

# Cyfrowe przetwarzanie obrazów i sygnałów

## Wykład 9

### AiR III

**Joanna Ratajczak**

KCiR (W4/K7)

Copyright © 2015 Joanna Ratajczak<sup>1</sup>

---

<sup>1</sup>Na podstawie materiałów autorstwa dra inż. Marka Wnuka.

Niniejszy dokument zawiera materiały do wykładu z przedmiotu Cyfrowe Przetwarzanie Obrazów i Sygnałów. Jest on udostępniony pod warunkiem wykorzystania wyłącznie do własnych, prywatnych potrzeb i może być kopiowany wyłącznie w całości, razem ze stroną tytułową.

# Aproksymacja krawędzi

Od wielu lokalnych cech (*edge elements*) do spójnej, jednowymiarowej cechy (*edge*).

## Różne podejścia:

- szukanie w pobliżu wstępnej aproksymacji,
- transformacja Hougha.

## Wiedza o obiektach:

- globalna forma brzegów
  - linie proste,
  - łuki,
  - krzywe stożkowe,
- ogólne założenia co do treści obrazu
  - możliwie krótka droga pomiędzy dwoma punktami,
  - ograniczona krzywizna linii.

## Metoda strojenia krawędzi znanych *a priori*

**Założenie:** znamy przybliżenie krawędzi (np. z obrazu o małej rozdzielczości).

Wzdłuż wstępnej aproksymacji krawędzi szukamy najbliższych lokalnych elementów krawędzi o podobnej orientacji (kierunku gradientu). Jeśli jest ich dostatecznie dużo, to przeprowadzamy aproksymację odpowiednim wielomianem w celu uzyskania ostatecznej krawędzi.

## Metoda korelacji w przestrzeni krawędzi

**Założenie:** mamy wzorzec (szablon) krawędzi.

Badamy zgodność lokalnych elementów krawędzi z szablonem:

- czy kierunek gradientu otrzymanego z lokalnego operatora jest prostopadły do modelowej krawędzi
- czy moduł gradientu przekracza zadany próg.

Liczba dopasowanych punktów decyduje o zgodności ze wzorcem bieżącego fragmentu krawędzi.

## Metoda kolejnych podziałów

**Założenie:** Krzywizna krawędzi jest mała.

Postępowanie rekurencyjne:

Wzdłuż symetralnej odcinka łączącego dwa punkty należące do aktualnego przybliżenia krawędzi szukamy lokalnego elementu krawędzi. Jeżeli jest dostatecznie blisko, to dołączamy go do nowego przybliżenia.



Warunek stopu: wielkość odchylenia otrzymanych punktów od prostej lub długość otrzymanych odcinków.

# Transformacja Hougha

U.S. Patent 3,069,654: "Method and means for recognizing complex patterns" (1962)

## Założenia:

- mamy informacje (lub czynimy założenia) o kształcie krawędzi,
- możliwe jest dokonanie prostej parametryzacji krzywej opisującej kształt krawędzi (w ogólnym przypadku – linii).

## Zalety:

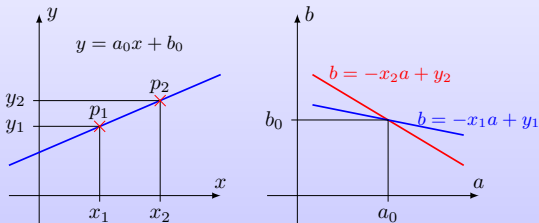
- mała wrażliwość na nieciągłości obrazu linii,
- mała wrażliwość na zaszumienie obrazu,
- prosta realizacja programowa.

## Transformacja Hougha

- Metoda detekcji kształtów w obrazie poprzez stopniowe kumulowanie wiarygodności.
- Na wejściu podawany jest obraz z wykrytymi punktami konturów poprzez zastosowanie detekcji krawędzi. Następnie poddawany jest transformacie.
- Woryginale służy do wykrywania prostych, później uogólniono na wykrywanie kształtów dających się opisać analitycznie, np. okręgów, dowolnych kształtów.

# Wykrywanie prostych

$$y = ax + b$$



$$p_1(x_1, y_1) \quad y_1 = ax_1 + b \quad b = -x_1a + y_1$$

$$p_2(x_2, y_2) \quad y_2 = ax_2 + b \quad b = -x_2a + y_2$$

$$(a_0, b_0) \quad y = a_0x + b_0 \quad \begin{cases} y_1 = a_0x_1 + b_0 \\ y_2 = a_0x_2 + b_0 \end{cases}$$



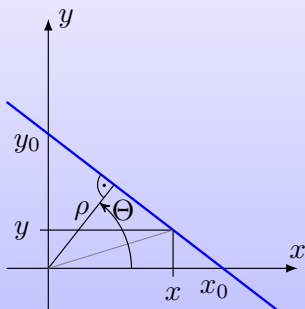
## Wykrywanie prostych

- Równanie kierunkowe pozwala na przedstawienie dowolnej prostej.
- Problem dla prostych pionowych lub niemalże pionowych – współczynnik kierunkowy przyjmuje nieskończoną wartość.
- Nieznane są zakresy wartości jakie mogą przyjmować zarówno współczynnik kierunkowy jak i przesunięcie.

Rozwiązanie: reprezentacja prostej przy pomocy równania kierunkowego.

## Lepsza parametryzacja prostej

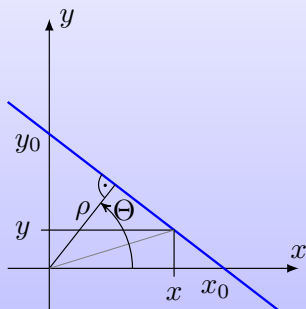
$$\rho = x \sin \Theta + y \cos \Theta$$



$$(x, y) \rightarrow (\rho, \Theta)$$

## Lepsza parametryzacja prostej

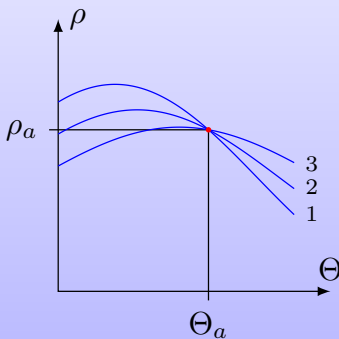
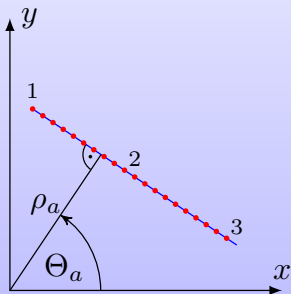
$$\rho = x \sin \Theta + y \cos \Theta$$



$$(x, y) \rightarrow (\rho, \Theta)$$

**Ważna cecha:** dla obrazu o ograniczonych rozmiarach współrzędne  $\rho$  i  $\Theta$  są ograniczone.

Wykrywanie linii opiera się o głosowanie. Pixel oddaje głos – dodaje jednostkową liczbę do wartości punktów w zbiorze prostych, które reprezentują proste przechodzące przez ten pixel. Wynikowy zbiór zawiera maksima, reprezentujące wykryte proste w oryginalnym obrazie.



## Algorytm transformaty Hougha – linia prosta

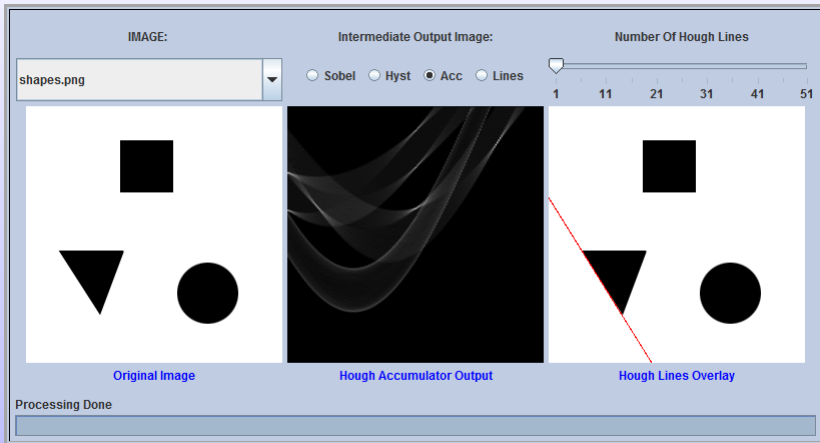
- 1 Wyznacz zakres parametrów  $\rho_{min}$ ,  $\rho_{max}$ ,  $\Delta\rho$  oraz  $\Theta_{min}$ ,  $\Theta_{max}$ ,  $\Delta\Theta$ .
- 2 Dla każdej wartości  $\Theta_{min} < \Theta_i < \Theta_{max}$  co  $\Delta\Theta$  wyznacz  $\rho$

$$\rho = x \cos \Theta_i + y \sin \Theta_i$$

i inkrementuj odpowiednią komórkę histogramu

$$H(\rho, \Theta_i) = H(\rho, \Theta_i) + 1$$

- 3 Wyznacz elementy transformaty o największych wartościach.



[http://users.ecs.soton.ac.uk/msn/book/new\\_demo/hough/](http://users.ecs.soton.ac.uk/msn/book/new_demo/hough/)

IMAGE: shapes.png

Intermediate Output Image:  Sobel  Hyst  Acc  Lines

Number Of Hough Lines: 1 11 21 31 41 51

Original Image      Hough Accumator Output      Hough Lines Overlay

Processing Done

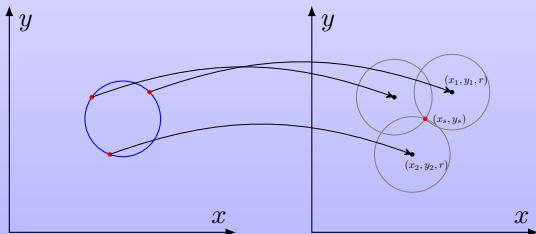
[http://users.ecs.soton.ac.uk/msn/book/new\\_demo/hough/](http://users.ecs.soton.ac.uk/msn/book/new_demo/hough/)

## Wykrywanie okręgów

Okrąg o promieniu  $r$  i środku  $(x_s, y_s)$

$$(x - x_s)^2 + (y - y_s)^2 = r^2$$

zwiększane będą w przestrzeni parametrów punkty  $(x_s, y_s, r)$  leżące na powierzchni stożka. Dla ustalonego  $r$  w przestrzeni parametrów  $(x_s, y_s)$  otrzymamy okrąg.





## Wykrywanie okręgów

- 1 Wyznacz obraz krawędziowy.
- 2 Dla każdego punktu krawędziowego
  - inkrementacja komórki  $(x_s, y_s)$  transformaty Hougha na podstawie  $(x, y)$ , ustalonego  $r$  oraz dla wszystkich wartości kąta  $\Theta \in (0, 360)$ .
- 3 Wyznacz elementy transformaty o największych wartościach.

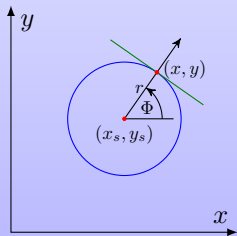
## Wykrywanie okręgów

Okrąg o promieniu  $r$  i środku  $(x_s, y_s)$

$$(x - x_s)^2 + (y - y_s)^2 = r^2.$$

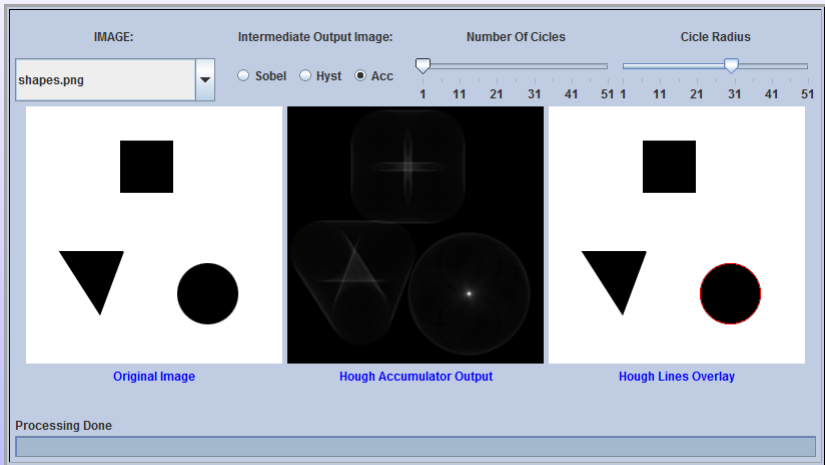
**Wykorzystanie kierunku gradientu:** zwiększane będą tylko punkty leżące w kierunku zgodnym z kierunkiem gradientu  $\Phi$  w odległości  $r$  od punktu  $(x, y)$

$$\begin{cases} x_s = x - r \cos \Phi \\ y_s = y - r \sin \Phi \end{cases}$$



## Wykrywanie okręgów – wersja 2

- 1 Wyznacz obraz krawędziowy wraz z kierunkami gradientów krawędzi.
- 2 Dla każdego punktu krawędziowego
  - wiedząc, że prosta prostopadła do kierunku krawędzi koła przechodzi przez środek okręgu o danym promieniu  $r$  wyznaczamy środek  $(x_s, y_s)$
  - inkrementacja komórki  $(x_s, y_s)$  transformaty Hougha na podstawie  $(x, y)$ , ustalonego  $r$  oraz dla kierunku zgodnego z kierunkiem gradientu  $\Phi$ .
- 3 Wyznacz elementy transformaty o największych wartościach.



[http://users.ecs.soton.ac.uk/msn/book/new\\_demo/houghCircles/](http://users.ecs.soton.ac.uk/msn/book/new_demo/houghCircles/)

IMAGE: shapes.png

Intermediate Output Image:  Sobel  Hyst  Acc

Number Of Cicles: 1 11 21 31 41 51

Cicle Radius: 1 11 21 31 41 51

The screenshot shows a software interface for Hough transform processing. It features three main panels: 'Original Image' on the left, 'Hough Accumulator Output' in the center, and 'Hough Lines Overlay' on the right. The 'Original Image' panel displays a white background with three black shapes: a square at the top, an inverted triangle at the bottom left, and a circle at the bottom right. The 'Hough Accumulator Output' panel shows a black background with three corresponding bright, semi-transparent regions: a cross-like shape for the square, a semi-transparent inverted triangle for the triangle, and a semi-transparent circle for the circle. The 'Hough Lines Overlay' panel shows the original shapes with red lines overlaid on them, representing the detected Hough lines. The interface also includes a file selection dropdown set to 'shapes.png', radio buttons for 'Sobel', 'Hyst', and 'Acc' (with 'Acc' selected), and two sliders for 'Number Of Cicles' and 'Cicle Radius', both set to 31. A 'Processing Done' status bar is at the bottom.

Original Image

Hough Accumulator Output

Hough Lines Overlay

Processing Done

[http://users.ecs.soton.ac.uk/msn/book/new\\_demo/houghCircles/](http://users.ecs.soton.ac.uk/msn/book/new_demo/houghCircles/)

IMAGE: shapes.png

Intermediate Output Image:  Sobel  Hyst  Acc

Number Of Cycles: 1 11 21 31 41 51

Cycle Radius: 11 21 31 41 51

Original Image

Hough Accumator Output

Hough Lines Overlay

Processing Done

[http://users.ecs.soton.ac.uk/msn/book/new\\_demo/houghCircles/](http://users.ecs.soton.ac.uk/msn/book/new_demo/houghCircles/)

IMAGE: shapes.png

Intermediate Output Image:  Sobel  Hyst  Acc

Number Of Cicles: 1 11 21 31 41 51

Cicle Radius: 11 21 31 41 51

The screenshot shows a software interface for Hough transform processing. It features three main panels: 'Original Image' on the left, 'Hough Accumulator Output' in the center, and 'Hough Lines Overlay' on the right. The 'Original Image' panel displays three black shapes on a white background: a square at the top, an inverted triangle at the bottom left, and a circle at the bottom right. The 'Hough Accumulator Output' panel shows the corresponding Hough space for these shapes, with bright spots indicating the parameters of the shapes. The 'Hough Lines Overlay' panel shows the original shapes with red lines overlaid, representing the detected Hough lines. The interface includes a file selection dropdown set to 'shapes.png', radio buttons for 'Sobel', 'Hyst', and 'Acc' (selected), and two sliders for 'Number Of Cicles' and 'Cicle Radius'.

Original Image

Hough Accumulator Output

Hough Lines Overlay

Processing Done

[http://users.ecs.soton.ac.uk/msn/book/new\\_demo/houghCircles/](http://users.ecs.soton.ac.uk/msn/book/new_demo/houghCircles/)

IMAGE: shapes.png

Intermediate Output Image:  Sobel  Hyst  Acc

Number Of Cicles: 1 11 21 31 41 51

Cicle Radius: 11 21 31 41 51

The screenshot shows a software interface for Hough transform processing. It features three main panels: 'Original Image' containing a square, an inverted triangle, and a circle; 'Hough Accumator Output' showing the corresponding sinusoidal curves for each shape; and 'Hough Lines Overlay' where the detected lines are overlaid on the original shapes. The interface includes a file selector set to 'shapes.png', radio buttons for 'Sobel', 'Hyst', and 'Acc' (selected), and two sliders for 'Number Of Cicles' and 'Cicle Radius'.

Original Image

Hough Accumator Output

Hough Lines Overlay

Processing Done

[http://users.ecs.soton.ac.uk/msn/book/new\\_demo/houghCircles/](http://users.ecs.soton.ac.uk/msn/book/new_demo/houghCircles/)



## Uogólnienie dla dowolnego kształtu

Wymaga stworzenia odpowiedniego modelu wykrywanego obiektu.

$$\Psi(X, A) = 0,$$

gdzie  $X$  jest wektorem na obrazie  $(x, y)$  a  $A$  jest wektorem w przestrzeni parametrów.

### Przykład:

- linia prosta o parametrach  $\rho, \Theta$

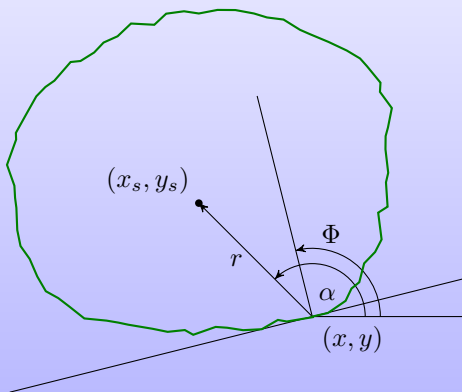
$$\rho = x \sin \Theta + y \cos \Theta; \quad X = (x, y), A = (\rho, \Theta)$$

- okrąg o trzech parametrach  $x_s, y_s, r$

$$(x - x_s)^2 + (y - y_s)^2 = r^2; \quad X = (x, y), A = (x_s, y_s, r)$$

## Uogólnienie dla dowolnego kształtu

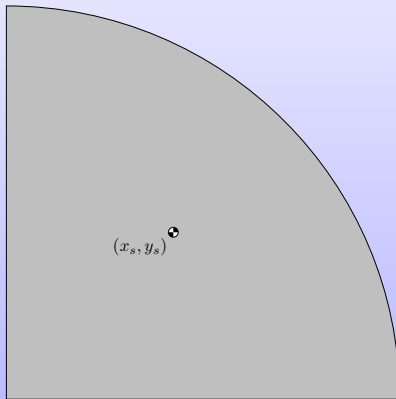
Parametryzacja kształtu krawędzi przy pomocy R–tablicy opisującej promień wodzący i kąt biegunowy funkcji kierunku gradientu



# Przykład

Faza I – Konstruowanie R-tablicy

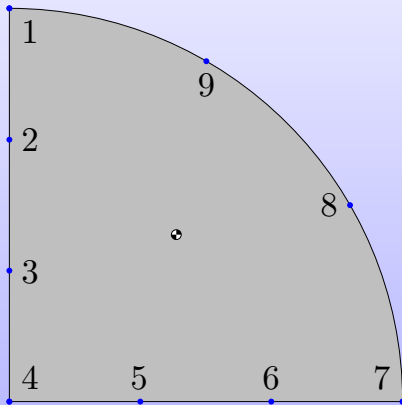
Określenie współrzędnych środka sylwetki.



# Przykład

Faza I – Konstruowanie R-tablicy

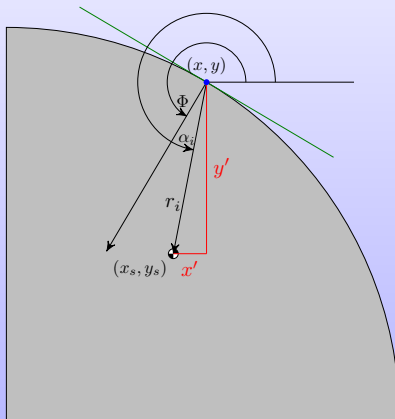
Wybranie pikseli na krawędzi obiektu.



## Przykład

Faza I – Konstruowanie R–tablicy

Dla punktu  $(x, y)$  wyznaczyć kierunek gradientu  $\Phi$ ,  $r_i$  oraz  $\alpha_i$ .



## Przykład

Faza I – Konstruowanie R–tablicy

Na podstawie wyznaczonych parametrów zbudować R–tablicę.

$\Phi_1$	$(r_1^1, \alpha_1^1), (r_2^1, \alpha_2^1), (r_3^1, \alpha_3^1), \dots$
$\Phi_2$	$(r_1^2, \alpha_1^2), (r_2^2, \alpha_2^2), (r_3^2, \alpha_3^2), \dots$
$\vdots$	
$\Phi_n$	$(r_1^n, \alpha_1^n), (r_2^n, \alpha_2^n), (r_3^n, \alpha_3^n), \dots$

$\Phi = 0$	$(r_2, \alpha_2), (r_3, \alpha_3)$
$\Phi = 45$	$(r_4, \alpha_4)$
$\Phi = 90$	$(r_5, \alpha_5), (r_6, \alpha_6)$
$\Phi = 135$	$(r_7, \alpha_7)$
$\Phi = 210$	$(r_8, \alpha_8)$
$\Phi = 240$	$(r_9, \alpha_9)$
$\Phi = 315$	$(r_1, \alpha_1)$

## Przykład

### Faza II – Detekcja

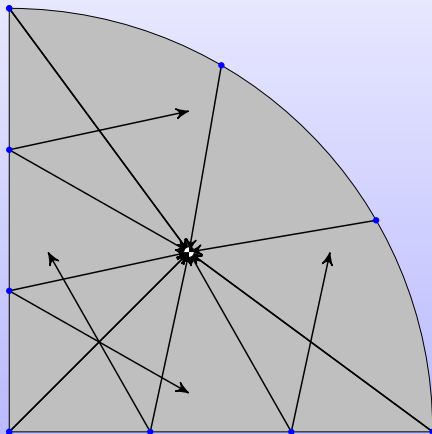
- 1 Dla punktu  $(x, y)$  w obrazie krawędzi wyznaczyć kierunek gradientu  $\Phi$ .
- 2 Przy pomocy R–tablicy wyznaczyć możliwe położenia środka sylwetki.

$$\begin{cases} x_s = x + r_i(\Phi) \cos(\alpha_i(\Phi)) \\ y_s = y + r_i(\Phi) \sin(\alpha_i(\Phi)) \end{cases} \quad A = (x_s, y_s)$$

- 3 Inkrementacja komórki  $(x_s, y_s)$  transformaty Hougha.

# Przykład

## Faza II – Detekcja





## R–tablica dla dowolnej orientacji i skali

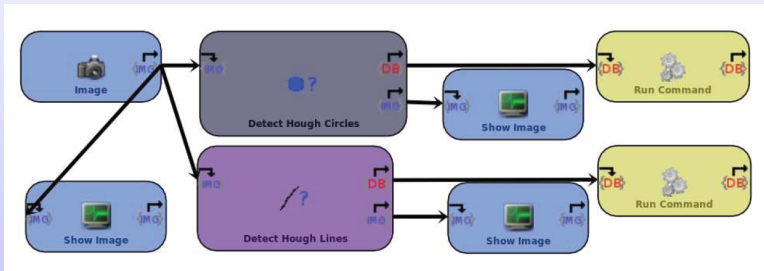
Dla czterowymiarowej przestrzeni parametrów

$$A = (x_s, y_s, S, \Theta),$$

gdzie  $S$  oznacza współczynnik skali, a  $\Theta$  – kąt obrotu sylwetki względem położenia wzorcowego (w R–tablicy)

$$\begin{cases} x_s = x + r_i(\Phi)S \cos(\alpha_i(\Phi) + \Theta) \\ y_s = y + r_i(\Phi)S \sin(\alpha_i(\Phi) + \Theta) \end{cases}$$

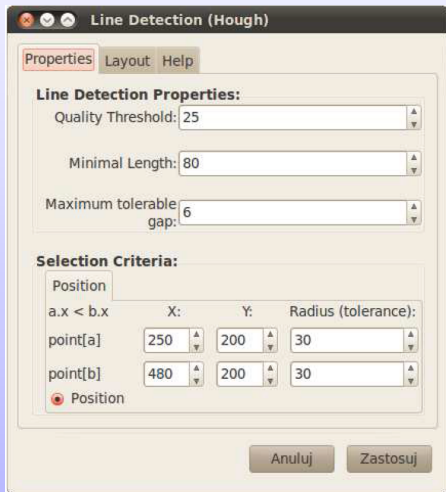
## Detektor Hougha w środowisku harpia



Wyjście bloku typu **IMG** zawiera obraz wejściowy z wykrytymi obiektami (naniesionymi kolorem czerwonym).

Wyjście typu **DB** podaje informację o fakcie wykrycia obiektów spełniających kryteria.

# Parametry dla detektora linii



# Parametry dla detektora okręgów

**Circle Detection (Hough)**

Properties Layout Help

**Circle Detection Parameters:**

Resolution (dp):

Minimal distance between two circles:

Threshold for edge detector:

Quality Threshold for detected circles:

**Selection Criteria**

Location + Radius **Number of Circles**

	Min X:	Max X:	Min Y:	Max Y:
Position:	<input type="text" value="0"/>	<input type="text" value="500"/>	<input type="text" value="0"/>	<input type="text" value="500"/>
	Min:		Max:	
Radius:	<input type="text" value="0"/>		<input type="text" value="500"/>	

Enable

Anuluj Zastosuj

## Parametry dla detektora okręgów

