

<b>UNIwersytet Zielonogórski</b>	<b>WYKONALI:</b>	<b>GRUPA</b>		<b>OCENA:</b>
<b>Rozpoznawanie obrazów</b>				
<b>ĆWICZENIE NUMER: 5</b>	<b>TEMAT:</b> Detekcja linii prostych.	<b>DATA WYKONANIA:</b>	<b>DATA ODDANIA:</b>	<b>PODPIS:</b>

### 1. Cel ćwiczenia:

Celem ćwiczenia było zapoznanie się z transformatą Radona wykorzystywaną do detekcji linii prostych.

### 2. Teoria:

Postać jednowymiarowa transformaty Radona ma postać:

$$R(\alpha, p) = \int_{-\infty}^{\infty} g(x, y) dx = \int_{-\infty}^{\infty} g\left(x, \frac{p - x \sin \alpha}{\sin \alpha}\right) dx \quad \alpha \neq k\pi$$

Postać dwuwymiarowa transformaty Radona ma postać:

$$R(\alpha, p) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \delta(x \cos \alpha + y \sin \alpha - p) dx dy$$

### 3. Algorytm:

Do realizacji tego ćwiczenia napisaliśmy skrypt w Matlabie który przetwarzał oryginalny obrazek. Skrypt wyznacza krawędzie obrazka oraz transformatę Radona a następnie odwrotną transformatę Radon'a.

```
% wczytanie obrazka
obraz=imread('obraz.bmp');

% wykrywanie krawędzi
krawedzie=edge(obraz);
imshow(krawedzie);

[wys, szer]=size(obraz);
krawedzie=double(krawedzie);
R=zeros(2*wys, 180);

%---- transformata Radona
for alfa=0:179,
    plicznik=1;
    for p=-wys:wys,
        czarny=zeros(szer, wys);

        if alfa==0, %kat zero
```

```
        if p<wys&p>0,
            czarny(wys-p, :)=1;
        end
    else %kat rozny od zera
        alfarad=alfa*2*pi/360;
        x=1:szer;
        y=(p+x.*cos(alfarad))./sin(alfarad);
        y=round(wys/2)+round(y);
        for i=wys:-1:1,
            if (y(i)<=szer)&(y(i)>0)
                czarny(x(szer+1-i),y(i))=1;
            end
        end
    end
    R(plicznik, alfa+1)=sum(sum(krawedzie.*czarny));
    plicznik=plicznik+1;
end
end
figure,
imagesc(1:180,1:2*wys,R);
colormap(hot)
colorbar
xlabel('kat alfa');
ylabel('przesuniecie');

% odwrotna transformata Radona
granica=30;

[y,x]=find(R>=granica);
krawedzie=zeros(szer,wys);

for i=1:length(x),

    if x(i)==1 ,
        p=y(i)-wys;
        krawedzie(wys-p, :)=1;
    else

        alfarad=(x(i)-1)*2*pi/360;
        xx=1:szer;
        p=y(i)-wys;

        yy=(p+xx.*cos(alfarad))./sin(alfarad);
        yy=round(wys/2)+round(yy);

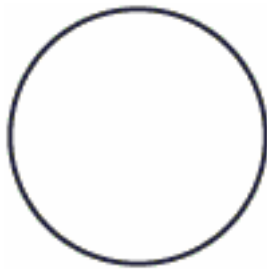
        for i=wys:-1:1,
            if (yy(i)<=szer)&(yy(i)>0)
                krawedzie(xx(szer+1-i),yy(i))=1;
            end
        end
    end
end

end

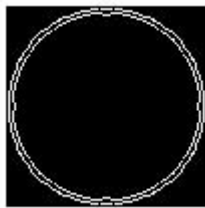
figure,
imshow(krawedzie);
```

#### 4. Wyniki:

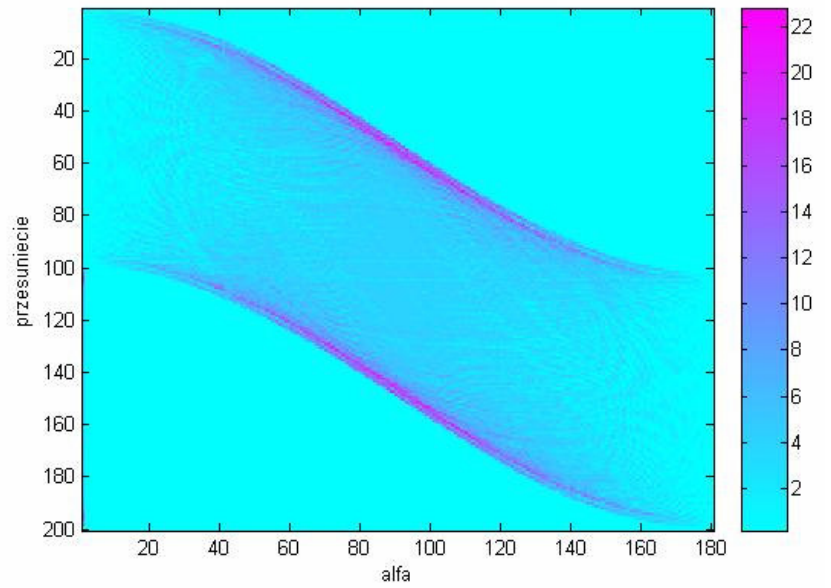
a) obraz nr 1






Rys. 1 Obraz wejściowy



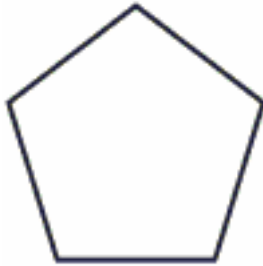
Rys. 2 Wykryte krawędzie



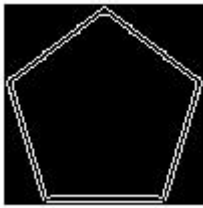
Rys. 3 Transformata Radona

Odtworzone obrazy		
granica = 20	granica = 35	granica = 50
		

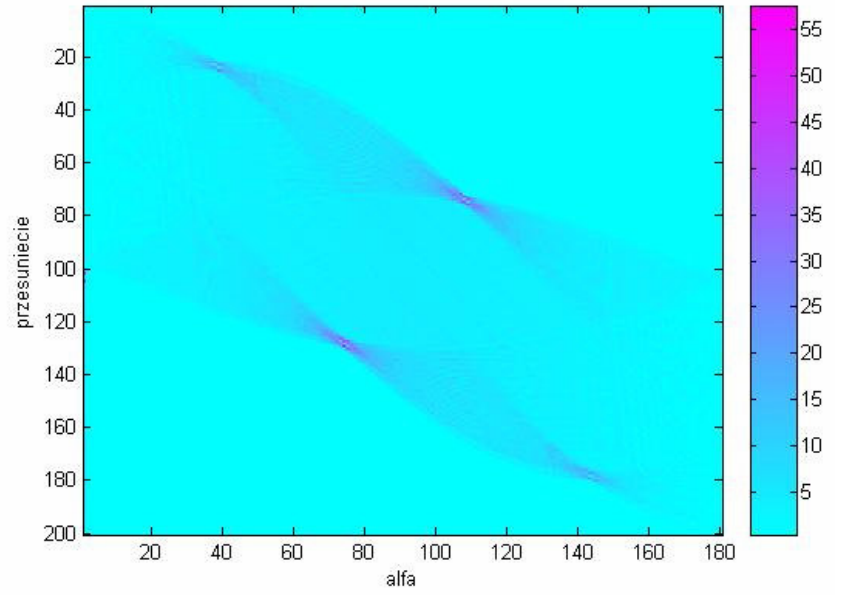
b) obraz nr 2



Rys. 4 Obraz wejściowy



Rys. 5 Wykryte krawędzie



Rys. 6 Transformata Radona

Odtworzone obrazy		
granica = 20	granica = 35	granica = 50

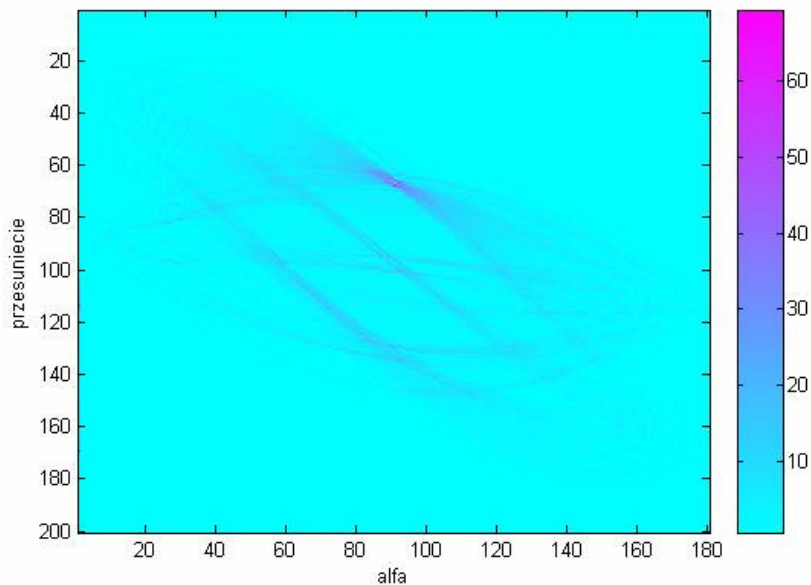
c) obraz nr 3



Rys. 7 Obraz wejściowy



Rys. 8 Wykryte krawędzie



Rys. 9 Transformata Radona

Odtworzone obrazy		
granica = 20	granica = 35	granica = 50

### 5. Wnioski:

W ćwiczeniu tym zajęliśmy się detekcją linii prostych przy wykorzystaniu transformaty Radona.

Do badań wykorzystaliśmy obrazu o rozmiarach 100x100 pikseli. Przy prawidłowo zoptymalizowanej implementacji algorytmu, transformata Radona jest bardzo szybkim narzędziem do rozpoznawania linii prostych.

Wstępnie obraz przepuszczany był przez filtr Sobela w celu wykrycia krawędzi. Jakość działania filtru Sobela miała duży wpływ na poprawną pracę algorytmu transformaty Radona. Przykładem może być obraz nr 3 (rysunek nr 8), gdzie wyraźnie widać, że spośród trzech zapalek poprawnie zostały wyznaczone krawędzie tylko dla pierwszej zapalki. Gwałtowne zmiany koloru tła również zostały rozpoznane przez filtr Sobela jako linie. Przerwy pozostawione przez wstępną filtrację nie mają znaczącego wpływu na poprawne działanie algorytmu transformaty Radona przy prawidłowo ustalonej granicy.