Graphics and Visual Computing.

sites.google.com/iiita.ac.in/course-of-pavan-iiita/gvc

* Pavan Chakraborthy :-

Screen size = 1920 X 1080 px
= 2073600 pixels.

Each pixel has RGB, each colour having 1 Byte size (8 bits)

$$\therefore \begin{array}{r} 2073600 \\ \times \quad 3 \\ \hline 6,220,800 \end{array}$$ Byte.

Also, to show the graphics per second, 50 frames are needed.

$$\therefore \begin{array}{r} 6220800 \\ \times \quad 50 \\ \hline 31,10,40,000 \end{array} \approx 31 \text{M Bytes/sec}$$

For this, we need separate dedicated processing unit, for Graphics processing. $\boxed{\text{GPU cards}}$

CPU
↓
Normal Tasks

GPU
↓
Graphics & maintaining screen
↓
Gaming Industry

TPU
↓
Tensor Flow
↓
Machine Learning

openCV → images
OpenGL → Graphics.

Tensor :- has effect in all directions.

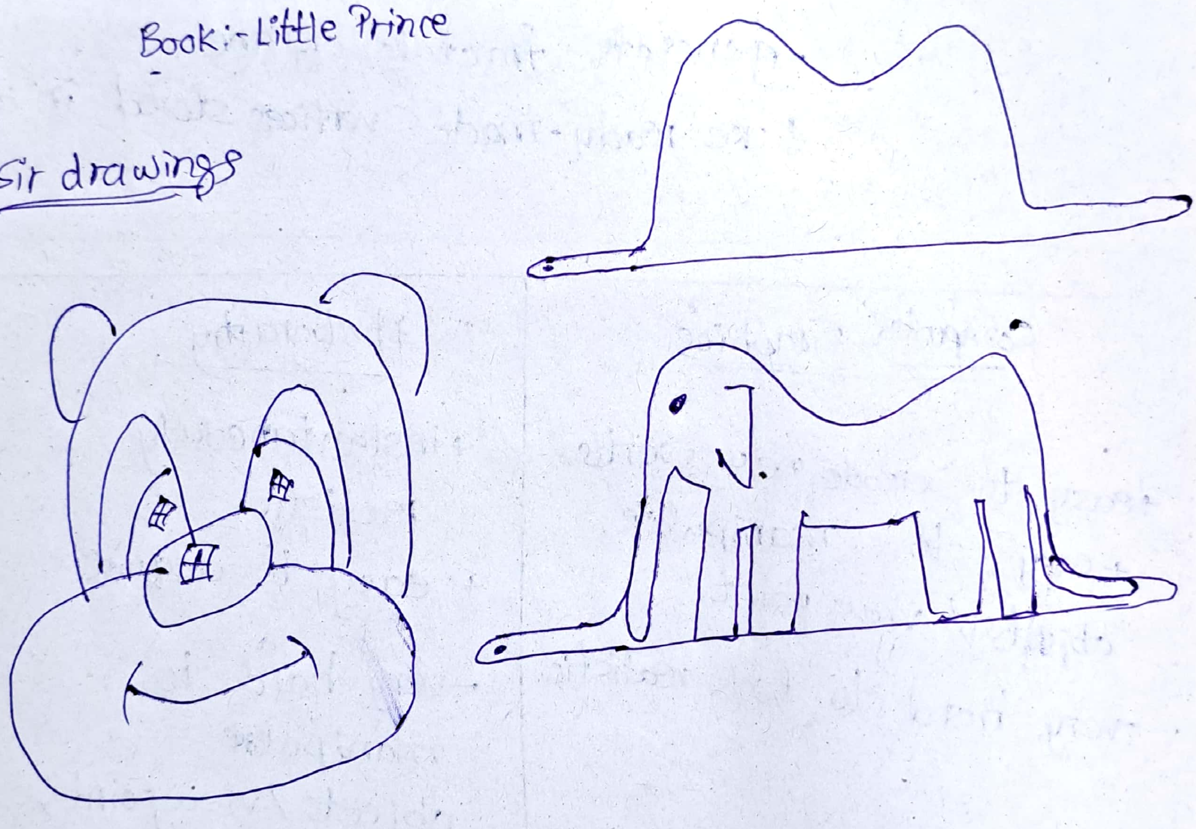
:- used by Einstein to develop Space-time curvature

Reflection Type → metallic
→ mirror-like

Green Apples > Red Apples.

Book :- Little Prince

Sir drawings



Lens Based Camera Obscura.

** Traditional Computer Graphics

Image $\xrightarrow[\text{breakdown}]{}$ wireframes

A wireframe has a similar surface & Texture.

Fractals - used to generate Trees
- but take a lot of time
- generate fractals offline
& use ready-made vertices stored in device

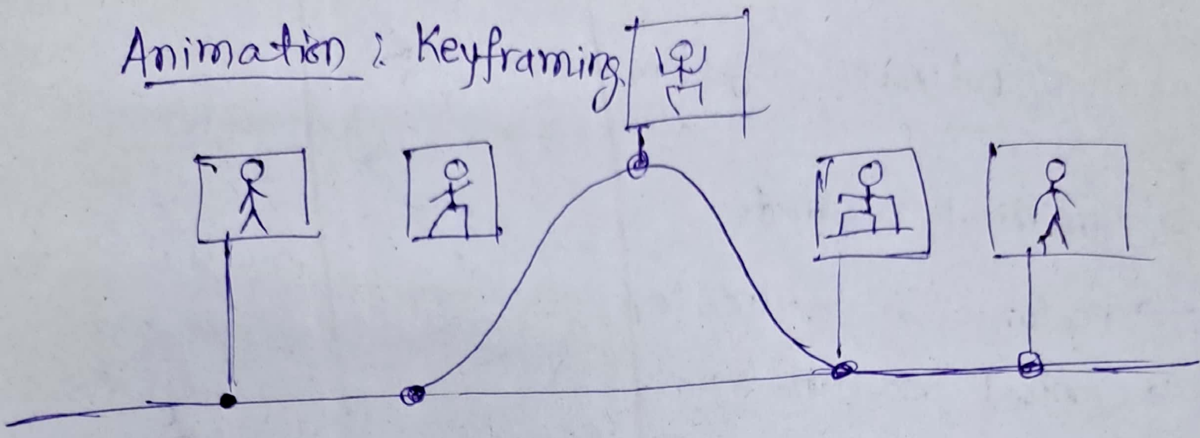
Computer Graphics	Photography
+ easy to create new worlds.	+ instantaneously realistic.
+ easy to manipulate objects / viewpoint.	+ easy to acquire.
- very hard to look realistic.	- very hard to manipulate objects / viewpoint.

Merging

Computer Graphics + Photography = Visual Computing

• Cheaper by Dozen - Pavan chakraborthy sir

Animation : Keyframing



Rigid Body Dynamics :- Learn torque & forces.

Face transitioning :- Face warping & morphing.

Cordinate systems

- Cylindrical cordinates
- Rectangular cordinates
- Conical cordinates
- Spherical Polar Cordinates
  - use in GPS.
  - has accuracy more than 10 metres.

India has Navik instead of GPS.

Object space

- local to each object.
- most natural cordinate system for the geometry object.
- Local cordinate system / modelling cordinate system.

World space

- common to all objects.
- The global cordinate system shared by all objects in world.

Eye Space / Camera space :-

- derived from view frustum

Screen Space

- indexed according

World Space → Eye Space → Screen space.

The world is Matrix!

model matrix → view matrix → projection matrix.

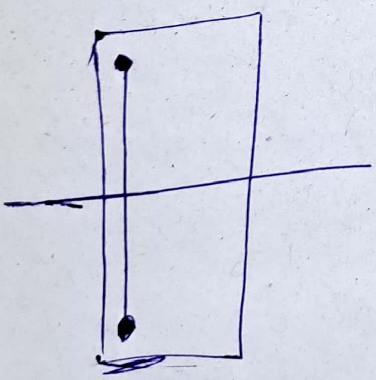
Raster :- • derived from TV system for a row of pixels.

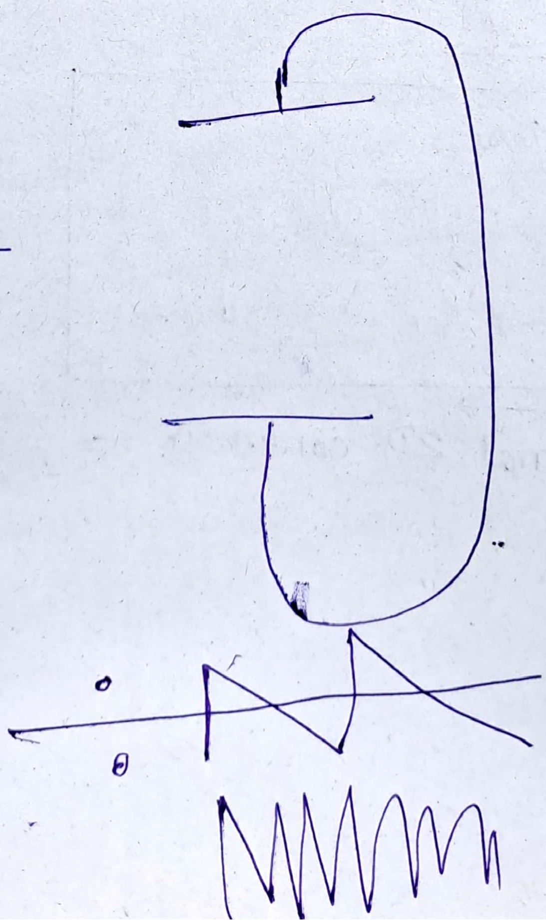
• Scanline.

• influence algorithms.

• is derivation of Rasterization or scanline algorithms.

Raster interlaced scanning :-





• colored CRT screen is achieved by hitting electron on florescent screen.

• LCD - Liquid crystal display, use view-voltage to change rotation, of Liquid crystal.

• OLED is better.

How 3D is achieved?

ans- using 3D polarized glasses

• screen also polarizes image.

Hence multiple dimensions can be felt.

* used in PVR.

• blue part of sky is polarized light

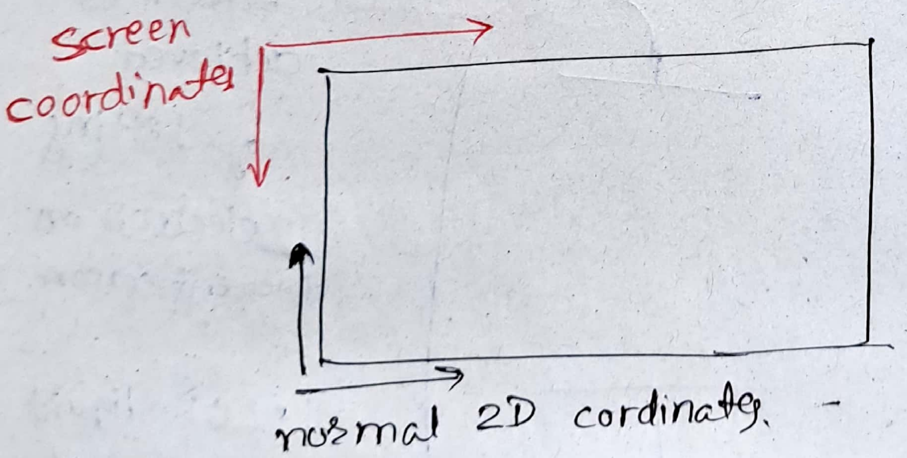
• Also used in seeing Eclipse.

Graphics Pipeline

- → Geometry or 3D pipeline
- → Imaging or 2D pipeline

• screen has pixels, which is quantized.

How to color, to make continous image



Geometry Pipeline

Animation / Interaction : time.

↓

Modelling : shapes.

↓

Shading : reflection & lighting.

↓

Transformation : viewing

↓

Hidden Surface Elimination

Geometry pipeline. →

Rasterization and sampling

↓

Texture Mapping.

↓

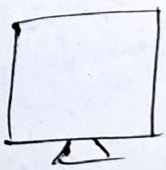
Image composition.

↓

Intensity and Colour quantisation

↓

Framebuffer / Display. →



This happens 50 times a second, to represent the fast view of screen.

Wireframe model - orthographic view.

3 views:- Top, side, front.

We, then place them into perspective view.

We do Depth cue, which tells which things are at what distance.

Q) Why not use sphere for sphere geometry, instead of using Triangles & Hexagons.

ans:- To ~~un~~ plot good reflections, easy to calculate surfaces normal to light

---

Then Hidden Line Removal - add color

we know foreground & background.

We remove the intersecting background lines.

---

we put flat illumination without shading.

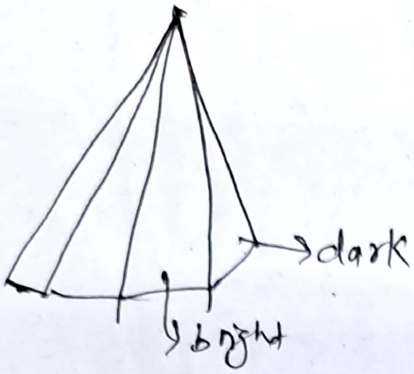
called constant shading.

• ambient illumination

- ex:- light coming from all directions equally

Faceted shading - Flat.

• light surfaces according to direction of Light source. Angles are measured, normal to surfaces



Gouraud Shading,

No specular Highlights.

• We do interpolating of shades between adjacent surfaces

(smoothening)

Specular Highlights Added

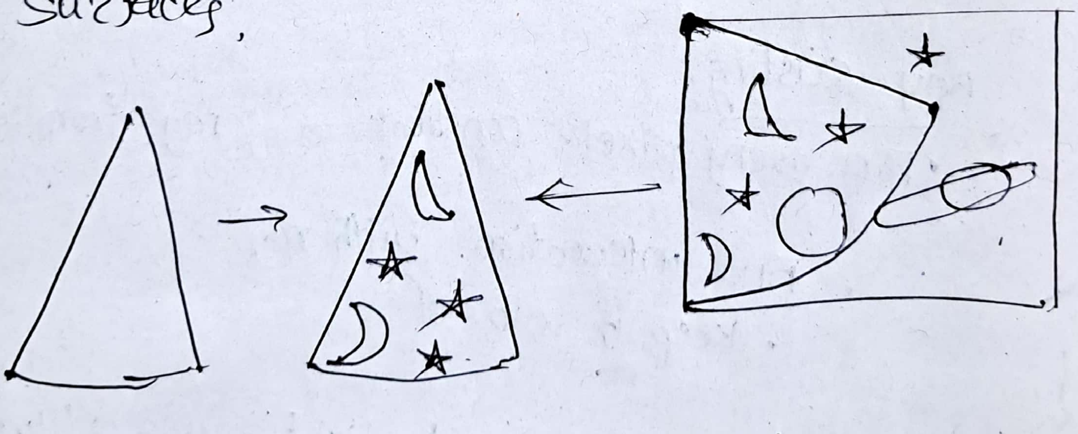
• add regions which reflect brightest

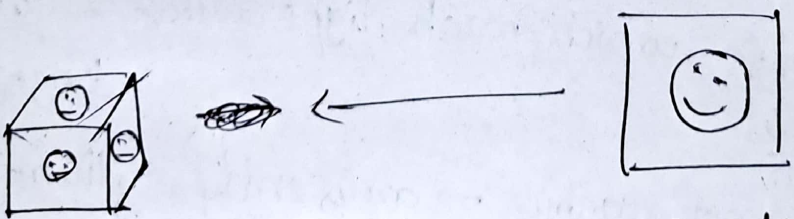
Phong Shading

Texture Mapping

• stick / map texture matrices onto the surfaces,

ex:-





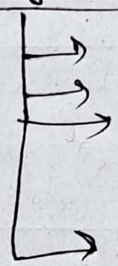
Finally polished & realistic images are made.

Why computer Graphics?

- movies
- CAD - CAM
- medical imaging
- astrophysics.

Graphics APIs (openGL, webGL)
modelling packages (3D studio MAX)

Rendering pipeline



Texture and modelling

Ray casting

- For every pixel, construct a ray from eye
  - Find intersection with ray
    - keep if closest.

- The objects which reach light source → illuminated brightly
- Those not reaching → ambiently illuminated.

Drawing lines & Antialising

Jurrasic park - done using openGL.

Line Equation

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

$i_1 = \text{Int}(x_1 + 0.5)$

$i_2 = \text{Int}(x_2 + 0.5)$

$j_1 = \text{Int}(y_1 + 0.5)$

$j_2 = \text{Int}(y_2 + 0.5)$

↳ 0.5 added to increase the ~~numbers~~ ~~don~~ (or) Round-off numbers.

* Floor and ceiling achieved without using special function.

~~While~~ Let $i = i_1$ and $j = j_1$

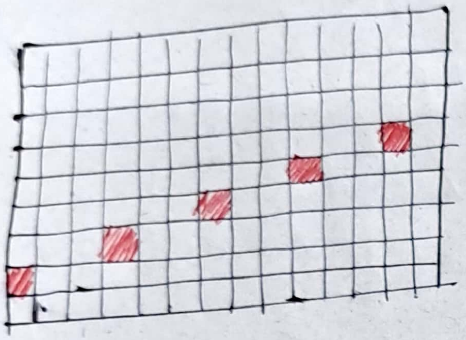
```
while ( i< i2 ) do
{ print (i,j);
  i=i+1
  y = m * i + c
  j = Int (y+0.5) }
```

Scanned with OKEN Scanner

Q] Is this good ?

ans:- ex:- $y = 0.25x + 0$

x	y	j
0	0	0
1	0.25	0
2	0.5	1
3	0.75	1
4	1.0	1
5	1.25	1
6	1.5	2
7	1.75	2
8	2.0	2
9	2.25	2.



Also if slope is more

$y = 10x + 0$

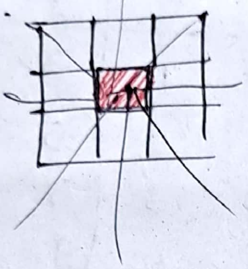
x	y	j
0	10	0
1	20	10
2	20	20
3	30	30
4	40	40
5	50	50
6	60	60

• we get dotted line in this case.

If we make smaller increments like $y + 0.01$?

- we will colour the same pixel many times

Solution :-

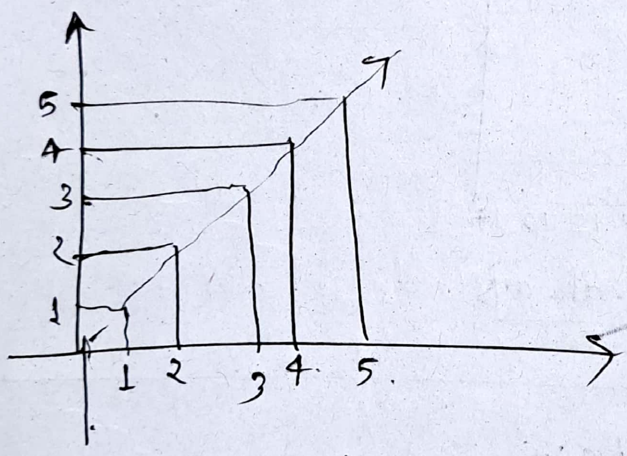


Just apply line equation such that u find adjacent out of 8 pixels.

Line drawing algo

$$y = mx + c$$

Line passing through $(1,1)$ and $(6,6)$.



$$y = mx + c$$

$$1 = m(1) + c$$

$$6 = m(6) + c$$

$$5 = 5m$$

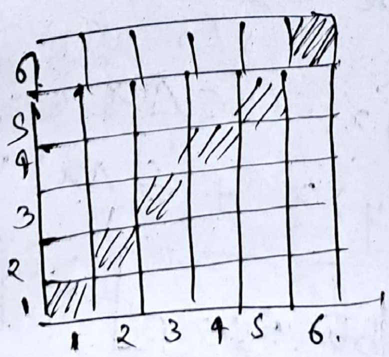
$$\boxed{m = 1}$$

$$1 = 1 + c$$

$$\boxed{c = 0}$$

$\therefore \boxed{y = x}$

x	y	eqn $y = x$
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	



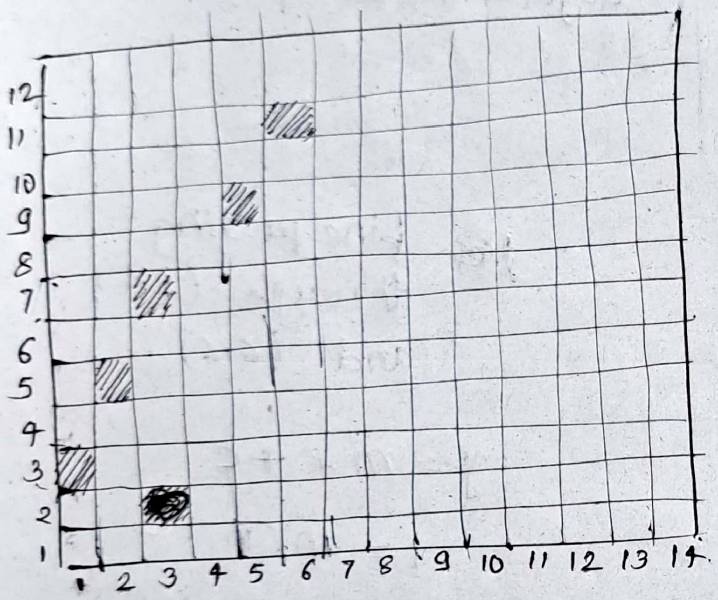
Q) Find Line passing through $P_1(1,3)$, $P_2(3,7)$.

ans:-

$$3 = m + c$$
$$7 = 3m + c$$
$$4 = 2m$$
$$\boxed{m = 2}$$

$$3 = 2 + c$$
$$\boxed{c = 1}$$
$$\boxed{y = 2x + 1}$$

x	y
1	3
2	5
3	7
4	9
5	11



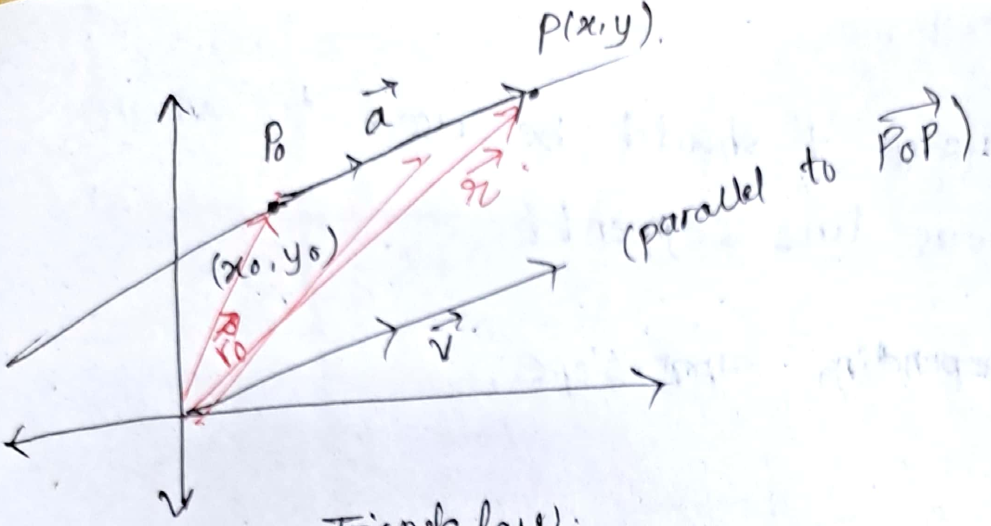
• Not continuous line always.

Any efficient Algorithm:-

- Yes, uses parametric line form

$$y = mx + c$$
$$y = \frac{\Delta y}{\Delta x} x + c$$
$$\Delta x y = \Delta y x + c \Delta x$$
$$0 = (\Delta y) x + (-\Delta x) y + c'$$

$$\boxed{f(x,y) = ax + by + c = 0}$$



Triangle law.

$$\vec{r_0} + \vec{a} = \vec{r}$$

$$\therefore \boxed{\vec{a} = \vec{r} - \vec{r_0}}.$$

Let constant 't'

since $\vec{a} \parallel \vec{v}$ $\therefore \vec{a} = t.\vec{v}$.

$$\therefore \boxed{\vec{r} = \vec{r_0} + t.\vec{v}}$$

DDA:-

$\langle x_0, y_0, z_0 \rangle + t \langle a, b, c \rangle$.

$\Rightarrow \langle x_0 + ta, \; y_0 + tb, \; z_0 + tc \rangle$

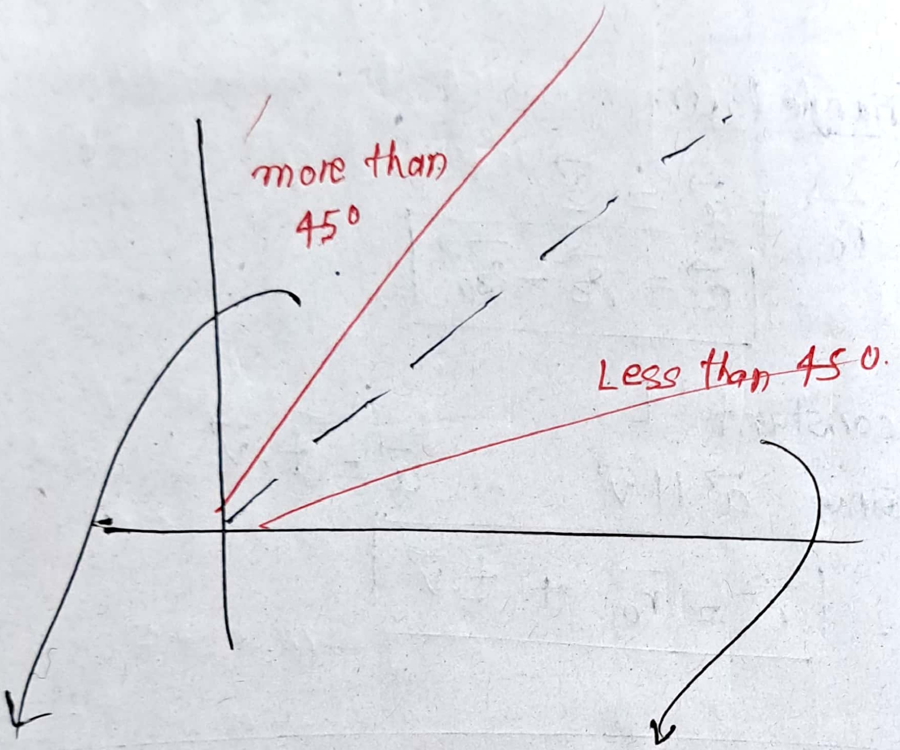
$$\left.\begin{aligned} x &= x_0 + ta \\ y &= y_0 + tb \\ z &= z_0 + tc \end{aligned}\right\} \quad \begin{aligned} x &= x_1 + t(x_2 - x_1) \\ y &= y_1 + t(y_2 - y_1) \\ z &= z_1 + t(z_2 - z_1). \end{aligned}$$

$\therefore$ ~~$t = \frac{\Delta y}{\Delta x}$~~

$$\left\{ \begin{aligned} &\therefore \frac{1}{dt} \\ &t = \frac{1}{\Delta x} \\ &t = \frac{1}{\Delta y} \end{aligned} \right\}$$

What value of 't' should be used to ensure continuous line segment?

ans:- Depending upon slope,



$m > 1$

$y = y + 1$

$x = \text{compute}$

$x + \{\}$

$x + \frac{dx}{dt}$ -

$\Rightarrow x = x + \frac{dx}{dy}$

$\boxed{x = x + \frac{1}{m}}$

$m < 1$

$x = x + 1$

$y = \text{compute}$

$y = y + \{\} \to \frac{dy}{dt} = \frac{dy}{dx},$

$y = y + \frac{dy}{dx}$

$\boxed{y = y + m}$

and if

$m = 1$

$x = x + 1$

$y = y + 1$

DDA algorithm :- (Digital differential analyser)

if $m < 1$	if $m > 1$	if $m = 1$
$x = x + 1$	$x = x + \frac{1}{m}$	$x = x + 1$
$y = y + m$	$y = y + 1$	$y = y + 1.$

ex:- $P_1(1, 3)$
$P_2(3, 7)$

$\boxed{m = 2}$.

$m > 1$

$x = x + \frac{1}{m}$

$y = y + 1.$

x	y	pixel to be plotted.
1	3	(1, 3)
1.5	4	(2, 4)
2	5	(2, 5)
2.5	6	(3, 6)
3	7.	(3, 7)

! simplest Algorithm !

Start :- $x = 1$, $y = 3$

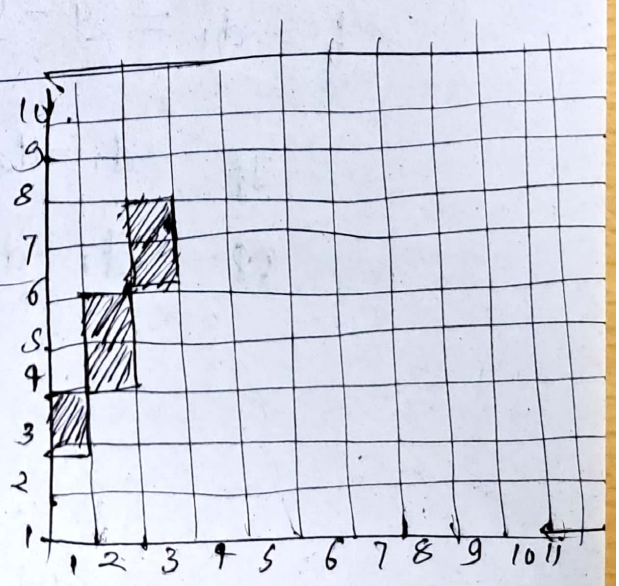
$x \Rightarrow x + \frac{1}{m} = 1 + \frac{1}{2} = \boxed{1.5}.$

$y \Rightarrow 3 + 1 = \boxed{4}$

$x = 1.5 + 0.5 = 2$
$y = 5$

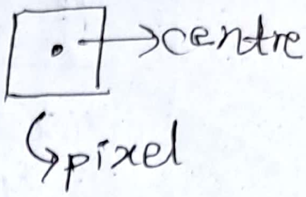
$x = 2.5$
$y = 6.$

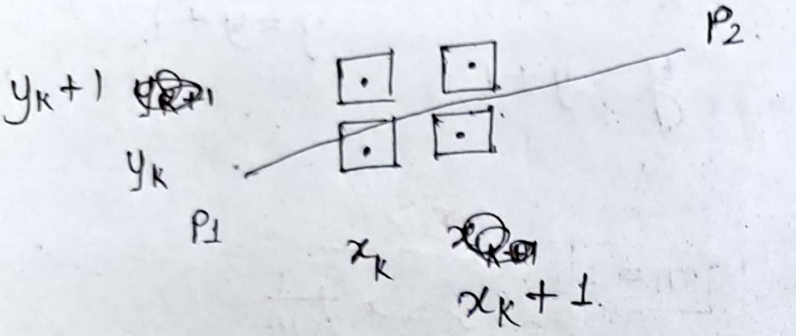
$x = 3$
$y = 7$

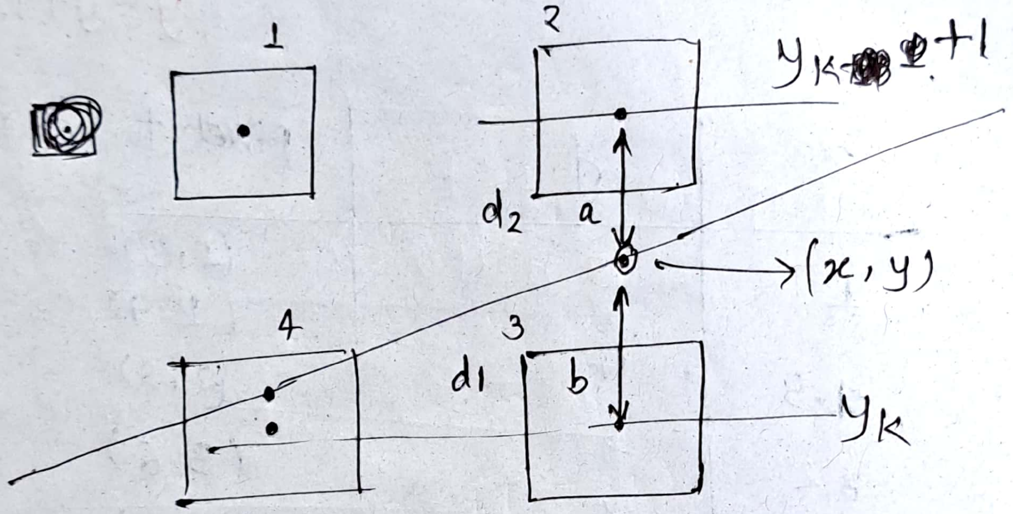


Beisener Algorithm

4 closely spaced pixels.







$a<b$:- Glow 2

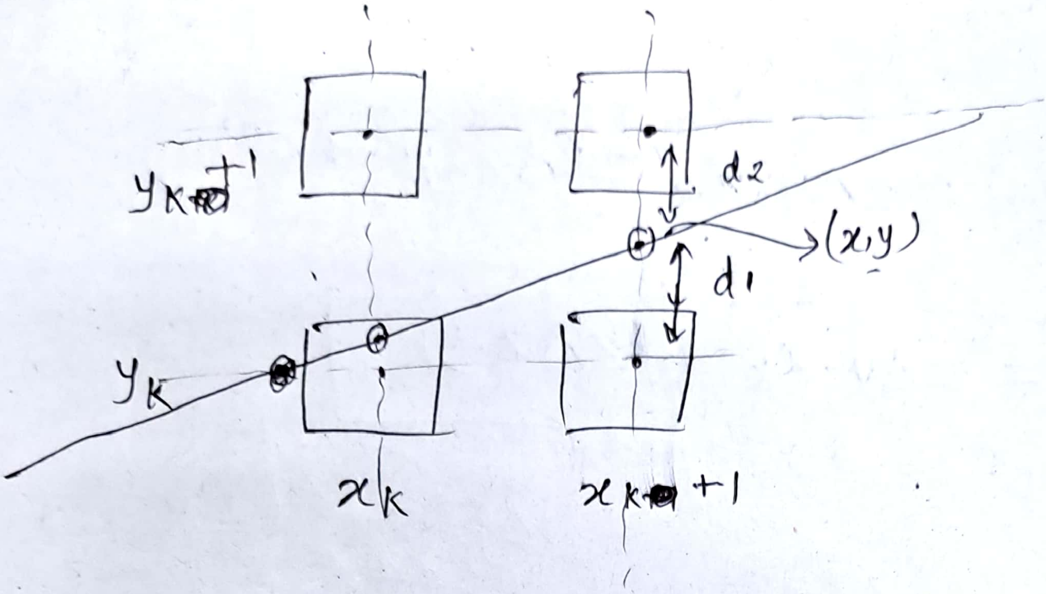
$a>b$:- Glow 3.

$\therefore \; a = d_2 = y_{k+1} - y$

$b = d_1 = y - y_k$.

If $d_1 - d_2 > 0$:- Glow 2

$d_1 - d_2 < 0$:- Glow 3.



$$d_1 = y - y_k$$

$$d_2 = y_{k+1} - y.$$

$$d_1 - d_2 = (y - y_k) - (y_{k+1} - y)$$

$$= y - y_k - y_k - 1 + y$$

$$\boxed{d_1 - d_2 = 2y - y_k - y_{k+1}} = 2y - 2y_k - 1$$

We are plotting for (x, y) which is (x_{k+1}, y).

$$x \rightarrow x_{k+1}$$

$$\therefore d_1 - d_2 = 2mx + 2c - y_k - y_{k+1}$$

$$= 2m(x_{k+1}) + 2c - y_k - y_{k+1}.$$

$$= 2m x_k + \underline{2m + 2c} - y_k - y_k - 1$$

$$= 2m x_k + c' - 2y_k - 1$$

$$d_1 - d_2 = 2 \frac{\Delta y}{\Delta x} (x_k) + c' - 2y_k - 1$$

$$\Delta x (d_1 - d_2) = 2 \Delta y \cdot x_k + \underline{\Delta x \cdot c' - \Delta x} - 2y_k \cdot \Delta x$$

$$\Delta x (d_1 - d_2) = 2 \Delta y \cdot x_k + c'' - 2y_k \cdot \Delta x.$$

Finally

$$\Delta x(d_1 - d_2) = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c''$$

$\hookrightarrow P_k$

$$\boxed{P_k = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k}$$

K, Knext

$$P_{next} = 2\Delta y \cdot x_{next} - 2\Delta x \cdot y_{next}$$

$$\therefore P_{next} - P_k = 2\Delta y\,(x_{next} - x_k) - 2\Delta x\,(y_{next} - y_k)$$

$$= \cancel{2\Delta y\,(x_k + 1 - x_k) - 2\Delta x\,(y_k + 1 - y_k)}$$

$$= \cancel{2\Delta y} - 2\Delta x.$$

$=$

If $P_{next} - P_k < 0$.

$x_k \to x_k + 1$

$y_k \to y_k$

Hence $P_{next} - P_k = 2\Delta y\,(x_k + 1 - x_k) - 2\Delta x\,(y_k - y_k)$

$$\therefore \boxed{P_{next} = P_k + 2\Delta y}$$

Scanned with OKEN Scanner

$\therefore$ If $P_{next} - P_k \geq 0$.

$y_{next} = y_k + 1$

$x_{next} = x_k + 1$

$\therefore$ $\boxed{P_{next} = P_k + 2\Delta y - 2\Delta x}$

$P_1 = 2\Delta y - \Delta x$

```
p= 2dy -dx
 x++
if (p<0)
{
   p= p + 2Δy
}
else{
   p= p+2dy - 2dx.
   y++
}.
```

Q] $P = 2dy - dx$

print $(1,1)$

print $(2,5)$

ans:-

$P = 2dy - dx$

$\Delta y = 5-1$
$= 4$

$\Delta x = 8-1 = 7$

$P = 2 \times 4 - 7$
$= 2 \cdot dy - dx$
$= 8-7$

$\boxed{P=1}$

else

{

$P>0$

Hence $y{+}{+} \rightarrow$

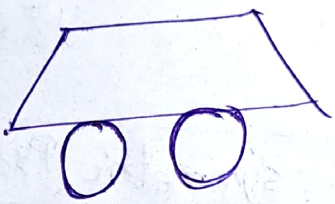
x	y	P
1	1	1
2	2	$P = 1+2\times4 - 2\times7$ $\boxed{P=-5}$
3		$(-5<0)$ $P = -5+2(4)-2(\bar{0}7)$ $P=3$

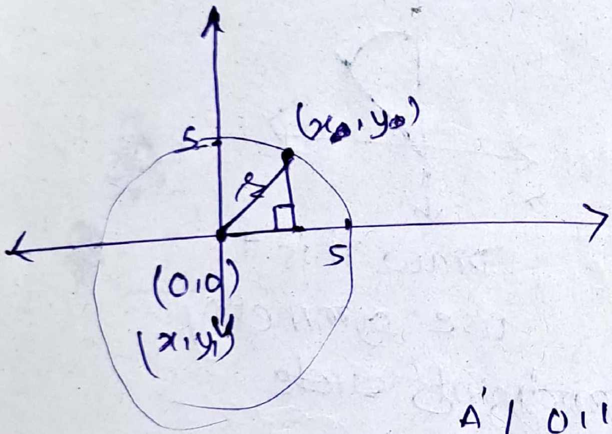
Hence, x always ++

* We have to choose which y to glow based on p at each step.

This will be used for all lines.

Draw this



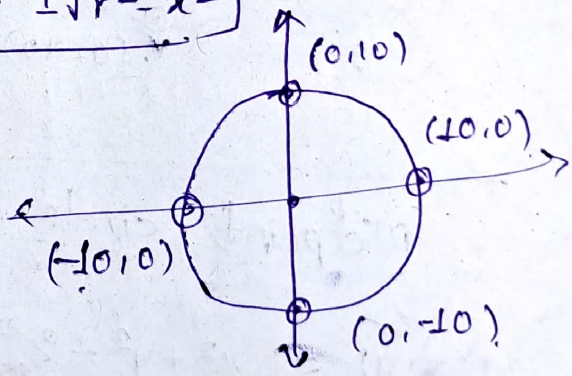


$$(x-x_1)^2+(y-y_1)^2=r^2$$

$$x^2+y^2=100$$

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

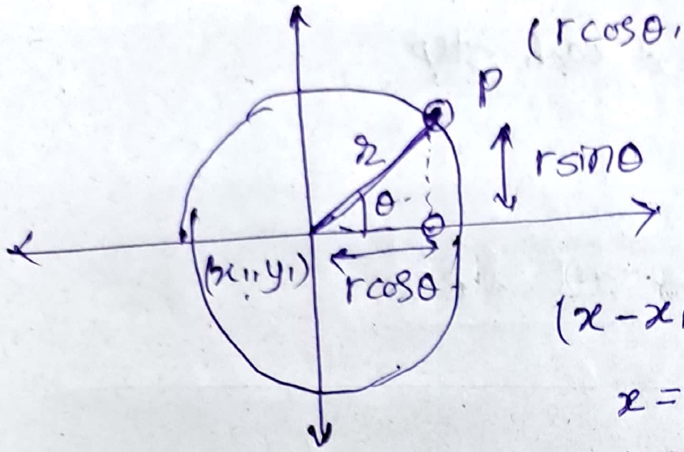
A(0,10)
B(10,0)
C(−10,0)
D(0,−10)



Drawback

1. Floating point no-s
2. square points.

Parametric Form :-

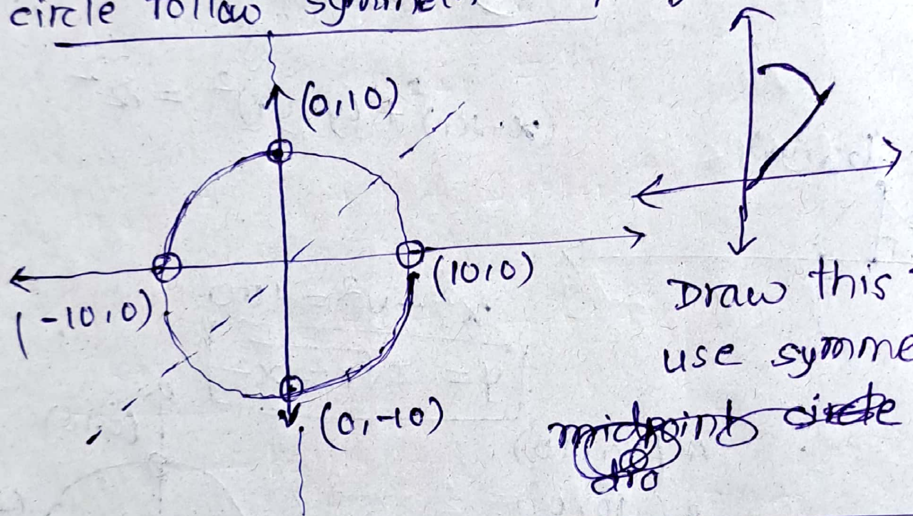


$$(x-x_1)^2 + (y-y_1)^2 = r^2$$

$$x = x_1 + r\cos\theta$$

$$y = y_1 + r\sin\theta$$

- circle follow symmetric Property



Draw this & use symmetry

~~midpoint circle~~

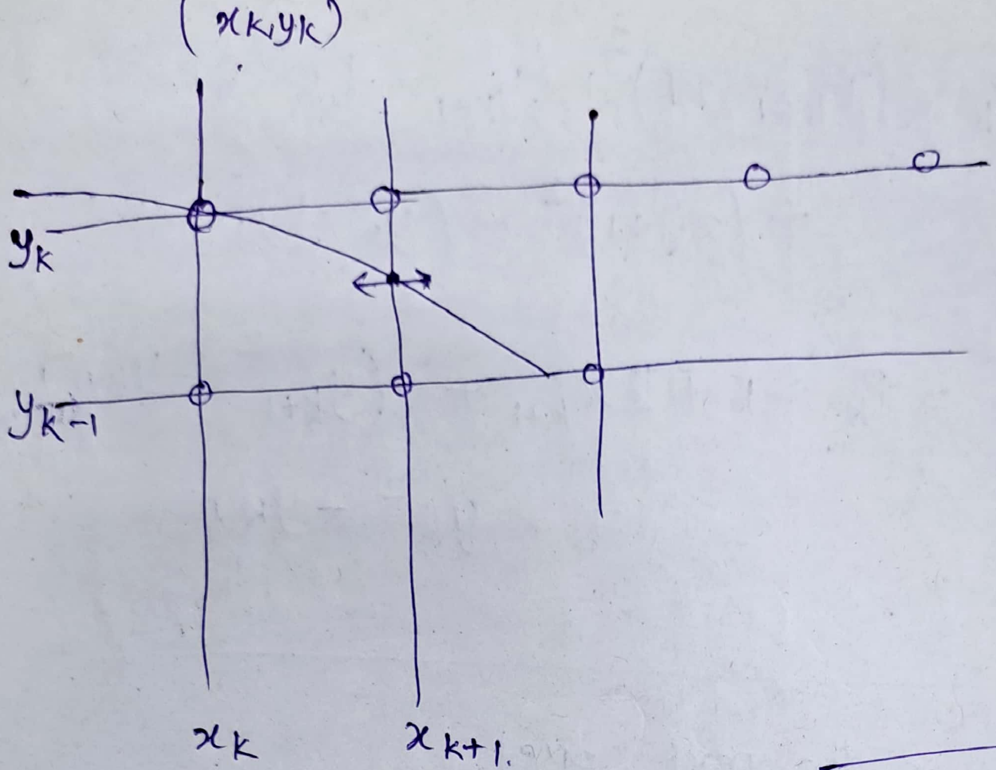
midpoint circle drawing algo

$$f(x,y) = x^2 + y^2 - r^2$$

$f(x,y) < 0 \rightarrow$ inside.

$f(x,y) > 0 \rightarrow$ outside.

$f(x,y) = 0 \rightarrow$ on the boundary



$$(x_{k+1}, y_{k+1}) \rightarrow y_k$$
$$(x_{k+1}, y_{k+1}) \rightarrow y_{k-1}$$

$$\boxed{x_{k+1} = x_k + 1}$$

$$\left.\begin{array}{l}(x_{k+1}, y_k) \\ (x_{k+1}, y_k - 1)\end{array}\right] \quad \begin{array}{l}(x_k + 1, y_k - \tfrac{1}{2}) < 0 \rightarrow (x_k + 1, y_k) \\ (x_k + 1, y_k - \tfrac{1}{2}) > 0 \rightarrow (x_k + 1, y_k - 1)\end{array}$$

$$P_{k+1} = (x_{k+1} + 1)^2 + (y_{k+1} - \tfrac{1}{2}) - r^2 \quad \text{——— (2)}$$

(2) − (1)

$$P_{k+1} - P_k = (x_{k+1} + 1)^2 + (y_{k+1} - \tfrac{1}{2})^2 - r^2$$
$$- \left[(x_k + 1)^2 + (y_k - \tfrac{1}{2})^2 - r^2 \right].$$

$$P_{k+1} - P_k = (x_{k+1} + 1)^2 + \left(y_{k+1} - \tfrac{1}{2}\right)^2 - (x_k + 1)^2 - \left(y_k - \tfrac{1}{2}\right)^2$$

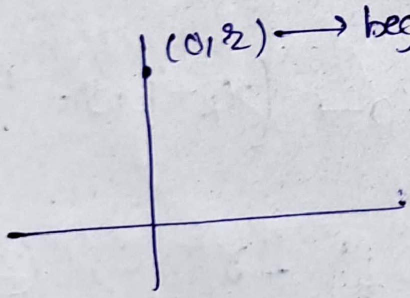
$$P_{k+1} = P_k + 1 + 2x_{k+1} + (y_{k+1}^2 - y_{k+1}) - (y_k^2 - y_k)$$

If y_k is the next pixel.

ie $y_{k+1} = y_k$

$$\therefore P_{k+1} = P_k + 1 + 2x_{k+1} + (y_k^2 - y_k) - y_k^2 + y_k$$

$$\therefore \boxed{P_{k+1} = P_k + 1 + 2x_{k+1}}$$



$(0, r)$

$(x_k + 1, y_k - \frac{1}{2})$

$(0+1, r - \frac{1}{2})$

$$x^2 + y^2 - r^2$$

$$1 + \left(r - \tfrac{1}{2}\right)^2 - r^2$$

$$1 + r^2 + \tfrac{1}{4} - r - r^2$$

$$\tfrac{5}{4} - r = 0$$

$$\therefore \boxed{r = \tfrac{5}{4}}$$

Scanned with OKEN Scanner

① $P_0 = 1 - r = \frac{5}{4} - r$

② $P_{k+1} = P_{k+1} + 2x_{k+1}$

③ $P_{k+1} = P_k + 1 + 2 \cdot (x_k + 1) \cdot - 2y_{k+1}$

---

concept: $P_0 = 1 - r$

$P_{k+1} = P_k + 1 + 2(x_k + 1) \longrightarrow P_k < 0 \quad (x{+}{+})$

$P_{k+1} = P_k + 1 + 2(x_k + 1) - 2y_{k+1} \longrightarrow P_k > 0$

Start with $(0, 10) \longrightarrow$ radius

k	P_k	x_{k+1}, y_{k+1}
0	-9	(1, 10)
1	-6	(2, 10)
2	-1	(3, 10)
3	-6	(4, 9)
4	-3	(5, 9)
5	8	(6, 8)
6	5	(7, 7)

$(x \geq y)$ stop

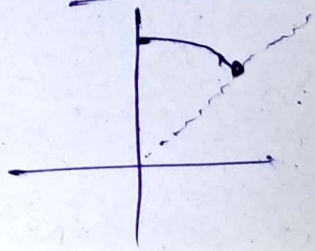
$P_0 = 1 - r = 1 - 10 = -9$ $\quad P_0 < 0$

$P_{0+1} = P_0 + 1 + 2(x_0 + 1)$

$= -9 + 1 + 2(0 + 1)$

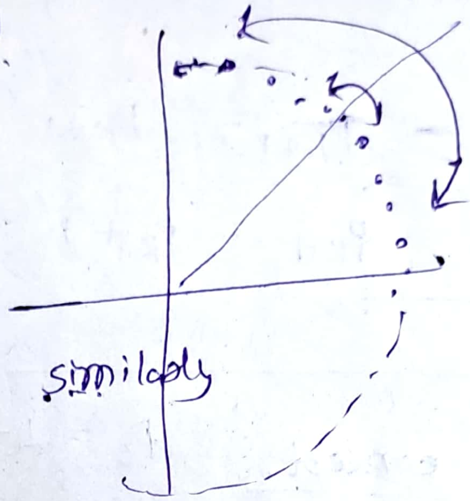
$\boxed{P_1 = -6}$

Arc complete



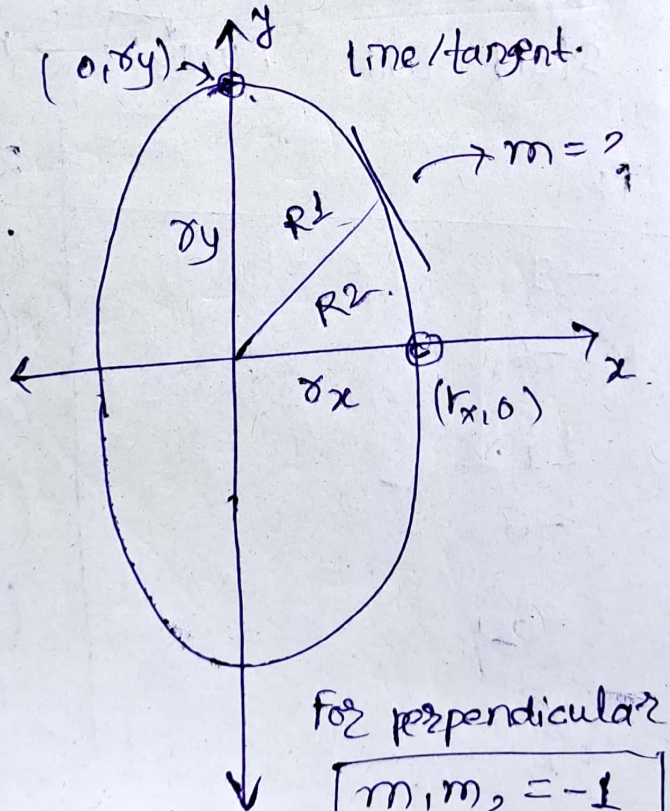
Arc

(x_{k+1}, y_{k+1})	(y_{k+1}, x_{k+1})
(1, 10)	(10, 1)
(2, 10)	(10, 2)
(3, 10)	(10, 3)
(4, 9)	(9, 4)
(5, 9)	(9, 5)
(6, 8)	(8, 6)
(7, 7)	(7, 7)

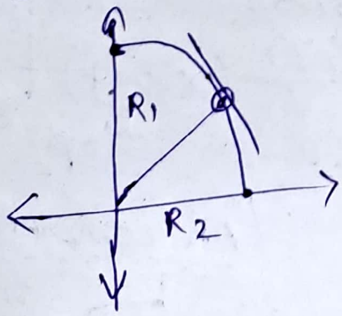


We use ellipse to draw this face.

$r_x \neq r_y$



By symmetry



For perpendicular lines:-

$$\boxed{m_1 m_2 = -1}$$

Region are divided by $m = 1$

tangent slope $= -1$

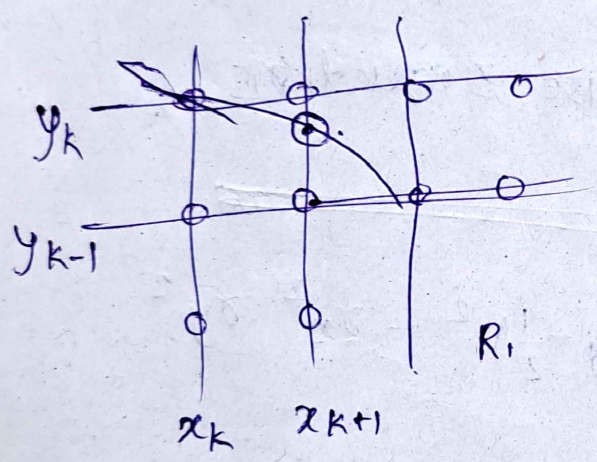
$$\frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} = 1$$

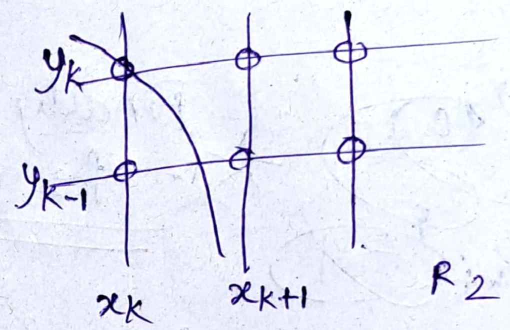
differentiate wrt x, to get slope.

$$\frac{dy}{dx} = \frac{-2 r_y^2 x}{2 r_x^2 y}$$

why 2 different regions?

ans:- due to difference in slope & point selection.





R_1	y_k
choice	y_{k-1}

R_2	x_k
choice	x_{k+1}

midpoint $\left(x_{k+1}, y_k - \frac{1}{2}\right)$ midpoint $\left(x_k + \frac{1}{2}, y_k - 1\right)$

$$P_k' = f\left(x_k + 1, y_k - \frac{1}{2}\right)$$

$$= r_y^2 (x_k + 1)^2 + r_x^2 \left(y_k - \frac{1}{2}\right) - r_x^2 r_y^2$$

$$\boxed{P_{k+1}' = P_k' + r_y^2 \left[1 + 2(x_k + 1)\right] + r_x^2 \left[y_{k+1}^2 - y_k^2 - (y_{k+1} - y_k)\right]}$$

$P_k' < 0$

$P_{k+1} = 1$

Finally use 3 equations for R1

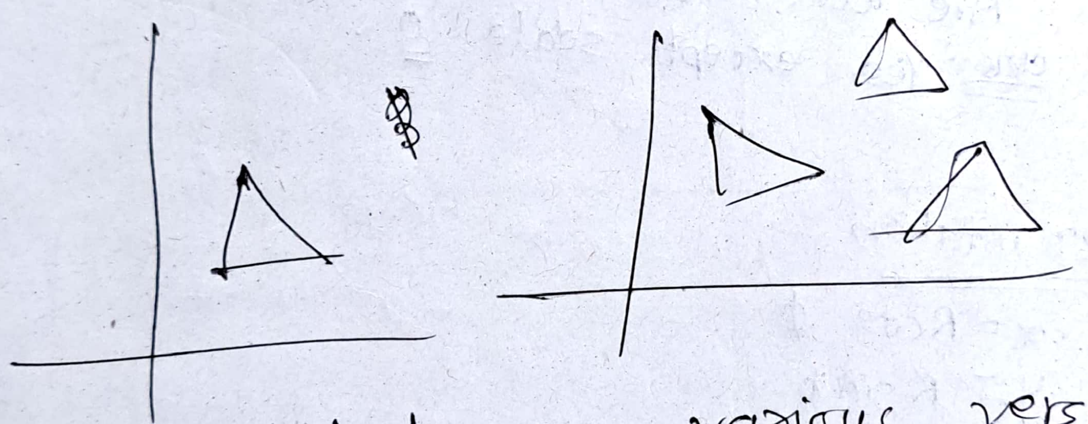
$$P_0' = r_y^2 + r_x^2 \left(r_y - \frac{1}{2}\right)^2 - r_x^2 r_y^2$$

$$P_k' < 0 \Rightarrow P_{k+1} = P_k' + r_y^2 (1 + 2(x_k + 1))$$

$$P_k' > 0 ; \quad P_{k+1} = P_k' + r_y^2 (1 + 2(x_k + 1)) + 2(r_x^2 (1 - y_k))$$

Region 2.

$$P_k^2 = \begin{cases} P_0^2 = r_y^2 \left(x_0 + \frac{1}{2}\right)^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2 \\ P_k^2 < 0 : (x_k + 1, y_k - 1) \\ P_{k+1}^2 = P_k^2 + 2 r_y^2 (x_k + 1) + r_x^2 (1 - 2(y_k - 1)) \end{cases}$$



If u want to copy various versions a figure, use Transformations.

3D

$$\begin{aligned} x' &= a_x x + b_x y + c_x z + \boxed{t_x} \\ y' &= a_y x + b_y y + c_y z + \boxed{t_y} \\ z' &= a_z x + b_z y + c_z z + \boxed{t_z} \end{aligned}$$

Translation

2D

$$\begin{aligned} x' &= a_x x + b_x y + t_p \\ y' &= a_y x + b_y y + t_q \end{aligned}$$

translation.

Transformation :- maps (x,y) in one cordinate system to points (x',y') in another cordinate system

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_x & b_x \\ a_y & b_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}.$$ 2D

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ a_z & b_z & c_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$ 3D.

Are all transformations reversible?

ans :- yes, except scale $\underline{0}$.

2D rotation

$x = R\cos\phi$

$y = R\sin\phi$

$x' = R\cos(\phi+\theta)$

$y' = R\cos(\phi+\theta)$

matrix :-

2D $$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

3D $$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$ Rotation around z-axis.

Scaling :- multiply.

scale a in x

scale by b in y

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Anti-Aliasing

Gaussian cross section

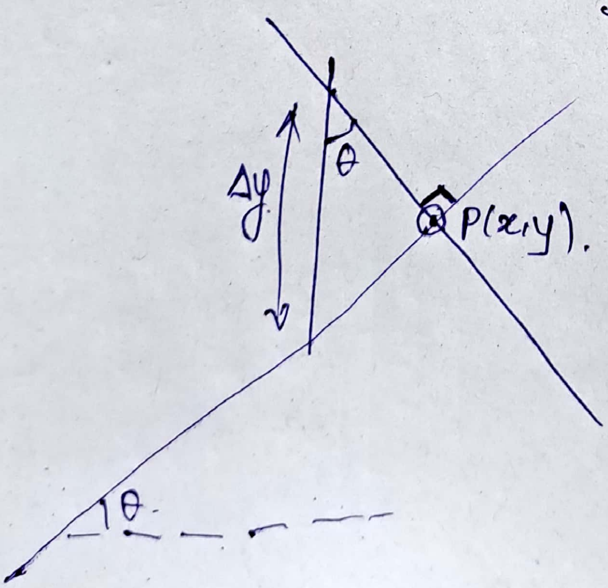
$$\cos\theta = \frac{1}{\sqrt{1+\tan^2\theta}} = \frac{1}{\sqrt{1+m^2}}$$

$$h = \frac{\Delta y}{\sqrt{m^2+1}} \qquad \sigma' = \sigma\sqrt{m^2+1}$$

$$I(h) = I_0 \cdot e^{\frac{-h^2}{2\sigma^2}}$$

$$= I_0 \cdot \exp\left(\frac{-\Delta y^2}{2\sigma^2(m^2+1)}\right)$$

$$\boxed{h = \Delta y \cos\theta}$$



$$I(\Delta y) = I_0 \cdot e^{\frac{-h^2}{2\sigma^2}} = I_0 \cdot e^{\frac{-\Delta y^2}{2\sigma^2(m^2+1)}}$$

$$= I_0 \cdot e^{\left(\frac{-\Delta y^2}{2\sigma_y^2}\right)}$$

$$I(h) = I_0 (1 - \alpha h) \Rightarrow I(\Delta y) = I_0 \left(1 - \frac{\alpha \Delta y}{\sqrt{m^2+1}}\right).$$

Linear Transformations :-

$x \in \mathbb{R}$.

const $a \in \mathbb{R}$.

$f(ax) = a f(x)$ — homogenity condition.

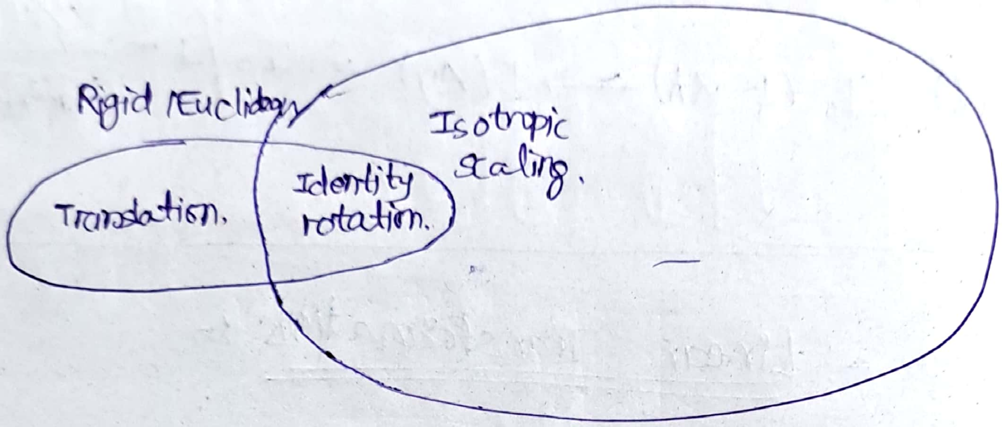
$x_1, x_2 \in \mathbb{R}$.

$$f(x_1 + x_2) = f(x_1) + f(x_2)$$

$$\frac{d}{dx}\left(a_1 f_1(x) + a_2 f_2(x)\right) = a_1 \cdot \frac{d}{dx} f_1(x) + a_2 \cdot \frac{d}{dx} f_2(x).$$

Scanned with OKEN Scanner

Thursday 7 Quiz



Q] What is an n-dimensional transformation matrix?

change of orthonormal basis:-

$$x = (\hat{x}.\hat{u})\hat{u} + (\hat{x}.\hat{v}).\hat{v} + (\hat{x}.\hat{n})\hat{n}$$

$$y = (\hat{y}.\hat{u})\hat{u} + (\hat{y}.\hat{v}).\hat{v} + (\hat{y}.\hat{n})\hat{n}$$

$$z = (\hat{z}.\hat{u})\hat{u} + (\hat{z}.\hat{v})\hat{v} + (\hat{z}.\hat{n})\hat{n}.$$

$$\begin{bmatrix} u \\ v \\ n \end{bmatrix} = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ n_x & n_y & n_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

where

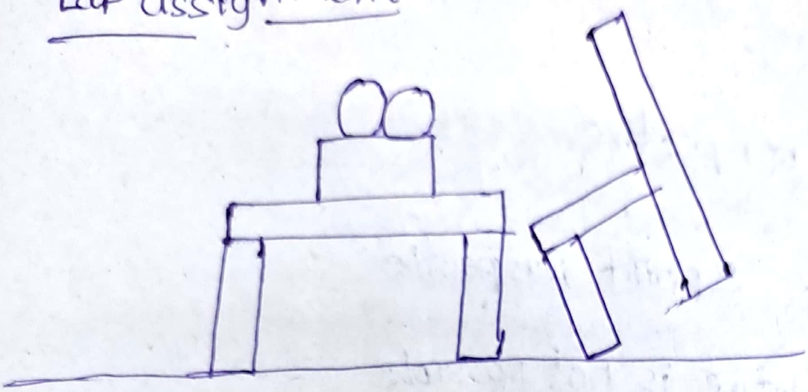
$u_x = \hat{x}.\hat{u} = \hat{x_u}$

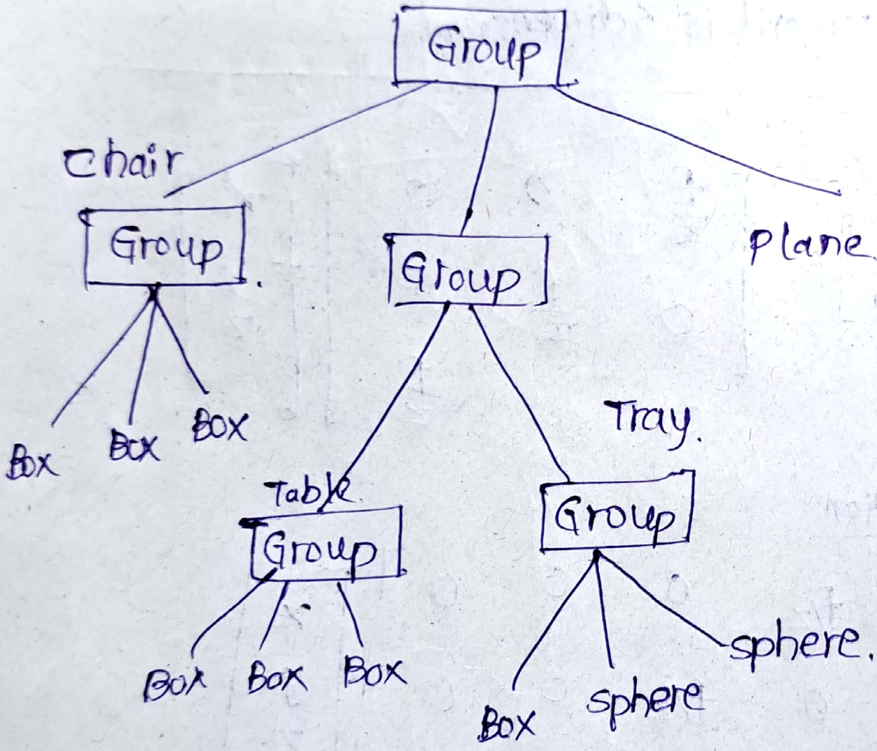
$u_y = \hat{y}.\hat{u} = \hat{y_u}$

Group:- group of 3Dshapes:-

Lab assignment

How chair hits Table & Table moves Forward?





If chair hits Table, Table moves
All objects in hierarchial Tree of Table
should move.

Donald Ham

Pavan Chakraboorthy Youtube videos

2 point, 3 point perspective

bent road has 2 point perspective

4 point perspective is not possible

$\because$ world is 3 dimensional.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \frac{f}{z} & 0 & 0 \\ 0 & \frac{f}{z} & 0 \\ 0 & 0 & \frac{f}{z} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

homogenisation

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} f/z & 0 & 0 & 0 \\ 0 & f/z & 0 & 0 \\ 0 & 0 & f/z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

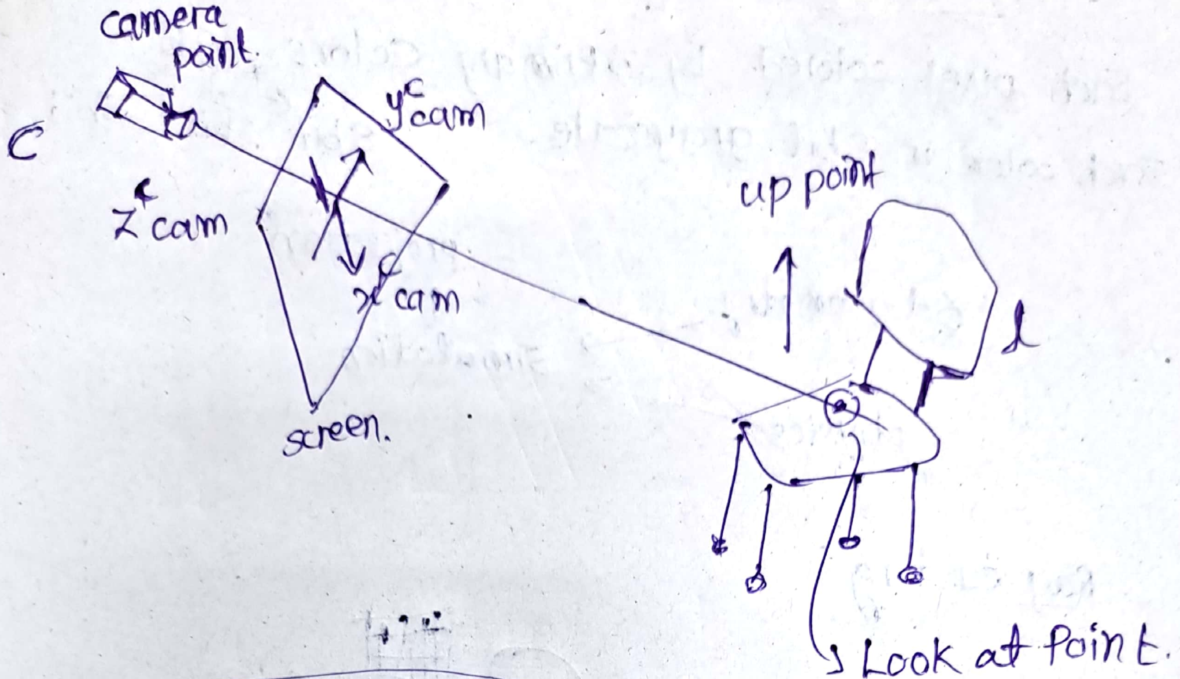
$x' = x$

$y' = y$

$z' = z$

$w' = \frac{z}{f}$

Scanned with OKEN Scanner



$\vec{c} = \{x_1^c, x_2^c, x_3^c \cdots x_n^c\}$

$\vec{l} = \{x_1^c, x_2^c, x_3^c, \cdots x_n^c\}$

Look at vector $\vec{l} - \vec{c}$.

$\vec{u} = \{x_1^n, x_2^n, \cdots\cdots, x_n^n\}$.

n-dimensional vector can be represented in 2D screen.

tsne → Large lang. models clustering