Datorgrafikas un Attēlu apstrādes pamati

1.Lekcija – Ievads Datorgrafikā un Attēlu apstrādē

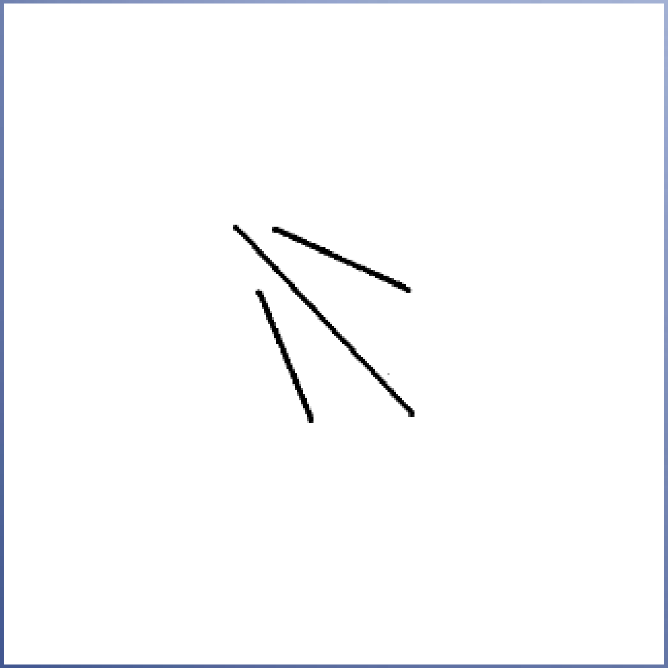
Datorgrafikas mērķi un pamatuzdevumi

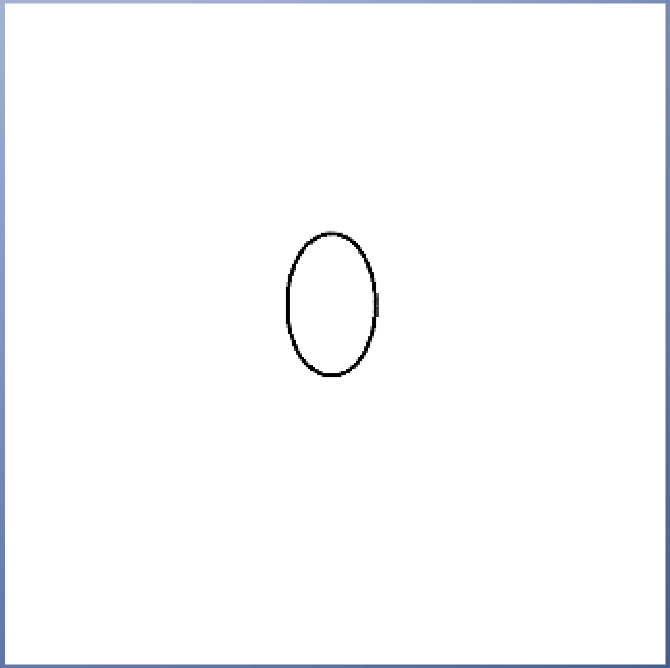
1) Grafisko objektu vai attēlu izveidošana (sintēze) 2D un 3D tēlpā.

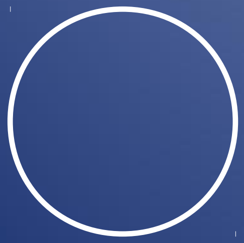
   Attēlu krāsas: melnbalti vai krasaini.

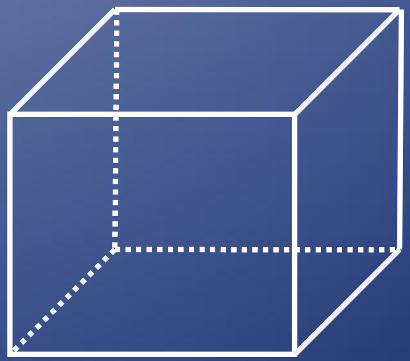
Ipatnība: grafiskie objekti veidojas no ģeometriskām pamatstruktūrām vai primitīviem – taisnas līnijas, riņķa līnijas, virsmas utt.

Piemēri







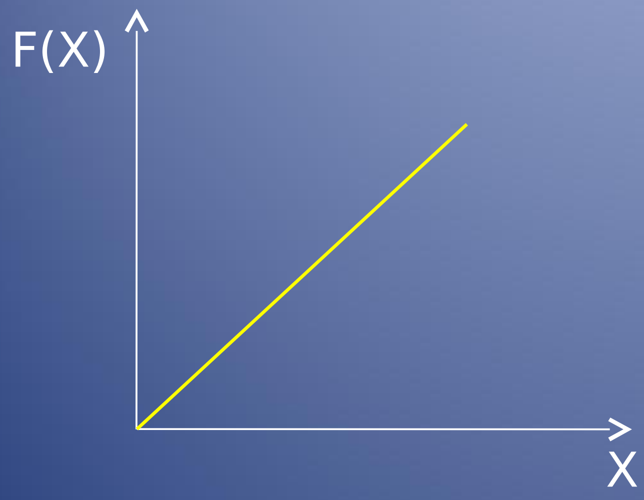


Pirmais uzdevums

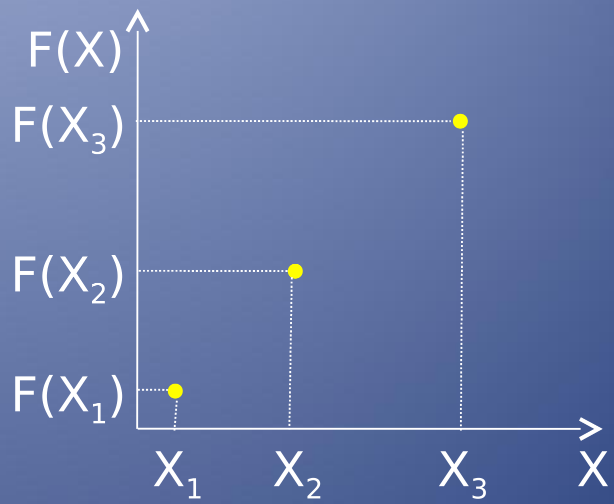
Izveidot grafiskos primitīvus no kuriem pēc tam izveidot grafisko objektu.

Taisnes veidošana

Nepartraukta funkcija



Diskrēta funkcija



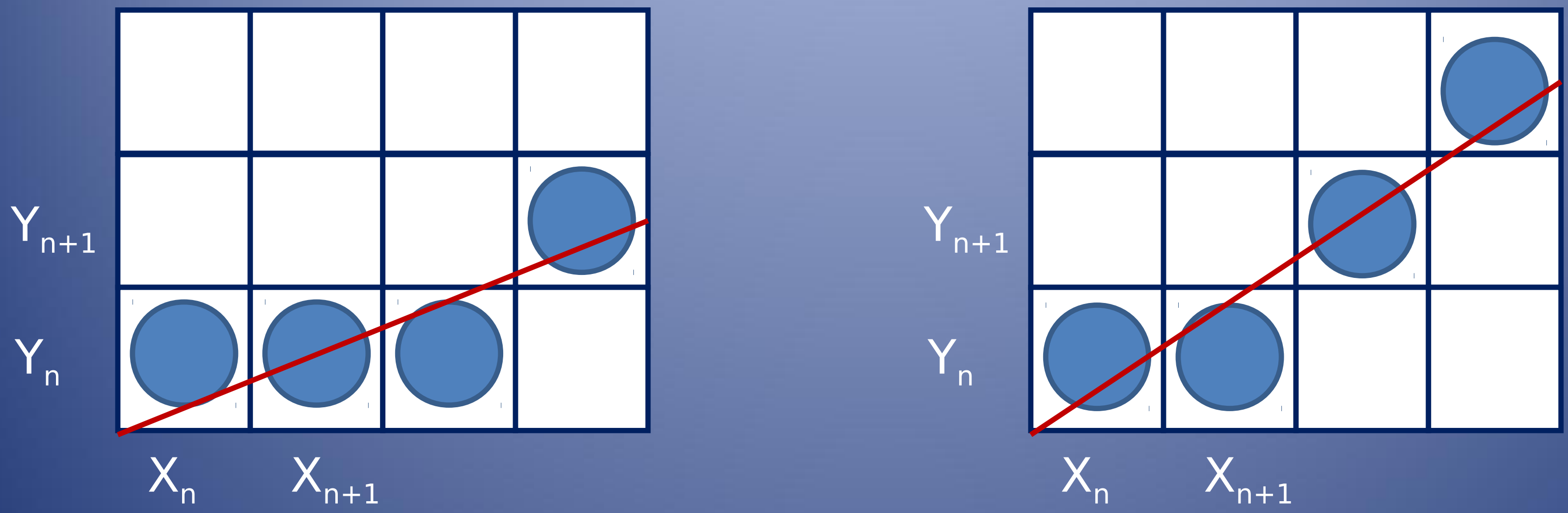
Taisnas līnijas vienādojums

partrauktā gadījumā, ja ir divi punkti (x_1, y_1), (x_2, y_2) :

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}, \qquad (1)$$

$$y = \frac{(x - x_1)(y_2 - y_1)}{x_2 - x_1} + y_1 \qquad (2)$$

Taisnes aproksimācija



Datorgrafikas mērķi un pamatuzdevumi

2) Grafisko objektu transformācija:

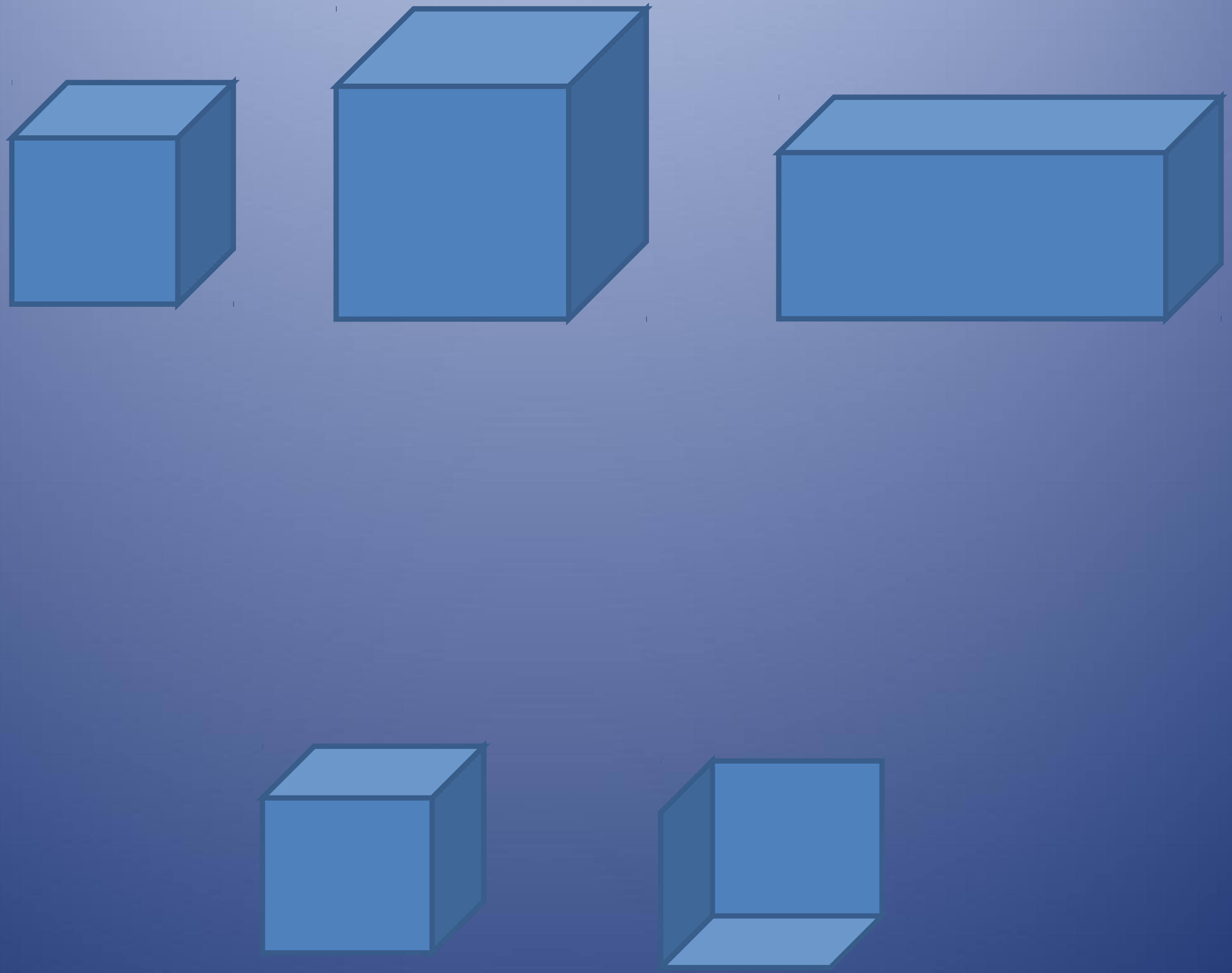
2.1 Mērogošana

2.2 Pārvietošana

2.3 Apgaismošana

utt.

Piemēri



Attēlu apstrādes mērķi un pamatuzdevumi

1) Attēlu kvalitātes uzlabošana – kontrasta izmaiņas, trokšņu attīrīšana utt.
2) Attēlu vai scēnu analīze – kontūru izdalīšana, segmentu izdalīšana, apgabalu atrašana utt.

Datorgrafikas un Attēlu apstrādes pamati

2.Lekcija – Taisnes līnijas veidošanas algoritms

Taisnes līnijas vienādojums vispārīgā gadījumā

$$y = kx + b\,, \qquad (3)$$

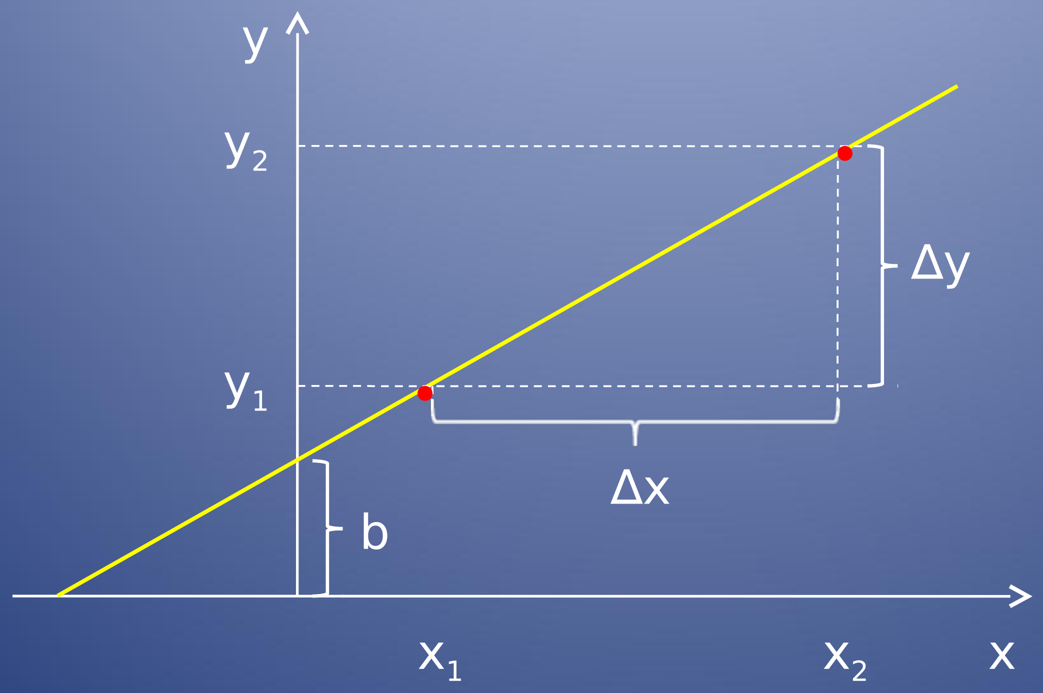
kur

$$k = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \qquad (4)$$

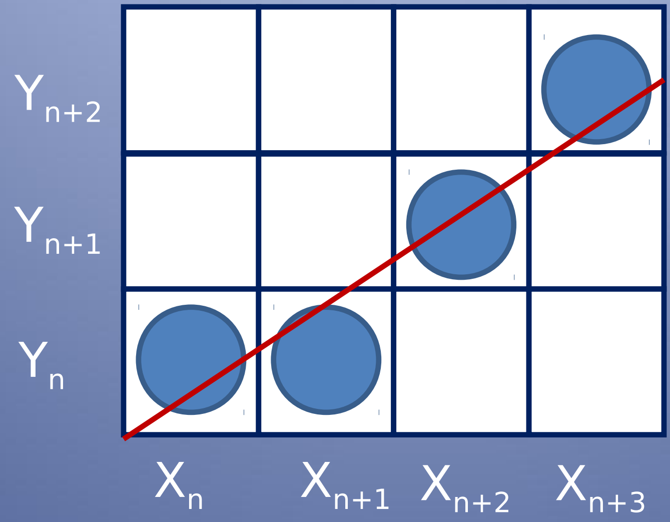
un

$$b = y_1 - kx_1 \qquad (5)$$

Taisnes līnijas vienādojums vispārīgā gadījumā



Pikseļa izvēlēšana

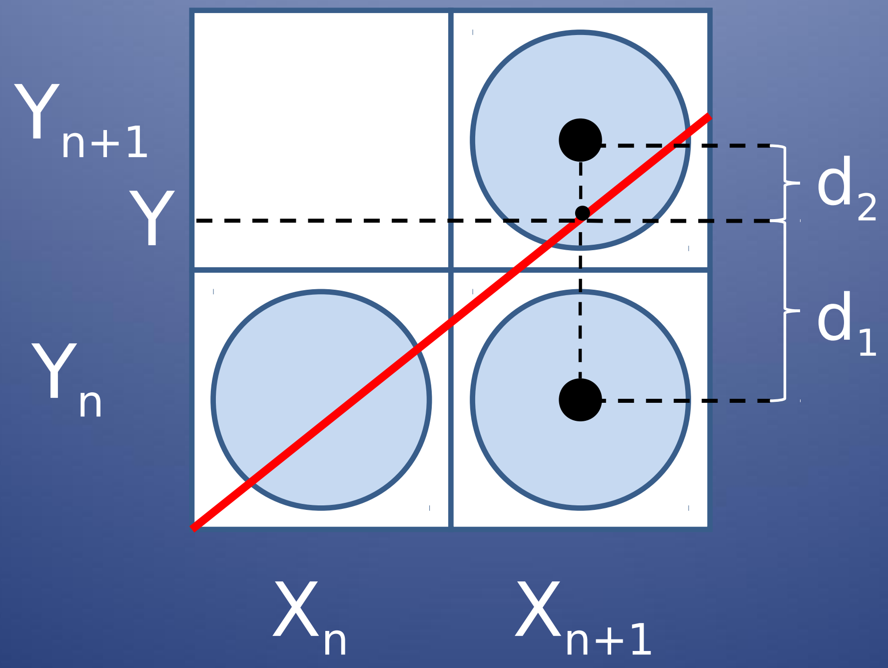


Pieņemsim, ka attālums starp tuvākiem pikseļiem ir vienāds ar 1 pa asi *x*, tad $x_{n+1} = x_n + 1$ un

$$y = k(x_n + 1) + b \qquad (6)$$

Attālumi līdz pikseļu centriem

Attālumi no punkta y, kas pieder taisnei līdz pikseļu (x_{n+1}, y_n), (x_{n+1}, y_{n+1}) centriem attiecīgi būs d_1 un d_2



Attālumi līdz pikseļu centriem

Izmantojot vienādojumu (6), iegūsim

$$d_1 = y - y_n = k(x_n + 1) + b - y_n \qquad (7)$$

$$d_2 = (y_n + 1) - y = y_n + 1 - k(x_n + 1) - b \qquad (8)$$

No tā izriet:

Ja $d_1 > d_2$, tad nākamais pikselis būs (x_{n+1}, y_{n+1})

Ja $d_1 < d_2$, tad nākamais pikselis būs (x_{n+1}, y_n)

Brezenhema algoritms

Starpība starp attālumiem d_1 un d_2 būs

$$d_1 - d_2 = 2k(x_n + 1) + 2b - 2y_n - 1 \qquad (9)$$

Brezenhems piedāvā algoritmu, kur tiek izmantots risinājošais *(decision)* parametrs

$$p_n = \Delta x(d_1 - d_2) \qquad (10)$$

Brezenhema algoritms

Tā kā mūsu piemērā $\Delta x > 0$, tad p_n zīme sakrīt ar $(d_1 - d_2)$. Ja $d_1 < d_2$, tad p_n zīme ir negatīva.

No iepriekšējā vienādojuma iegūsim

$$p_n = 2\Delta y x_n - 2\Delta x y_n + c, \qquad (11)$$

kur

$$c = 2\Delta y + \Delta x(2b - 1) \qquad (12)$$

Brezenhema algoritms

Nākamā (n+1) solī risinājušais parametrs p_{n+1} saskaņā ar vienādojumu (11) būs

$$p_{n+1} = 2\Delta y \cdot x_{n+1} - 2\Delta x \cdot y_{n+1} + c \qquad (13)$$

Izmantojot formulu (11), atradīsim starpību

$$p_{n+1} - p_n = 2\Delta y(x_{n+1} - x_n) - 2\Delta x(y_{n+1} - y_n) \qquad (14)$$

un tā kā $x_{n+1} = x_n + 1$, tad iegūsim

$$p_{n+1} = p_n + 2\Delta y - 2\Delta x(y_{n+1} - y_n), \qquad (15)$$

kur $(y_{n+1} – y_n)$ pieņem vērtību 0 vai 1, atkarīgi no p_n vērtības

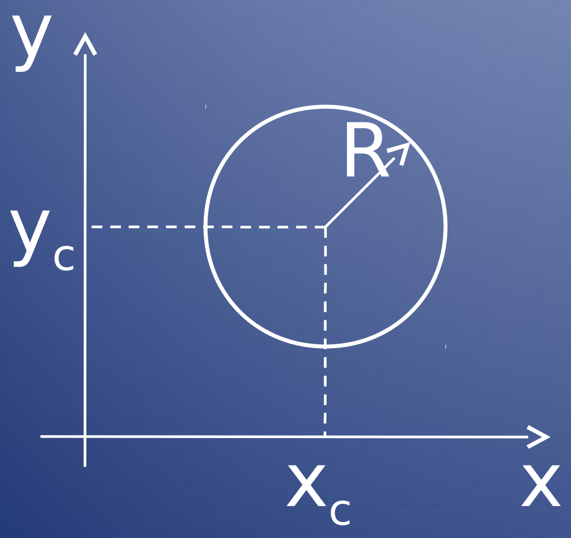
Datorgrafikas un Attēlu apstrādes pamati

3.Lekcija – Riņķa līnijas veidošanas algoritms

Ģeometriskie pamati

Dekarta koordināšu sistēmā riņķa vienādojums:

$$(x - x_c)^2 + (y - y_c)^2 = R^2 \qquad (16)$$



Ģeometriskie pamati

No vienādojuma (16) izriet, ka intervalā

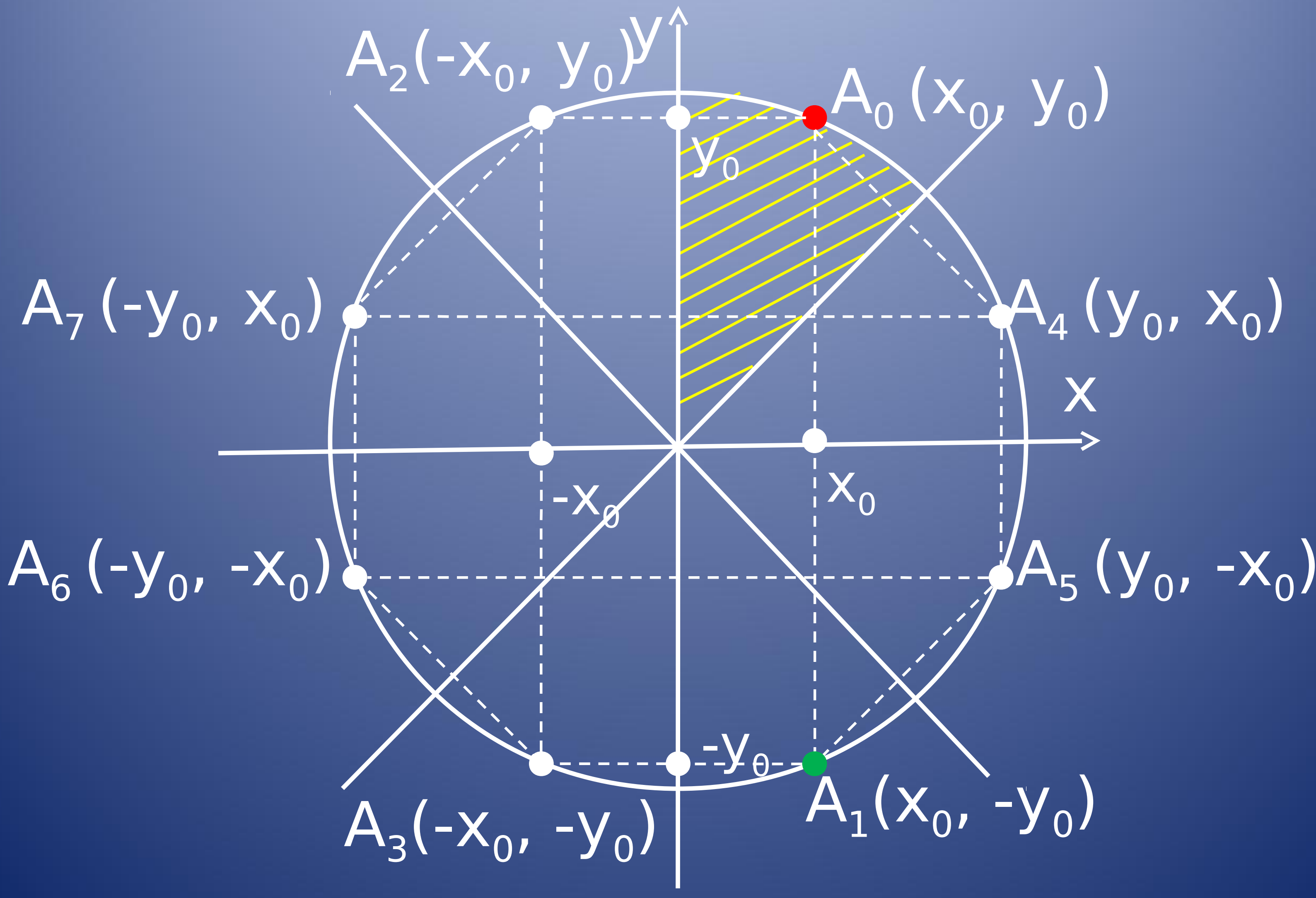
$$x_c - R \le x \le x_c + R$$

var atrast attiecīgi vērtību y:

$$y = y_c \pm \sqrt{R^2 - (x_c - x)^2}$$

Ja izmantot simetrijas īpašības aprēķinājumu apjomu var samazināt

Ģeometriskie pamati



Riņķa veidošanas algoritms ar vidējā punkta izmantošanu

Noteiksim riņķa funkciju gadījumā kad centrs atrodas koordināšu sistēmas sākumā:

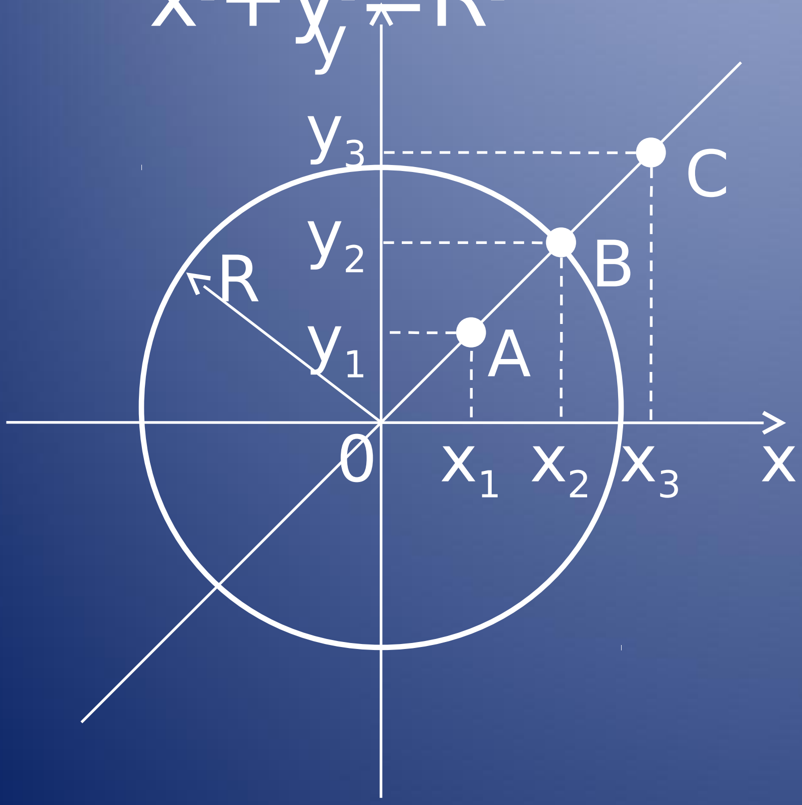
$$f_r(x, y) = x^2 + y^2 - R^2 \qquad (17)$$

Viegli pārliecināties, ka

$$f_r(x, y) = \begin{cases} < 0, & \text{ja punkts (x, y) ir iekšā riņķī} \\ = 0, & \text{ja punkts (x, y) atrodas riņķa līnijā} \\ > 0, & \text{ja punkts (x, y) atrodas ārpus riņķa} \end{cases}$$

Piemērs

Pieņemsim, ka riņķa vienādojums ir $x^2+y^2=R^2$



No zīmējuma izriet, ja R=1, tad iegūsim:

$$A: x_1^2 + y_1^2 < 1,\ \ x_1^2 + y_1^2 - 1 < 0$$

$$B: x_2^2 + y_2^2 = 1,\ \ x_2^2 + y_2^2 - 1 = 0$$

$$C: x_3^2 + y_3^2 > 1,\ \ x_3^2 + y_3^2 - 1 > 0$$

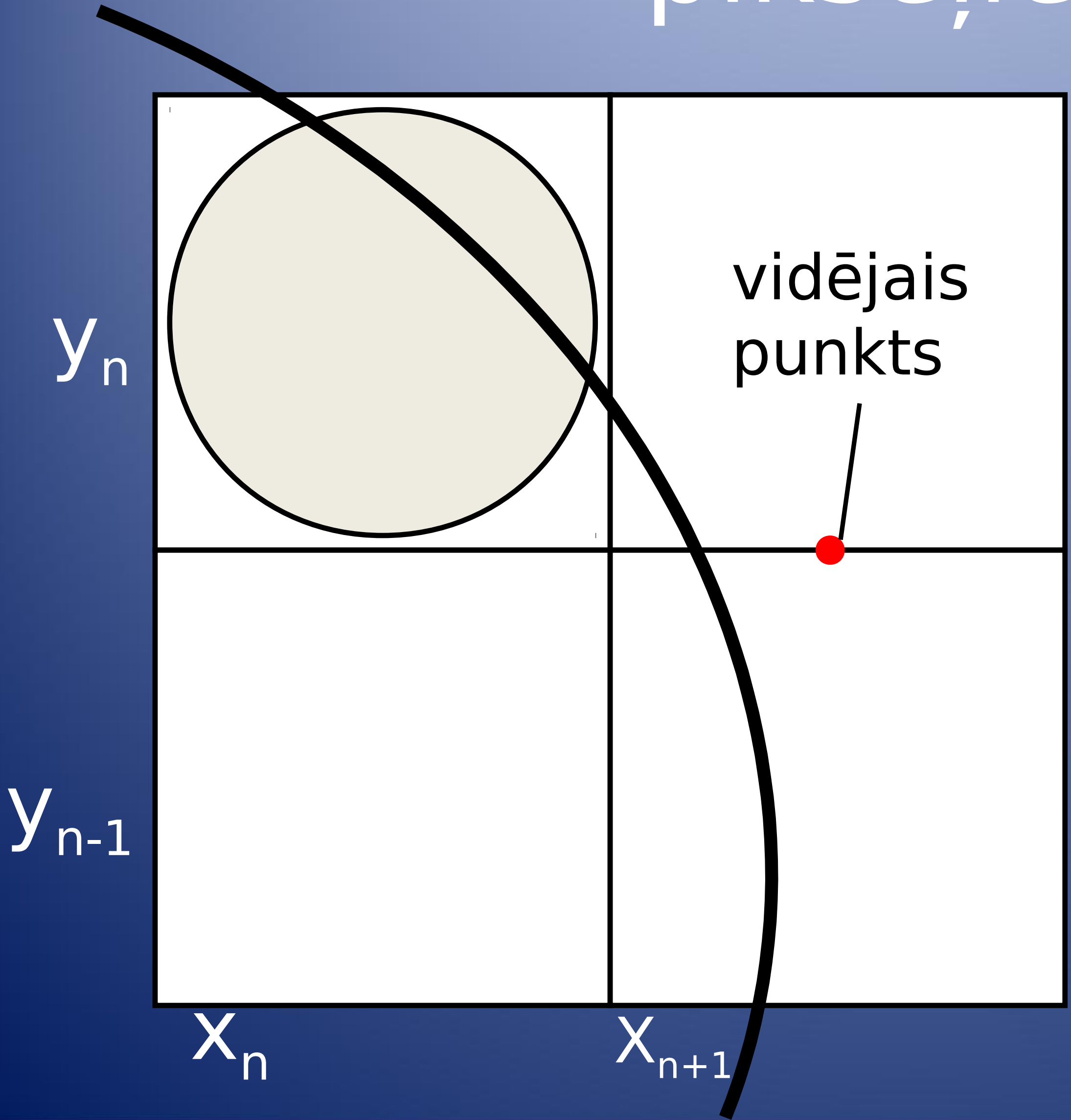
Riņķa veidošanas algoritms ar vidējā punkta izmantošanu

Secinājums:

Atkarīgi no riņķa funkcijas zīmes var noteikt kur atrodas punkts (x, y)

No tā izriet, ka šo funkciju var izmantot kā risinajušu parametru, t.k. tas ļauj noteikt vidējā punkta vietu attiecībā pret riņķi (sk. zīm).

Vidējais punkts starp pikseļiem



$$(x_n, y_n) \rightarrow (x_{n+1}, y_{n+1}),$$

$$kur\ y_{n+1} = \begin{cases} y_n \\ y_{n-1} \end{cases}$$

Vidējais punkts starp pikseļiem un riņķa līniju.

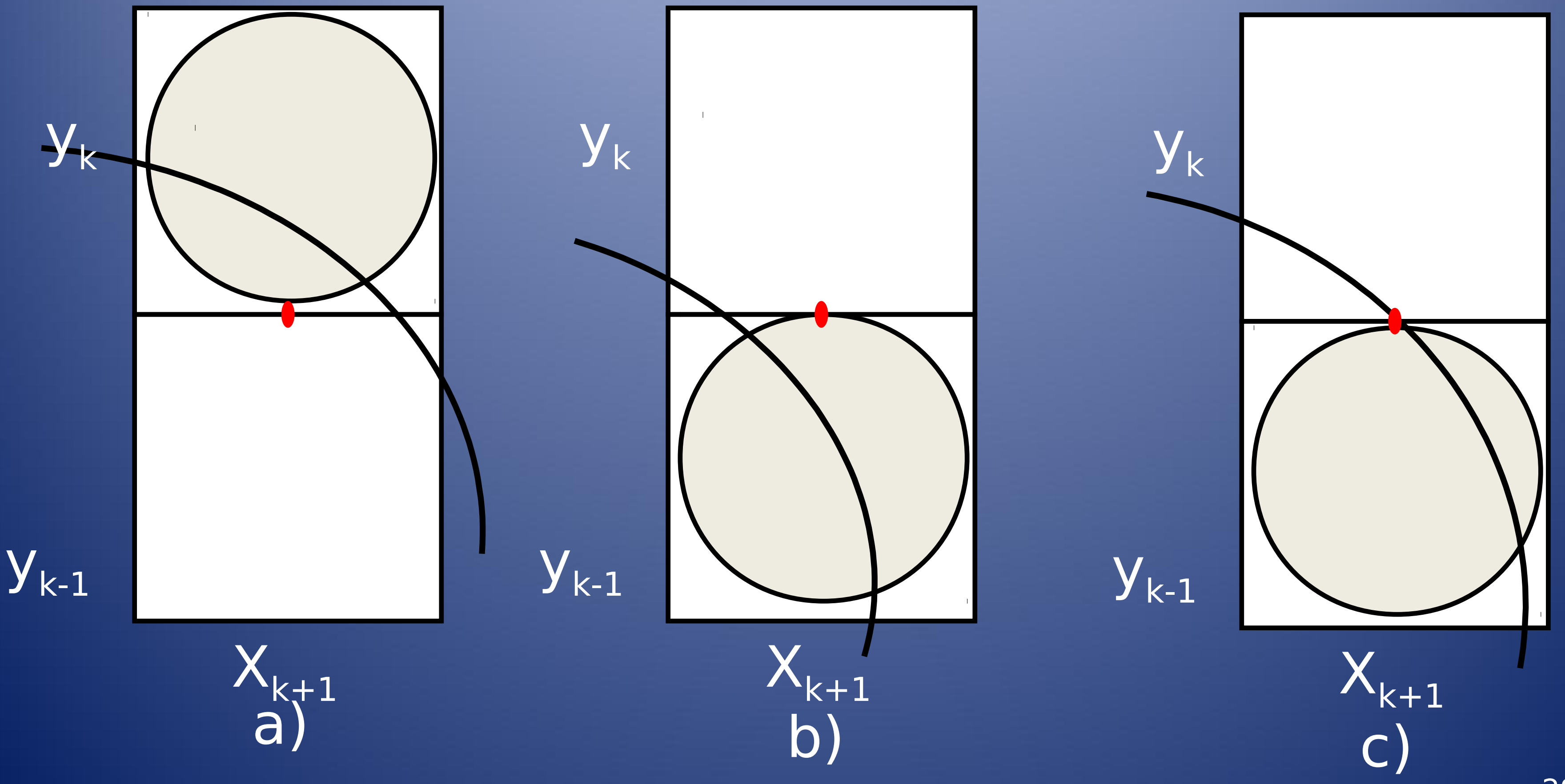
Ja iepriekšējais pikselis (x_k, y_k), tad nākamā solī x_k+1 ir nepieciešams noteikt sekojošo pikseli, kā vienu no diviem, attiecībā pret vidējo punktu: vai nu pikselis (x_k+1,y_k), vai nu pikselis (x_k+1, $y_{k\text{-}1}$).

Šeit tuvumu pie riņķa nosaka vidējā parametra vieta. Šajā gadījumā risinājušais parametrs ir riņķa funkcija, kura nosaka vidējā punkta vietu attiecībā pret riņķi.

$$p_k = f_r\left(x_k+1, y_k-\frac{1}{2}\right) = (x_k+1)^2 + \left(y_k-\frac{1}{2}\right)^2 - R^2 \quad (18)$$

Pikseļu izvēle attiecīgi no vidējā punkta vietas

No izteiksmes (17) var secināt:



Pikseļu izvēle attiecīgi no vidējā punkta vietas

a) Ja $p_k<0$, tad vidējais punkts atrodas iekšā un pikselis y_k tuvākais pie riņķa robežas

b) Ja $p_k>0$, tad vidējais punkts atrodas ārpus riņķa, tad tuvākais pikselis ir y_{k-1}

c) Ja $p_k=0$, tad vidējais punkts atrodas uz robežas

Pikseļu izvēle attiecīgi no vidējā punkta vietas

Nākamā solī x_{k+1}+1 iegūsim:

$$p_{k+1} = f_r(x_{k+1}+1, y_{k+1}-\frac{1}{2}) = [(x_k+1)+1]^2 + (y_{k+1}-\frac{1}{2})^2 - R^2 =$$

$$= (x_k+1)^2 + 2(x_k+1) + 1 + (y_{k+1}-\frac{1}{2})^2 - R^2$$

Pieskaitīsim un atņemsim $(y_k - \frac{1}{2})^2$, rezultātā iegūsim:

$$p_{k+1} = p_k + 2(x_k+1) + (y_{k+1}{}^2 - y_k{}^2) - (y_{k+1} - y_k) + 1, \quad (19)$$

kur y_{k+1} ir vienāds ar y_k vai y_{k-1}, atkarīgi no p_k zīmes:

Pikseļu izvēle attiecīgi no vidējā punkta vietas

1) Ja $p_k<0$, tad $y_{k+1}=y_k$ un saskaņā ar (19), iegūsim:

$$p_{k+1} = p_k + 2(x_k + 1) + 1$$ (20)

2) Ja $p_k \geq 0$, tad $y_{k+1}=y_{k-1}$ un saskaņā ar (19), iegūsim:

$$p_{k+1} = p_k + 2x_{k+1} - 2y_{k-1} + 1$$ (21)

Pikseļu izvēle attiecīgi no vidējā punkta vietas

Sākotnējā punktā (0,R), iegūsim

$$p_0 = f_r(0+1, R-\frac{1}{2}) = 1^2 + (R-\frac{1}{2})^2 - R^2 = \frac{5}{4} - R$$ (22)

Ja rādiusa vērtība ir vesels skaitlis, pēc noapaļošanas iegūsim:

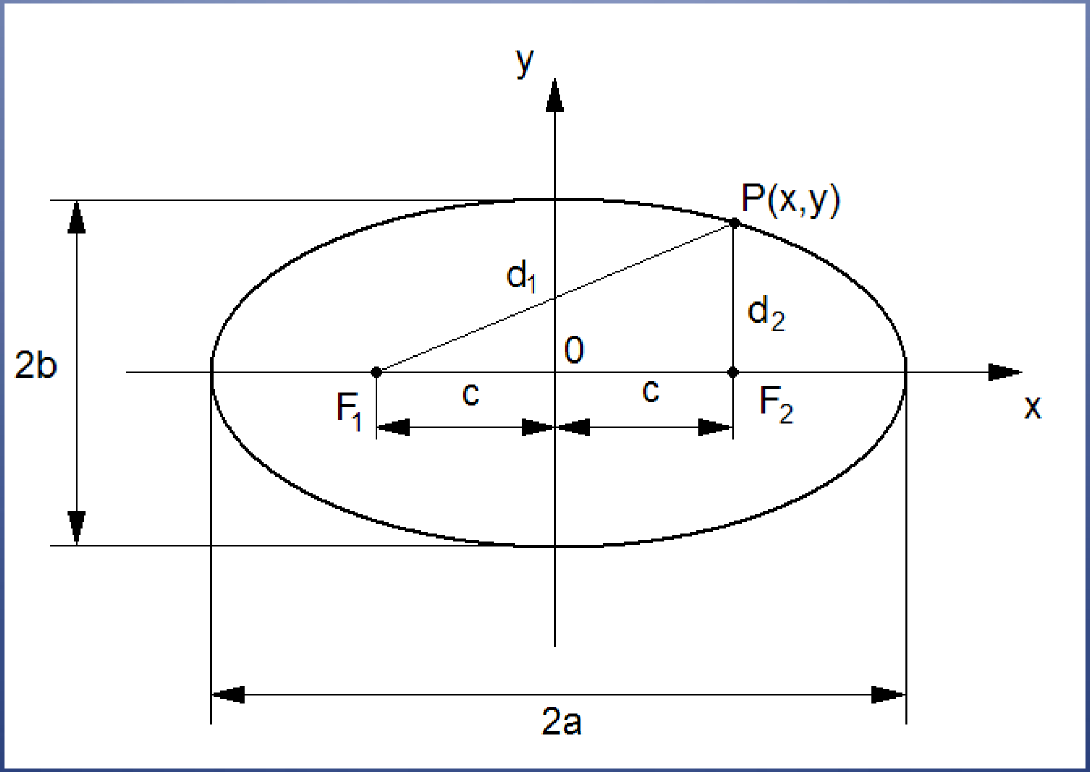
$$p_0 = 1 - R$$ (23)

Datorgrafikas un Attēlu apstrādes pamati

4.Lekcija – Elipses līnijas veidošanas algoritms

Elipses līnija

Elipse – tā ir punktu ģeometriskā vieta un tās punkti atrodas vienādā summārā attālumā d_1+d_2, kur d_1 – attālums no fokusa F_1 un d



Elipses līnija

Ja elipses centrs sakrīt ar koordināšu sistēmas sākumu (0, 0) tad fokusa attālums no centra

$$c = \sqrt{a^2 - b^2}$$

kur a, b – lielā un mazā pusass.

Ja fokusa koordinātes $F_1(x_1,y_1)$ un $F_2(x_2,y_2)$ tad Dekartu koordināšu sistēmā elipses vienādojums būs:

$$\sqrt{(x-x_1)^2+(y-y_1)^2}+\sqrt{(x-x_2)^2+(y-y_2)^2}=2a$$

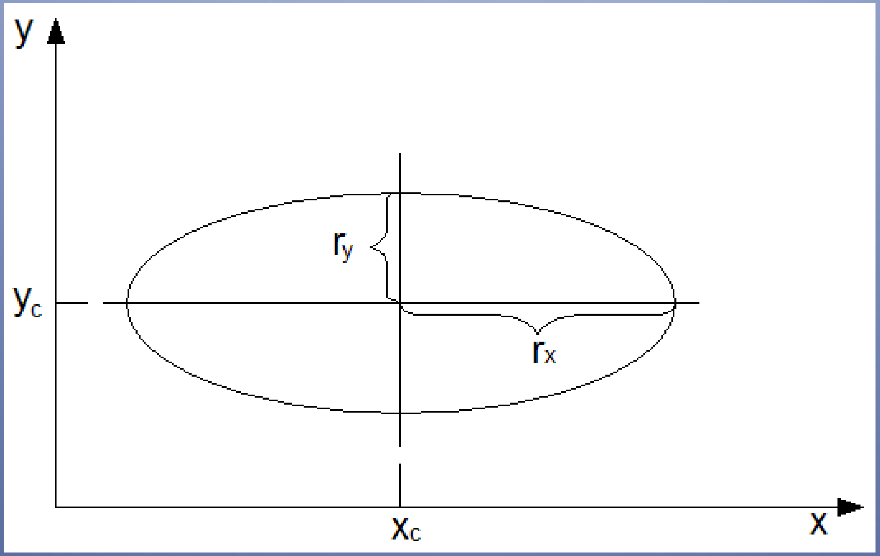
Elipses līnija

Kanoniskā formā punktu koordinātes, kas atrodas elipses līnijā, nosaka vienādojums

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \qquad (24)$$

Gadījumā, ja elipses centrs (x_c, y_c) nesakrīt ar koordināšu sistēmas centru:

Elipses līnija



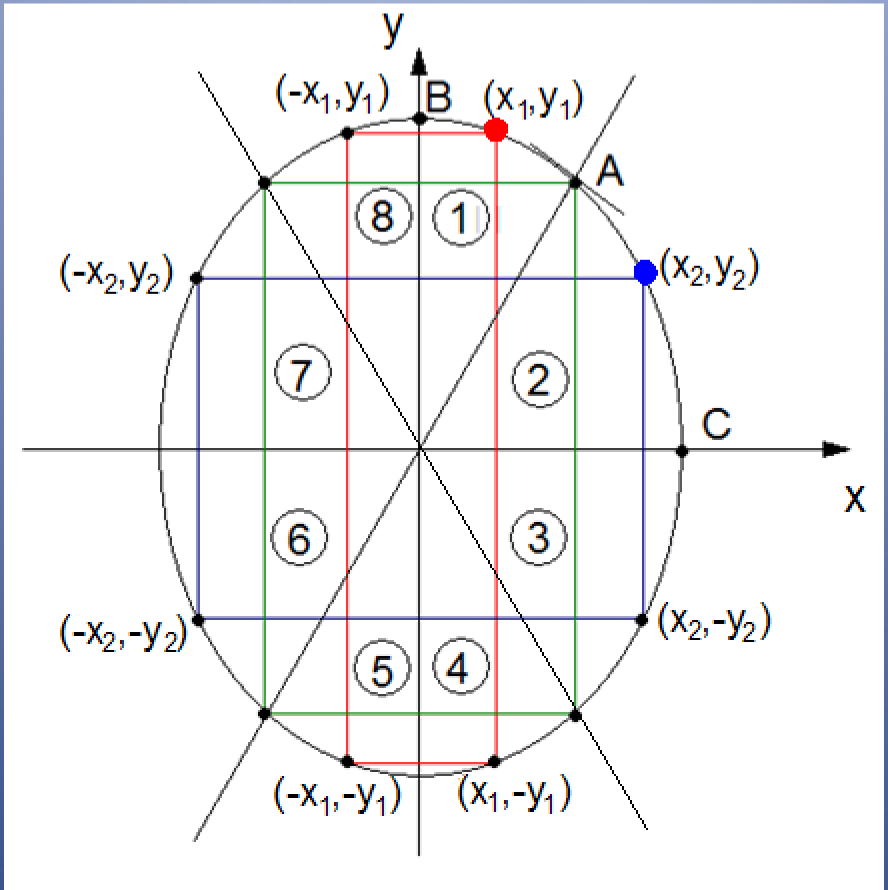
$$\frac{(x-x_c)^2}{r_x^2}+\frac{(y-y_c)^2}{r_y^2}=1 \qquad (25)$$

kur r_x=a un r_y=b

Elipses līnija

- No vienādojumiem (24) un (25) jebkurai vērtībai x var noteikt vērtību y. Bet šajā gadījumā ir nepieciešams katrai veselai vērtībai x noteikt, kāds pikselis būs tuvāks līnijai.
- Atzīmēsim, ka tagad nebūs tāda pilna simetrija, ka iepriekšējā gadījumā riņķa līnijai. Pieņemsim, ka $r_x<r_y$ un sadalīsim pirmo kvadrantu divās daļās (1) un (2).

Elipses līnija



Punkta A pieskares slīpums būs vienāds ar -1

Elipses līnija

Atradīsim punktus pirmā daļā:

Sāksim no punkta B (0, r_y) un soļosim pulksteņa rādītāja virzienā pa līkni līdz punktam A. Pēc tam, kad slīpums paliks mazāks par -1, soļosim uz y virzienu līdz punktam C.

Tālāk atradīsim simetriskus punktus. No daļas (1) var atrast simetriskus punktus (4), (5), un (8) daļās, no daļas (2) var atrast simetriskus punktus (3), (6) un (7) daļās.

Elipses līnija

Aplūkosim elipses veidošanas algoritmu ar vidējā punkta izmantošanu.

Pieņemsim, ka $x_c=y_c=0$ un pēc reizināšanas ar $r_x^2 r_y^2$, no vienādojuma (25) iegūsim

$$r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2 = 0 \qquad (26)$$

Viegli pārliecināties, ka elipses punkta pieskares slīpums

$$\frac{dy}{dx} = -\frac{2r_y^2 x}{2r_x^2 y} \qquad (27)$$

Elipses līnija

un robežā starp daļām (1) un (2) slīpums būs

$$\frac{dy}{dx} = -1, \;\; t.i. \; 2r_y^2 x = 2r_x^2 y \qquad (28)$$

Tālāk, saskaņā ar (26), noteiksim elipses funkciju

$$f_{el}(x,y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2 \qquad (29)$$

Nav grūti pārliecināties, ka:

$$f_{el}(x,y) = \begin{cases} < 0, & \text{ja punkts (x,y) atrodas elipses iekšpusē} \\ = 0, & \text{ja punkts (x,y) atrodas elipses līnijā} \\ > 0, & \text{ja punkts (x,y) atrodas ārpus elipses līnijas} \end{cases} \qquad (30)$$

Elipses līnija

No tā izriet, ka šo funkciju var izmantot ka risinājušo parametru, tā kā funkcija (29) ļauj noteikt vidējā punkta vietu attiecīgi elipses līnijai.

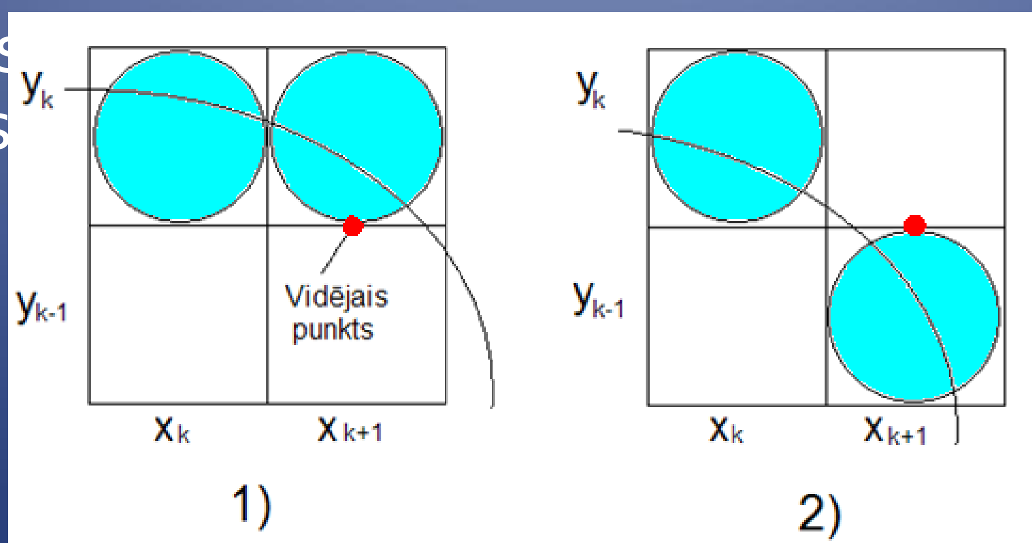
Pirmajā daļā vidējā punkta koordinātes būs (x_k+1,y_k-1/2), risinājušais parametrs pirmā daļā ir

$$p1_k = f_{el}(x_k + 1, y_k - \tfrac{1}{2}) =$$
$$= r_y^2(x_k + 1)^2 + r_x^2(y_k - \tfrac{1}{2})^2 - r_x^2 r_y^2 \qquad (31)$$

Elipses līnija

Saskaņā ar (30), iegūsim:

1) Ja $p1_k<0$, tad vidējais punkts atrodas elipses līnijas iekšpusē un pikselis y_k būs tuvāk elipses līnijai
2) Ja $p1_k \geq 0$, tad vidējais punkts atrodas ārpus elipses … un pikselis y_{k-1} būs …



Elipses līnija

Nākamo (k+1)-a soli risinājušu parametru var pierakstīti šādi:

$$p1_{k+1} = f_{el}(x_{k+1}+1, y_{k+1}-\tfrac{1}{2}) =$$

$$= r_y^2[(x_k+1)+1]^2 + r_x^2(y_{k+1}-\tfrac{1}{2})^2 - r_x^2 r_y^2 =$$

$$= p1_k + 2r_y^2(x_k+1) + r_y^2 + r_x^2[(y_{k+1}-\tfrac{1}{2})^2 - (y_k-\tfrac{1}{2})^2]$$

(32)

kur y_{k+1} vai vienāds y_k, vai y_{k-1}, atkarībā no $p1_k$ zīmes.

Elipses līnija

Pierakstīsim $p1_{k+1}$ šādā veidā:

$$p1_{k+1} = p1_k + \Delta$$

kur

$$\Delta = \begin{cases} 2r_y^2(x_k+1) + r_y^2, \ \ ja \quad p1_k < 0 \\ 2r_y^2(x_k+1) + r_y^2 - 2r_x^2(y_k - 1), \ \ ja \quad p1_k > 0 \end{cases}$$

Elipses līnija

Pirmā daļā sākotnējā punktā (0,r_y) iegūsim

$$p1_0 = f_{el}(0+1, r_y - \tfrac{1}{2}) = r_y^2 - r_x^2 r_y + \tfrac{1}{4} r_x^2 \quad (33)$$

Tālāk katrā solī, saskaņā ar (28) ir nepieciešams pārbaudīt nevienādību

$$2r_y^2 x \geq 2r_x^2 y$$

un, ja tā ir spēkā, pāriet pie otrās daļas.

Elipses līnija

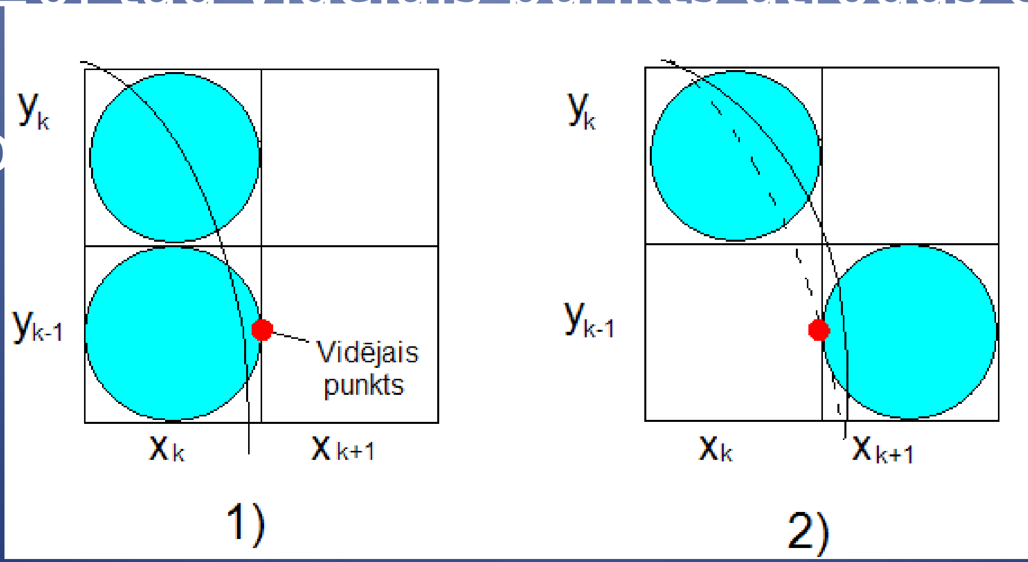
Otrā daļā vidējā punkta koordinātes būs (x_k+1/2,y_k-1), attiecīgi risinājušu parametru var pierakstīt sekojošā veidā:

$$p2_k = f_{el}(x_k + \tfrac{1}{2}, y_k - 1) =$$
$$= r_y^2 (x_k + \tfrac{1}{2})^2 + r_x^2 (y_k - 1)^2 - r_x^2 r_y^2 \qquad (34)$$

Elipses līnija

Saskaņā ar (30), iegūsim:

1) Ja $p2_k>0$, tad vidējais punkts atrodas ārpus elipses līnijas un pikselis x_k būs tuvāk elipses līnijai
2) Ja $p2_k\leq0$, tad vidējais punkts atrodas elipses līnijas un pikselis x_k+1 b



Elipses līnija

Nākamā solī risinājušā parametra lielums būs

$$p2_{k+1} = f_{el}(x_{k+1} + \tfrac{1}{2}, y_{k+1} - 1) =$$

$$= r_y^2 (x_k + \tfrac{1}{2})^2 + r_x^2 [(y_k - 1) - 1]^2 - r_x^2 r_y^2 =$$

$$= p2_k + 2r_x^2 (y_k - 1) + r_x^2 + r_y^2 [(x_{k+1} + \tfrac{1}{2})^2 - (x_k + \tfrac{1}{2})^2] \qquad (35)$$

kur x_{k+1} vai vienāds x_k, vai x_{k+1}, atkarībā no $p2_k$ zīmes.

Datorgrafikas un Attēlu apstrādes pamati

5.Lekcija – Līknes veidošanas algoritms

Līknes veidošana

Līkne ir viens no pamatelementiem (primitīviem) datorgrafikā.

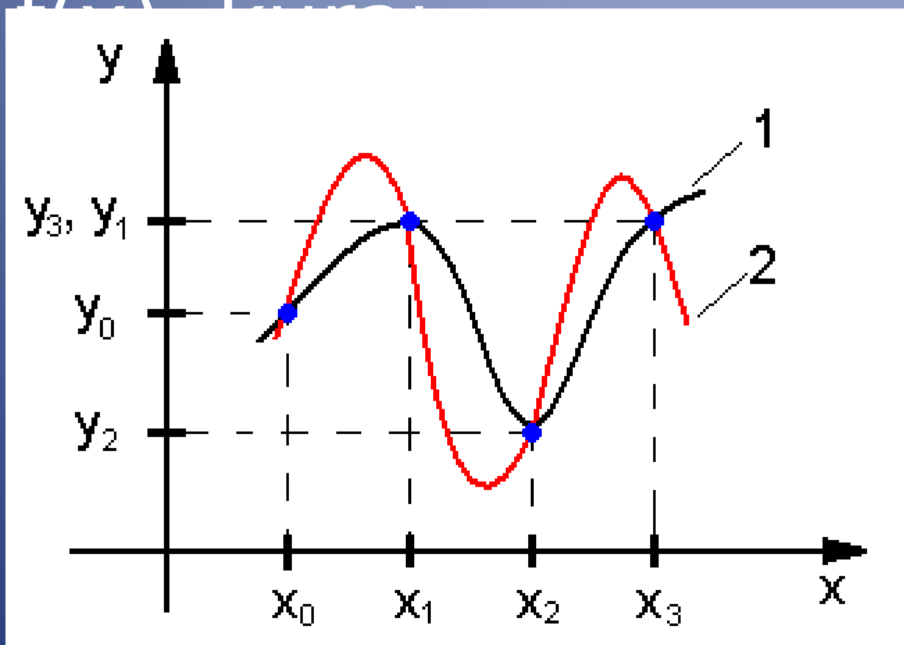
Ja mēs gribam attēlot līkni, kad uzdoti kontrolpunkti (vai vadošie punkti):

$$(x_0,y_0),\ (x_1,y_1),\ \ldots,\ (x_n,y_n)$$

rodas divas iespējas:

Līknes veidošana

1) Interpolācija – atrast tādu funkciju f(x), kurai



$$f(x_0) = y_0$$

$$f(x_1) = y_1$$

$$f(x_2) = y_2$$

$$f(x_3) = y_3$$

....

$$f(x_i) = y_i$$

(x_i, y_i) – kontrolpunkti, $i \in [1:n]$

(x_0, y_0), (x_n, y_n) – sākuma un beigu punkti

Līknes veidošana

Šīs problēmas risināšanai var izmantot Lagranža interpolācijas polinomu.

Tiek uzdoti kontrolpunkti

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

un ir zināms

$$f(x_0) = y_0, f(x_1) = y_1, \ldots, f(x_n) = y_n$$

Līknes veidošana

Ir nepieciešams izveidot polinomu

$L_n(x)$, kuram pakāpe ≤ n

un

$L_n(x_i) = y_i$, $i \in [0:n]$

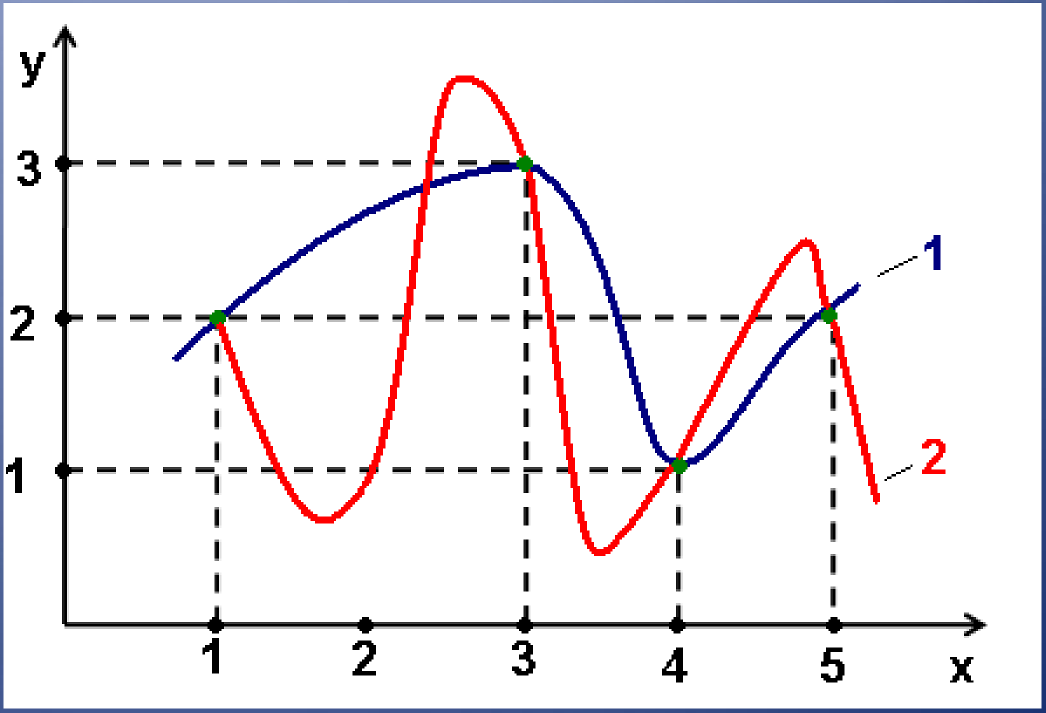
Lagranža polinoms:

$$L_n(x) = \sum_{i=0}^{n} y_i \frac{(x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)(x_i - x_1)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}$$

Līknes veidošana

Piemērs:

i	x_i	y_i
0	1	2
1	3	3
2	4	1
3	5	2



Līknes veidošana

$$L_n(x) = \sum_{i=0}^{n} y_i P_i,$$

kur

$$P_i = \frac{(x - x_0)(x - x_1)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n)}{(x_i - x_0)(x_i - x_1)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)}$$

$$P_0 = \frac{(x-3)(x-4)(x-5)}{(1-3)(1-4)(1-5)} = \frac{(x-3)(x-4)(x-5)}{-24}$$

Līknes veidošana

$$P_1 = \frac{(x-1)(x-4)(x-5)}{(3-1)(3-4)(3-5)} = \frac{(x-1)(x-4)(x-5)}{4}$$

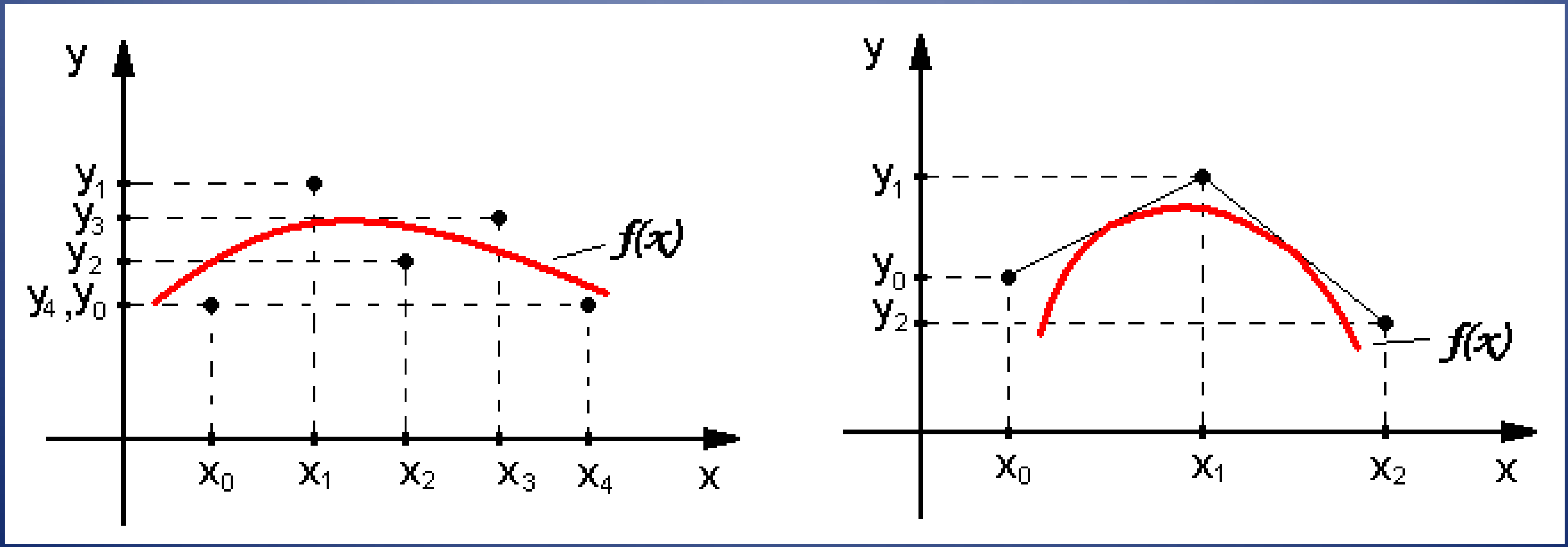
$$P_2 = \frac{(x-1)(x-3)(x-5)}{(4-1)(4-3)(4-5)} = \frac{(x-1)(x-3)(x-5)}{-3}$$

$$P_3 = \frac{(x-1)(x-3)(x-4)}{(5-1)(5-3)(5-4)} = \frac{(x-1)(x-3)(x-4)}{8}$$

Viegli pārliecināties, ka i-tā punktā P_i=1, bet P_k=0, k∈[1:n], k≠i un $L_n(x_i) = y_i$

Līknes veidošana

2) Nogludināšana (aproksimācija) – atrast tādu gludu funkciju, kura būs “tuvāka” kontrolpunktiem.



Līknes veidošana

Vispārīgā gadījumā, nezināmu funkciju var aprakstīt ar n-tās pakāpes polinomu

$$p(t) = a_n t^n + a_{n-1} t^{n-1} + \ldots + a_1 t + a_0$$

Ja gribam atrast tādu funkciju, kura būs “tuvāka” kontrolpunktiem, var pielietot mazāko kvadrātu metodi.

Līknes veidošana

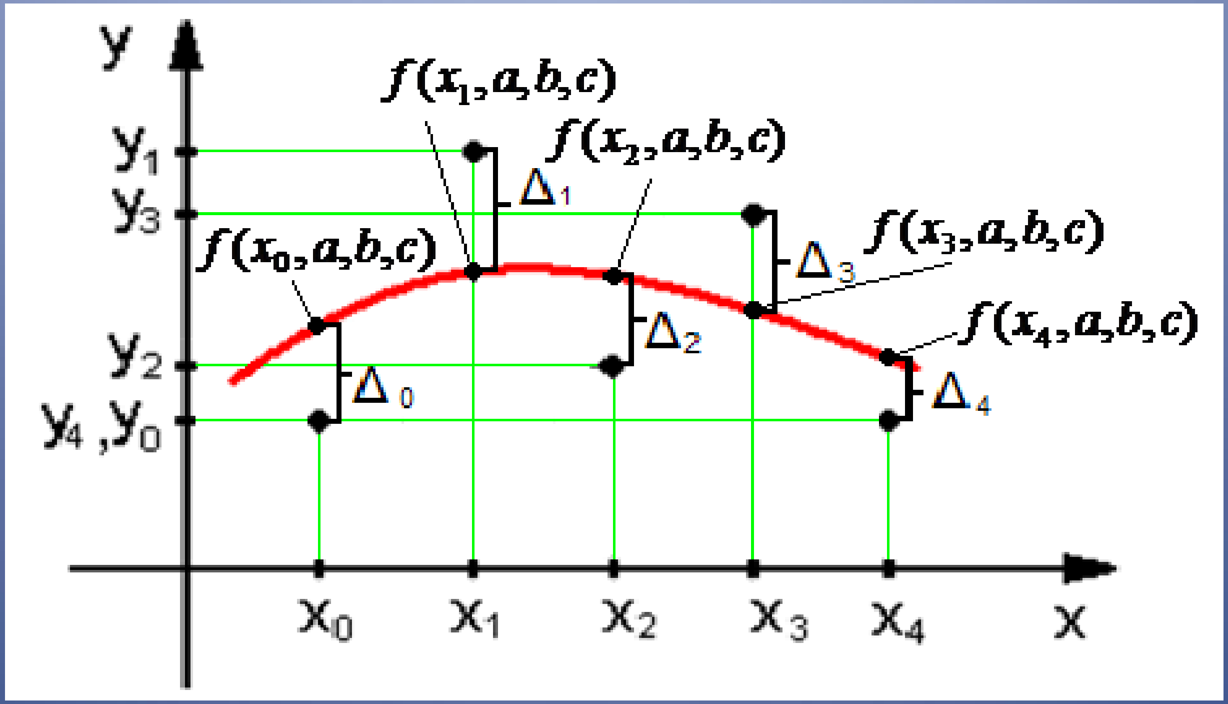
Mazāko kvadrātu metode

$$f(x) \to f(x_i, a, b, c)$$

$$f(x_i, a, b, c) = ax_i^2 + bx_i + c$$

$$\sum_{i=0}^{n} [y_i - f(x_i, a, b, c)]^2 \to \min_{a,b,c}$$

Līknes veidošana



$$\Delta_i = y_i - f(x_i, a, b, c)$$

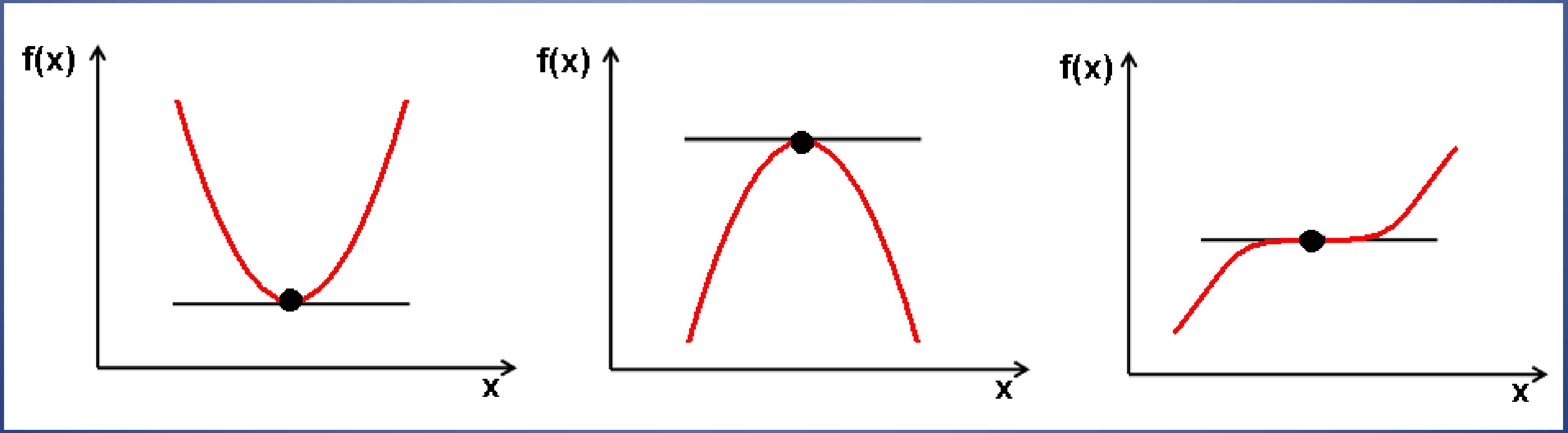
$$\sum_{i=0}^{n} \Delta_i^2 \to \min_{a,b,c}$$

Līknes veidošana

$$\begin{cases} \sum_{i=0}^{n} [y_i - f(x_i, a, b, c)] \frac{\partial f}{\partial a}\Big|_i = 0 \\ \sum_{i=0}^{n} [y_i - f(x_i, a, b, c)] \frac{\partial f}{\partial b}\Big|_i = 0 \\ \sum_{i=0}^{n} [y_i - f(x_i, a, b, c)] \frac{\partial f}{\partial c}\Big|_i = 0 \end{cases}$$

kur $\frac{\partial f}{\partial a} = x^2, \; \frac{\partial f}{\partial b} = x, \; \frac{\partial f}{\partial c} = 1$

Līknes veidošana



Bezier līknes

Tiek uzdoti vadošie (kontrol) punkti

$P_0, P_1, P_2, \ldots, P_n$

kas dotu sākotnējo tuvinājumu nākošās līknes izskatam. Šos punktus savā starpā saista matemātiskā sakarība ar Bernšteina polinomu palīdzību. No tā tālāk izriet Bezier funkcija.

Bezier līknes

$$P(t) = \sum_{i=0}^{n} P_i B_{i,n}(t),$$

kur Bernšteina n-tas pakāpes polinoms:

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i},$$

kur

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

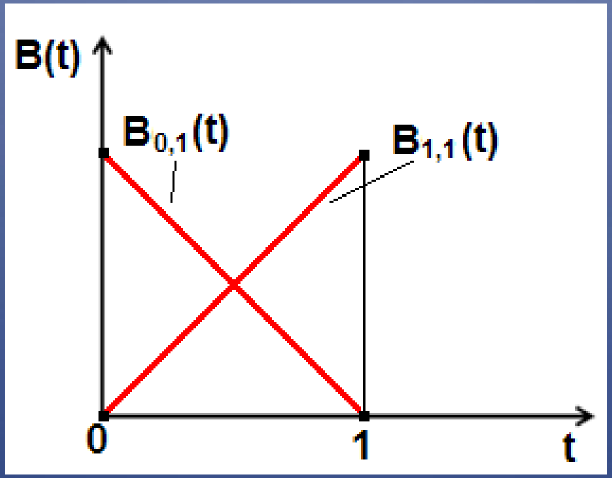
ir kombināciju skaits no *n* pa *i*

Bezier līknes

Pirmās pakāpes polinomu iegūsim, ja t mainās no 0 lidz 1

$$B_{0,1}(t) = (1 - t)$$

$$B_{1,1}(t) = t$$



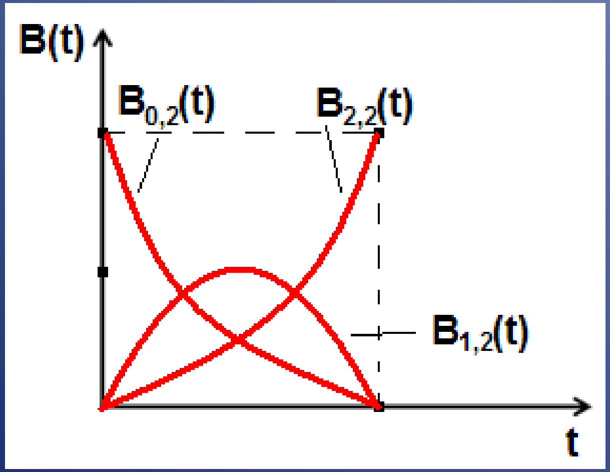
Bezier līknes

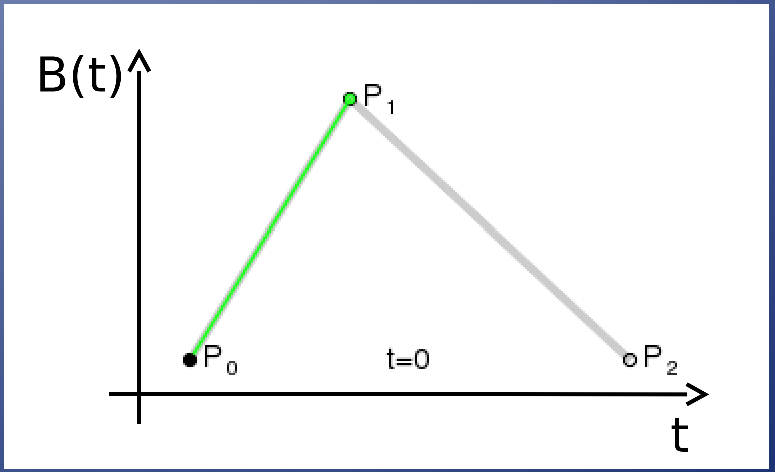
Otrās pakāpes polinomu iegūsim šādi:

$$B_{0,2}(t) = (1-t)^2$$

$$B_{1,2}(t) = 2t(1-t)$$

$$B_{2,2}(t) = t^2$$





Bezier līknes

Trešās pakāpes polinoms:

$$B_{0,3}(t) = (1-t)^3$$

$$B_{1,3}(t) = 3t(1-t)^2$$

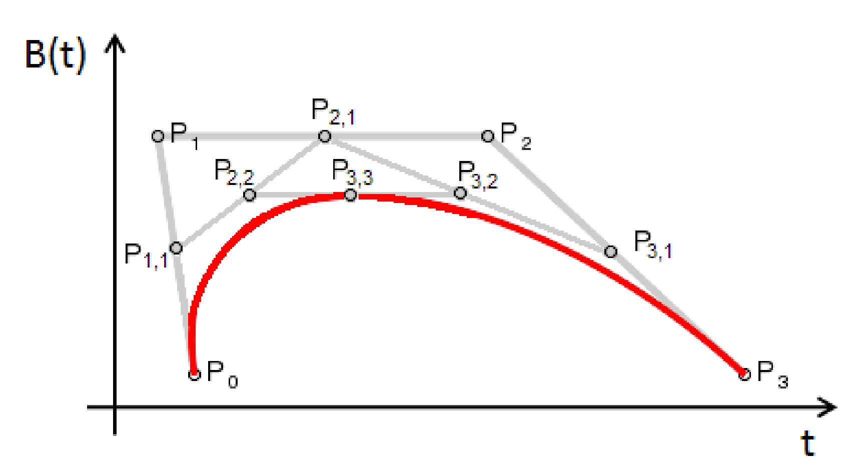
$$B_{2,3}(t) = 3t^2(1-t)$$

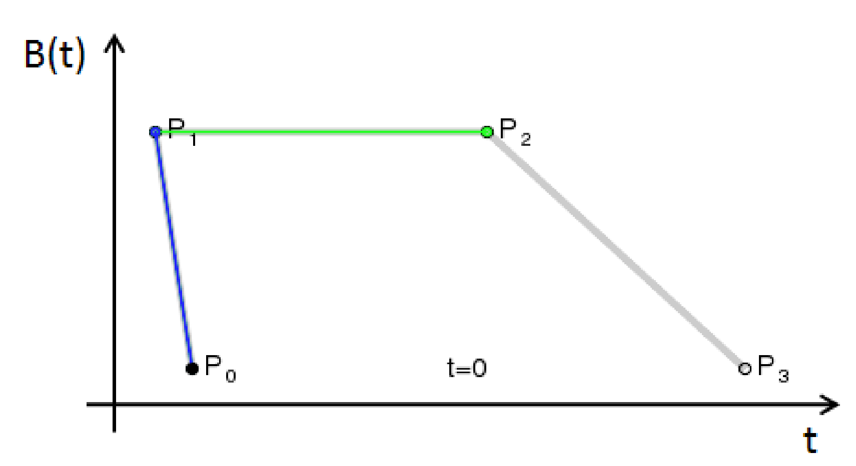
$$B_{3,3}(t) = t^3$$

$$P(t) = t^3 P_3 + 3t^2(1-t)P_2 + 3t(1-t)^2 P_1 + (1-t)^3 P_0$$

Bezier līknes

Kubiskas Bezier līknes veidošanas rezultāt



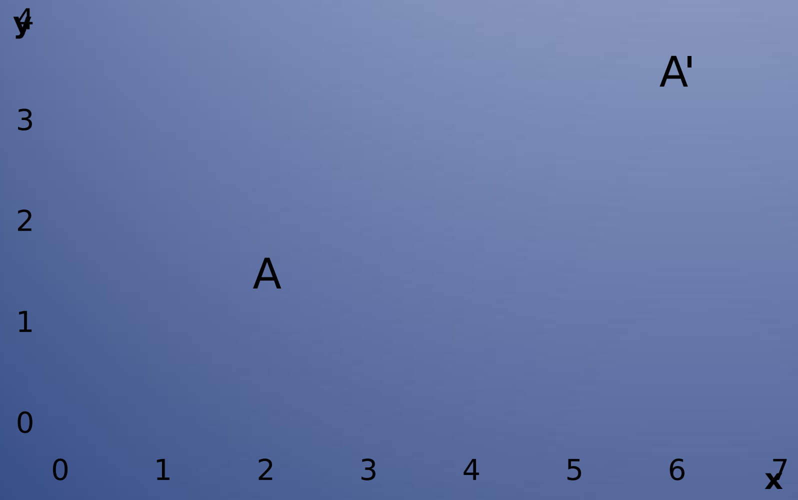


Datorgrafikas un Attēlu apstrādes pamati

6.Lekcija – Matemātiskas operācijas ar grafiskiem objektiem

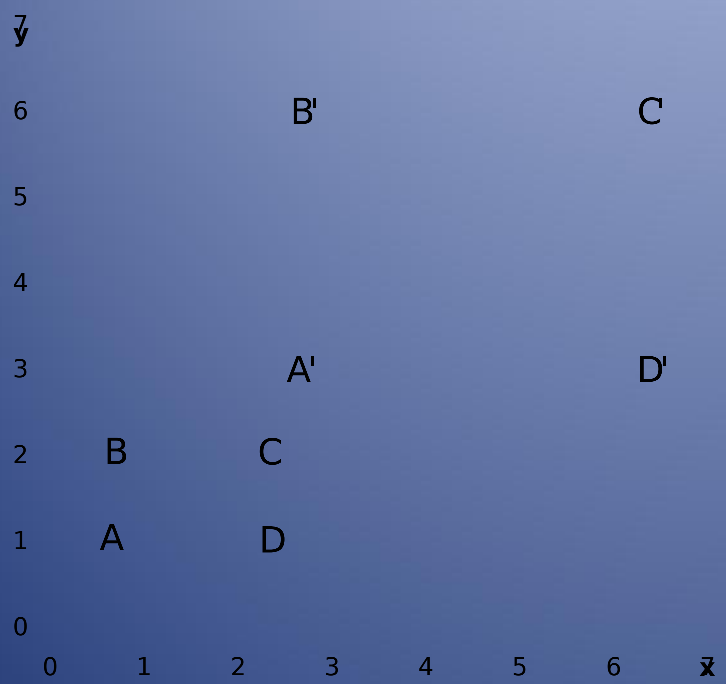
Matemātiskās operācijas ar 2D objektiem

1. 2D mērogošana



$$\begin{pmatrix} 2 & 1 \end{pmatrix} \begin{vmatrix} 3 & 0 \\ 0 & 3 \end{vmatrix} = \begin{pmatrix} 6 & 3 \end{pmatrix}$$

Matemātiskās operācijas ar 2D objektiem



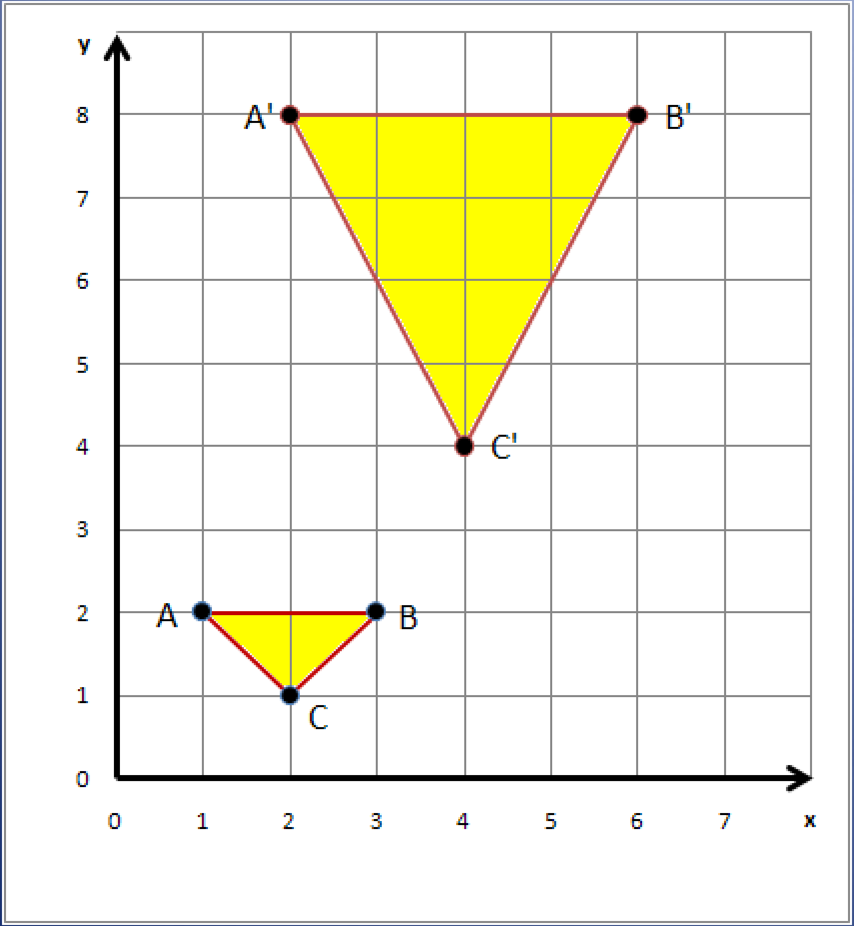
$$A' : \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{vmatrix} 3 & 0 \\ 0 & 3 \end{vmatrix} = \begin{pmatrix} 3 & 3 \end{pmatrix}$$

$$B' : \begin{pmatrix} 1 & 2 \end{pmatrix} \begin{vmatrix} 3 & 0 \\ 0 & 3 \end{vmatrix} = \begin{pmatrix} 3 & 6 \end{pmatrix}$$

$$C' : \begin{pmatrix} 2 & 2 \end{pmatrix} \begin{vmatrix} 3 & 0 \\ 0 & 3 \end{vmatrix} = \begin{pmatrix} 6 & 6 \end{pmatrix}$$

$$D' : \begin{pmatrix} 2 & 1 \end{pmatrix} \begin{vmatrix} 3 & 0 \\ 0 & 3 \end{vmatrix} = \begin{pmatrix} 6 & 3 \end{pmatrix}$$

Matemātiskās operācijas ar 2D objektiem



$$A' : \begin{pmatrix} 1 & 2 \end{pmatrix} \begin{vmatrix} 2 & 0 \\ 0 & 4 \end{vmatrix} = \begin{pmatrix} 2 & 8 \end{pmatrix}$$

$$B' : \begin{pmatrix} 3 & 2 \end{pmatrix} \begin{vmatrix} 2 & 0 \\ 0 & 4 \end{vmatrix} = \begin{pmatrix} 6 & 8 \end{pmatrix}$$

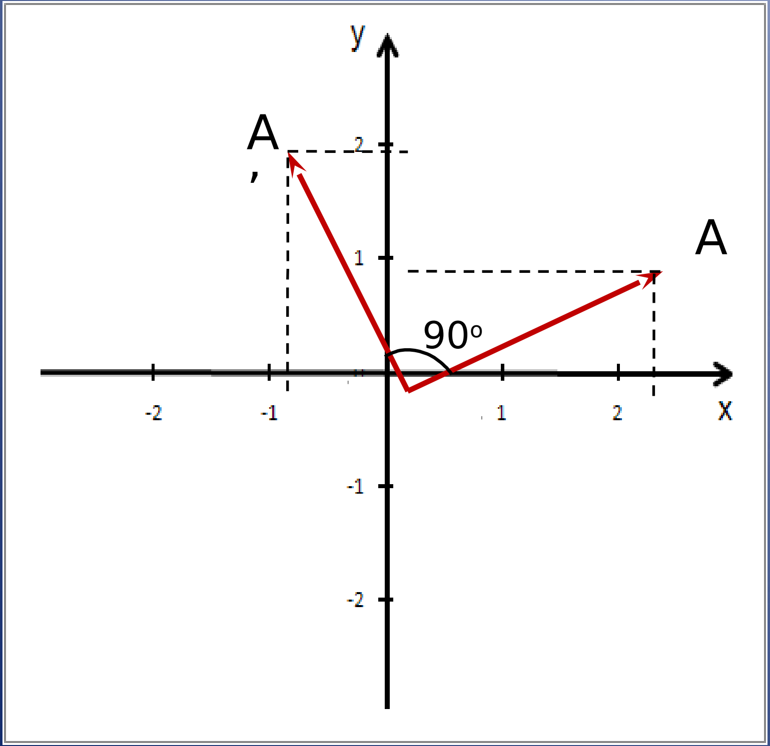
$$C' : \begin{pmatrix} 2 & 1 \end{pmatrix} \begin{vmatrix} 2 & 0 \\ 0 & 4 \end{vmatrix} = \begin{pmatrix} 4 & 4 \end{pmatrix}$$

$$\begin{vmatrix} 1 & 2 \\ 3 & 2 \\ 2 & 1 \end{vmatrix} \cdot \begin{vmatrix} 2 & 0 \\ 0 & 4 \end{vmatrix} = \begin{vmatrix} 2 & 8 \\ 6 & 8 \\ 4 & 4 \end{vmatrix}$$

Matemātiskās operācijas ar 2D objektiem

2. 2D pagriešana (rotācija)

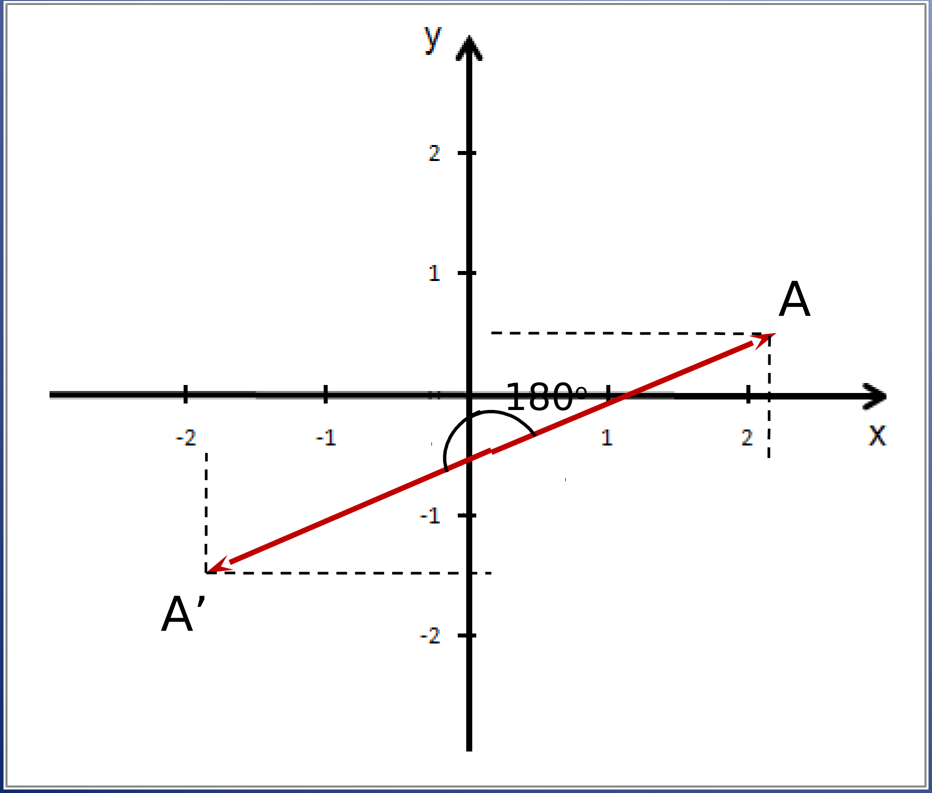
2.1 pagrieziens uz leņķi 90⁰



$$A': \begin{pmatrix} 2 & 1 \end{pmatrix} \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix} = \begin{pmatrix} -1 & 2 \end{pmatrix}$$

Matemātiskās operācijas ar 2D objektiem

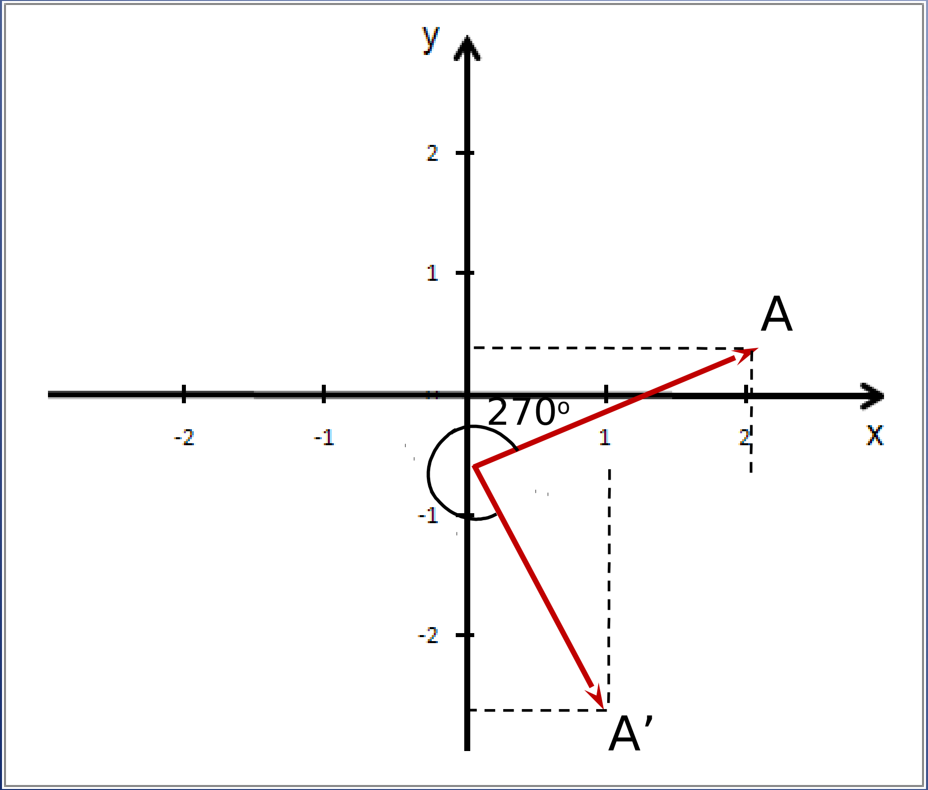
2.2 pagrieziens uz leņķi 180⁰



$$A': \begin{pmatrix} 2 & 1 \end{pmatrix} \begin{vmatrix} -1 & 0 \\ 0 & -1 \end{vmatrix} = \begin{pmatrix} -2 & -1 \end{pmatrix}$$

Matemātiskās operācijas ar 2D objektiem

2.3 pagrieziens uz leņķi 270⁰

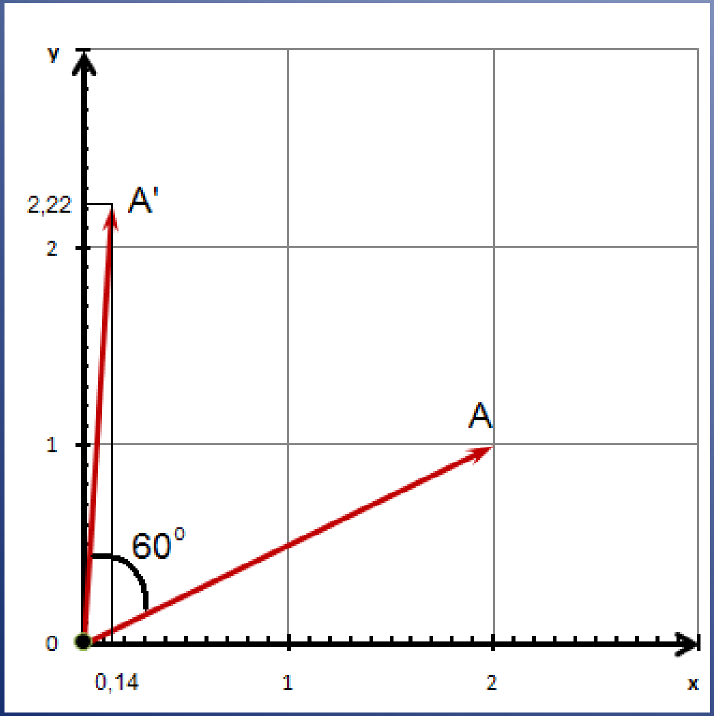


$$A' : \begin{pmatrix} 2 & 1 \end{pmatrix} \begin{vmatrix} 0 & -1 \\ 1 & 0 \end{vmatrix} = \begin{pmatrix} 1 & -2 \end{pmatrix}$$

Matemātiskās operācijas ar 2D objektiem

2.4 Pagrieziens uz patvaļīgu leņķī θ.

Parveidojumu matrica: $\begin{vmatrix} \cos\Theta & \sin\Theta \\ -\sin\Theta & \cos\Theta \end{vmatrix}$



$$A' : \begin{pmatrix} x & y \end{pmatrix} \begin{vmatrix} \cos\Theta & \sin\Theta \\ -\sin\Theta & \cos\Theta \end{vmatrix} =$$

$$= \begin{pmatrix} x\cos\Theta - y\sin\Theta & x\sin\Theta + y\cos\Theta \end{pmatrix}$$

$$Ja \quad \Theta = 60°, \ \cos 60° = 0{,}5, \sin 60° = 0{,}86,$$

$$un \ \ x = 2, y = 1, \ \ ieg\bar{u}sim:$$

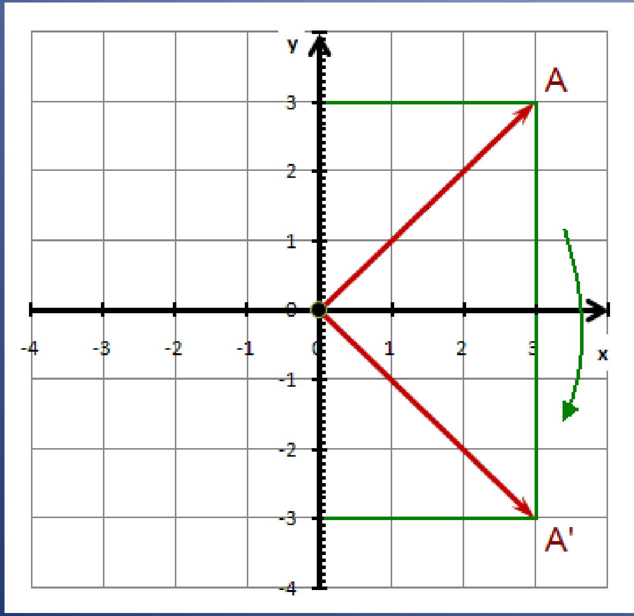
$$A' = \begin{pmatrix} 2 \cdot 0{,}5 - 1 \cdot 0{,}86 & 2 \cdot 0{,}86 + 1 \cdot 0{,}5 \end{pmatrix} =$$

$$= \begin{pmatrix} 0{,}14 & 2{,}22 \end{pmatrix}$$

Matemātiskās operācijas ar 2D objektiem

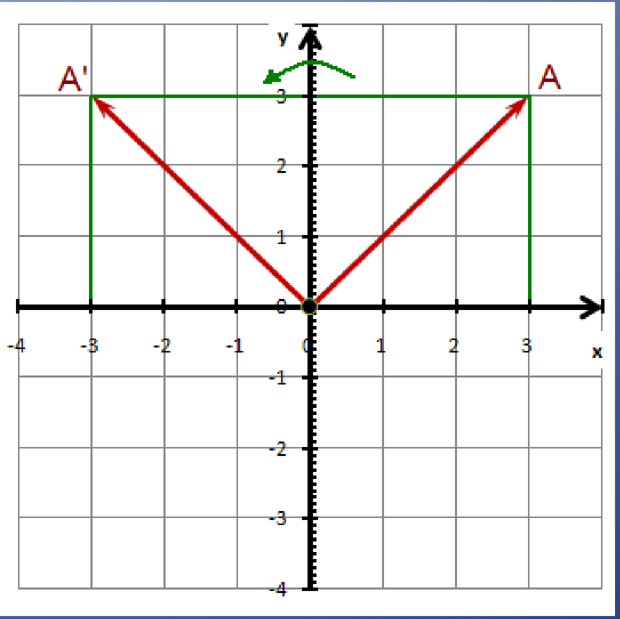
2.5 Pagriešana ap asi

2.5.1 Ap asi x



$$A': \begin{pmatrix} 3 & 3 \end{pmatrix} \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} = \begin{pmatrix} 3 & -3 \end{pmatrix}$$

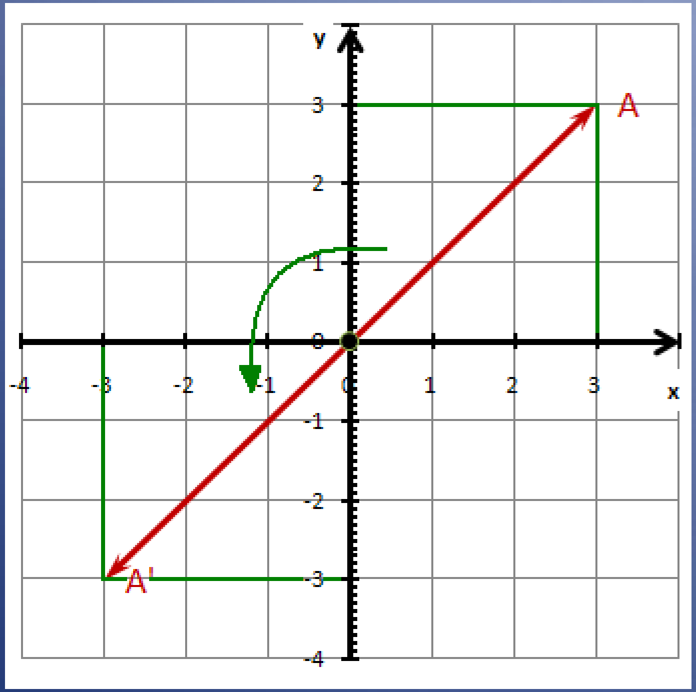
2.5.2 Ap asi



$$A': \begin{pmatrix} 3 & 3 \end{pmatrix} \begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix} = \begin{pmatrix} -3 & 3 \end{pmatrix}$$

Matemātiskās operācijas ar 2D objektiem

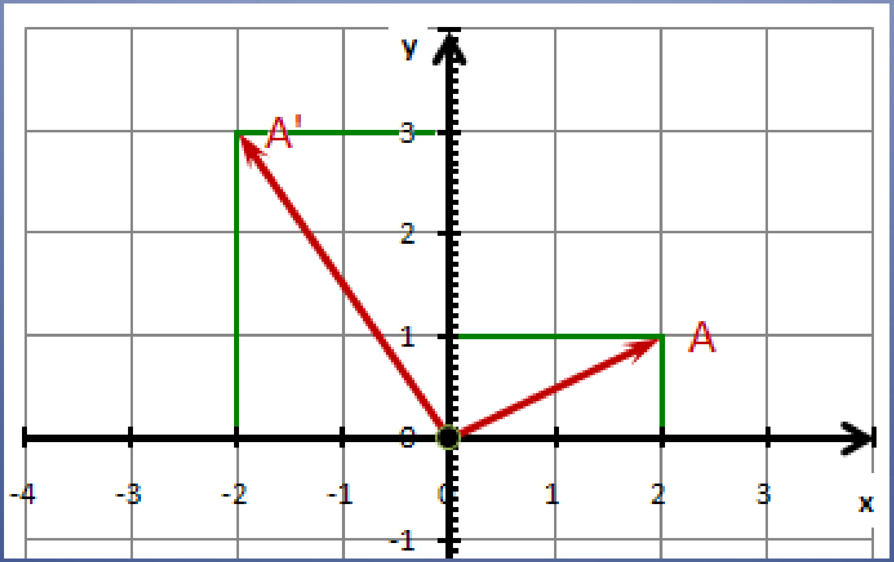
2.5.3 ap x un y asīm



$$A' : \begin{pmatrix} 3 & 3 \end{pmatrix} \begin{vmatrix} -1 & 0 \\ 0 & -1 \end{vmatrix} = \begin{pmatrix} -3 & -3 \end{pmatrix}$$

Matemātiskās operācijas ar 2D objektiem

2.5.4 Pagrieziens un mēroga maiņa



$$A':\begin{pmatrix}2 & 1\end{pmatrix}\begin{vmatrix}-1 & 0\\ 0 & 3\end{vmatrix}=\begin{pmatrix}-2 & 3\end{pmatrix}$$

Homogēnas koordinātes

$$\left(x, y\right) \rightarrow \left(x, y, 1\right)$$

Pārveidojumu matrica (3x3)

$$\left|\begin{array}{cc:c} a & b & 0 \\ c & d & 0 \\ \hdashline 0 & 0 & 1 \end{array}\right|$$

Augšējā kreisā (2x2) matricas daļa nosaka lineārus pārveidojumus (mērogošana, pagriešana, utt.)

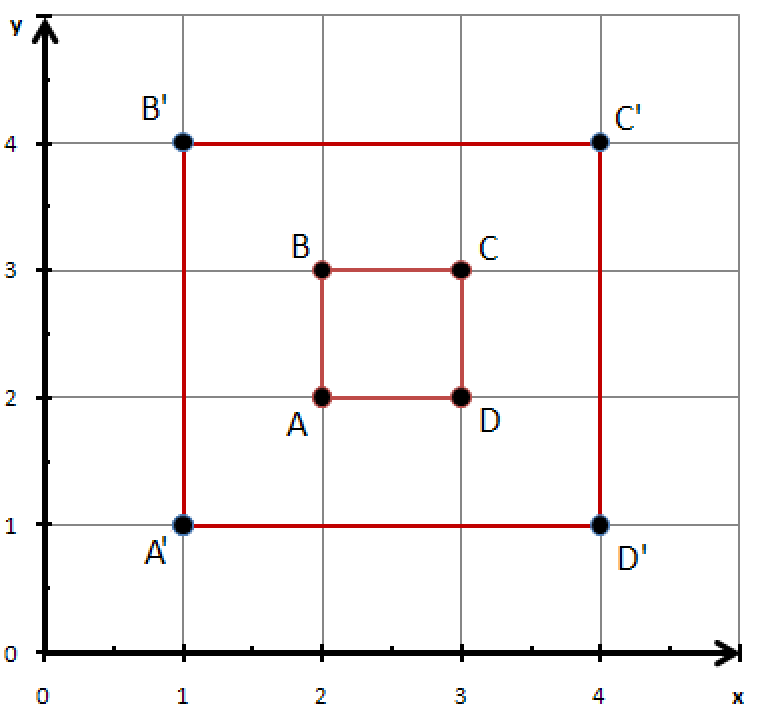
Homogēnas koordinātes

Apakšējā kreisā (1x2) matricas daļa nosaka pārvietojumu:

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \left|\begin{array}{cc:c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \hdashline m & n & 1 \end{array}\right| = \begin{pmatrix} x+m & y+n & 1 \end{pmatrix}$$

Homogēnas koordinātes

Piemērs. Pārvietojumi.



$$C':\begin{pmatrix}3 & 3 & 1\end{pmatrix}\begin{vmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 1 & 1 & 1\end{vmatrix}=\begin{pmatrix}4 & 4 & 1\end{pmatrix}$$

$$A':\begin{pmatrix}2 & 2 & 1\end{pmatrix}\begin{vmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -1 & -1 & 1\end{vmatrix}=\begin{pmatrix}1 & 1 & 1\end{pmatrix}$$

$utt.$

Homogēnas koordinātes

Vispārīgā gadījumā homogēnas koordinātes 2D telpā

$$\left(hx, hy, h\right) \rightarrow \left(x', y', h\right),$$

kur

$$x' = hx, \quad y' = hy,$$

un

$$x = \frac{x'}{h}, \quad y = \frac{y'}{h},$$

Homogēnas koordinātes

Vispārīgā gadījumā pārveidojumu matricu var attēlot sekojošā veidā

$$\left|\begin{array}{cc:c} a & b & p \\ c & d & q \\ \hdashline m & n & s \end{array}\right|$$

Matricas daļa $\left|\begin{matrix} p \\ q \end{matrix}\right|$ nosaka projekcijas

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \left|\begin{matrix} 1 & 0 & p \\ 0 & 1 & q \\ 0 & 0 & 1 \end{matrix}\right| = \begin{pmatrix} x & y & (px+qy+1) \end{pmatrix}$$

Homogēnas koordinātes

Matricas daļa |s| nosaka proporcionālu mērogošanu

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{vmatrix} = \begin{pmatrix} x & y & s \end{pmatrix}$$

Parastās koordinātes (x*,y*)

$$x^* = \frac{x}{s} \qquad y^* = \frac{y}{s}$$

Matemātiskās operācijas ar 3D objektiem

Homogēnās koordinātes

$$\begin{pmatrix} x & y & z \end{pmatrix} \rightarrow \begin{pmatrix} x' & y' & z' & h \end{pmatrix},$$

kur pāriet no homogēnām koordinātēm uz parastajām

$$x = \frac{x'}{h}, \; y = \frac{y'}{h}, \; z = \frac{z'}{h}$$

Matemātiskās operācijas ar 3D objektiem

Vispārējo pārveidojumu 4x4 dimensiju matricu var attēlot šādā veidā

$$\left|\begin{array}{ccc:c} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ \hdashline l & m & n & s \end{array}\right|$$

Matemātiskās operācijas ar 3D objektiem

Matricas daļa

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & i & j \end{vmatrix}$$

nosaka lineārus pārveidojumus.

Matricas dāļa $\begin{vmatrix} l & m & n \end{vmatrix}$ nosaka pārvietojumu.

Daļa $\begin{vmatrix} p \\ q \\ r \end{vmatrix}$ nosaka perspektīvas projekcijas.

Pēdējā matricas dāļā $\begin{vmatrix} s \end{vmatrix}$ nosaka globālo mērogošanu.

Datorgrafikas un Attēlu apstrādes pamati

7.Lekcija – Telpiskie pārveidojumi un projekcijas

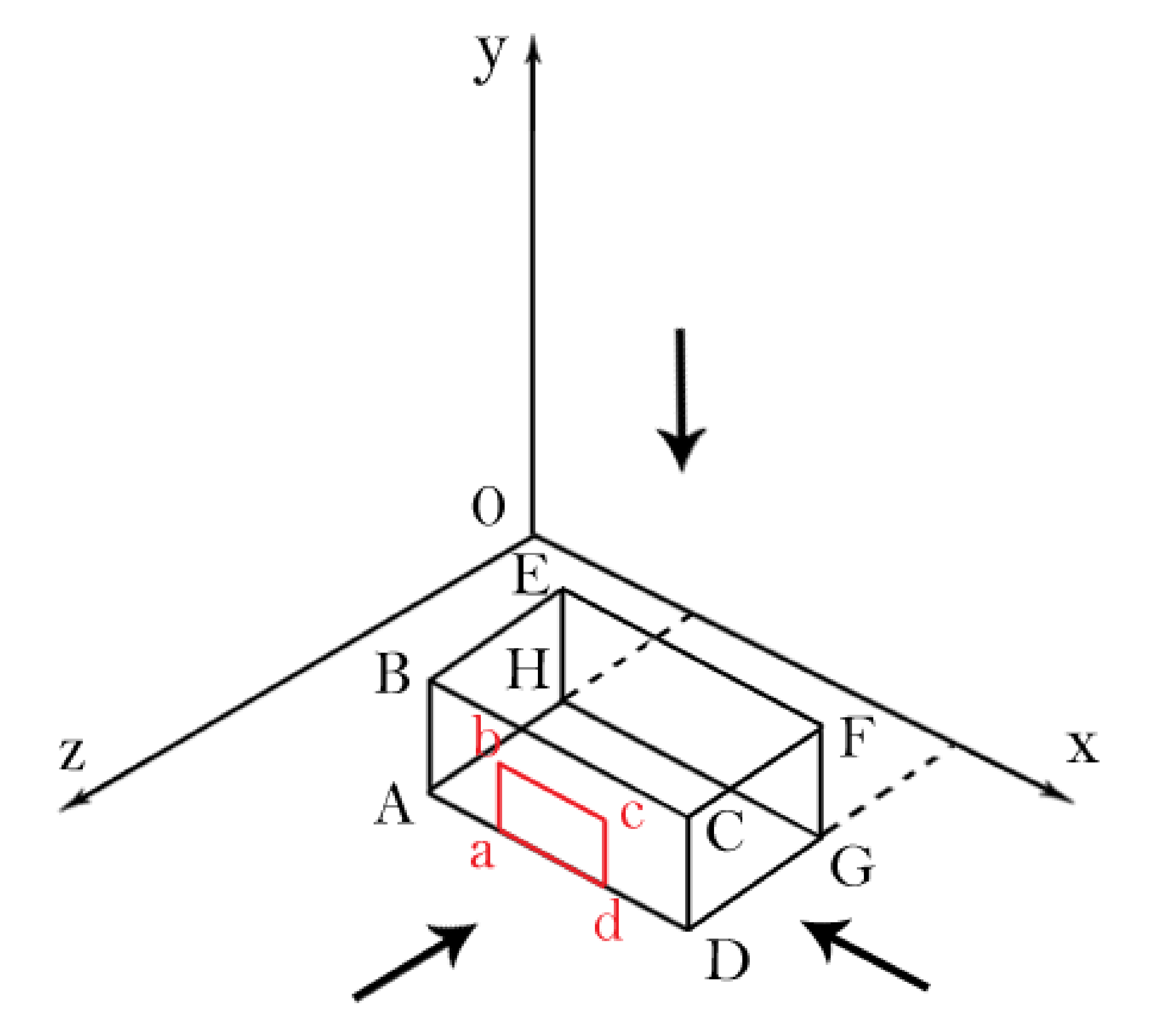
Telpiskie pārveidojumi un projekcijas

- Projekcijas
  - Paralēlās
    - Ortogrāfiskās
    - Aksonometriskās
  - Perspektīvās
    - Vienpunkta
    - Divpunkta
    - Trīspunkta

Ortogrāfiskās projekcijas

Ortogrāfisko projekciju var iegūt ja būs paralēli pārveidojumi. Tas nozīmē, ka projicēšanas stari (paralēlas taisnes) nekrustojās. Šajā gadījumā var saglabāt objekta formu un izmēru.

Ortogrāfiskās projekcijas

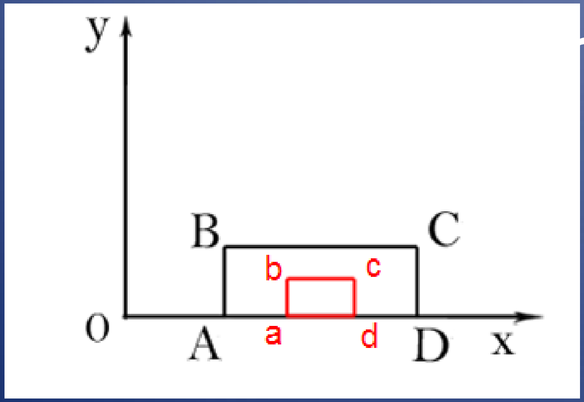


Ortogrāfiskās projekcijas

1. Ortogrāfiskā projekcija uz x0y plakni (z=0)

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{pmatrix} x' & y' & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x & y & 0 \end{pmatrix}$$



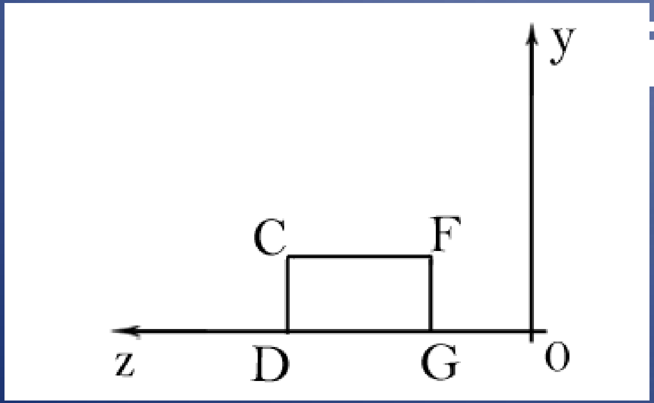
rdinātes

Ortogrāfiskās projekcijas

2. Ortogrāfiskā projekcija uz y0z plakni (x=0)

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \cdot \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{pmatrix} 0 & y' & z' & 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & y & z \end{pmatrix}$$



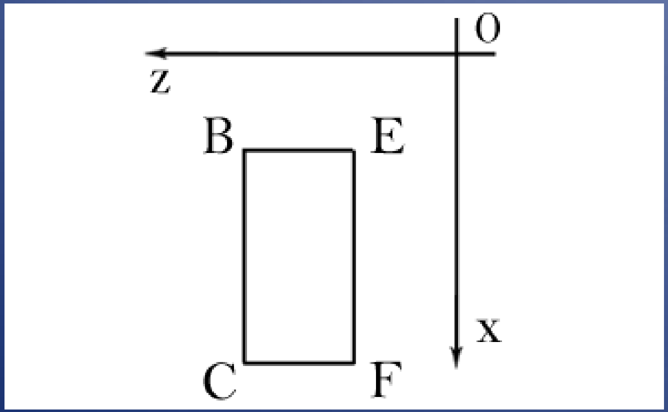
inātes

Ortogrāfiskās projekcijas

3. Ortogrāfiskā projekcija uz x0z plakni (y=0)

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{pmatrix} x' & 0 & z' & 1 \end{pmatrix}$$

$$\begin{pmatrix} x & 0 & z \end{pmatrix}$$



nātes

Perspektīvie pārveidojumi

Pārveidojumu matrica

$$\left|\begin{array}{ccc|c} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ \hline h & k & l & s \end{array}\right| \qquad p, q, r \neq 0$$

Perspektīvie pārveidojumi notiek tad, kad vispārēja pārveidojuma matricā ceturtās kolonnas kaut viens no pirmajiem 3 elementiem (vai nu divi, vai nu visi trīs) nav vienāds ar 0

Perspektīvie pārveidojumi

Atšķirībā no paralēlajiem pārveidojumiem šajā gadījumā paralēlas taisnes krustojas, objekta izmērs samazinās pieaugot attālumam līdz projicēšanas centram un notiek nehomogēna objekta līniju izkropļošana, kas atkarīga no objekta orientācijas un attāluma līdz projicēšanas centram (skatīšanas punktam).

Tas viss palīdz telpiskuma uztverei, bet nesaglabā objekta formu

Perspektīvie pārveidojumi

Ja viens no p, q, r elementiem nav vienāds ar nulli, bet pārējie divi ir vienādi ar nulli, tad šādus gadījumus sauc par vienpunkta perspektīvas pārveidojumiem.

Apskatīsim vienpunkta perspektīvos pārveidojumus ar projicēšanas centru uz:

Perspektīvie pārveidojumi

1. x asi:

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{pmatrix} x & y & z & px+1 \end{pmatrix}$$

Parastās koordinātes

$$\begin{pmatrix} x' & y' & z' & 1 \end{pmatrix} = \begin{bmatrix} \frac{x}{px+1} & \frac{y}{px+1} & \frac{z}{px+1} & 1 \end{bmatrix}$$

un projicēšanas centrs punktā

$$\begin{pmatrix} -\frac{1}{p} & 0 & 0 & 1 \end{pmatrix}$$

Perspektīvie pārveidojumi

2. y asi:

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{pmatrix} x & y & z & qy+1 \end{pmatrix}$$

Parastās koordinātes

$$\begin{pmatrix} x' & y' & z' & 1 \end{pmatrix} = \begin{bmatrix} \dfrac{x}{qy+1} & \dfrac{y}{qy+1} & \dfrac{z}{qy+1} & 1 \end{bmatrix}$$

un projicēšanas centrs punktā

$$\begin{pmatrix} 0 & -\dfrac{1}{q} & 0 & 1 \end{pmatrix}$$

Perspektīvie pārveidojumi

3. z asi:

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{pmatrix} x & y & z & rz+1 \end{pmatrix}$$

Parastās koordinātes

$$\begin{pmatrix} x' & y' & z' & 1 \end{pmatrix} = \begin{bmatrix} \frac{x}{rz+1} & \frac{y}{rz+1} & \frac{z}{rz+1} & 1 \end{bmatrix}$$

un projicēšanas centrs punktā

$$\begin{pmatrix} 0 & 0 & -\frac{1}{r} & 1 \end{pmatrix}$$

Perspektīvie pārveidojumi

Divpunktu perspektīvie pārveidojumi

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{pmatrix} x & y & z & px+qy+1 \end{pmatrix}$$

ar parastajām koordinātēm

$$\begin{pmatrix} x' & y' & z' & 1 \end{pmatrix} = \begin{bmatrix} \dfrac{x}{px+qy+1} & \dfrac{y}{px+qy+1} & \dfrac{z}{px+qy+1} & 1 \end{bmatrix}$$

satur divus projicēšanas centrus:

Pirmo uz x ass punktā $\begin{pmatrix} -\frac{1}{p} & 0 & 0 & 1 \end{pmatrix}$

Otro uz y ass punktā $\begin{pmatrix} 0 & -\frac{1}{q} & 0 & 1 \end{pmatrix}$

Datorgrafikas un Attēlu apstrādes pamati

8.Lekcija – Attēlu apstrādes pamati

1. Attēlu apstrādes īpatnības

Kas ir kopīgs datorgrafikai un attēlu apstrādei?

1. Pirmkārt tas, kā attēli vai grafiskie objekti izveidojas uz ekrāna kā pikseļu matrica vai vesela skaitļa režģis (rastru režģis)
2. Cita īpašība ir tā, ka datorgrafikas uzdevumos objekti veidojas no ģeometriskām pamatstruktūrām vai primitīviem, no kuriem pēc tam veidojas attēls. Attēlu apstrādes uzdevumos tieši otrādi – ir jāizdala attēlu pamatdaļas (kontūri, segmenti, apgabali utt.)

Tādēļ datorgrafikas un attēlu apstrādes uzdevumus var uzskatīt kā tiešus vai pretējus uzdevumus.

Ļoti bieži attēlā vizuāli nav iespējams novērot dažādas detaļas. Tas var būt saistīts ar dažādiem iemesliem, piemēram: slikts kontrasts, troksnis un citi. Sakarā ar to ir nepieciešams pielietot attēla kvalitātes uzlabošanas metodes.

Bet no sākuma rodas attēlu kodēšanas problēma.

a) 

 b)

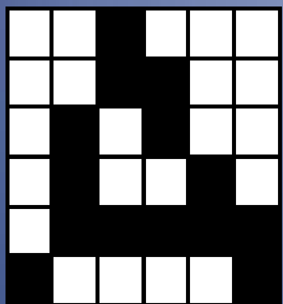
c) 

 d)

a) automašīnas valsts numurs atrodas koka ēnā, un b) parādīts apstrādāts attēls, kur valsts numurs labi redzams.
c) attēlota automašīna, kurai tieši valsts numurā atspīd saule un tāpēc ciparus uz numura noteikt nav iespējams, bet d) parādīts apstrādāts attēls, kur cipari un burti uz valsts numura ir labi redzami.

2. Attēlu kodēšana

1) Melnbalts attēls

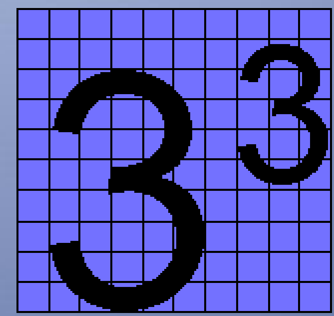


X=||

$i \in [1 : n]$

$j \in [1 : m]$

$x_{ij} = \begin{cases} 1 \\ 0 \end{cases} \rightarrow 1$ bits



n = m = 6

Kodēšana tiks izpildīta četros virzienos:

1) Pa x ass virzienu (nepārtrauktio iekrāsojumu skaits);
2) Pa y ass virzienu;
3) Pamatdiagonāles virzienā;
4) Palīgdiagonāles virzienā

Ja n x n tad iegūsim 6n - 2 skaitļus:

n + n + 2(2n - 1) = 6n-2

Tālāk katrā virzienā tiek veikta saspiešana

0 0 1 1 1 1 1 2 2 2 0 0 0

0 1 2 0

Jauno kodu var pārveidot binārā formā

00 01 10 00

Pēc tam katram virzienam var piešķirt bināro kodu:

1) x ass virzienā – 01

2) y ass virzienā – 10

. . .

Rezultātā iegūsim saspiesto bināro kodu.

2) Attēls ar pustoņiem

Ja attēlam būs pustoņi, tad katram matricas elementam būs vērtības

$0 \leq x_{ij} \leq 255$ -> 1 baits

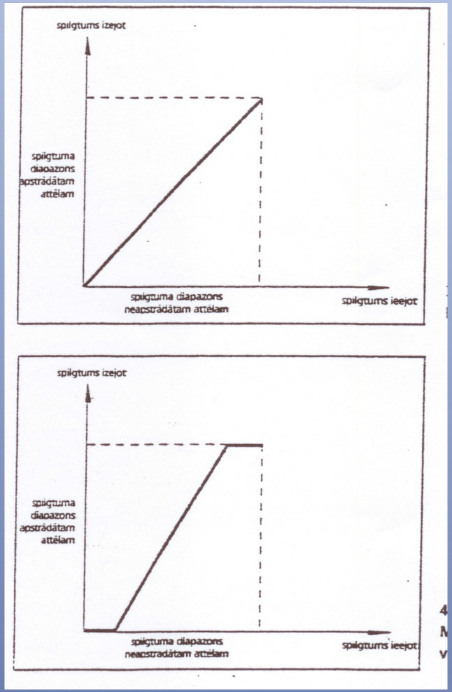
Vispārīgā gadījumā katram elementam (i,j) atbilst intensitātes funkcijas vērtība g(i,j).

3. Kontrasta raksturojums

Lai sāktu pētīt, kas īsti ir kontrasts, kādas ir tā izmaiņas, kā tās aprakstīt, vispirms jādefinē, kas ir kontrasts? Vienkāršākā atbilde – kontrasts ir starptoņu skaits attēlā; ļoti kontrastainā bildē starptoņi vai pustoņi būs tikai divi – melns un balts. Jo vairāk tonālu nianšu – jo mazāks kontrasts (bet vairāk informācijas!). Tas pats attiecas arī uz krāsu attēliem.

Palielinot kontrastu, attēla pustoņi paliek gaišāki vai tumšāki.

Praktiski kontrasta izmaiņas lieto, gadījumos, kad jāuzlabo bildes vizuālais izskats (piemēram, sliktas kvalitātes, neskaidras utt.); gadījumos, ja to prasa attēla izmantošanas mērķi (piemēram, attēlu redzēs tikai pa lielu attālumu, tad kontrastu liek lielāku, lai attēls labāk “nolasītos”), ja attēlus tālāk apstrādā dators (lai labāk nolasītu objektus un kontūras), un, gadījumos, ja to prasa mākslinieka iecere. Visizplatītākais defekts ir neasi attēli ar zemu kontrastu (miglaini, izplūduši).

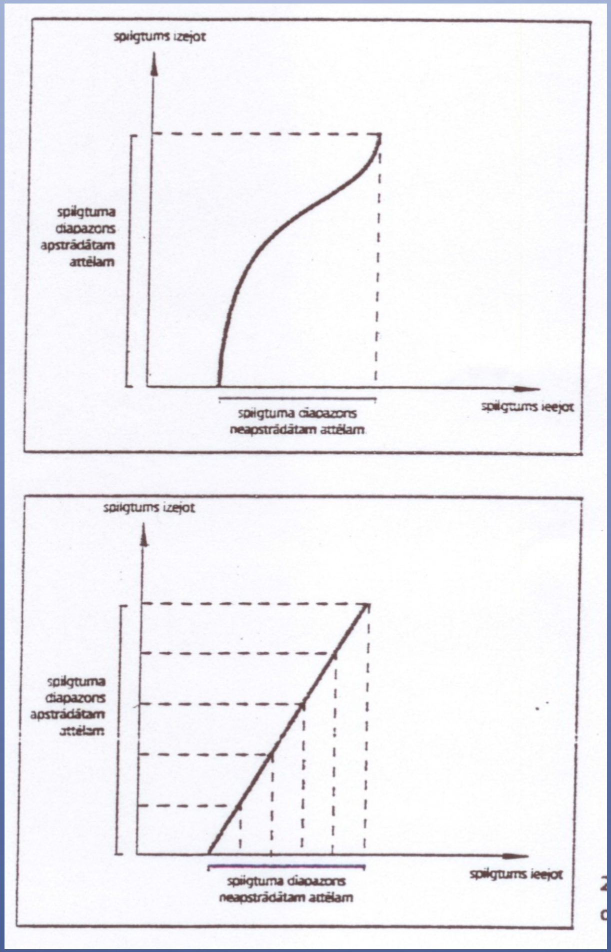


1. Kontrasta izmaiņas
Pilnībā izmantoti visi punkti.

2. Kontrasta izmaiņas
Nemainīti visgaišākie un vistumšākie punkti

1. zīmējumā redzams attēls, kurš pilnībā izmanto visu savu spilgtuma diapazonu. 2. zīmējums rāda attēlu, kuram nav apstrādāti tikai 2 līmeņi – maksimālais un minimālais, t.i. visgaišākās vietas attēlā un vistumšākās. Vienkāršiem vārdiem runājot – gaišākās vietas paliek gaišākas, tumšākās – tumšākas. Šis paņēmiens ir visvienkāršākais (arī vissubjektīvākais) kontrasta izmaiņu vai kontrasta pieauguma panākšanai attēlā.

Visi pieminētie paņēmieni attiecas gan uz melnbaltiem, gan uz krāsainiem attēliem. Dažādiem gadījumiem tiek izmantotas dažādas lineārās funkcijas. Iespējams pielietot pārtrauktas un nepārtrauktas



3. Kontrasta izmaiņas nepārtrauktam attēlam

4. Kontrasta izmaiņas digitālajam attēlam

Piemēram, digitālajam attēlam ir nokvantēti J līmeņi ieejā un J līmeņi izejā.

4. Kontūru izdalīšana

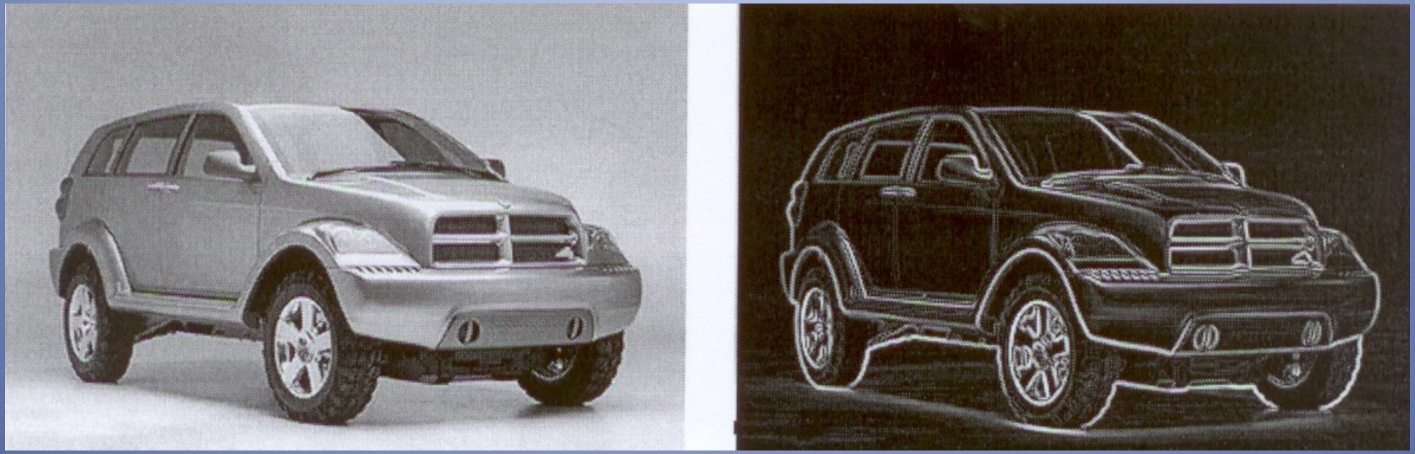
Diskrētas funkcijas g(i, j) gradients punktā (i, j) var tikt vērtētas ar Robertsa krustoperātoru

$$\text{grad}(i,j) \approx [g(i,j)-g(i+1,j+1)]^2+[g(i,j+1)-g(i+1,j)]^2$$

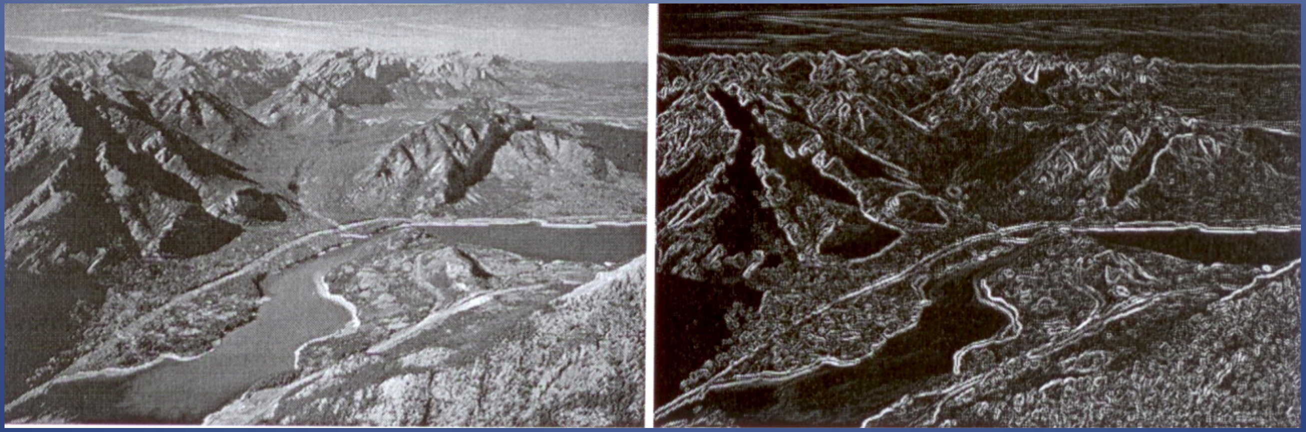
t.i. Katram elementam (I,j) tiek aplūkots logs 2 x 2, kurā diagonālie elementi ir saistīti ar atņemšanas operāciju.

i+1, j	i+1, j+1
i, j	i, j+1

Operatora pielietošanas rezultātā iegūto attēlu parasti sauc par gradiento attēlu, bet šādu gradiento transformācijas procesu sauc par kontūra izdalīšanu.



5. Attēls ar mašīnu pirms un pēc kontūru izdalīšanas



6. Attēls ar dabu pirms un pēc kontūru izdalīšanas

5. Trokšņu attīrīšanas algoritms

Trokšņu attīrīšanas metodē izrēķinām vidējo lielumu no elementiem izvēlētajā logā 3 x 3

g(i-1), j-1)	g(i-1, j)	g(i-1, j+1)
g(i, j-1)	g(i, j)	g(i, j+1)
g(i+1, j-1)	g(i+1, j)	g(i+1, j+1)

$$g'_{ij} = \begin{cases} \frac{1}{8}\left(\sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} g_{kl} - g_{ij}\right), \\ ja \quad \left[g_{ij} - \frac{1}{8}\left(\sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} g_{kl} - g_{ij}\right)\right] > \varepsilon; \\ g_{ij}, ja \quad \left[g_{ij} - \frac{1}{8}\left(\sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} g_{kl} - g_{ij}\right)\right] \le \varepsilon \end{cases}$$

kur ε uzdotais slieksnis.