

## 7. DIAGRAMY VORONOI'A

---

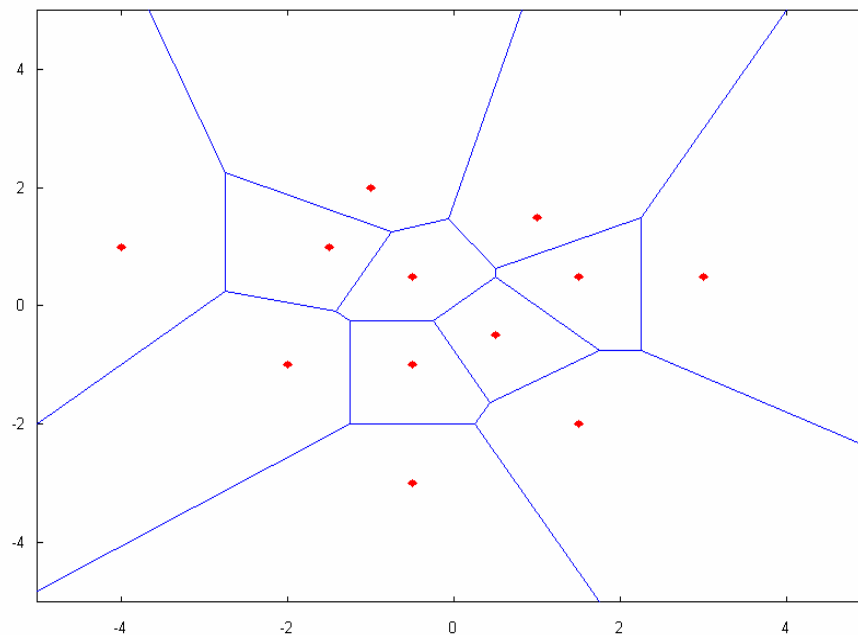
Niniejsze ćwiczenie ma na celu zapoznanie studentów z tytułowymi diagramami Voronoi'a. Mając do dyspozycji punkty wejściowe należy je w ten sposób przetworzyć, aby w pliku wyjściowym znalazły się wierzchołki odcinków otaczający dany punkt oraz sam punkt.

W ćwiczeniu wykorzystywany jest program **voronoi**, który jest odniesieniem do implementacji własnego algorytmu wyznaczania diagramów Voronoi'a.

### Wprowadzenie teoretyczne

---

Diagramy Voronoi'a są to komórki, które wewnątrz posiadają tylko jeden punkt wejściowy. Krawędź danego diagramu jest w równej odległości od dwóch sąsiednich punktów wejściowych, tzn. że linia oddzielająca dwa punkty leży w połowie odległości między nimi. Przykład diagramów Voronoi'a znajduje się poniżej (rysunek 1):



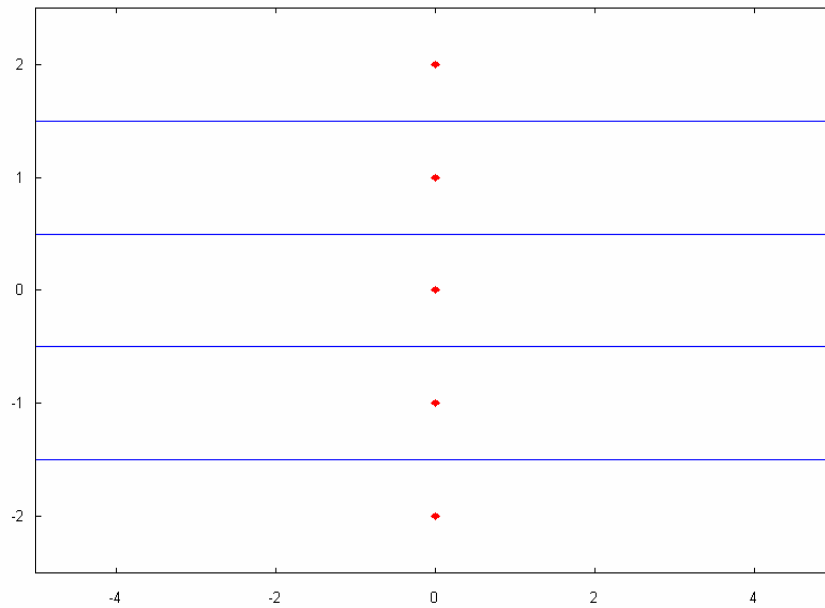
Rysunek 1

Wyznaczanie diagramów Voronoi'a możemy podzielić na trzy przypadki:

1. Liczba punktów wejściowych jest równa 1.
2. Liczba punktów wejściowych jest równa 2 lub wszystkie punkty wejściowe leżą na jednej linii.
3. Liczba punktów wejściowych jest równa lub większa od 3.

W przypadku pierwszym nie możemy wyznaczyć diagramów Voronoi'a, ponieważ na płaszczyźnie znajduje się tylko jeden punkt. Można powiedzieć, że istnieje tylko jedna komórka, która posiada swoje granice w nieskończoności.

Przypadek drugi jest już bardziej złożony. Punkty leżące na jednej linii mają krawędzie oddzielające je, które są równoległe i nieskończenie długie (rysunek 2)



Rysunek 2

W przypadku tym, aby wyznaczyć proste oddzielające punkty wejściowe, należy określić punkt leżący w połowie odległości między nimi.

Niech punktami wejściowymi będą punkty  $P_1$  i  $P_2$ , a punktem leżącym między nimi  $S$  takimi, że:

$$P_1 = (x_1, y_1);$$

$$P_2 = (x_2, y_2);$$

$$S = (x_s, y_s);$$

Wówczas:

$$x_s = \frac{x_1 + x_2}{2};$$

$$y_s = \frac{y_1 + y_2}{2};$$

Następnym krokiem jest określenie prostej prostopadłej do prostej przechodzącej przez punkty  $P_1$  i  $P_2$ , do której należy punkt  $S$ .

Nich prostą przechodzącą przez punkty  $P_1$  i  $P_2$  będzie prosta  $l$ .

$$l: y = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) + y_1;$$

gdzie

$$a = \frac{y_2 - y_1}{x_2 - x_1};$$

Prosta prostopadła do prostej  $l$  to prosta  $k$ :

$$k: y = cx + d;$$

która jest związana z prostą l następującą zależnością:

$$ac = -1;$$

$$c = \frac{-1}{a};$$

$$c = \frac{x_1 - x_2}{y_1 - y_2};$$

Do wyznaczenia niewiadomej d skorzystamy z faktu, że do prostej k należy punkt S, tzn. że spełnia równanie tej prostej:

$$S \in k;$$

$$y_S = cx_S + d;$$

$$d = y_S - cx_S;$$

$$d = y_S - x_S \frac{x_1 - x_2}{y_1 - y_2};$$

Ostatecznie prosta k ma postać:

$$k : y = \frac{x_1 - x_2}{y_1 - y_2} (x - x_S) + y_S;$$

Zauważmy, że w przypadku tym  $y_1$  musi być różny od  $y_2$ .

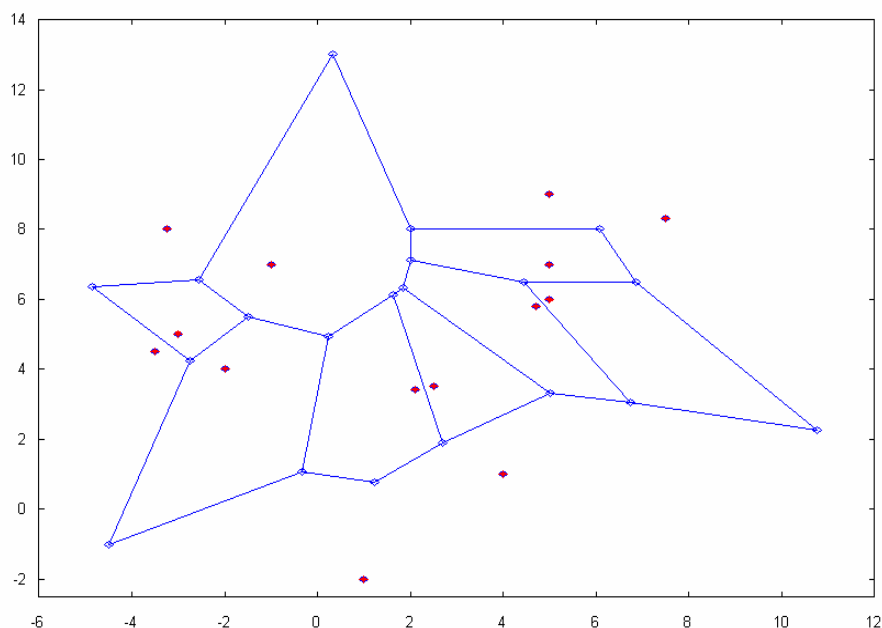
Jeżeli okaże się, że punkty  $P_1$  i  $P_2$  mają taką samą współrzędną x lub y (jednocześnie nie zajdzie przypadek gdy  $x_1=x_2$  i  $y_1=y_2$ , bo wówczas oznaczałoby to, że mamy do czynienia z tym samym punktem.), to wtedy prosta k przyjmuje postać:

$$k : y = y_S \quad \text{dla } x_1 = x_2;$$

$$k : x = x_S \quad \text{dla } y_1 = y_2;$$

Ostatnim przypadkiem jest przypadek, gdy liczba punktów wejściowych jest równa bądź większa od trzech. W tym wypadku najprościej jest skorzystać z metody „pustych okręgów”.

Metoda pustych okręgów polega na wyznaczeniu wszystkich możliwych okręgów na bazie trzech dowolnych punktów wejściowych. Gdy określimy równanie okręgu należy sprawdzić czy jakiś inny punkt wejściowy nie leży wewnątrz tego okręgu. Jeżeli okaże się, że okrąg jest pusty to jego środek wyznaczy nam punkt leżący w równej odległości od trzech punktów wejściowych. Postępując w ten sposób dla wszystkich punktów wejściowych, otrzymamy dla każdego punktu pewien zbiór środków okręgów. Łącząc je otrzymujemy krawędzie diagramów Voronoi'a. Rysunek 3 ukazuje ten przypadek.



Rysunek 3

Wykres powyżej ukazuje pewną niedoskonałość tej metody. Punkty wejściowe leżące na granicy obrazu nie mają określonych krawędzi, które by je rozdzielały. Ten problemem został rozwiązany i opisany w dalszej części tekstu. W tym miejscu chciałbym zająć się wyznaczeniem środka okręgu, zbudowanego na bazie trzech punktów wejściowych.

Niech punktami wejściowymi będą punkty  $P_1$ ,  $P_2$  i  $P_3$  takie, że:

$$P_1 = (x_1, y_1);$$

$$P_2 = (x_2, y_2);$$

$$P_3 = (x_3, y_3);$$

a środkiem okręgu punkt  $S$ :

$$S = (x_s, y_s);$$

Równanie okręgu ma postać:

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

gdzie  $r$  jest promieniem okręgu.

Punkty  $P_1$ ,  $P_2$ ,  $P_3$  należą do tego okręgu, czyli spełniają powyższe równanie.

$$(x_1 - x_s)^2 + (y_1 - y_s)^2 = r^2;$$

$$(x_2 - x_s)^2 + (y_2 - y_s)^2 = r^2;$$

$$(x_3 - x_s)^2 + (y_3 - y_s)^2 = r^2;$$

Mamy trzy równania i trzy niewiadome, tak więc możliwe jest określenie szukanych wartości  $x_s$ ,  $y_s$ ,  $r^2$ .

Gdy od pierwszego i trzeciego równania odejmiemy drugie równanie i przekształcimy je odpowiednio to otrzymamy:

$$2x_s(x_2 - x_1) + 2y_s(y_2 - y_1) = x_2^2 - x_1^2 + y_2^2 - y_1^2;$$

$$2x_s(x_2 - x_3) + 2y_s(y_2 - y_3) = x_2^2 - x_3^2 + y_2^2 - y_3^2;$$

Stosując teraz metodę wyznaczników mamy:

$$W = 4(x_2 - x_1)(y_2 - y_3) - 4(x_2 - x_3)(y_2 - y_1);$$

$$W_{x_s} = 2(y_2 - y_3)(x_2^2 - x_1^2 + y_2^2 - y_1^2) - 2(y_2 - y_1)(x_2^2 - x_3^2 + y_2^2 - y_3^2);$$

$$W_{y_s} = 2(x_2 - x_1)(x_2^2 - x_3^2 + y_2^2 - y_3^2) - 2(x_2 - x_3)(x_2^2 - x_1^2 + y_2^2 - y_1^2);$$

czyli

$$x_s = \frac{W_{x_s}}{W};$$

$$y_s = \frac{W_{y_s}}{W};$$

$$r^2 = (x_1 - x_s)^2 + (y_1 - y_s)^2;$$

Wyznaczenie niewiadomych  $x_s$ ,  $y_s$ ,  $r^2$  jest możliwe gdy  $W \neq 0$ . Przypadek, gdy  $W = 0$  występuje wówczas, gdy trzy punkty wejściowe mają którąś ze współrzędnych równą (punkty leżą w jednej linii).

Kolejnym krokiem jest odpowiednie uporządkowanie otrzymanych punktów (środków okręgów). W tym celu należy przejść na współrzędne biegunowe, ale względem punktu wejściowego, do którego owe środki okręgów należą (współrzędne danego punktu wejściowego są współrzędnymi początku układu współrzędnych dla jego środków okręgów). Następnie porządkujemy punkty zgodnie z regułą – punkt o największym kącie znajduje się na początku zbioru a o najmniejszym kącie na końcu. Podczas implementacji okazało się, że konieczna jest również zmiana kąta środków okręgów uzależniona położeniem punktu wejściowego względem innych punktów wejściowych. W tym celu określany jest „wirtualny” środek układu. Wyznamy go w ten sposób, że współrzędna  $x$  jest środkiem odległości między punktami o największej i najmniejszej współrzędnej  $x$ , a współrzędna  $y$  w analogiczny sposób dla punktów o największej i najmniejszej współrzędnej  $y$ . Następnie przechodzimy na współrzędne biegunowe ze wszystkimi punktami wejściowymi tak jakby środek układu współrzędnych leżał w „wirtualnym” środku. Na tej podstawie określimy zmienną offset, która jest odejmowana od środków okręgów (każdy punkt ma swój własny offset).

Określenie zmiennej offset:

$$\frac{3\pi}{4} < kat < \frac{\pi}{4} \Rightarrow offset = \frac{\pi}{2};$$

$$\frac{5\pi}{4} < kat < \frac{3\pi}{4} \Rightarrow offset = \pi;$$

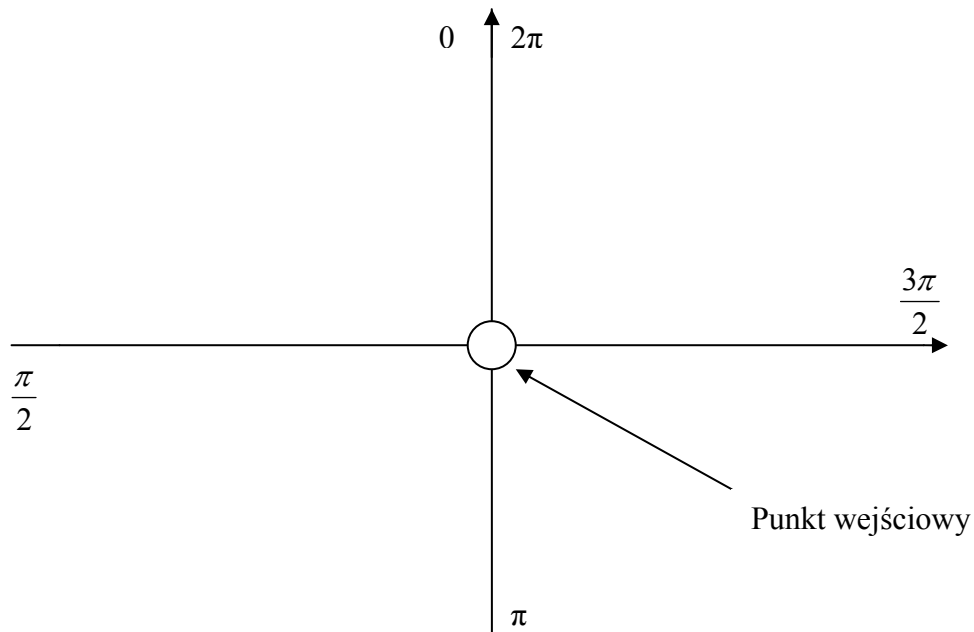
$$\frac{5\pi}{4} < kat < \frac{7\pi}{4} \Rightarrow offset = \frac{3\pi}{2};$$

$$kat > \frac{7\pi}{4} \vee kat < \frac{\pi}{4} \Rightarrow offset = 0;$$

$$kat = \left\{ \frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4} \right\} \Rightarrow offset = kat;$$

Ostatecznie środek okręgu ma kąt jaki wynika z jego położenia względem punktu wejściowego pomniejszony o  $(2\pi - offset)$ . W ten sposób dla punktu wejściowego znajdującego się w górnej części obrazu środek okręgu mający najmniejszy kąt to środek

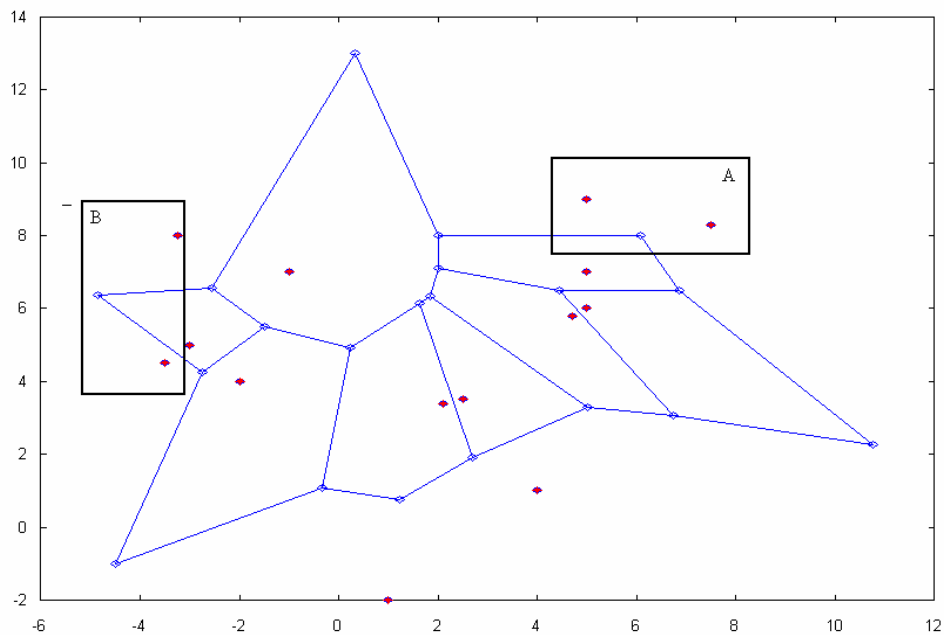
leżący na dodatniej półosi OY (względem punktu wejściowego). Sytuację tą ilustruje rysunek 4.



Rysunek 4

Zajmijmy się teraz punktami leżącymi na granicy obrazu i wyznaczeniem krawędzi rozdzielających je. Wyznaczenie owych krawędzi możemy podzielić na dwa przypadki:

1. Punkty wejściowe „widzą się” (rysunek 5, punkty A).
2. Punkty wejściowe są „zasłonięte” przez środek okręgu leżący między nimi (rysunek 5, punkty B).



Rysunek 5.

W przypadku pierwszym postępujemy analogicznie do jak w przypadku, gdy liczba punktów wejściowych jest równa 2 lub punkty wejściowe leżą w jednej linii. Oznacza to, że

wyznaczymy punkt  $S$  leżący na środku odległości między punktami, a następnie określamy prostą prostopadłą do prostej przechodzącej przez dwa punkty wejściowe, taką że punkt  $S$  należy do niej. Zauważmy, że punkt  $S$  może znajdować się na początku lub końcu zbioru środków okręgów. To kryterium wraz z faktem, że punkt  $S$  musi spełnić je dla jednego i drugiego punktu wejściowego, jest wykorzystywane do określenia, czy punkt  $S$  jest punktem środkowym punktów leżących na granicy obrazu.

Przypadek drugi jest trochę bardziej skomplikowany niż pierwszy. Zauważmy, że środek okręgu powstał z trzech punktów wejściowych, które tworzą trójkąt rozwartokątny. W czasie wyznaczania równania okręgu, gdy czynność ta się powiodła, należy sprawdzić czy punkty na bazie, których powstał dany okrąg nie tworzą trójkąta rozwartokątnego. Jeżeli zachodzi taki przypadek to wówczas punkty leżące na ramionach kąta rozwartego muszą zapamiętać współrzędne drugiego punktu oraz środek okręgu. Niech punktami tworzącymi okrąg i trójkąt rozwartokątny będą punkty:

$$P_1 = (x_1, y_1);$$

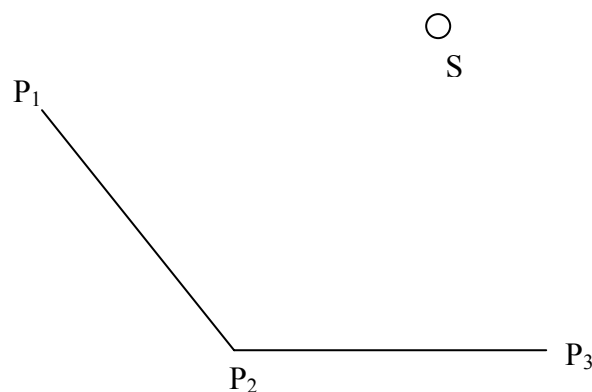
$$P_2 = (x_2, y_2);$$

$$P_3 = (x_3, y_3);$$

a środkiem okręgu

$$S = (x_S, y_S);$$

wówczas punkt  $P_1$  zapamiętuje w odpowiednim zbiorze punkty  $P_3$  i  $S$ , a punkt  $P_3$  punkty  $P_1$  i  $S$ .



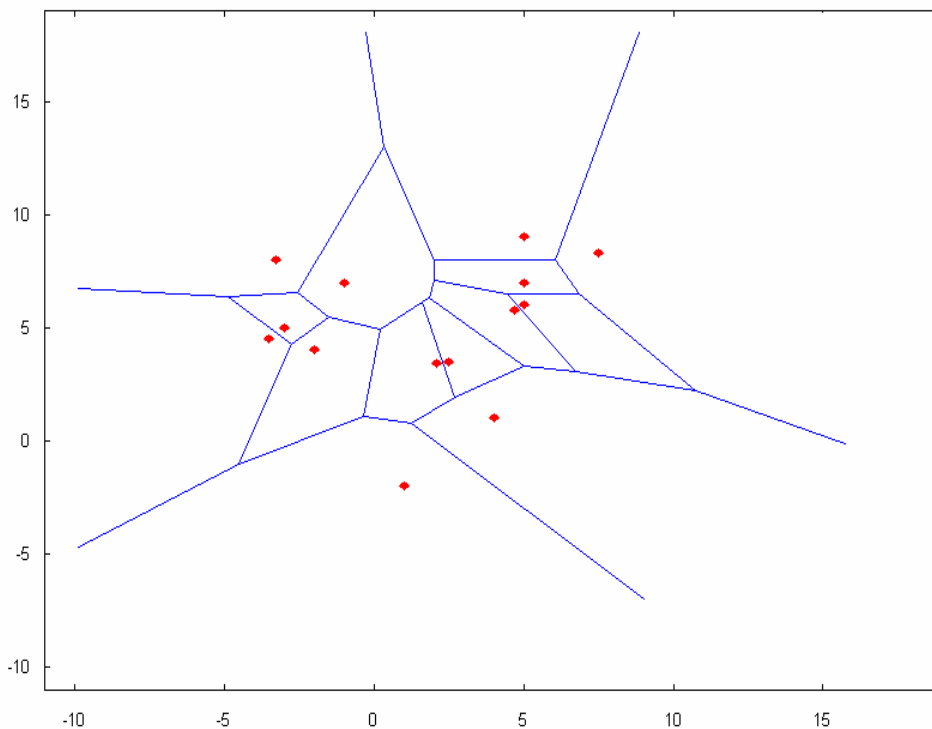
Rysunek 6.

Następnie, po wyznaczeniu wszystkich środków okręgów oraz punktów krawędzi z przypadku pierwszego, sprawdzamy dla wszystkich punktów wejściowych (tych, które tworzyły trójkąt rozwartokątny) czy zapamiętany punkt  $S$  znajduje się na początku lub na końcu zbioru wyznaczonych prędkiej punktów komórki. Jeżeli taka sytuacja ma miejsce wówczas wyznaczamy prostą prostopadłą do punktów  $P_1$  i  $P_2$ , taką że należy do niej punkt  $S$ .

Na zakończenie sprawdzamy, czy wszystkie punkty wejściowe spełniają następujące warunki:

- zawierają dwa punkty określające krawędzie;
- nie zawierają żadnych punktów określających krawędzie.

Jeżeli okaże się, że któryś z punktów posiada tylko jeden taki punkt, to wówczas należy go usunąć ze zbioru. Tu pojawia się problem, gdyż punkt ten należy usunąć również ze zbioru drugiego punktu wejściowego. Dlatego też punkt określający krawędź znajdujący się w zbiorze punktu  $P_1$  zapamiętuje punkt  $P_3$  i na odwrót. W ten sposób można określić ten drugi punkt wejściowy. Jednocześnie ze zbioru punktów zapamiętanych (tych, które tworzą trójkąt rozwartokątny) punktu  $P_1$  usuwamy punkty  $P_3$  i  $S$  i odwrotnie. Jeżeli wystąpiła operacja usuwania punktu to wówczas sprawdzamy pozostałe punkty zapamiętane, czy czasami one nie utworzą punktu krawędzi. Czynność tą powtarzamy tak długo, aż wszystkie punkty wejściowe nie będą spełniać powyższych dwóch warunków. Ostatecznie otrzymujemy wynik taki jak na rysunku 7.



Rysunek 7

Zauważmy, że punkty określające krawędzie dla punktów leżących na granicy obrazu ograniczają długość tychże krawędzi. Teoretycznie powinny one być nieskończenie długie, ale w praktyce stosuje się ich określoną długość, którą można ustalić.

W ten sposób otrzymaliśmy diagramy Voronoi'a dla zadanych punktów wejściowych.

## Obsługa programu:

---

Program wymaga podania pliku z punktami wejściowymi. W wyniku działania programu otrzymamy również plik zawierający współrzędne wyznaczonych punktów. Przykładowe wywołanie programu:

```
voronoi -i [nazwa_pliku_wejściowego] -o [nazwa_pliku_wyjściowego]
```



W przypadku, gdy nie podano nazwy pliku wejściowego lub nazwy pliku wyjściowego to domyślnie zostaną przyjęte odpowiednie nazwy `punkty.txt` i `wynik.vor`.

## Polecenia:

---

### 1. Ilustracja wyznaczania diagramów Voronoi'a

Przeprowadzić dla poniższych punktów wejściowych wyznaczenie diagramów Voronoi'a za pomocą programu **voronoi**.

a)

Punkty wejściowe:

1, 1

4, 3

2, 2

5, 4

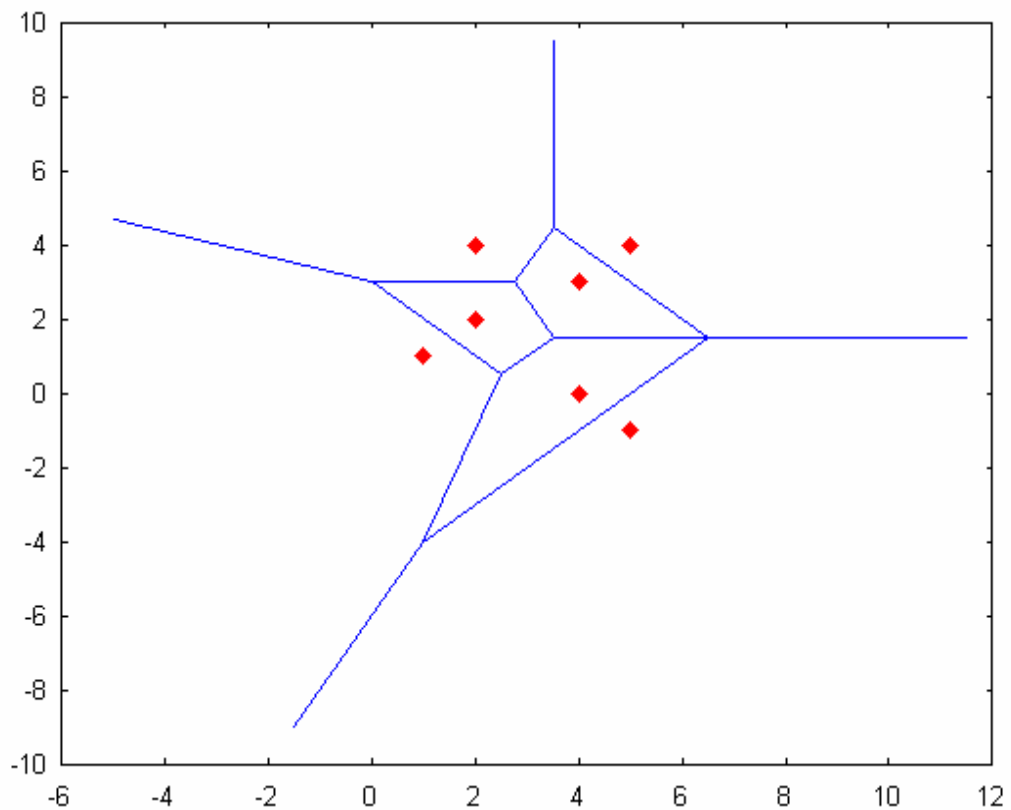
2, 4

2, 2

5, -1

4, 0

Wynik działania:

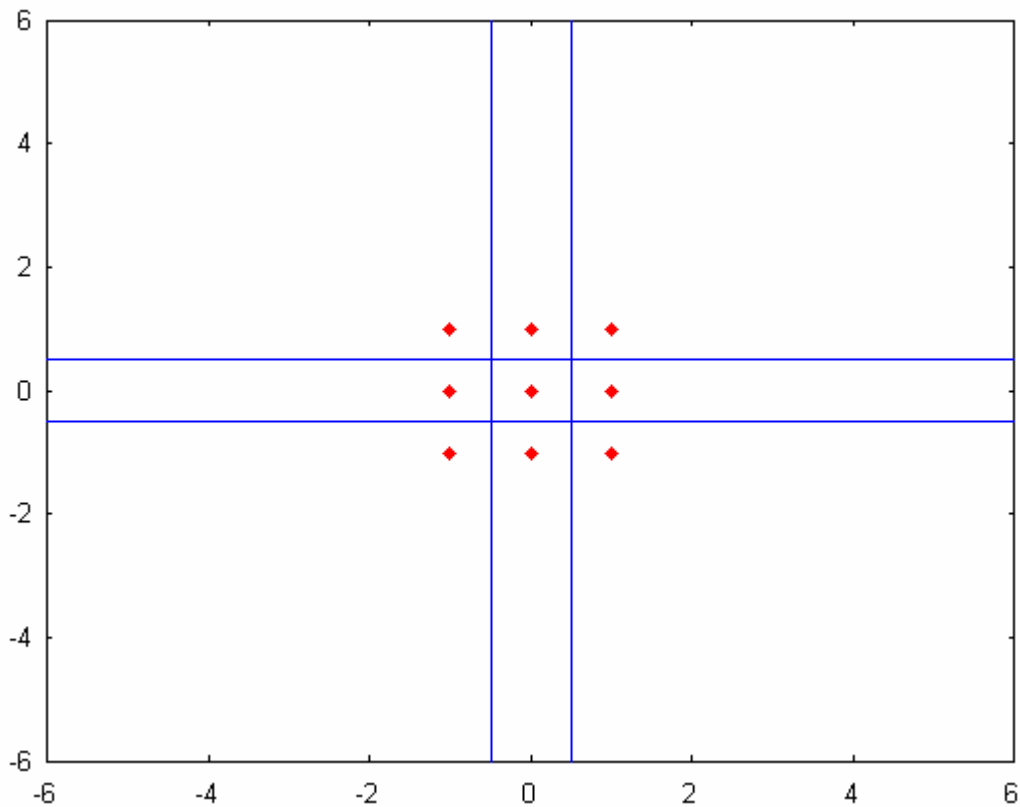


Przypadek ten jest ciekawy z tego względu, że wszystkie punkty wejściowe leżące na granicy obrazu nie „widzą się”.

b)

Punkty wejściowe:

-1, 1  
0, 1  
1, 1  
-1, 0  
0, 0  
1, 0  
-1, -1  
0, -1  
1, -1



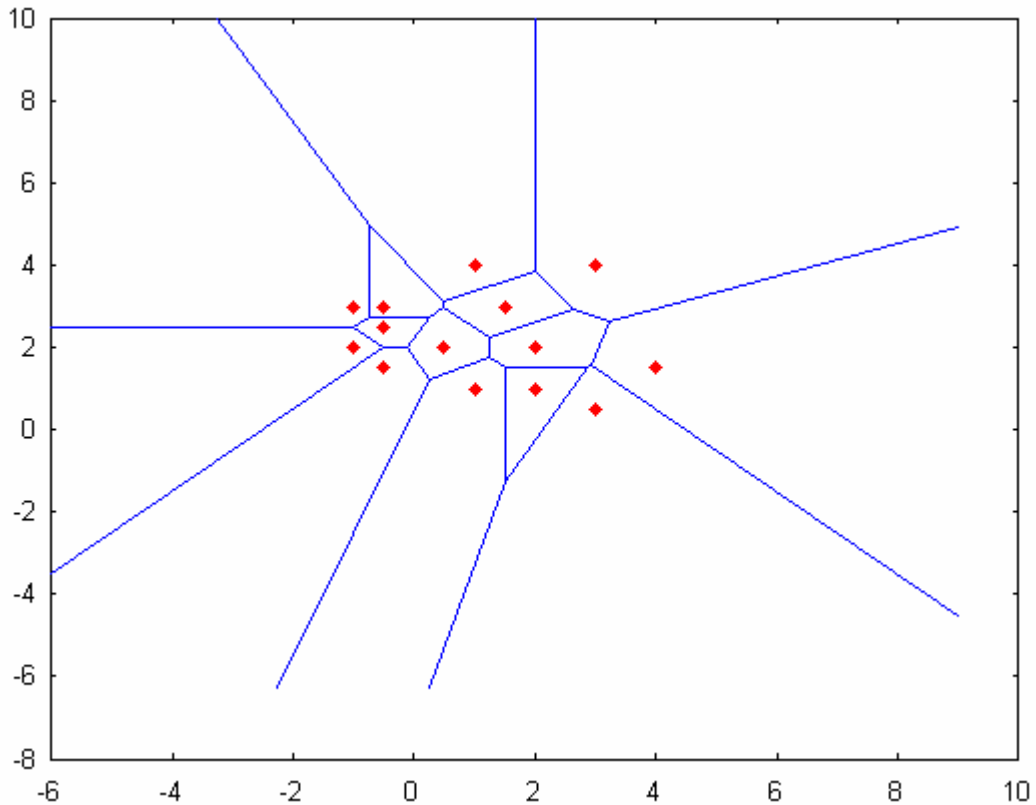
Przykład punktów regularnie ułożonych na płaszczyźnie.

c)

Punkty wejściowe:

1, 4  
3, 4  
-1, 3  
-0.5, 3  
1.5, 3  
-0.5, 2.5  
-1, 2

0.5, 2  
2, 2  
-0.5, 1.5  
1, 1  
2, 1  
4, 1.5  
3, 0.5



Przykład punktów o ułożeniu losowym.

## 2. Implementacja algorytmu wyznaczania diagramów Voronoi'a w Matlabie

Dokonać implementacji algorytmu wyznaczania diagramów Voronoi'a w Matlabie według opisu zawartego w części teoretycznej bazując na kodzie przedstawionym poniżej. Dokonać porównania wyników działania programu z programem **voronoi** dla punktów wejściowych zdefiniowanych w punkcie 1 zbioru poleceń. Wyjaśnić różnice w uzyskanych wynikach? Przygotować tak kod źródłowy programu, aby plik wynikowy można było prezentować w programie 'gnuplot'.

My\_vor, główna funkcja realizująca dzielenie płaszczyzny na komórki. Po zakończeniu obliczeń, funkcja wyświetla podział na ekranie, przy pomocy funkcji „plot”, oprócz tego zapisuje współrzędne punktów do pliku, dzięki czemu możemy je wyświetlić przy pomocy programu GnuPlot. Zmienna hm określa, który z przypadków chcemy użyć, są do wyboru (1-3) zbiory liczb podane w punkcie pierwszym, oraz (0) –generowane są przypadkowo liczby z przedziału 0-200, parametr hmt określa ile takich liczb ma być wygenerowane.

```
function [ vx, vy ] = my_vor(hm,hmt)
```

```

switch hm
case 0
    X = round(200 * rand(hmt,1));
    Y = round(200 * rand(hmt,1));
case 1
    X=[1; 4 ;2 ;5 ;2 ;2 ;5 ;4];
    Y=[1 ;3 ;2 ;4; 4 ;2; -1 ;0];

case 2
    X=[-1 ;0 ;1; -1; 0; 1 ;-1; 0 ;1];
    Y=[ 1 ;1 ;1;0 ;0 ;0 ;-1 ;-1; -1];
case 3
    X=[1; 3 ;-1 ;-0.5; 1.5 ;-0.5 ;-1; 0.5 ;2 ;-0.5 ;1; 2 ;4; 3];
    Y=[4; 4 ;3; 3 ;3; 2.5; 2; 2; 2 ;1.5 ;1; 1 ;1.5; 0.5 ];
end

function tri = delaunay ( X, Y );
    ls = '';

    xt = X(tri);
    yt = Y(tri);

    if ( size ( xt, 2 ) == 1 )
        xt = xt';
        yt = yt';
    end

    ot = xt(:,1) .* ( yt(:,2) - yt(:,3) ) + ...
        xt(:,2) .* ( yt(:,3) - yt(:,1) ) + ...
        xt(:,3) .* ( yt(:,1) - yt(:,2) );

    bt = find ( ot < 0 );
    tri(bt,[1 2]) = tri(bt,[2 1]);

    c = tr_cn ( tri, X, Y );

    n = numel ( X );
    t = repmat ( (1:size(tri,1))', 1, 3 );
    T = sparse ( tri, tri(:,[3 1 2]), t, n, n );

    E = ( T & T' ) .* T;
    F = xor ( T, T' ) .* T;

    [ i, j, v ] = find ( triu ( E ) );
    [ i, j, vv ] = find ( triu ( E' ) );

    vx = [ c(v,1) c(vv,1) ]';
    vy = [ c(v,2) c(vv,2) ]';

    [ i, j, z ] = find ( F );

    dx = X(j) - X(i);
    dy = Y(j) - Y(i);

    rx = max(X) - min(X);
    ry = max(Y) - min(Y);

    cx = ( max(X) + min(X) ) / 2 - c(z,1);
    cy = ( max(Y) + min(Y) ) / 2 - c(z,2);

    nm = sqrt ( rx .* rx + ry .* ry ) + sqrt ( cx .* cx + cy .* cy );

    scale = nm ./ sqrt ( ( dx .* dx + dy .* dy ) );

    ex = [ c(z,1) c(z,1) - dy .* scale ]';
    ey = [ c(z,2) c(z,2) + dx .* scale ]';

    vx = [ vx ex ];
    vy = [ vy ey ];

```

```

cax = gca;

if ( isempty(ls) )
    co = get(ancestor(cax, 'figure'), 'defaultaxescolororder');
    h1 = plot(X,Y, '.', 'color', co(1,:), 'parent', cax);
    h2 = line(vx,vy, 'color', co(1,:), 'parent', cax, 'yliminclud', 'off', ...
        'xliminclud', 'off' );
else
    [l,c,m,msg] = colstyle(ls);
    error(msg)
    if ( isempty(m) )
        m = 'none';
    end
    h1 = plot(X,Y,[c '.'], 'parent', cax);
    h2 = line(vx,vy, 'color', c, 'linestyle', l, 'marker', m, 'parent', ...
        cax, 'yliminclud', 'off', 'xliminclud', 'off' );
end

file = fopen('d.dat','w');
for(m=1:2:(numel(vx)-1))

    fprintf(file, '%d\t%d\n', vx(m), vy(m));
    fprintf(file, '%d\t%d\n', vx(m+1), vy(m+1));
    fprintf(file, '\n');
end
fclose(file);

end;

```

Tr\_cn, funkcja ta zwraca współrzędne środka okręgu opisanego na trójkącie.

```

function c = tr_cn ( tri, x, y )

x1 = x(tri(:,1));
x2 = x(tri(:,2));
x3 = x(tri(:,3));

y1 = y(tri(:,1));
y2 = y(tri(:,2));
y3 = y(tri(:,3));

a11 = x2 - x1;
a12 = y2 - y1;
a21 = x3 - x1;
a22 = y3 - y1;

idet = a11 .* a22 - a21 .* a12;

d = find ( idet == 0 );

if ( ~isempty ( d ) )

    delta = sqrt ( eps );
    x1(d) = x1(d) + delta * ( rand ( size ( d ) ) - 0.5 );
    x2(d) = x2(d) + delta * ( rand ( size ( d ) ) - 0.5 );
    x3(d) = x3(d) + delta * ( rand ( size ( d ) ) - 0.5 );
    y1(d) = y1(d) + delta * ( rand ( size ( d ) ) - 0.5 );
    y2(d) = y2(d) + delta * ( rand ( size ( d ) ) - 0.5 );
    y3(d) = y3(d) + delta * ( rand ( size ( d ) ) - 0.5 );

    a11 = x2 - x1;
    a12 = y2 - y1;
    a21 = x3 - x1;
    a22 = y3 - y1;

    idet = a11 .* a22 - a21 .* a12;

end

b1 = a11 .* (x2+x1) + a12 .* (y2+y1);
b2 = a21 .* (x3+x1) + a22 .* (y3+y1);

idet = 0.5 ./ idet;

```

```
xc = ( a22 .* b1 - a12 .* b2) .* idet;  
yc = (-a21 .* b1 + a11 .* b2) .* idet;  
c = [ xc yc ];
```

Do sprawozdania proszę dołączyć kod źródłowy w Matlabie oraz odpowiedzi na zadane pytania.

## Uwagi organizacyjne:

---

1. Katalogiem roboczym do umieszczania własnych plików jest **x:\**. Przed przystąpieniem do ćwiczeń należy usunąć wszystkie pliki z tego katalogu.
-