UNIVERSITATEA TEHNICĂ A MOLDOVEI
Facultatea Calculatoare, Informatică şi Microelectronică
Departamentul Informatică şi Ingineria Sistemelor

GRAFICA PE CALCULATOR

TEMA 9. ALGORITMI FUNDAMENTALI DE SINTEZĂ A IMAGINILOR

l. u., dr. NASTAS Andrei

CONȚINUT

9.3. Algoritmi de generare a elipselor
9.3.1. Calculul punctelor de pe o elipsă folosind ecuațiile parametrice ale elipsei

O elipsă este determinată prin centrul său, raza și mărimea celor două semiaxe paralele cu axele Ox și respectiv cu Oy.

Poate fi definită analitic în: Coordonate polare (prin ecuații parametrice).

Ecuațiile parametrice ale elipsei sunt:

$$x = xc + a \cdot \cos(t)\,,\quad y = yc + b \cdot \sin(t)\,,$$ unde $0 \le t \le 6{,}28\ (2 \cdot \pi)$, (9.25)

(xc, yc) este centrul elipsei,

a este mărimea semiaxei paralele cu Ox,

iar b este mărimea semiaxei paralele cu Oy.

Se poate obține o secvență de puncte care aproximează circumferința elipsei, dându-i lui t valori de la 0 la 6,28 cu un pas constant.

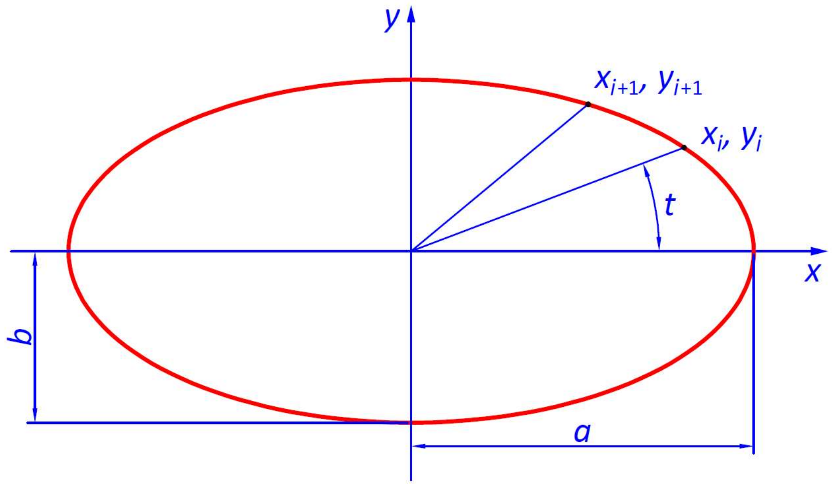

Fig. 9.7. Calculul punctelor de pe elipsă

9.3.1. Calculul punctelor de pe o elipsă folosind ecuațiile parametrice ale elipsei

Fie (x_i, y_i) ultimul punct calculat pe circumferința elipsei:

$$\begin{aligned} x_i &= xc + a \cdot \cos(t)\,, \\ y_i &= yc + b \cdot \sin(t)\,. \end{aligned} \tag{9.26}$$

Următorul punct (x_{i+1},y_{i+1}), este distanțat de cel curent cu pasul unghiular (notat cu pas):

$$\begin{aligned} x_{i+1} &= xc + a \cdot \cos(t + \mathrm{pas})\,, \\ y_{i+1} &= yc + b \cdot \sin(t + \mathrm{pas})\,. \end{aligned} \tag{9.27}$$

Notăm:

$dx = \cos(t)$ și $dy = \sin(t)$,

$c = \cos(\text{pas})$ și $s = \sin(\text{pas})$.

Știm că:

$\cos(t + \text{pas}) = \cos(t)\cdot\cos(\text{pas}) - \sin(t)\cdot\sin(\text{pas})$,

$\sin(t + \text{pas}) = \cos(t)\cdot\sin(\text{pas}) + \sin(t)\cdot\cos(\text{pas})$.

Iar

$$\begin{aligned} dx' &= \cos(t + \mathrm{pas}) = dx \cdot c - dy \cdot s, \\ dy' &= \sin(t + \mathrm{pas}) = dy \cdot c + dx \cdot s. \end{aligned} \tag{9.28}$$

Înlocuind în relațiile (9.27) pentru x_{i+1} și y_{i+1}, obținem:

$$\begin{aligned} x_{i+1} &= xc + a \cdot dx' = xc + a \cdot (dx \cdot c - dy \cdot s), \\ y_{i+1} &= yc + b \cdot dy' = yc + b \cdot (dy \cdot c + dx \cdot s). \end{aligned} \tag{9.29}$$

9.3.2. Generarea unei elipse rotite

Algoritmul prezentat mai înainte poate fi folosit pentru generarea elipselor care au axele paralele cu axele sistemului de coordonate carteziene 2D.

Fie *R* o elipsă cu centrul în originea sistemului de coordonate și semiaxele *a*, *b* rotite cu unghiul *u*, față de axele de coordonate.

Fie *E* elipsă cu centrul în origine și semiaxele *a* și *b* suprapuse peste axele sistemului de coordonate.

Punctele de pe elipsa *R* pot fi obținute rotind punctele de pe elipsa *E* în jurul originii cu unghiul *u*.

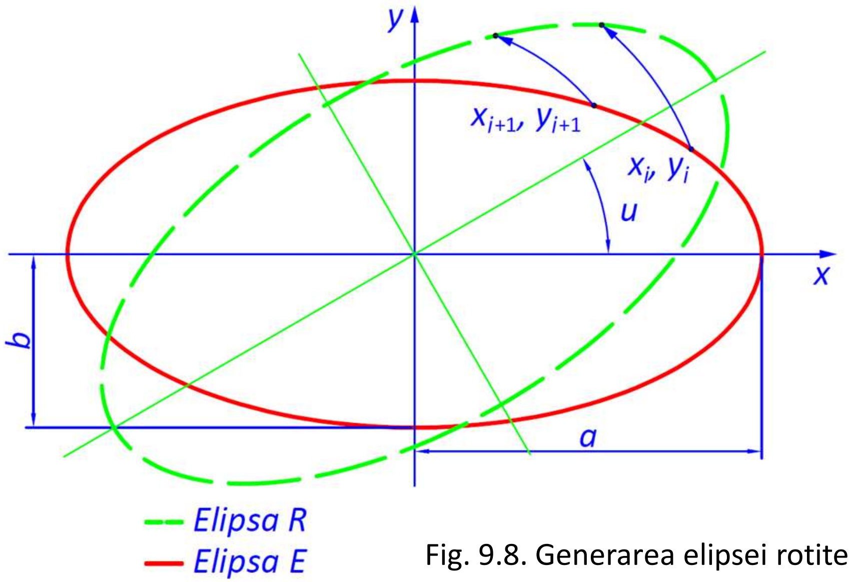

Fig. 9.8. Generarea elipsei rotite

9.3.2. Generarea unei elipse rotite

Dacă centrul elipsei *R* este în (*xc*, *yc*), atunci după rotație se aplică fiecărui punct translația cu (*xc*, *yc*). Fie:

$$x' = a \cdot dx', \qquad (9.30)$$
$$y' = b \cdot dy'.$$

punctul curent de pe elipsa *E*, unde [*dx*', *dy*'] s-a obținut prin aplicarea formulei de recurență (9.28).

Notăm:

$$dx_1 = a \cdot dx', \qquad (9.31)$$
$$dy_1 = b \cdot dy'.$$

Punctul corespunzător de pe elipsa *R* este:

$$xr = xc + dxr, \qquad (9.32)$$
$$yr = yc + dyr.$$

unde

$$dxr = dx_1 \cdot \cos(u) - dy_1 \cdot \sin(u)\,, \qquad (9.33)$$
$$dyr = dx_1 \cdot \sin(u) + dy_1 \cdot \cos(u).$$

9.3.3. Generarea elipselor în spațiul discret

Algoritmul următor pornește de la ecuația corespunzătoare unei elipse cu centrul în origine și axele aliniate cu cele ale sistemului de coordonate carteziene 2D.

Elipsa intersectează axa Ox în a și –a, iar axa Oy în b și –b (figura 9.9).

Se considera elipsa definita prin ecuația implicita:

$$f(x, y) = b^2 \cdot x^2 + a^2 \cdot y^2 + a^2 \cdot b^2 = 0 \tag{9.34}$$

Algoritmul generează aproximarea discretă a elipsei în primul cadran (adică în octanții 1 și 2), știind că celelalte puncte de pe elipsă se pot obține prin simetrie.

De asemenea, pentru o elipsă cu centrul în (*xc*, *yc*) se aplică o translație cu (*xc*, *yc*) (figura 9.10).

Algoritmul constă în alegerea între punctele din spațiul discret față de distanțele la care se află acestea de elipsa teoretică.

Aceste distanțe se măsoară pe orizontală în primul octant și pe verticală în al doilea octant.

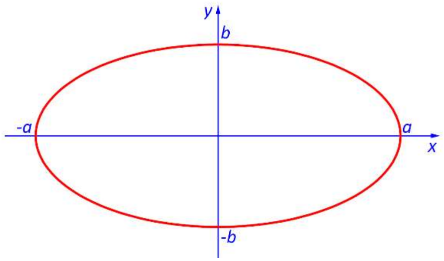

Fig. 9.9. Punctele de intersecție a elipsei cu axele

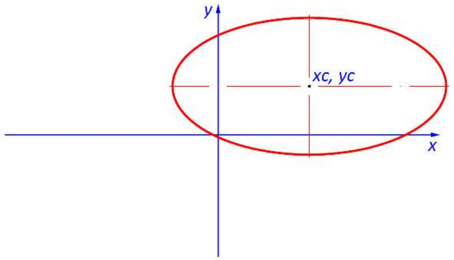

Fig. 9.10. Translația elipsei cu *xc, yc*

9.3.2. Generarea unei elipse rotite

Fie P_1 și P_2 două puncte adiacente ale spațiului discret, între care trebuie să se aleagă.

P_1 este exterior elipsei, iar P_2 este interior.

Se determină care dintre ele este mai apropriat de elipsă în funcție de poziția față de elipsă a punctului aflat la mijlocul distanței dintre cele două puncte.

Dacă punctul de mijloc este în interiorul elipsei, atunci punctul P_1 este mai apropriat de elipsă decât P_2, și invers.

Se calculează valoarea funcției f în punctul de mijloc (x_m, y_m):

– pentru primul octant,

$$(x_m, y_m) = (x_i - 0{,}5,\ y_i + 1), \tag{9.35}$$

– pentru al doilea octant,

$$(x_m, y_m) = (x_i - 1,\ y_i + 0{,}5). \tag{9.36}$$

Valoarea $f(x_m, y_m)$ poate fi:

> 0, dacă (x_m, y_m) este exterior elipsei;
< 0, dacă (x_m, y_m) este în interiorul elipsei;
= 0, dacă (x_m, y_m) este pe elipsă.

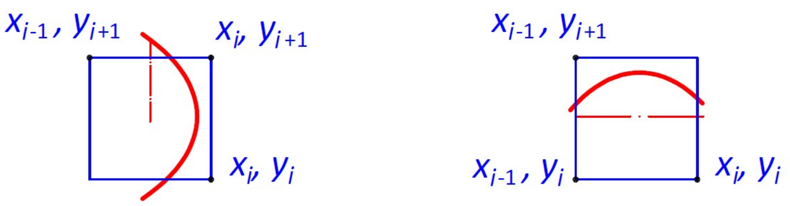

Fig. 9.11. Alegerea punctelor pentru reprezentarea elipsei

9.3.2. Generarea unei elipse rotite

Situația în care este nevoie de a detecta punctele de trecere dintr-un octant în celălalt este studiată în continuare.

Considerăm că panta curbei definite în ecuația (9.34) este dată de relația:

$$\frac{dx}{dy} = -\frac{fx}{fy} = -\frac{2 \cdot b^2 \cdot x}{2 \cdot a^2 \cdot y}, \qquad (9.37)$$

unde *fx și fy sunt derivatele parțiale ale funcției f în raport cu x și y*.

Trecerea din octantul 1 în octantul 2 are loc atunci când panta devine mai mare decât –1:

adică: $-\frac{2 \cdot b^2 \cdot x}{2 \cdot a^2 \cdot y} > -1$,

sau: $2 \cdot b^2 \cdot x < 2 \cdot a^2 \cdot y$.

Apoi efectuăm următoarele notații:

$$\begin{aligned} dx &= 2 \cdot b^2 \cdot x, \\ dy &= 2 \cdot a^2 \cdot y. \end{aligned} \qquad (9.38)$$

Aceste valori se calculează prin recurență, fiind actualizate, atunci când se modifică *x*, respectiv *y*.

9.3.2. Generarea unei elipse rotite

Valoarea funcției f în punctul de mijloc, o notăm cu fm_i, și este calculată de asemenea prin recurență.

Pentru primul pas, avem $x_m = a - 0{,}5$ și $y_m = 1$,

de unde obținem $fm_i = f(x_m, y_m) = b^2 \cdot (0{,}25 - a) + a^2$.

Pentru punctele din primul octant, înlocuim în ecuația (9.34) și obținem:

$$f\left(x_i - \frac{1}{2}, y_i + 1\right) = b^2 \cdot \left(x_i^2 - x_i + 0{,}25\right) + a^2 \cdot \left(y_i^2 + 2y_i + 1\right) - a^2 \cdot b^2. \tag{9.39}$$

Pentru punctele din al doilea octant, înlocuim în ecuația (9.34) și obținem:

$$f\left(x_i - 1, y_i + \frac{1}{2}\right) = b^2 \cdot \left(x_i^2 - 2x_i + 1\right) + a^2 \cdot \left(y_i^2 + y_i + 0{,}25\right) - a^2 \cdot b^2. \tag{9.40}$$

Diferența dintre valoarea funcției în punctul de mijloc pentru al doilea octant și valoarea pentru primul octant este:

$$b^2 \cdot \left(-x_i + \frac{3}{4}\right) + a^2 \cdot \left(-y_i - \frac{3}{4}\right).$$

9.4. Generarea suprafețelor
9.4.1. Generarea poligoanelor

Suprafețele pot fi definite în trei moduri :

– prin contur – se cunoaște culoarea pixelilor care alcătuiesc conturul suprafeței și un punct interior;

– prin interior – se cunoaște culoarea pixelilor și un punct interior;

– geometric – se cunosc coordonatele punctelor care determină geometric conturul suprafeței; în acest caz conturul este un poligon oarecare.

1) Pentru generarea unei suprafețe definite prin contur este necesar ca în memoria raster să fie deja înscris conturul.

Pornind de la punctul interior cunoscut, se schimbă culoarea tuturor pixelilor interiori conturului în culoarea și conform șablonului care sunt date.

2) Generarea unei suprafețe definite prin interior presupune existența în memoria raster a suprafeței.

Pornind din punctul interior cunoscut, se schimbă culoarea tuturor pixelilor suprafeței în culoarea și cu șablonul care sunt date.

Există două tipuri de algoritmi folosiți la generarea suprafețelor definite prin contur și a celor definite prin interior :

– *algoritmi bazați pe verificarea culorii punctelor învecinate punctului curent;*

– *algoritmi bazați pe parcurgerea liniilor raster care traversează suprafața*.

9.4. Generarea suprafețelor
9.4.1. Generarea poligoanelor

A) Suprafețele definite prin contur sau prin interior pot fi:

– conexe de ordinul 4 (figura 9.12, *a*),

– sau conexe de ordinul 8 (figura 9.12, *b*).

O suprafață este conexă de ordinul 4 dacă se consideră că fiecare pixel al suprafeței poate fi atins, pornind dintr-un punct interior, printr-o combinație de deplasări în 4 direcții: stânga, dreapta, sus, jos.

O suprafață este conexă de ordinul 8 dacă se consideră că fiecare pixel al suprafeței poate fi atins, pornind dintr-un punct interior, printr-o combinație de deplasări în 8 direcții – pe orizontală, pe verticală și pe diagonale.

Exemple:

În figura 9.13 sunt prezentate suprafețe:

a – cu colorare totală,

b – cu colorare parțială, conexă de ordinul 8,

c – cu colorare parțială, conexă de ordinul 4.

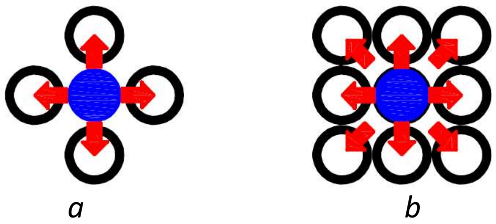

Fig. 9.12. Suprafețe definite prin contur

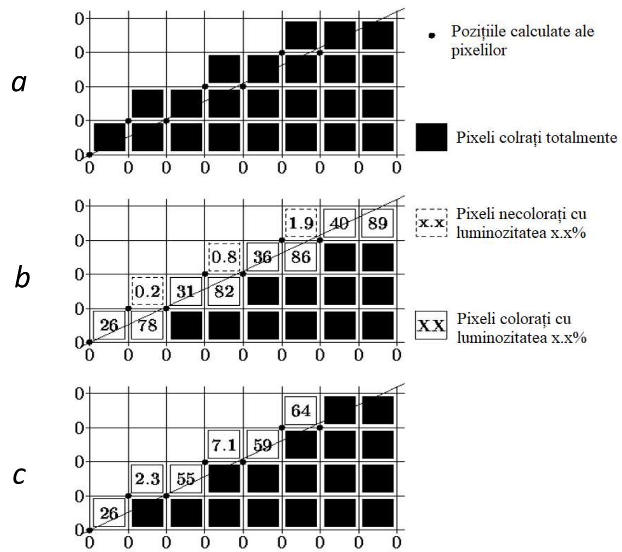

Fig. 9.13. Exemple de colorare cu conexiuni de ordinul 4 și 8

9.4. Generarea suprafețelor
9.4.1. Generarea poligoanelor

B) Algoritmul necesită rezervarea unui spațiu mare de memorare pentru stiva program.

De aceea, pentru generarea suprafețelor definite prin contur sau prin interior sunt preferați algoritmii bazați pe baleierea suprafeței linie cu linie.

Prezentăm în continuare un astfel de algoritm particularizat pentru suprafețe definite prin contur.

O suprafață poate fi convexă sau concavă și poate avea găuri.

Deci o linie orizontală care o traversează poate întâlni conturul suprafeței de mai multe ori.

Considerăm următoarea suprafață:

Un grup compact de pixeli situați pe aceeași linie de baleiere, care nu au culoarea conturului și nici culoarea de umplere, formează un interval;

Fie $P(x_{int}, y_{int})$ punctul interior dat. El este primul punct de start pentru colorare.

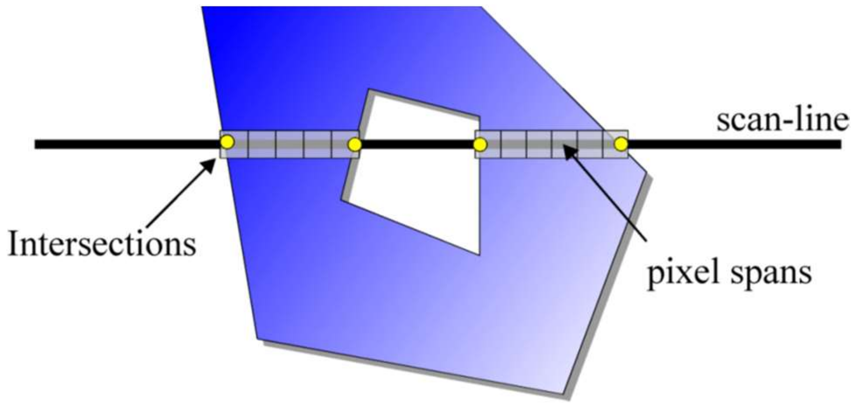

Fig. 9.14. Colorarea unui poligon

9.4. Generarea suprafețelor
9.4.1. Generarea poligoanelor

În cadrul algoritmului se execută ciclic următorii pași:

1) Pornind din punctul de start, $P(x_i, y_i)$, se colorează toți pixelii din dreapta sa până se ajunge la contur, apoi toți pixelii din stânga sa până se ajunge la contur.

Se memorează extremitatea dreapta a intervalului colorat = x_{max}, și extremitatea sa stânga = x_{min}.

2) Se examinează pixelii din intervalul $x_{min} - x_{max}$ de pe linia de deasupra liniei curente, pentru a se determina dacă include numai pixeli de contur sau pixeli deja colorați.

Intervalul poate conține mai multe subintervale. Se memorează extremitatea din dreapta a fiecărui subinterval într-o stivă.

Aceeași prelucrare se execută pentru linia de sub cea curentă.

3) Se extrage din vârful stivei noul punct de start pentru colorare, (x_i, y_i), și se revine la pasul 1.

Deoarece ultimul punct introdus în stivă este de pe linia de sub cea curentă, prelucrarea va continua cu această linie.

Execuția algoritmului se termina atunci când stiva este vidă.

Colorarea poligonului din figura de mai înainte începe din punctul *I*.

După colorarea liniei $y = y_I$ vor fi colorate liniile din zona 1.

La prima linie din zona 2 se schimbă x_{max}.

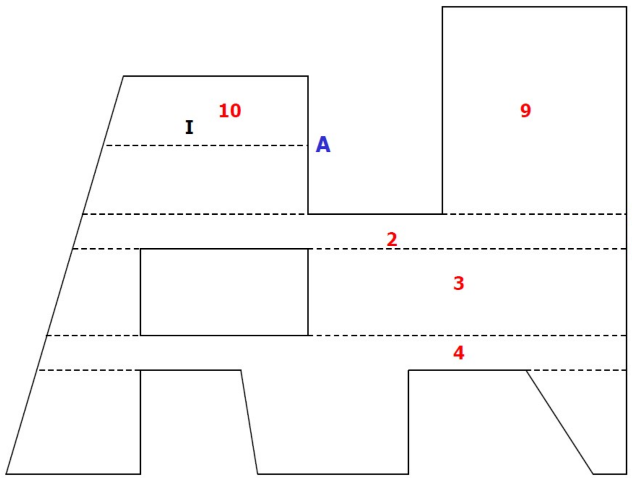

Fig. 9.15. Exemplu de colorare a unui poligon

9.4. Generarea suprafețelor
9.4.1. Generarea poligoanelor

Când se examinează pixelii de pe linia ce conține latura de sus a dreptunghiului interior poligonului, se vor memora în stiva punctul C și apoi punctul D.

Punctul D este extras imediat din stivă, devenind punct de start.

Parcurgând pentru colorare linia $y = y_D$, de la dreapta la stânga, se va întâlni latura din dreapta a dreptunghiului, și se va modifica x_{min}.

Colorarea se continuă cu liniile din zona 3, apoi cu cele din zona 4 și din zona 4 și din zona 5.

Când se ajunge la limita de jos a zonei 5, punctul din vârful stivei este *F*, de aceea colorarea se continuă cu zona 6 și așa mai departe.

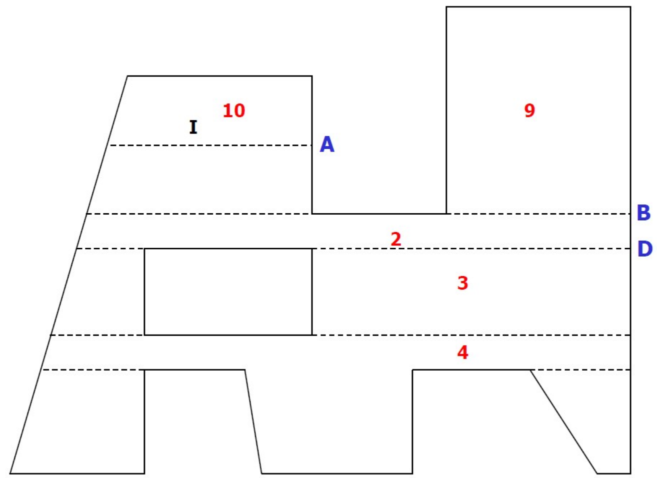

Fig. 9.15. Exemplu de colorare a unui poligon

9.4.2. Generarea suprafețelor circulare și eliptice

Algoritmii de generare a cercurilor și elipselor pot fi extinși foarte ușor pentru afișarea tuturor punctelor interioare.

De exemplu, pentru fiecare punct determinat în algoritmul Bresenham se pot trasa patru linii orizontale determinate de cele opt puncte simetrice de pe cerc.

9.4.3. Generarea interiorului cu un șablon

Un șablon este o matrice de numere întregi, reprezentând culori.

Generarea interiorului unei suprafețe presupune ca fiecare element al matricei șablon este pus în corespondență cu un singur pixel.

Șablonul se aplică începând fie dintr-un punct interior dat, fie din originea ecranului, colorându-se numai punctele interioare suprafeței.

O metodă foarte simplă de generare a unui interior cu șablon se bazează pe algoritmul de colorare prin linii de hașurare.

Astfel, în loc să se afișeze toți pixelii de pe un segment de hașurare cu aceeași culoare, se va extrage culoarea fiecărui pixel de pe linia matricei șablon corespunzătoare liniei de hașurare.

Considerăm cazul în care șablonul se aplică începând din punctul (0, 0).

Notăm cu *hs – numărul de linii ale matricei șablon* și cu *ls – numărul de coloane*.

Atunci, pixelii de pe linia de hașurare *y* = *yh* vor fi colorați folosind elementele de pe linia *lin* = *yh* % *hs* a matricei șablon.

Pixelul de coordonate (*x*, *yh*) va fi colorat folosindu-se valoarea elementului de pe linia lin și coloana *x* % *ls* al matricei șablon.

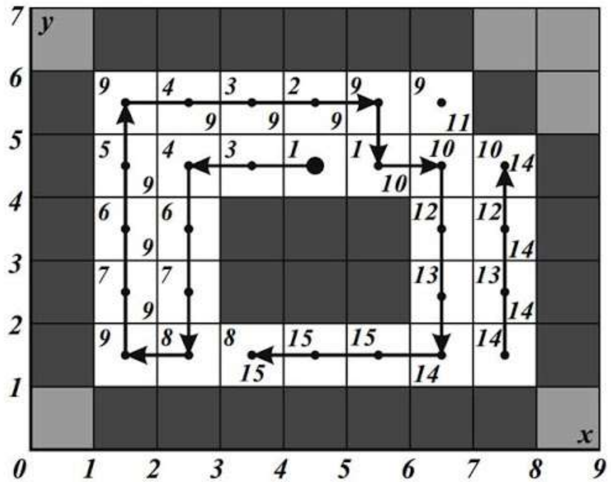

Fig. 9.16. Exemplu de colorare a unui poligon prin metoda conexă de ordinul 4

ÎNTREBĂRI