape(ones(64,1)  
             *reshape(jet(64),1,192),[64,64,3]);  
R = RGB(:, :, 1);  
G = RGB(:, :, 2);  
B = RGB(:, :, 3);  
figure , imshow(R)  
figure , imshow(G)  
figure , imshow(B)  
figure , imshow(RGB)
```

Wczytywanie obrazu

Funkcja imread

Wczytywanie obrazu do macierzy w matlabie sprowadza się do wywołania funkcji `imread` z nazwą obrazu. Po zawartości pliku `imread` rozpoznaje jego typ i zwraca macierz/-e obrazu. Można oczywiście w odpowiednim parametrze wyspecyfikować ten typ.

Example

```
RGB = imread( 'football.jpg' );  
[X,map] = imread( 'trees.tif' );
```

Wczytywanie obrazu - przykłady

Example

```
A = imread( 'ngc6543a.jpg' );  
image(A);  
A(300, 300, 1)  
A(300, 300, 2)  
A(300, 300, 3)
```

```
[X,map] = imread( 'corn.tif' );  
if ~isempty(map)  
    Im = ind2rgb(X,map);  
end  
whos Im;
```

Zapisywanie obrazu

Funkcja imwrite

Funkcja `imwrite` zapisuje podaną macierz w postaci obrazu, którego typ rozpoznaje z podanej nazwy pliku albo przy użyciu odpowiedniego parametru.

Example

```
imwrite(X,map,'clown.bmp')  
imwrite(X,map,'clown.jpg')
```

Zadanie

```
X/100  
imwrite(X/100,'clown3.jpg')
```

Konwertowanie pomiędzy formatami obrazów

Konwersja

Nie ma specjalnych funkcji konwertujących pomiędzy różnymi formatami. Używamy już omówionych `imread` i `imwrite` do odpowiednich konwersji.

Pozyskiwanie informacji o pliku

Funkcja imfinfo

Do odczytania typu obrazu i jego różnych właściwości możemy użyć funkcji `imfinfo`, w argumencie podając nazwę pliku.

Example

```
info = imfinfo( 'ngc6543a.jpg ' )
```


Wyświetlanie i badanie obrazów

Display and exploration tools

Funkcje i aplikacje do wyświetlania i badania obrazów służą do oglądania zdjęć, pozyskiwania szczegółowych informacji o poszczególnych pikselach, ustawiania kontrastu czy też pomiaru odległości. Wprowadzimy w tym referacie najbardziej podstawowe narzędzia.

Funkcja imshow

Funkcja imshow służy do otwierania w oknie Matlaba zdjęcia. Jeżeli nie mieści się ono na ekranie jest przeskalowywane. Standardowo po wyświetleniu jeden piksel na ekranie - jeden piksel zdjęcia.

Example

```
moon = imread('moon.tif');  
imshow(moon);  
figure , imshow(moon, 'Border', 'tight');
```

Wyświetlanie i badanie obrazów - przykłady

Przykłady

Prezentujemy jeszcze przykład wyświetlenia dwóch obrazów w jednej figurze. Subplot dzieli figurę, a subimage działa jak imshow, z tą jednak różnicą, że przed wyświetleniem konwertuje obraz na format truecolor. Przy wyświetlaniu zdjęć w jednej figurze musiałyby mieć w przeciwnym wypadku tę samą colormapę.

Example

```
[X1, map1]=imread('forest.tif');  
[X2, map2]=imread('trees.tif');  
subplot(1,2,1), subimage(X1, map1)  
subplot(1,2,2), subimage(X2, map2)
```

Image Viewer App

Image Viewer App

Image Viewer App to prosty tool do wyświetlania obrazów i wykonywania podstawowych operacji na obrazach. To zintegrowane środowisko zawiera w sobie wiele pomocnych tooli:

- ▶ Pixel Information Tool - do uzyskiwania informacji o bieżącym pikselu.
- ▶ Pixel Region Tool - do uzyskiwania informacji o grupie pikseli.
- ▶ Distance Tool - do pomiaru odległości między dwoma pikselami.
- ▶ Adjust Contrast Tool - do zmiany kontrastu wyświetlanego obrazu i modyfikowania danych.
- ▶ Crop Image Tool - do przycinania.
- ▶ Display Range Tool - do zmiany zakresu natężeń obrazu.

Image Viewer App - użycie

Funkcja imtool

Aby uruchomić Image Viewer App, wybieramy odpowiednią opcję z poziomu menu File albo używamy funkcji `imtool`.

Example

```
imtool( 'moon.tif' );  
imtool( 'moon.tif', 'InitialMagnification', 150)
```

Transformacje geometryczne obrazu, spatial referencing, rejestracja obrazu

Plan

Zajmiemy się na początku prostymi przekształceniami geometrycznymi obrazu by później skończyć na bardziej zaawansowanym przykładzie rejestracji obrazu.

Skalowanie obrazu

Funkcja imresize

Aby zmienić rozmiar obrazu używamy funkcji `imresize`. Kluczowy jest magnification factor, który podajemy jako argument. Dla wartości większych od 1 obraz wyjściowy będzie większy niż oryginalny.

Example

```
I = imread('circuit.tif');  
J = imresize(I,1.25);  
imshow(I)  
figure , imshow(J)
```

Skalowanie obrazu - dodatkowe parametry

Interpolacja przy powiększaniu obrazu

Do funkcji `imresize` możemy podać w apostrofach argument mówiący o metodzie interpolowania pikseli powstałych przy powiększeniu obrazu. Opcja `'bicubic'` jest defaultowa.

Zjawisko aliasingu podczas zmniejszania rozmiaru obrazu

Domyślnie podczas zmniejszania rozmiaru obrazu, funkcja `imresize` używa antialiasingu. Możemy tę opcję wyłączyć dodając argumenty: `'Antialiasing', false`.

Example

```
I = imread('rice.png');  
J = imresize(I, 0.5, 'Antialiasing', false);  
figure, imshow(I), figure, imshow(J)  
J = imresize(I, 0.5);  
figure, imshow(J)
```

Obracanie obrazu

Funkcja imrotate

Do obracania obrazu używamy funkcji `imrotate` podając w argumencie oprócz macierzy obrazu, kąt obrócenia. Dodatni kąt obraca w ruchu przeciwnym do ruchu wskazówek zegara.

Example

```
I = imread('circuit.tif');  
J = imrotate(I,35,'bilinear');  
imshow(I)  
figure , imshow(J)
```


Przycinanie obrazu

Funkcja `imcrop`

Do przycinania obrazu używamy funkcji `imcrop`. Możemy wywołać ją interaktywnie lub podając konkretne parametry w postaci `[xmin, ymin, width, height]`.

Example

```
I = imread('circuit.tif');  
J = imcrop(I,[60 40 100 90]);
```

Geometryczna transformacja 2D

Transformacja geometryczna

Wykonanie transformacji w Image Processing Toolbox składa się z dwóch etapów.

- ▶ Stworzenie obiektu transformacji `geometric transformation object`
- ▶ Użycie funkcji `imwarp` podając obraz do transformacji i obiekt transformacji

Geometryczna transformacja 2D

Macierze transformacji

Macierze transformacji służą temu by stosować rachunek macierzowy przy składaniu przekształcenia. Każdy piksel reprezentuje się przy pomocy macierzy 3×1 postaci $[x, y, 1]$. Przekształcenie to macierz, która po pomnożeniu przez punkt wskaże pozycję tego punktu. I tak macierz translacji o współrzędne $[t_x, t_y]$ będzie miała postać:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

Inne przykłady macierzy transformacji

Skalowanie

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotacja

$$R = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Przykład prostej translacji obrazu

Translacja 2D obrazu

Najpierw odcytujemy obraz, który użyjemy do przekształcenia. Następnie tworzymy z macierzy transformacji obiekt transformacji funkcją `affine2d`. Ostatnim krokiem jest użycie funkcji `imwarp` podając obraz i uzyskany wcześniej obiekt. Wszystko co zaraz zrobimy można zrobić też jedną funkcją `imtranslate`.

Przykład prostej translacji obrazu

Example

```
cb = checkerboard;  
tform = affine2d([1 0 0; 0 1 0; 20 20 1]);  
Rcb = imref2d(size(cb))  
Rout = Rcb;  
Rout.XWorldLimits(2) = Rout.XWorldLimits(2)+20;  
Rout.YWorldLimits(2) = Rout.YWorldLimits(2)+20;  
[out,Rout] = imwarp(cb,tform,'OutputView',Rout);  
figure , subplot(1,2,1);  
imshow(cb,Rcb);  
subplot(1,2,2);  
imshow(out,Rout)
```

Rejestracja obrazu

Rejestracja obrazu

Rejestracja obrazu jest procesem przenoszenia kilku zbiorów danych do jednego układu współrzędnych. Wykonamy przykładową rejestrację dwóch obrazów używając spatial referencing.

Przykładowa rejestracja obrazu

- ▶ Odczytanie dwóch przykładowych obrazów
- ▶ Wczytujemy gotowe wspólne punkty dla obu obrazów
- ▶ Tworzymy obiekt transformacji typu projective
- ▶ Używamy funkcji imwarp z gotowym obiektem do wykonania transformacji na przestawionym obrazie
- ▶ Wyświetlamy obrazy w odpowiednim układzie

Przykładowa rejestracja obrazu

Example

```
fixed = imread('westconcordorthophoto.png');  
moving = imread('westconcordaerial.png');  
load westconcordpoints  
tform = fitgeotrans(movingPoints, fixedPoints,  
                    'projective');  
registered = imwarp(moving, tform,  
                    'FillValues', 255);  
figure, imshow(registered);  
Rfixed = imref2d(size(fixed));  
registered1 = imwarp(moving, tform,  
                    'FillValues', 255,  
                    'OutputView', Rfixed);  
  
figure, imshowpair(fixed, registered1, 'blend');
```

Rejestracja obrazu

O

czywiście istnieje cały szereg funkcji i opcji umożliwiających wykonywanie automatycznych rejestracji. Jedną z nich jest `imregister`. Tu jednak nie będziemy ich omawiać. Zainteresowanych odsyłamy do dokumentacji Matlab.

Ustawianie kontrastu

T

echniki, które teraz zaprezentujemy mogą być opisane wspólnie mianem technik retuszujących, które mają za zadanie "poprawić" w jakiś subiektywny sposób obraz. Pierwszą jaką zaprezentujemy to zwiększenie kontrastu.

Example

```
I = dicomread( 'CT-MONO2-16-ankle.dcm' );  
imshow(I, 'DisplayRange', [])
```

Ustawianie kontrastu

funkcja imadjust

Funkcja `imadjust` zwiększa kontrast w taki sposób, że rozszerza skalę szarości do całego zakresu. Powoduje to, że obraz staje się wyraźniejszy i wykorzystywane są wszystkie odcienie szarości.

Ustawianie kontrastu

Example

```
I = imread('pout.tif');  
imshow(I);  
imtool(I);  
imtool(J);  
J = imadjust(I);  
figure , imshow(J);
```

Ustawianie kontrastu

Funkcja imadjust

Zwykłe użycie imadjust - czy to z ograniczeniem czy na całym zakrasie powoduje, że wartości rozciągają się liniowo na zadanym przedziale, jeśli byśmy jednak chcieli by było inaczej, musimy jako 4. argument podać współczynnik gamma.

Example

```
[X,map] = imread('forest.tif');  
I = ind2gray(X,map);  
J = imadjust(I,[],[],0.5);  
K = imadjust(I,[],[],2);  
imshow(I)  
%figure , imshow(J)  
%figure , imshow(K)  
imtool(J)  
imtool(K)
```

Histogram equalization

Example

```
I = imread('pout.tif');  
J = histeq(I);  
imshow(J)  
figure , imhist(J,64)
```

Filtrowanie obrazu

Funkcja `imfilter`

Filtrowaniu obrazu nie poświęcimy dużo czasu. Pokażemy tylko kilka z wielu różnych możliwości:

Example

```
originalRGB = imread('peppers.png');  
imshow(originalRGB)  
h = fspecial('motion', 50, 45);  
filteredRGB = imfilter(originalRGB, h);  
figure, imshow(filteredRGB)  
boundaryReplicateRGB = imfilter(originalRGB, h, 'replicate');  
figure, imshow(boundaryReplicateRGB)
```


Filtrowanie obrazu

Filtrowanie obrazu

Filtrowanie obrazu to proces, w którym wartość dla danego piksela jest ustalana na podstawie wartości z funkcji biorącej jako argumenty wartości dla pikseli znajdujących się w sąsiedztwie. Jeżeli jest to kombinacja liniowa wartości wtedy filtrowanie nazywamy filtrowaniem liniowym. Omówimy dwa podstawowe filtrowania:

- ▶ Convolution - liniowe filtrowanie z wagami podanymi w postaci macierzy (wyjaśnimy na przykładzie)
- ▶ Correlation - właściwie to samo co konwolucja, jedyna różnica, to że kernel korelacji nie jest obracany

Użycie gotowego filtra

Użycie filtra

Użyjemy teraz predefiniowanego filtra na jednym z obrazów.

Example

```
I = imread('moon.tif');  
h = fspecial('?')  
I2 = imfilter(I,h);  
imshow(I)  
figure  
imshow(I2)
```

Zadanie

Wiedząc że: $h = \begin{bmatrix} -0.1667 & -0.6667 & -0.1667 \\ -0.6667 & 4.3333 & -0.6667 \\ -0.1667 & -0.6667 & -0.1667 \end{bmatrix}$

Powiedz jaki efekt uzyskamy na Księżycu używając tego filtra.

Arytmetyka na obrazach

Wprowadzenie

Arytmetyka na obrazach to po prostu operacje jakie znamy: dodawanie, odejmowanie, mnożenie dzielenie, tyle, że wykonywane na macierzach będącymi obrazami.