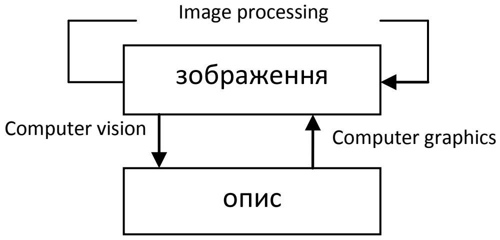
Вступ до комп’ютерної графіки

Комп’ютерна графіка це наука предметом вивчення якої є створення, зберігання та обробка моделей та їх зображень за допомогою комп’ютерних засобів.



Класифікація зображень:

1. За кольоровістю
   a. монохромні
   b. відтінки сірого
   c. кольрові
2. За динамікою
   a. Статичні
   b. Динамічні
3. За внутрішнім поданням
   a. Растрові
   b. Векторні
4. За об’ємністю
   a. Двовимірні
   b. Тривимірні

Кольори в комп’ютерній графіці

Колір – суб’єктивна характеристика світла, яка відображає здатність людського зору розрізняти частоту електромагнітних коливань у області видимого світла.

Формування відчуття кольору залежить від таких складових: джерела освітлення та умов розповсюдження випромінювання, властивостей об’єктів що сприймаються, зорової системи людини.

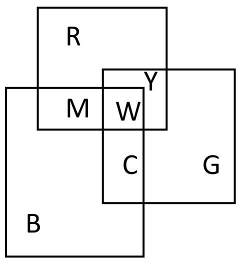
Характеристики кольору:

1. Яскравість - насичення кольору
2. Насиченість - віддалення сірого кольору
3. Світлота (ясність) – ступінь віддалення від білого кольору
4. Кольоровий тон

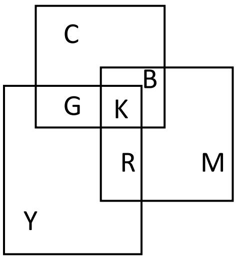
Фізичні принципи формування відтінків об’єктів

В комп’ютерній графіці розрізняють 2 види об’єктів: випромінюючі (монітори, світло-діодні матриці) та не випромінюючі (папір, світлофільтри).

Для випромінюючих об’єктів характерне адитивне формування відтінків, коли потрібний колір отримується як результат змішування трьох базових кольорів R G B.

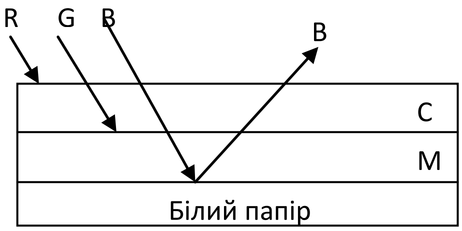


Адитивна модель



Субтрактивна модель

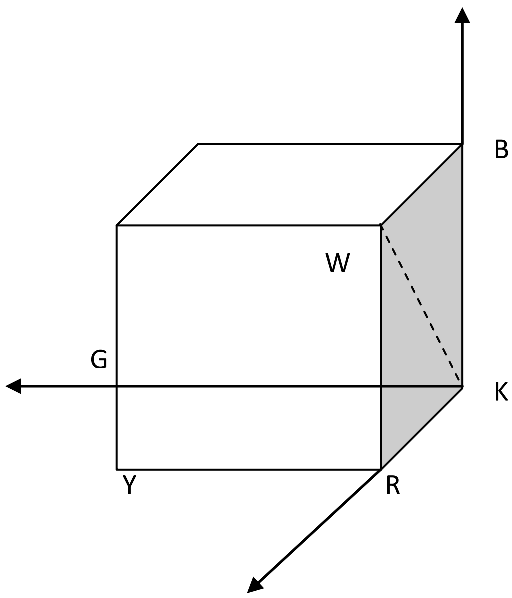
Кольори однієї моделі доповнюють кольори іншої моделі. Доповнюючий колір – це колір, який доповнює даний колір до білого.



Кольорові моделі

Призначення кольорової моделі - надати засіб опису відтінку кольору у межах деякої кольорової гамми, у тому числі і для виконання інтерполяції кольорів.

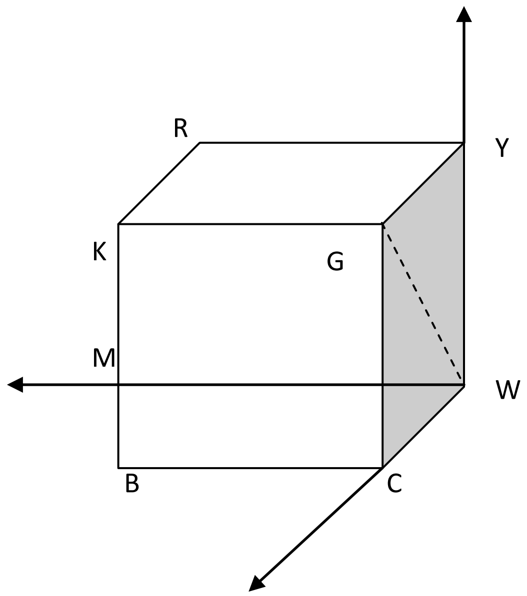
1. Модель RGB – апаратна орієнтована модель, яка використовується для адитивного формування відтінків зображень.



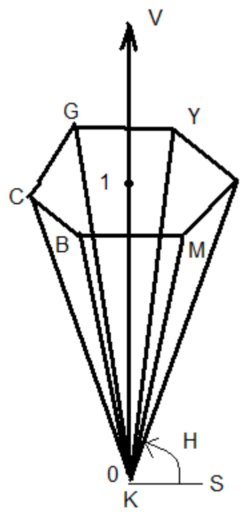
Трикутник паскаля об'єднує найбільш яскраві кольори. Він об'єднує точки R, G, B.

Кольори, що знаходяться біля штрихованої лінії – відтінки сірого.

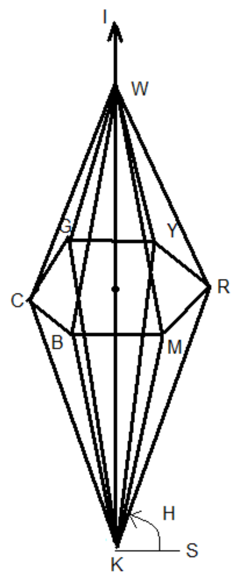
2. Модель CMY - апаратно орієнтована модель, яка використовується у поліграфії. Відповідає субтрактивному способу формування відтінків.



3. Модель HSV (Hue Saturation Value / Lightness)



4. Модель HSI (Hue Saturation Intensity)/HLS (Hue Lightness Saturation)/BHS (Brightness Hue Saturation)



Hue – кольоровий тон
Saturation - насиченість

$$I = \frac{R+G+B}{3}$$

$$S = 1 - \frac{3}{R+G+B} * \min(R,G,B)$$

$$H = \begin{cases} \theta \ \text{ при } B \le G \\ 360^\circ - \theta \ \text{ при } B > G \end{cases}$$

$$\theta = \arccos\left(\frac{(R-B)+(R-G)}{2\sqrt{(R-G)^2+(R-B)(G-B)}}\right)$$

- 0 <= H < 120
  (RG сектор)
  R=I*(1+(S*cos H)/cos(60-H))
  G=3*I-(R+B)
  B=I(1-S)
- 120 <= H < 270
  (GB sector)
  H'=H-120
  R=I(1-S)
  G=I*(1+(S*cos H)/cos(60-H'))
- 240 <= H < 360
  (BR sector)
  H' = H – 240
  R = 3*I – (G + B)
  G = I (1 - S)
  B = I(1 + (S*cos H')/cos(60 - H))

$$I = \frac{1}{3}$$

$$S = 1$$

$$H = \theta = 0^\circ$$

5. Модель YIQ
   Y – яскравість
   I, Q – кольоро-різнісні сигнали
   $Y = Rr + Gg + Bb$
   Маленькі літери – еталони сприйняття людиною різних базових кольорів.
   Аппаратно-орієнтована модель яка використовується в телебаченні та слугує для скорочення смуги частот за рахунок врахування психофізіологічних особливостей зору людини.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,274 & -0,322 \\ 0,211 & -0,522 & 0,311 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1,0 & 0,956 & 0,623 \\ 1,0 & -0,272 & -0,648 \\ 1,0 & -1,105 & 0,705 \end{bmatrix} * \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

Растрова графіка

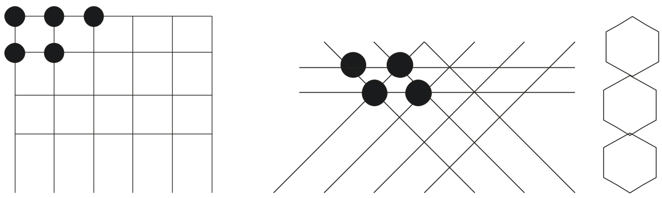
Якість растрової графіки залежить від оптичної роздільності, яка вимірюється кількістю точок на одиницю довжини. Розрізняють:

- роздільність оригінала [dpi – dots per inch]
- роздільність екранного зображення [pixel]. Елементарну точку растра для екранного зображення називають пікселом. Розмір пікселу залежить від екранної роздільної здатності, роздільності оригіналу та масштабу відображення.
- Роздільність друкованого зображення [lpi – lines per inch, лініатура]

В поліграфії при формуванні зображення можуть бути застосовані: амплітудна модуляція та частотна модуляція. Вважається що більш якісним виглядає зображення з частотною модуляцією.

<u>Растр</u> :

1. У поліграфії – подання графічної інформації за допомогою точок різної величини або таких що знаходяться на різній відстані.
2. У комп'ютерній графіці – порядок розташування точок, які називають растровими елементами. Розрізняють прямокутний (квадратний), трикутний та гексагональний (шестикутний) растр.



Генерація графічних примітивів

Генерація відрізків

Загальні вимоги до зображення відрізків:

1. Кінці відрізка мають знаходитись в заданих точках
2. Відрізки повинні мати вигляд прямих
3. Яскравість уздовж відрізка повинна бути постійною та не залежати від довжини та нахилу

Жодна з цих умов не може бути повністю виконана на растровому моніторі:

1. Кінці відрізка у загальному випадку розташовуються на пік селах, які є лише наближенням до математичних координат відрізка
2. Відрізок апроксимується набором пік селів і лише в окремих випадках буде виглядати прямим
3. Яскравість для різних відрізків та уздовж того самого відрізка у загальному випадку різна тому що відстань між центрами пік селів буде відрізнятись

Розглянемо задачу побудови растрового зображення відрізка що з'єднує точки з координатами (x1, y1),(x2, y2). Будемо вважати що має місце нерівність: 0 < y2-y1 <= x2-x1

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

$$y = kx + b$$

$$k = \frac{y_2 - y_1}{x_2 - x_1}$$

$$b = y_1 - kx_1$$

$$k = (y_2 - y_1)/(x_2 - x_1)$$

```
for (x=x1; x<=x2; x++)
{
      putpixel(x, (int)(kx+b), colour);
}
```

Приклад: нехай потрібно побудувати растровий відрізок в точках (2,3),(6,6)

k=3/4   b=3/2

X	2	3	4	5	6
Y	3	3,75	4,5	5,25	6
Y окр	3	4	4	5	6

Оскільки при збільшенні координати x на 1 координата y збільшується на значення k то можна вдосконалити алгоритм щоб зменшити кількість операцій що виконується.

```
k = (y2-y1)/(x2-x1);
y = y1;
for (x = x1; x <= x2; x++ , y += k)
{
      putpixel(x, (int)(y), colour);
}
```

X	2	3	4	5	6
Y	3	3,75	4,5	5,25	6
Y окр	3	4	4	5	6

Наведені найпростіші алгоритми мають наступні недоліки:

1. Виконують операції над числами з плаваючою точкою
2. На кожному кроці виконується операція округлення

Алгоритм ЦДА/DDA (Цифровий Диференціальний Аналізатор/Digital Differential Analyzer)

$$\frac{dy}{dx} = \frac{Py}{Px}$$

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{y - y_1}{x - x_1}$$

Алгоритм ЦДА формує дискретну апроксимацію безперервного рішення рівняння прямої, проведеної через 2 відомі точки.

Розрізняють 2 види алгоритму ЦДА: звичайний та несиметричний.

У звичайному алгоритмі ЦДА на початку задають кількість кроків N, за яку потрібно зробити апроксимацію відрізка.

$$\begin{cases} x_{i+1} = x_i + \dfrac{P_x}{N} \\ y_{i+1} = y_i + \dfrac{P_y}{N} \end{cases}$$

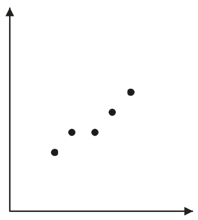
X	2	2.8	3.6	4.4	5.2	6
X окр..	2	3	4	5	5	6
У	3	3.6	4.2	4.8	5.4	6
У окр..	3	4	4	5	5	6

В несиметричному ЦДА використовується різний крок по осям X та У.

Нехай $P_x > P_y$, то якщо X збільшується на 1, то У має збільшуватись на величину $\dfrac{Py}{Px}$.

$$\begin{cases} x_{i+1} = x_i + 1 \\ y_{i+1} = y_i + \dfrac{P_y}{P_x} \end{cases}$$

X	2	3	4	5	6
У	3	4	4	5	6



<u>Параметричне рівняння</u>:

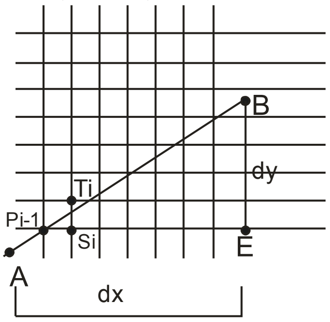
$$\begin{cases} x_{i+1} = x_i + t(x_2 - x_1) \\ y_{i+1} = y_i + t(y_2 - y_1) \end{cases}$$

$$t \in [0,1]$$

$$t = \frac{1}{\max(|x_2 - x_1|, |y_2 - y_1|)}$$

X	2	3	4	5	6
У	3	4	4	5	6

Алгоритм Брезенхема



Визначимо координати точки C, через яку проходить ідеальний відрізок: $C(r+1, q+S)$

$$\frac{Py}{Px} = \frac{dy}{dx}$$

$$\frac{y}{x} = \frac{dy}{dx} \qquad (*)$$

$$\frac{q+S}{r+1} = \frac{dy}{dx} \qquad (**)$$

$$S = \frac{dy}{dx}(r+1) - q$$

$$T = 1 - \frac{dy}{dx}(r+1) + q$$

$$S - T = \frac{dy}{dx}(r+1) - q - 1 + \frac{dy}{dx}(r+1) - q = 2\frac{dy}{dx}(r+1) - 2q - 1$$

$$dx(S-T) = 2dy(r+1) - 2dxq - dx \qquad (***)$$

$$dx > 0$$

Оскільки величина dx є завжди додатньою то вибір наступної точки буде залежати від лівої частини рівняння (***).

Після спрощення повернемось до точки Pi-1 та її координат

$$di = 2dyr + 2dy - 2dxq - dx = 2dy(r+1) - dx(2q+1) =$$

$$= 2(dyr - dxq) + 2dy + dx = 2(x_{i-1}dy - y_{i-1}dx) + 2dy - dx \qquad (V)$$

$$d_{i+1} = 2(x_i dy - y_i dx) + 2dy - dx \qquad (VV)$$

$$d_{i+1} - d_i = 2(x_i dy - y_i dx) - 2(x_{i-1}dy - y_{i-1}dx) = 2dy(x_i - x_{i-1}) - 2dx(y_i - y_{i-1})$$

$$d_{i+1} = d_i - 2dx(y_i - y_{i-1}) + 2dy$$

Дане рівняння дозволяє ітеративно обчислювати управляючий коефіцієнт d за допомогою якого в якості наступного вибирається або піксель Si або Ti. Якщо di величина додатня, то вибирається піксель Ti , при цьому координата y збільшується на 1, а значення di+1 :

$$d_{i+1} = d_i - 2(dx - dy)$$

Інакше вибирається піксель Si

$$y_i = y_{i-1}$$

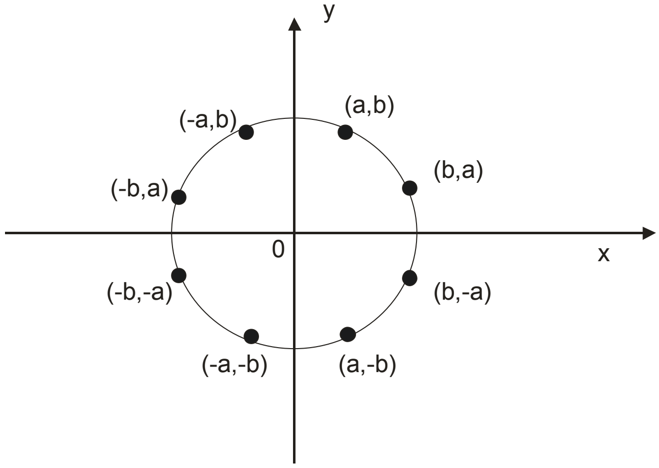
$$d_{i+1} = d_i + 2dy$$

Растрова розгортка кіл

$$1)\begin{cases} x = R * \cos\alpha \\ y = R * \sin\alpha \end{cases}$$

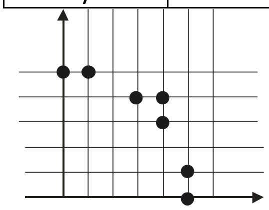
$$2) x^2 + y^2 = R^2$$

1:



$$R = 5, \quad 0^\circ \le \alpha \le 90^\circ, \quad \Delta\alpha = 15^\circ$$

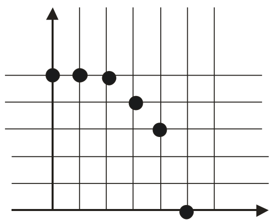
α	90°	75°	60°	45°	30°	15°	0°
X	0	]1.3[= 1	]2.5[= 3	]3.5[=4	4	5	5
y	5	]4.8[= 5	]4.3[= 4	]3.5[=4	3	1	0



2:

$$y = \pm\sqrt{R^2 - x^2}$$

X	0	1	2	3	4	5
Y	5	]4.8[=5	]4.6[=5	4	3	0



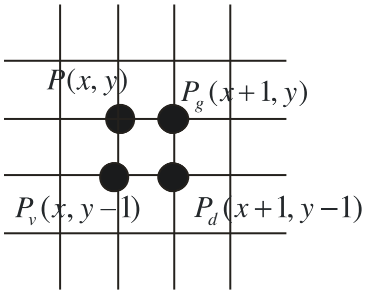
Алгоритм Брезенгхема

Коло не буде абсолютно ідеальним.

$$x^2 + y^2 - R^2 = 0$$

$$\varepsilon_i = x^2{}_i + y^2{}_i - R^2 = \min$$

$$P_i(x_i, y_i)$$



$\varepsilon_g = (x+1)^2 + y^2 - R^2$

$\varepsilon_d = (x+1)^2 + (y+1)^2 - R^2$

$\varepsilon_v = x^2 + (y-1)^2 - R^2$

Аналіз значень похибки починають зі значення ε_d

Якщо ε_d =0 то вибирають діагональну точку. Якщо ε_d <0 то точка знаходиться в середині кола. В цьому випадку вибирають з горизонтальної або діагональної точки. Для цього обчислюють d_i. Якщо d_i <=0 то вибирають точку P_g, інакше P_d.

$d_i = \left|\varepsilon_g\right| - \left|\varepsilon_d\right| = \left|(x+1)^2 + y^2 - R^2\right| - \left|(x+1)^2 + (y-1)^2 - R^2\right|$

Якщо ε_d >0, тоді вибираємо з точок P_d або P_v.

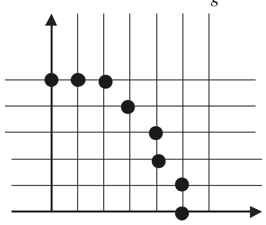
$d_i = \left|\varepsilon_d\right| - \left|\varepsilon_v\right| = \left|(x+1)^2 + (y-1)^2 - R^2\right| - \left|x^2 + (y-1)^2 - R^2\right|$

Якщо d_i<=0 то вибираємо діагональну точку, інакше P_v.

$(0,0), \quad k=5, \quad P_i(0,k)$

X	0	1	2	3	4	4	5	5
Y	5	5	5	4	3	2	1	0
ε_g	1+25-25=1	4	-------	-------	-------	-------	-------	-------
ε_d	1+16-25=-8 <0	-5 < 0	0	0	4 > 0	1 > 0	9 > 0	-------
ε_v	---------	-------	-------	-------	0	-8	0	-------
d_i	1-|-8|=-7 <0	-4 <0	-------	-------	4 > 0	-7 < 0	9 > 0	-------

$P_g \quad P_g \quad P_d \quad P_d \quad P_v \quad P_d \quad P_v$



Криві Без'є

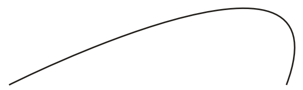
Це поліноміальні криві, форму яких визначають контрольні точки. Чим більша кількість контрольних точок тим більший ступінь поліноміального опису кривої і тим точніша його побудова. На практиці застосовують криві третього ступеня, які визначаються 4-ма контрольними точками. Рівняння кривої Без'є в параметричному вигляді :

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

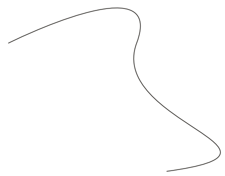
$$0 \le t \le 1$$

Властивості кривих Без'є:

1. Крива завжди проходить через першу та останню контрольні точки
2. Порядок контрольних точок має значення
3. Лінія, що з'єднує перші дві точки дотична до кривої в першій точці
4. Лінія, що з'єднує дві останні точки дотична до кривої в останній точці
5. Зміна розміщення будь-якої контрольної точки змінює форму кривої



Для отримання кривих більш складного вигляду необхідно об'єднувати криві третього порядку. Необхідна умова для отримання безперервності – це розміщення двох останніх точок першої частини та двох перших точок другої частини уздовж тієї ж лінії, крім цього остання точка першої частини має бути першою точкою другої частини.



$$x(t) = (1-t)^3 x_{P_0} + 3t(1-t)^2 x_{P_1} + 3t^2(1-t)x_{P_2} + t^3 x_{P_3}$$
$$y(t) = (1-t)^3 y_{P_0} + 3t(1-t)^2 y_{P_1} + 3t^2(1-t)y_{P_2} + t^3 y_{P_3}$$
$$0 \le t \le 1$$

Задача: намалювати ескіз кривої Без'є, яка визначена 4ма контрольними точками:

$P_0(2,2), \quad P_1(3,5), \quad P_2(6,6), \quad P_3(4,3)$

$\Delta t = 0.25$

$x(0) = 2$

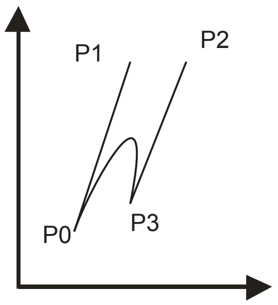
$y(0) = 2$ $\qquad x(0.75) = 4.65$

$x(0.25) = 3$ $\qquad y(0.75) = 4.5$

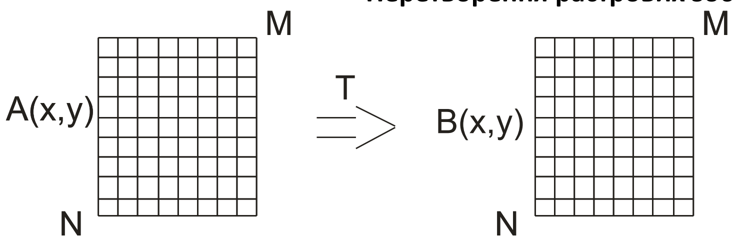
$y(0.25) = 3.8$ $\qquad x(1) = 4$

$x(0.5) = 4.1$ $\qquad y(1) = 3$

$y(0.5) = 4.75$

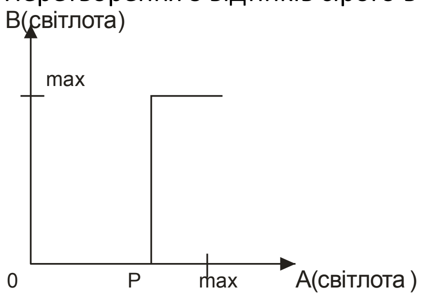


Перетворення растрових зображень



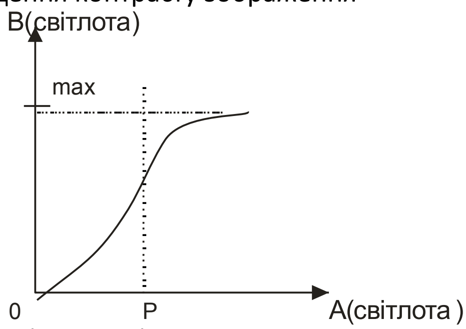
B=T(A)

1. Перетворення з відтінків сірого в чорно-біле зображення

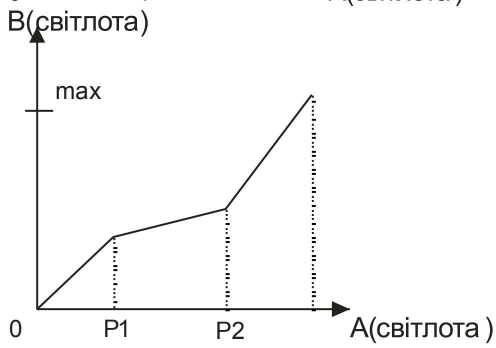


$$B = \begin{cases} 0, & A \le P \\ \max, & A > P \end{cases}$$

2. Покращення контрасту зображення

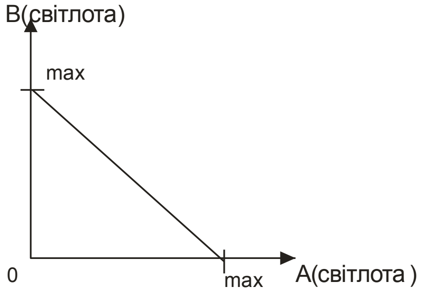


a.



b.

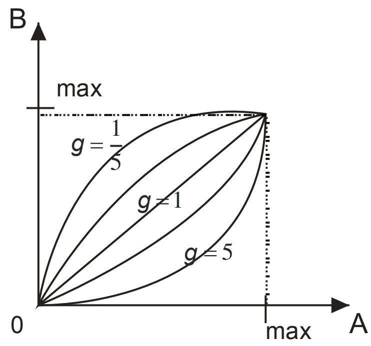
3. Негатив



$$B(x, y) = \max - A(x, y)$$

4. γ -корекція

$$B = A^{\gamma}$$



Завдання на самостійну роботу: Для чого може бути застосоване логарифмічне перетворення?

Фільтрація зображень

Фільтрація зображення A(x,y) розміром M*N виконується із застосуванням маски – підматриці $\mu(x,y)$, $k*l$, елементи якої задають вид фільтрації.

$k = 2a+1$

$l = 2b+1$

$k = l$

$a,b > 0$

Зазвичай вибирають $a = b = 1$

В ряді випадків застосовують матрицю 2*2 або більше ніж 3*3. Але 3*3 є більш поширеним випадком.

$$B(x,y) = \sum_{i=-a}^{a}\sum_{j=-b}^{b}\mu(i,j) * A(x+i, y+i)$$

Приклади масок:

1. Фільтр із застосуванням лаплассіану

$$\nabla^2 A = \frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2}$$

$$\frac{dA}{dx} = A(x+1) - A(x)$$

$$\begin{cases} \dfrac{\partial^2 A}{\partial x^2} = A(x+1,y) - A(x,y) - A(x,y) + A(x-1,y) = A(x+1,y) + A(x-1,y) - 2A(x,y) \\ \dfrac{\partial^2 A}{\partial y^2} = A(x,y+1) + A(x,y-1) - 2A(x,y) \end{cases}$$

0	1	0
1	-4	1
0	1	0

$$\nabla^2 A = A(x+1,y) + A(x-1,y) + A(x,y+1) + A(x,y-1) - 4A(x,y)$$

1	1	1
1	-8	1
1	1	1

2. Фільтр Собела

-1	-2	-1
0	0	0
1	2	1

Або

0	-1	2
1	0	-1
-2	1	0

Загальна властивість фільтрів: сума всіх елементів дорівнює нулю.

3. Згладжувальна фільтрація

$$B(x,y)=\frac{\sum_{i=-a}^{a}\sum_{j=-b}^{b}\mu(i,j)*A(x+i,y+i)}{\sum_{i=-a}^{a}\sum_{j=-b}^{b}\mu(i,j)}$$

Приклади:

1/9*

1	1	1
1	1	1
1	1	1

1/16*

1	2	1
2	4	2
1	2	1

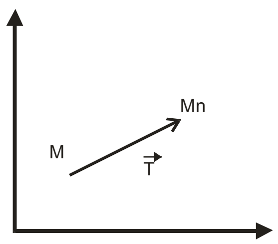
Геометричні перетворення

Перетворення, які відбуваються над точкою М можна розглядати у двох варіантах: зміна координатної системи, а точка зберігається; координатна система зберігається незмінною, а змінюється точка, її положення та координати.

Перетворення, під час якого точці М ставиться у відповідність точка М', при чому точка М належить одній системі координат, а точка М' – новій (перетвореній) системі координат, але має в ній такі самі координати називається афінним перетворенням.

До основних перетворень відносяться:

1. Зсув на вектор

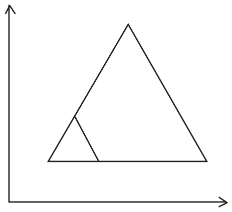


$$\begin{cases} x_n = x + T_x \\ y_n = y + T_y \end{cases}$$

$$\overrightarrow{M_n} = \overrightarrow{M} + \overrightarrow{T}$$

$$\overrightarrow{T} = [T_x T_y]$$

2. Масштабування

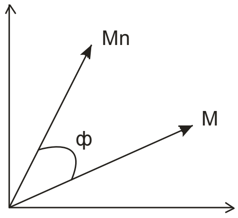


$$\begin{cases} x_n = x * S_x \\ y_n = y * S_y \end{cases}$$

$$\overrightarrow{M_n} = \overrightarrow{M} * S$$

$$S = \begin{vmatrix} S_x & 0 \\ 0 & S_x \end{vmatrix}$$

3. Поворот на кут



$$\begin{cases} x_n = x\cos\varphi - y\sin\varphi \\ y_n = x\sin\varphi + y\cos\varphi \end{cases}$$

$$R = \begin{vmatrix} \cos\varphi & \sin\varphi \\ \text{-}\sin\varphi & \cos\varphi \end{vmatrix}$$

<u>Задача</u>: даний трикутник ABC, координати вершин A(1,1), B(2,3), C(3,1). Виконати такі перетворення: 1. Поворот на кут п/2; 2. Розтягнення з коефіцієнтом $\alpha = \beta = 2$; 3. Зсув на вектор (3,5).

1.

$$\left.\begin{aligned} x'(A) &= 1*\cos\frac{\pi}{2} - 1*\sin\frac{\pi}{2} = -1 \\ y'(A) &= 1*\sin\frac{\pi}{2} + 1*\cos\frac{\pi}{2} = 1 \end{aligned}\right\} A'(-1,1)$$

$$\left.\begin{aligned} x'(B) &= -3 \\ y'(B) &= 2 \end{aligned}\right\} B'(-3,2)$$

$$\left.\begin{aligned} x'(C) &= -1 \\ y'(C) &= 3 \end{aligned}\right\} C'(-1,3)$$

2.

$$\left.\begin{aligned} x''(A) &= 2*1 = 2 \\ y''(A) &= 2*1 = 2 \end{aligned}\right\} A'(2,2)$$

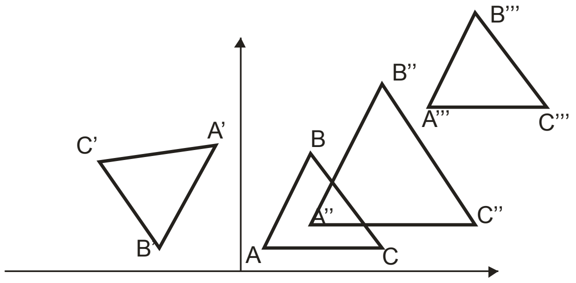
$$\left.\begin{aligned} x''(B) &= 4 \\ y''(B) &= 6 \end{aligned}\right\} B'(4,6)$$

$$\left.\begin{aligned} x''(C) &= 6 \\ y''(C) &= 2 \end{aligned}\right\} C'(6,2)$$

$$\left.\begin{aligned} x'''(A) &= 1+3=4 \\ y'''(A) &= 1+5=6 \end{aligned}\right\} A'(4,6)$$

3. $$\left.\begin{aligned} x'''(B) &= 5 \\ y'''(B) &= 8 \end{aligned}\right\} B'(5,8)$$

$$\left.\begin{aligned} x'''(C) &= 6 \\ y'''(C) &= 6 \end{aligned}\right\} C'(6,6)$$



Двовимірні перетворення в однорідних координатах

Оскільки всі розглянуті перетворення виконуються по різному це ускладнює комплексні перетворення. Для усунення цього недоліку перейдемо до однорідних координат які в площинному випадку мають вигляд: $[X\ \ Y\ \ W]$, де $\mathrm{W} \neq 0$.

<u>Однорідними</u> називають такі координати, що при їх множенні на одне число об’єкт, який визначається цими координатами не змінюється.

Декартові та однорідні координати зв’язуються формулами:

$$x = \frac{X}{W}$$

$$y = \frac{Y}{W}$$

Таким чином однорідні координати можна представити як про масштабовані з коефіцієнтом W значення двовимірних координат, які розміщені у площині z = W.

З огляду на довільність значення W в однорідних координатах не існує єдиного подання точки в заданих декартових координатах.

$M' = M * P$ (змінюється матриця перетворення)

1. Для зсуву: $M' = M * \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{vmatrix}$ $\qquad M' = M * T$
2. Масштабування: $M' = M * \begin{vmatrix} \mathrm{S}_x & 0 & 0 \\ 0 & \mathrm{S}_y & 0 \\ 0 & 0 & 1 \end{vmatrix}$ $\qquad M' = M * S$
3. Розтягнення: $M' = M * \begin{vmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{vmatrix}$ $\qquad M' = M * R$

$$P = \begin{vmatrix} \mathrm{A} & \mathrm{B} & \mathrm{P} \\ \mathrm{D} & \mathrm{E} & \mathrm{Q} \\ \mathrm{L} & \mathrm{M} & \mathrm{S} \end{vmatrix}$$

Коефіцієнти A, B, D, E визначають зміну масштабу та поворот. Коефіцієнти L, M відповідають за зсув. S визначає загалюну зміну масштабу, а P та Q визначають загальну проекцію.

Покажемо що S визначає загальну зміну масштабу, а P, Q визначають проеціювання. Для цього розглянемо перетворення:

4. $$\begin{vmatrix} X' & Y' & h \end{vmatrix} = \begin{vmatrix} X & Y & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & S \end{vmatrix}$$

$$\begin{aligned} X' &= X \\ Y' &= Y \qquad \Rightarrow \quad x' = \frac{X}{S} \quad y' = \frac{Y}{S} \\ h &= S \end{aligned}$$

5. $$\begin{aligned} X' &= X \\ Y' &= Y \qquad \Rightarrow \quad x' = \frac{X'}{P*X'+Q*Y'+1} \quad y' = \frac{Y'}{P*X'+Q*Y'+1} \\ h &= P*X'+Q*Y'+1 \end{aligned}$$

Композиція двовимірних перетворень

Розглянемо зсув точки P0 на відстань $[T_{x1}\ T_{y1}]$

$P_0 \to P_1 \to P_2$

$[T_{x1}\ T_{y1}]$

$[T_{x2}\ T_{y2}]$

$T_1,\ T_2$

$P_1 = P_0 * T_1$

$P_2 = P_1 * T_2 = (P_0 * T_1) * T_2 = P_0 * (T_1 * T_2)$

$$T = T_1 * T_2 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_{x1} & T_{y1} & 1 \end{vmatrix} * \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_{x2} & T_{y2} & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_{x1}+T_{x2} & T_{y2}+T_{x2} & 1 \end{vmatrix}$$

$P_1 = P_0 * S_1$

$P_2 = P_1 * S_2 = P_0 * S$

$$S = S_1 * S_2 = \begin{vmatrix} S_{x1} & 0 & 0 \\ 0 & S_{y1} & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} S_{x2} & 0 & 0 \\ 0 & S_{y2} & 0 \\ 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} S_{x1} * S_{x2} & 0 & 0 \\ 0 & S_{y1} * S_{y2} & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Розглянемо приклад з виконанням перетворення в однорідних координатах. Дано трикутник ABC. A(1,1), B(2,3), C(3,1).

1. $\frac{3\pi}{2}$ P(5,2)
2. $\alpha = \beta = 2$ P(5,2)

Розв’язання:

1. А) $T = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -5 & -2 & 1 \end{vmatrix}$

Б) $R = \begin{vmatrix} \cos\frac{3\pi}{2} & \sin\frac{3\pi}{2} & 0 \\ -\sin\frac{3\pi}{2} & \cos\frac{3\pi}{2} & 0 \\ 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix}$

В) $T' = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 2 & 1 \end{vmatrix}$

Г) $K = T * R * T' = \begin{vmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 3 & 7 & 1 \end{vmatrix}$

$A'(x', y', 1) = [1\ 1\ 1] * K = (4, 6, 1)$

Д) $B'(x', y', 1) = [2\ 3\ 1] * K = (6, 5, 1)$

$C'(x', y', 1) = [3\ 1\ 1] * K = (4, 4, 1)$

2. А) $T = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -5 & -2 & 1 \end{vmatrix}$

Б) $S = \begin{vmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{vmatrix}$

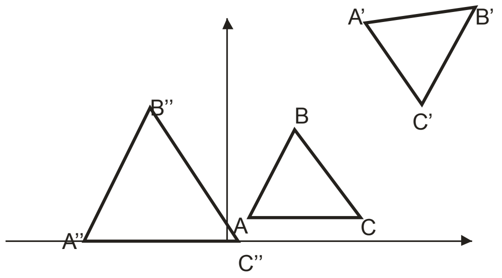
В) $T' = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 2 & 1 \end{vmatrix}$

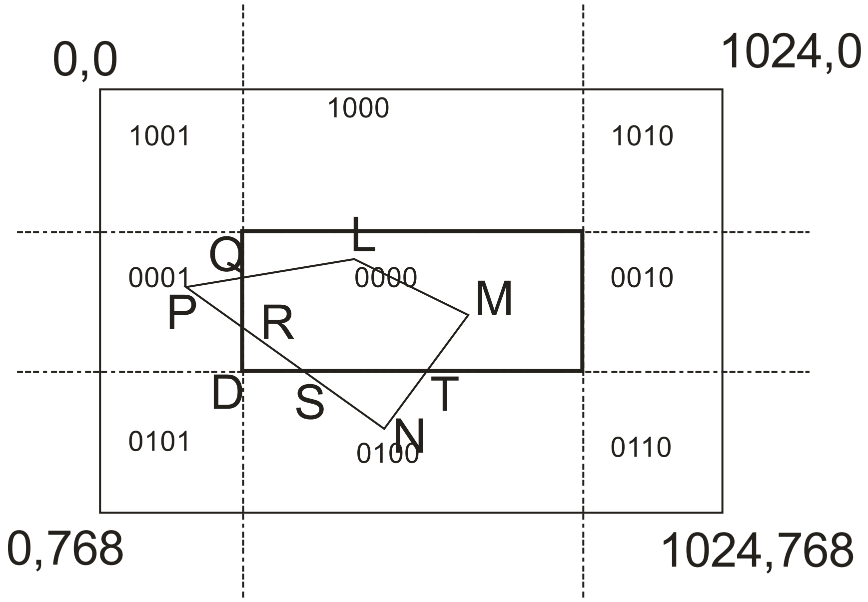
Г) $K = T * S * T' = \begin{vmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ -5 & -2 & 1 \end{vmatrix}$

$A''(-3, 0, 1)$

Д) $B''(-1, 4, 1)$

$C''(1, 0, 1)$





A (210, 120)
B (720, 120)
C (720, 540)
D (210, 540)
Визначаємо коди точок багатокутника:
Потребує відтинання лівою стороною вінка. Тоді координати точки перетину:

1. Вв
2. PL
   a. Вв
      $X_Q = X_A = 210$
   b. $Y_Q = 360 + \frac{210-160}{280-160} * (128-360) = -263$
      $Q(210, 263)$
   c. Замінюємо точку P на точку Q. $k_Q = 0000$
3. Аналізуємо коди: $k_Q = k_L = 0$. Висновок: відрізок QL візуалізується.
4. $k_M = k_L = 0$. Висновок: віздрізок LM візуалізується.
5. MN
      $k_M = 0$
   a. $k_N \neq 0$
      $k_M \wedge k_N = 0$
      Потребує відтинання нижньою стороною вікна з заміною на відрізок MN
   b. T(621, 540)
   c. $k_T = 0000$
6. TM
   $k_T = k_M = 0000$, отже відрізок візуалізується
7. NP
      $k_N \neq 0$
   a. $k_P \neq 0$
      $k_N \wedge k_P = 0$
      Потребує відтинання нижньою та лівою сторонами вікна.
   b. Визначаємо координати точки перетину з нижньою стороною вінка: S(540, 497)
   c. $k_S = 0000$
8. SP

$k_S = 0$

a. $k_P \neq 0$

$k_S \wedge k_P = 0$

Потребує відтинання лівою стороною вікна з заміною на відрізок PS

b. R (210, 387)

c. $k_R = 0000$

9. RS. $k_R = k_S = 0$, отже відрізок RS візуалізується.

Алгоритми з параметричними рівняннями

Припустимо що прямокутне вікно повернуте відносно системи координат або має іншу довільну форму. Для того щоб виконувати відсікання невидимих частин об'єктів вікна довільної форми застосовується алгоритм Кіруса-Бека (CB-алгоритм).

Спочатку розглянемо відсікання параметрично заданого відрізка прямокутним вікном. Параметричне рівняння відрізка має вигляд:

$$\vec{p} = \vec{p}_1 + (\vec{p}_2 - \vec{p}_1)t \qquad (*)$$

, де $\vec{p}$ з'єднує початок координат та довільну точку P на прямій P_1P_2. $\vec{p}_1$ та $\vec{p}_2$ - вектори, які з'єднують точку початку координат та початок та кінець відрізка P_1P_2 даної прямої. t – параметр, що знаходиться в межах від 0 до 1 для відрізка P_1P_2.

Параметричний опис не залежить від вибору системи координат. Ця властивість робить параметричне представлення зручним для знаходження перетину відрізка з стороною довільного опуклого багатокутника.

Розглянемо це на прикладі прямокутного вікна. В двовимірній декартовій системі координат рівняння (*) зводиться до пари рівнянь виду:

$$x(t) = x_1 + (x_2 - x_1)t$$

$$y(t) = y_1 + (y_2 - y_1)t$$

В випадку прямокутного відтинаючого вікна одна з координат перетину відрізка з кожною стороною відома. Необхідно обчислити лише другу координату.

$$t = \frac{\vec{p} - \vec{p}_1}{\vec{p}_2 - \vec{p}_1}$$

Значення параметра t, що відповідають перетинам зі сторонами вікна дорівнюють:

Для лівої сторони: $t = \dfrac{x_Л - x_1}{x_2 - x_1}$

Для нижньої сторони: $t = \dfrac{y_Н - y_1}{y_2 - y_1}$

Для правої сторони: $t = \dfrac{x_П - x_1}{x_2 - x_1}$

Для верхньої сторони: $t = \dfrac{y_В - y_1}{y_2 - y_1}$

Якщо рішення цих рівнянь дають значення t за границями інтервалу від 0 до 1, то такі рішення відкидаються, оскільки вони відповідають точкам, що лежать поза відрізками.

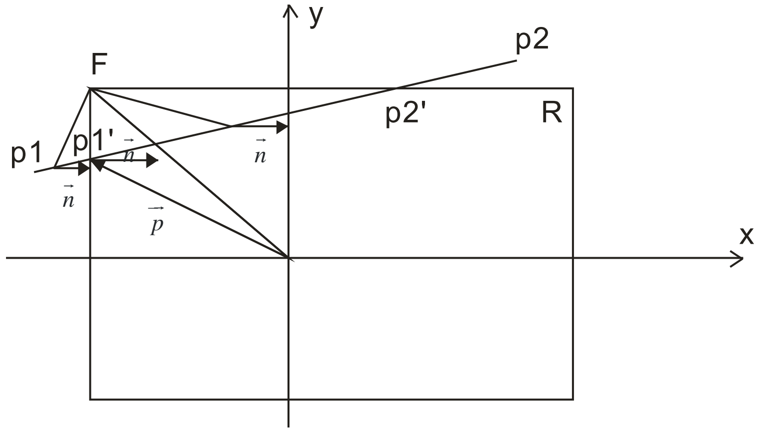
Алгоритм Кіруса-Бека (CB-алгоритм)

Відрізок прямої, який перетинає вікно довільної форми в параметричному вигляді може бути заданий рівнянням: $\vec{p} = \vec{p}_1 + (\vec{p}_2 - \vec{p}_1) * t$

$0 <= t <= 1$

R – опукла область (вікно відтинання)

Розглянемо граничну точку F опуклої області R та внутрішню нормаль N до одного з відрізків прямих, що обмежують цю область.



Для будь-якої точки відрізку p1p2 можливі такі випадки для значення скалярного добутку векторів.

$\vec{n}(\vec{p}-\vec{F})<0$ - точка знаходиться поза вікном

$\vec{n}(\vec{p}-\vec{F})=0$ - точка знаходиться на границі вінка

$\vec{n}(\vec{p}-\vec{F})>0$ - точка знаходиться в середині вікна

$\vec{n}(\vec{p_1}-(\vec{p_2}-\vec{p_1})*t-\vec{F})=0$ (**)

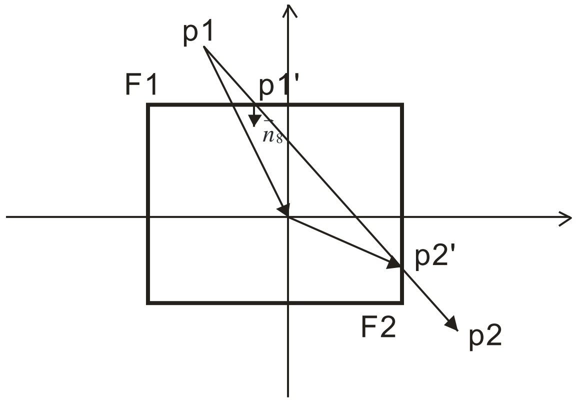
Якщо t лежить за межами інтервалу (0, 1), то ним можна знехтувати. Рівняння (**) може дати більше ніж 2 рішення для параметра t .

Для знаходження дійсних значень параметра t всі отримані значення t потрібно впорядкувати та поділити на 2 групи: «нижню» та «верхню».

Треба знайти найбільше з нижніх та найменше з верхніх значень. Якщо $(\vec{p_2}-\vec{p_1})\vec{n}>0$, то знайдене значення t розглядається в якості нижньої границі, інакше в якості верхньої границі діапазону точок відрізка, які знаходяться в області R відтинаючого вікна.

Розглянемо <u>приклад</u> застосування:

Задані точки: P1(-3; 6.5), P2(6; -4.5). Задане прямокутне вінко точками: F1(-5; 4), F2(4; -3)



$\vec{p}=(-3*9t)\vec{i}+(6.5-11t)\vec{j}$ (***)

$0<=t<=1$

$\vec{p}-\vec{f_1}=(2+9t)\vec{i}+(2.5-11t)\vec{j}$

$\vec{p}-\vec{f_2}=(-7+9t)\vec{i}+(9.5-11t)\vec{j}$

$\vec{p_2}-\vec{p_1}=[6-4.5]-[-3+6.5]=9\vec{i}-11\vec{j}$

$7-9t_П=0$

$t_П=7/9$

Перевіряємо якою, верхньою чи нижньою границею діапазону значень параметру t може бути знайдене значення.

$(\vec{p_2}-\vec{p_1})\vec{n_n}=-9<0$ - це ймовірно верхня границя.

Умова перетину з нижньою границею:

$(\vec{p}-\vec{f}_2)\overrightarrow{n_H}=0$

$9.5-11t=0$

$t_H=19/22$

Перевіримо якою, верхньою чи нижньою границею діапазону значень параметра t може бути знайдене значення:

$(\overrightarrow{p_2}-\overrightarrow{p_1})\overrightarrow{n_H}=-11<0$ - знайдене значення є можливою верхньою границею.

Для того щоб визначити яке з отриманих двох значень є верхньою границею порівняємо їх і виберемо менше значення. Отже t = 7/9. Це означає що відрізок p1p2 відтинається правою границею вікна в точці p2’ (4; -2.05).

Умова перетину лівої границі вікна:

$(\vec{p}-\vec{f}_1)\overrightarrow{n_Л}=0$

$2+9t=0$

$t_Л=-2/9<0$

T < 0 отже воно не розглядається.

Умова перетину верхньої границі вікна:

$(\vec{p}-\vec{f}_1)\overrightarrow{n_B}=0$

$2.5-11t=0$

$t_B=5/22>0$

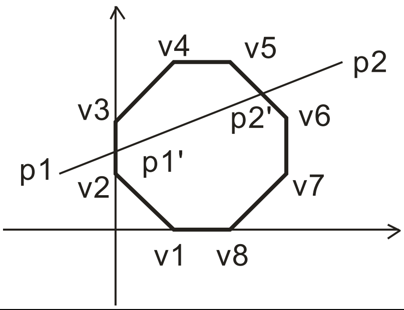
$(\overrightarrow{p_2}-\overrightarrow{p_1})\overrightarrow{n_H}=11>0$ - нижня границя.

p1’(-0.95; 4)

5/22 <= t <= 7/9

Розглянемо приклад застосування алгоритму Кіруса-Бека у випадку відтинаючого вікна довільної форми.

P1(-1; 1), P2(3; 3)



ребра	$\vec{n}$	f	$\vec{p}_1-\vec{f}_1$	$(\vec{p}_1-\vec{f})\vec{n}$	$(\overrightarrow{p_2}-\vec{p}_1)\vec{n}$	t_H	t_B
V1V2	[1 1]	(1,0)	[2,1]				
V2V3	[1; 0]	(0, 2)	[-1 -1]				
V3V4	[1; -1]	(0, 2)	[-1 -1]				
V4V5	[0; -1]	(2,3)	[-3 -2]				
V5V6	[-1; -1]	(2,3)	[-3 -2]				
V6V7	[-1; 0]	(3,1)	[-4 0]				
V7V8	[-1; 1]	(3,1)	[-4 0]				
V8V1	[0; 1]	(1,0)	[-2 1]				

В якості приклада розглянемо ребро v5v6

$\vec{p}_2-\vec{p}_1=[3;3]-[-1;1]=[4;2]$

$f(2;3)$

$\vec{p}_1-\vec{f}=[-1;1]-[2;3]=[-3;-2]$

$\vec{n}=[-1;-1]$

$(\overrightarrow{p_2}-\vec{p}_1)\vec{n}=-6$

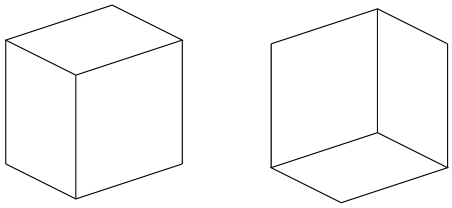
$(\vec{p}_1 - \vec{f})\vec{n} = 5$

Знайдена верхня границя значення t.

Аналіз таблиці показує що максимальне значення t_H =1/4, а мінімальне t_B =5/6. ¼<=t<=5/6 .

Видалення невидимих ліній та поверхонь

Алгоритми видалення невидимих ліній та поверхонь слугують для визначення ліній, ребер, поверхонь та об’ємів які можуть бути видимими, невидимими або частково видимими спостерігачу який знаходиться в заданій точці простору. Від видимості або невидимості певних ліній залежить інтерпретація об’єкту спостерігачем.



Розглянемо класифікацію методів видалення невидимих частин об’єктів. Їх можна поділити за такими ознаками:

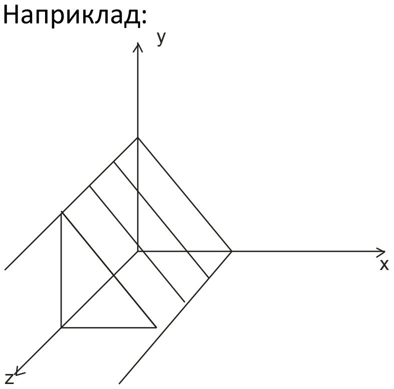
1. За вибором частин, які видаляються (видалення ребер, поверхонь, об’ємів)
2. За порядком обробки елементів сцени (видалення у довільному порядку, видалення у порядку, що визначається процесом візуалізації)
3. За системою координат (Алгоритми які виконуються у просторі зображення коли для кожного піксела зображення визначається яку з n граней об’єкта видно. Для цього кожна з граней порівнюється з позиціями пік селів в системі координат екрану. Прикладом такого алгоритму є алгоритм плаваючого горизонту; Алгоритми які виконуються у просторі об’єктів, коли кожна з n граней об’єкта порівнюється з n-1 гранню що залишились. Прикладом є алгоритм Робертса.)

Алгоритм плаваючого горизонту

Алгоритм плаваючого горизонту частіше за все використовується для видалення невидимих ліній тривимірного представлення функцій, які описують поверхню у вигляді: $F(x, y, z) = 0$.

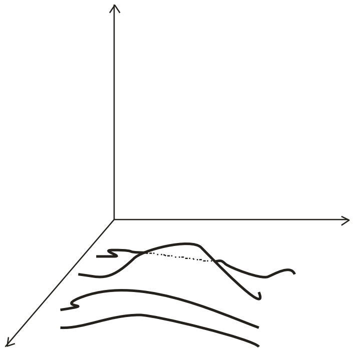
Основна ідея даного алгоритму полягає у зведенні тривимірної задачі до двовимірної шляхом перетину вихідної поверхні послідовністю паралельних січних площин, які мають сталі значення координат x, y або z.

Наприклад:

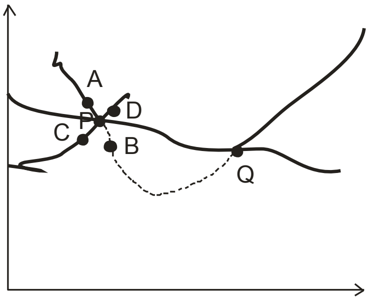


$y = f(x, z)$

$z = const$



<u>Критерій видимості:</u> для кожного значення координати х кожної кривої визначимо відповідне значення, тоді критерій видимості ліній, що спроецьовані на певну площину полягає в наступному: якщо на заданій площині при певному значенні х, відповідне значення у на кривій, видимість якої аналізується більше максимуму (виявлення «гір»), або менше мінімуму (виявлення «ям») по у для всіх попередніх кривих для цього х, то поточна крива є видимою.
Для того щоб візуалізувати отриманий горизонт необхідно визначити точки перетину ліній, які інтерполюють поточну та попередню криві.



$$\begin{cases} x_i = \dfrac{\square x_2(y_{n1}\square x_2 + x_{n1}\square y_1) - \square x_2(y_{n2}\square x_2 + x_{n2}\square y_2)}{\square y_1\square x_2 - \square y_2\square x_2} \\ y_i = y_{n1} + \dfrac{\square y_1(x_i - x_{n1})}{\square x_1} \end{cases}$$

$$\square x_i = x_{(n+1)j} - x_{nj}$$

$$\square y_i = y_{(n+1)j} - y_{nj}$$

У загальному вигляді алгоритм плаваючого горизонту зводиться до таких кроків:

1. Якщо на поточній площині при певному заданому значенні х відповідне значення у на кривій видимість якої аналізується більше максимуму, або менше мінімуму по у для всіх попередніх кривих для цього х, то поточна крива візуалізується
2. Якщо на проміжку від попереднього (x_n) до поточного (x_{n+k}) значення х видимість кривої змінюється, то обчислюється точка перетину x_i
3. Якщо на проміжку (x_n , x_{n+k}) сегмент кривої повністю видимий, то він візуалізується цілком. Якщо він став невидимим, то візуалізується фрагмент (x_n , x_i). Якщо він став видимим, то візуалізується фрагмент (x_i , x_{n+k}).
4. Якщо крива цілком невидима то виконується перехід до наступної кривої.

Види дипломних робіт по комп’ютерній графіці:
Інженерна розробка;
наукова робота з фільтрації зображень;
обробка звуку;

Алгоритм Робертса

В алгоритмі Робертса вимагається щоб всі об’єкти були опуклими. Опуклий багатогранний об’єкт з пласкими гранями має задаватись набором площин, які перетинаються. В алгоритмі передбачається що об’єкти складаються з пласких полігональних граней, які складаються з ребер, а ті в свою чергу задаються набором вершин.

Алгоритм Робертса складається з трьох основних етапів:

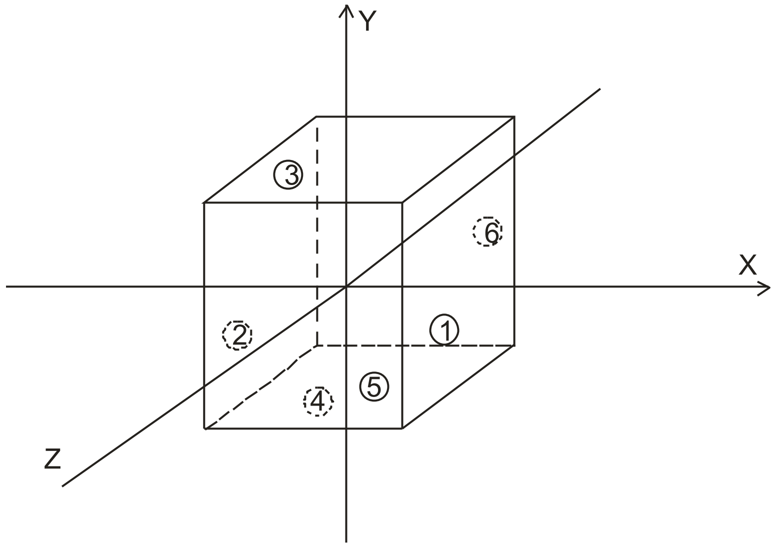
1. З кожного об’єкта вилучаються ті ребра та грані, які екрануються самим об’єктом
2. Перевіряється чи екрануються ребра та грані, що залишились, іншими об’єктами
3. Обчислюються відрізки, які утворюють нові ребра, якщо об’єкти протикають один-одного

В загальному вигляді алгоритм Робертса зводиться до наступного:

1. Видалення невидимих площин для кожного об’єкта в сцені.
    a. Сформувати багатокутники, виходячи зі списку вершин об’єкта.
    b. Обчислити рівняння площини для кожної полігональної грані об’єкта.
    c. Перевірити знак рівняння площини
        i. Взяти будь-яку точку в середині об’єкта
        ii. Обчислити скалярний добуток рівняння площини та точки в середині об’єкта
        iii. Якщо цей скалярний добуток від’ємний, то змінити знак рівняння площини
        iv. Сформувати матрицю об’єкту

Приклад: формування матриці об’єкту:

Нехай заданий одиничний куб з центром в початку координат.



$x_1 = 1/2$

$x_2 = -1/2$

$y_3 = 1/2$

$y_4 = -1/2$

$z_5 = 1/2$

$z_6 = -1/2$

Повна матриця об’єкта має вигляд:

$$V = \begin{vmatrix} 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \\ -1 & 1 & -1 & 1 & -1 & 1 \end{vmatrix}$$

Оскільки в алгоритмі Робертса передбачається що всі точки, що лежать в середині об’єкта діють додатній скалярний добуток, то отриману матрицю потрібно перевірити за допомогою однієї з внутрішніх точок об’єкта.
Мета перевірки – упевнитись, що знаки кожного рівняння відповідають спільній ознаці об’єкта – додатному скалярному добутку матриці об’єкта та вектора координат спостерігача. Якщо під час перевірки виявляється, що для якоїсь з площин скалярний добуток від’ємний, то рівняння відповідної площини необхідно помножити на -1.

$$S = [\frac{1}{4} \;\; \frac{1}{4} \;\; \frac{1}{4} \;\; 1] = [1 \;\; 1 \;\; 1 \;\; 4]$$

Скалярний добуток цього вектора на матрицю складає:

$$\vec{S} * V = [-2 \;\; 6 \;\; \text{-}2 \;\; 6 \;\; \text{-}2 \;\; 6]$$

$$V' = \begin{vmatrix} -2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{vmatrix}$$

v. Помножити матрицю об’єкта на матрицю, що є зворотною до матриці геометричного перетворення
vi. Обчислити та запам’ятати габарити прямокутної осяжної оболонки перетвореного об’єкта.
$x_{\max}, x_{\min}, y_{\max}, y_{\min}$
vii. Визначити невидимі площини
1. Обчислити скалярний добуток пробної точки на перетворену матрицю об’єкта
2. Якщо цей скалярний добуток від’ємний то площина невидима
3. Видалити весь багатокутник, який лежить в цій площині

<u>Приклад</u>: нехай точка спостереження знаходиться на позитивній півосі і має однорідні координати. Погляд спостерігача направлений на початок координат.

$$\vec{E} * V = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 2 \;\; \text{-}2]$$

Це означає що грань з номером 6 є невидимою для спостерігача. Нульові результати відповідають площинам, які є паралельними напряму погляду

2. Видалення з кожного об’єкта ребер, які екрануються іншими об’єктами в сцені
   a. Якщо заданий лише один об’єкт, то алгоритм завершується
   b. Сформувати пріоритетний список об’єктів сцени. Для цього виконати сортування по Z. Сортування виконується за максимальним значенням координати Z вершин перетворених об’єктів. Першим в упорядкованому списку та таким що має найбільший пріоритет буде той об’єкт, у якого значення координати Z буде мінімальним серед максимальних. Цей об’єкт в просторі буде найвіддаленішим від точки спостереження, яка розташована у безкінечності уздовж на осі Z.
   c. Для кожного об’єкта з пріоритетного списку потрібно перевірити екранування всіх фронтальних ребер усіма іншими об’єктами сцени. Об’єкт, ребра якого перевіряються назвемо пробним об’єктом, а об’єкт, відносно якого в даний момент виконується перевірка назвемо екрануючим. Необхідно перевіряти екранування пробного об’єкта тими екрануючими об’єктами, у яких пріоритети нижчі.
   d. Провести перевірку екранування для прямокутних осяжних оболонок пробного об’єкта та екрануючого об’єкта.

$$x_{\min}(екран.\ об'єкта) > x_{\max}(пробн.\ об'єкта)$$
$$x_{\min}(екран.\ об'єкта) < x_{\max}(пробн.\ об'єкта)$$
$$y_{\min}(екран.\ об'єкта) > y_{\max}(пробн.\ об'єкта)$$
$$y_{\min}(екран.\ об'єкта) < y_{\max}(пробн.\ об'єкта)$$

Якщо виконується хоча б одна з цих умов, то екрануючий об’єкт не може екранувати жодне з ребер пробного об’єкта.

Виконати перевірку для всіх об’єктів сцени.

3. Провести попередні перевірки протикання щоб визначити чи перетинається екрануючий об’єкт пробним об’єктом, та чи існує можливість часткового екранування.
    a. Порівняти максимальне значення координати Z та мінімальне її значення для екрануючого об’єкта.

       Якщо $z_{\max}(екран.\ об'єкта) < z_{\min}(пробн.\ об'єкта)$, то протикання неможливе.

       Перейти до наступного об’єкту.
    b. Перевірити видиме протикання.
        i. Якщо $z_{\max}(пробн.\ об'єкта) > z_{\max}(екран.\ об'єкта)$, то пробний об’єкт може проткнути передню грань екрануючого об’єкта.
        ii. Встановити ознаку видимого протикання, і занести об’єкт в список протикань.
    c. Якщо

       $$x_{\max}(пробн.\ об'єкта) > x_{\max}(екран.\ об'єкта)$$

       $$x_{\min}(пробн.\ об'єкта) < x_{\max}(екран.\ об'єкта)$$

       То пробний об’єкт може проткнути бокову сторону екрануючого об’єкта.
    d. Встановити ознаку протикання та внести об’єкт в список протикань
    e. Якщо

       $$y_{\max}(пробн.\ об'єкта) > y_{\max}(екран.\ об'єкта)$$

       $$y_{\min}(пробн.\ об'єкта) < y_{\max}(екран.\ об'єкта)$$

       То пробний об’єкт може проткнути верхню або нижню грань екрануючого об’єкта.
    f. Встановити ознаку протикання та внести об’єкт в список протикань.