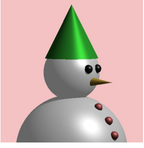
PREPARANDO A COLA PRA CG



INTRODUÇÃO

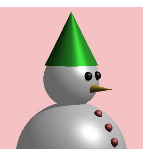
Vamos lá!

Bem vindos ao material que eu estou criando. Vamos explicar o conceito geral da renderização de objetos para dar significado a todos os passos seguintes.

O Raycasting uma técnica que independe do OpenGL que tenta representar visualmente objetos representados matematicamente da forma mais realista possível, ou tão realista quanto desejar. Apesar do OpenGL poder ser utilizado para renderizar um pixel na tela, no caso do raycasting o OpenGL não é o responsável por representar, processar, ou renderizar os objetos matematicamente definidos. Por que essa informação é importante? Pois o OpenGL tem representação de objetos, pode também renderizar objetos na tela, simular a ação de um ponto luminoso em um objeto entre outras funções. Mas nenhuma dessas funcionalidades é implementada usando o conceito de raycasting e o objetivo desse material é ensinar o conceito básico que pode ser implementado utilizando qualquer ferramenta que possibilite a renderização de um pixel, inclusive o terminal hahahah, sem depender desse método em si, ou seja, o método utilizado para processar o valor do pixel é mais importante que o método utilizado para pintar o pixel em si.

Por causa disso acho mais interessante expor aqui o pseudo-código para a implementação do raycasting na linguagem de preferência, apesar de que seja aconselhável utilizar o C++ por ser utilizado na aula.

Não sei se ficou claro, mas espero que sim.



RAYCASTING

Ideia geral do raycasting:

A partir de um observador (um ponto no sistema de coordenadas) e utilizando um plano de projeção (um plano finito entre o observador e o cenário a ser renderizado que é proporcional a janela de pixels do computador onde a imagem deve aparecer). São lançados raios (retas) que saem do observador e atravessam pontos do plano de projeção (cada ponto no plano de projeção representa um pixel a ser pintado na janela do computador) em direção ao cenário. O ponto onde a reta atravessa o objeto mais próximo ao plano de projeção é o ponto que o observador consegue ver e o qual deve ser calculada a cor.

Responsável pela renderização dos objetos na tela. Utiliza equações matemáticas de interseção com uma reta para decidir qual cor deve ser pintada no pixel do monitor.

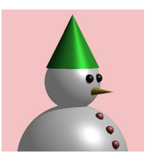
Explicando melhor: No raycasting a cor de cada pixel da tela é definida através de uma série de cálculos matemáticos que passam por:

- Descobrir o objeto que está sendo visto pelo pixel e o ponto exato que está sendo visto;
- Descobrir a cor do pixel através da interação desse objeto com a iluminação que incide nele

Para isso, algumas definições devem ser feitas para que a renderização proceda de forma desejada. São elas:

- Objetos
- Cenário
- Iluminação
  - Fontes
  - Materiais
- Observador (Câmera)
- Renderizar

Objetos (Aqui entende-se por objeto qualquer forma que deve ser renderizada, por exemplo: Se deseja-se renderizar um boneco de neve,



o boneco em si é um objeto, mas a cabeça, o corpo, o nariz e a cartola que compõem esse boneco também são objetos.)

Tem duas formas de renderizar um objeto:

Uma é renderizar cada parte do objeto utilizando formas geométricas ao qual descobrir o ponto a ser renderizado possui um cálculo já definido. Iremos aprender a renderizar as formas geométricas:

Esfera

Cone

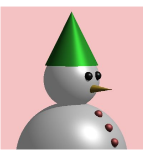
Cilindro

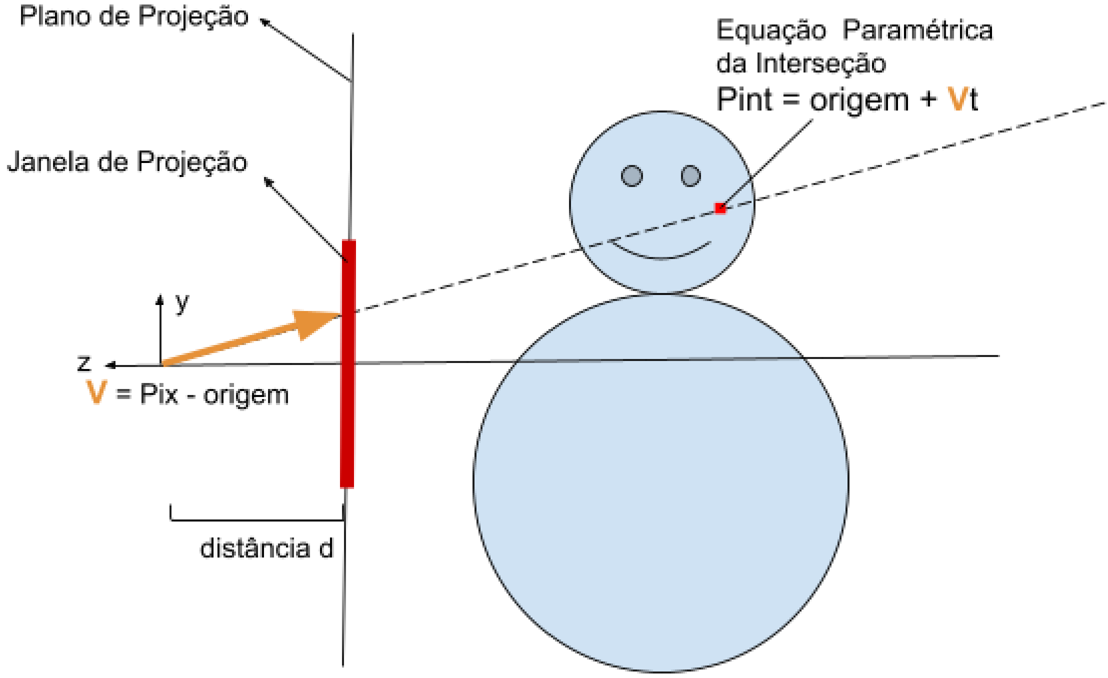
Outra forma é renderizar o objeto, independente de qual seja, a partir de aproximação da sua superfície com face triangulares. Isso dá mais liberdade de criação de objetos e é a principal forma de renderizar objetos. Iremos aprender a renderizar faces triangulares também.

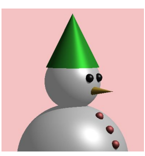
A representação dos objetos em código pode variar bastante dependendo da forma como foi projetado.

Sobre programação: Recomendo utilizar a orientação a objetos para o projeto daí é possível representar as formas geométricas e o objeto formado por aproximação de faces triangulares utilizando classes.

Os objetos podem ser definidos a partir de uma função matemática que varia de acordo com a forma geométrica desejada ou podem ser definidos a partir




Plano de Projeção
Equação Paramétrica
da Interseção
Pint = origem + Vt
Janela de Projeção
y
z
V = Pix - origem
distância d



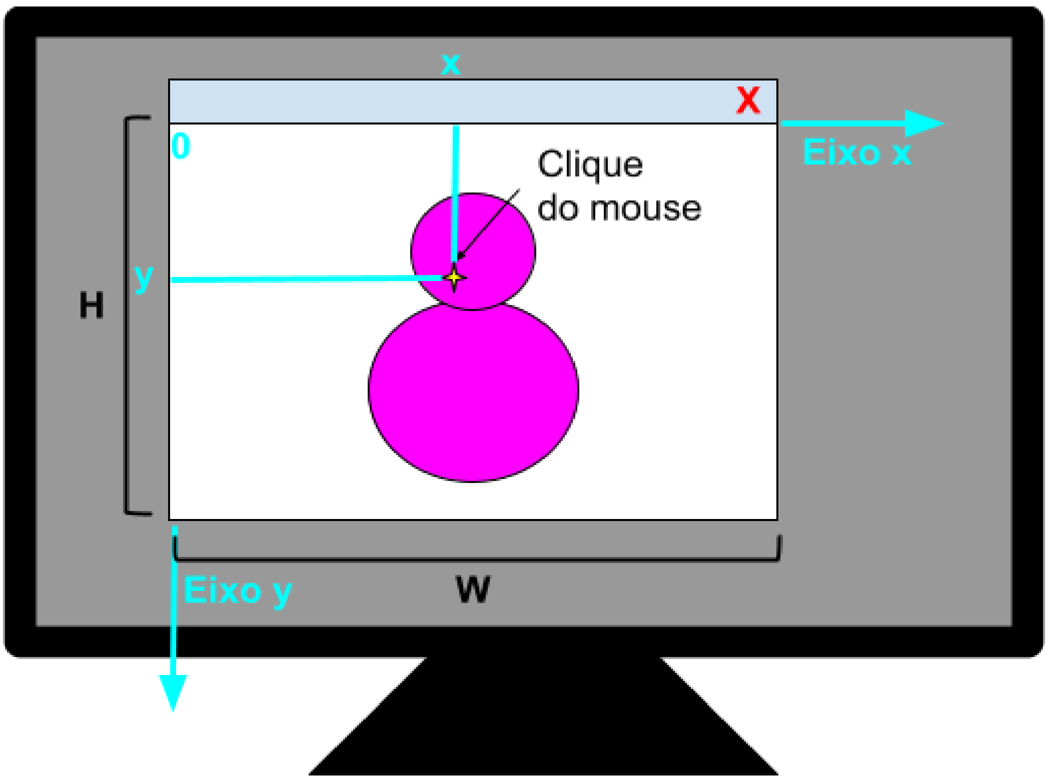
INTRODUÇÃO A MATRIZES E VETORES

Aqui vai vir toda uma revisão de soma de vetores, produto vetorial, diagonal, determinante, propriedade de vetores, produto escalar, produto vetorial.

REPRESENTAÇÃO DE OBJETOS

Aqui vai ficar a representação de ponto, vetor, face, objetos complexos, objetos matematicamente definidos, enumeração de vetores e os paranaeu lá.

MOUSE PICKING

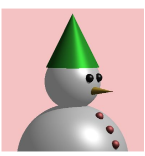


Converter o valor de (x, y) em coordenada de Janela de Projeção.

Observação: A fórmula abaixo serve apenas para a representação do desenho. Caso os eixos mudem e iniciem em outro valor é necessário modificar

Para isso, vamos usar a equação que dada uma linha e uma coluna ela nos dá o centro do **Quadradinho** da Janela de Projeção:

$$Projeção_x = \frac{Wprojeção}{2} - \frac{\Delta x}{2} - x * \Delta x$$

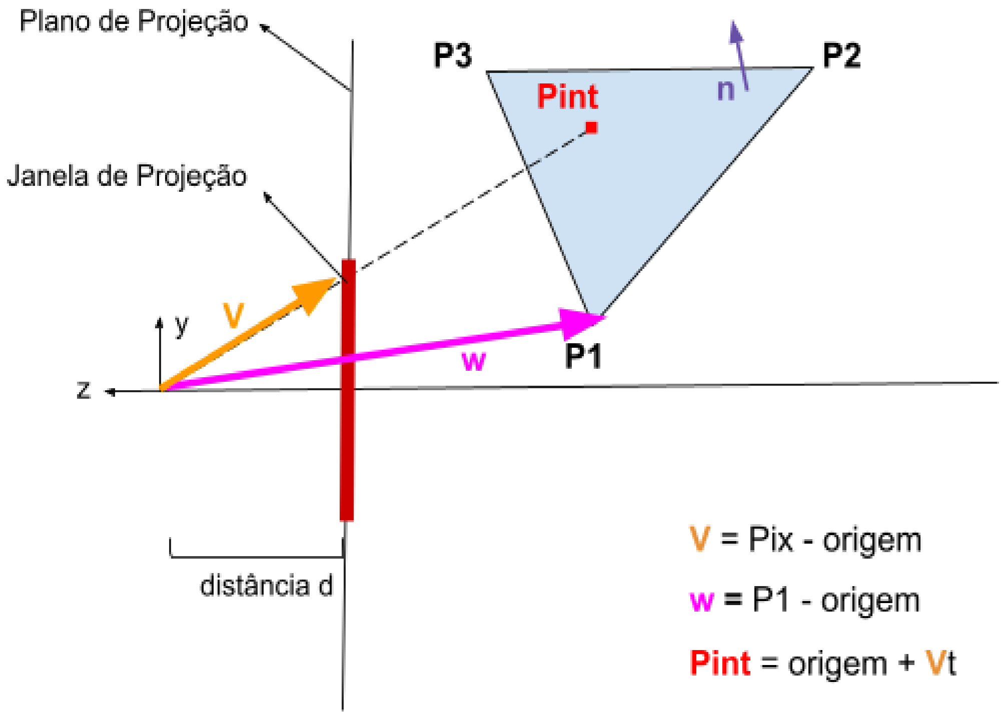


$$Projeção_y = \frac{Hprojeção}{2} - \frac{\Delta y}{2} - y * \Delta y$$

Com x e y os valores do mouse no monitor, considerando que o primeiro pixel tem coordenadas (0, 0) e $\Delta y = \frac{Hprojeção}{H}$ e $\Delta x = \frac{Wprojeção}{W}$

Daí a gente encontra os valores de V na equação: $P_{int} = O + Vt$. Daí é só encontrar os valores do raycasting normal, lembrando que a coordenada z do vetor V é - d, com d sendo a distância da janela ao plano xy.

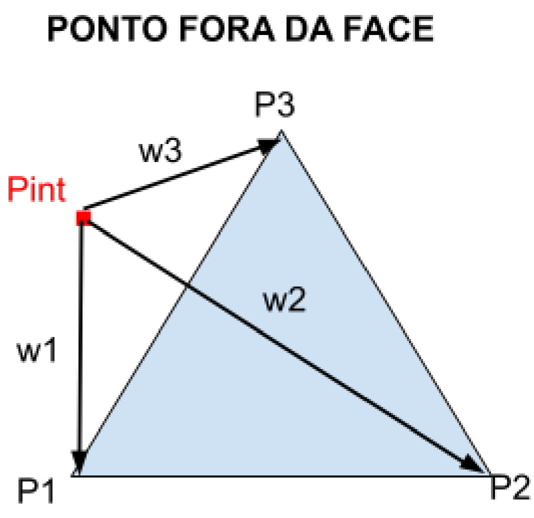
EQUAÇÕES DE INTERSECÇÃO - PLANO



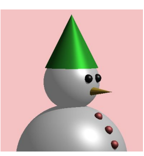
A equação para descobrir o valor t que aplicado a equação paramétrica da reta intercepta plano infinito contendo a face triangular.

$$t = \frac{w.n}{v.n}$$

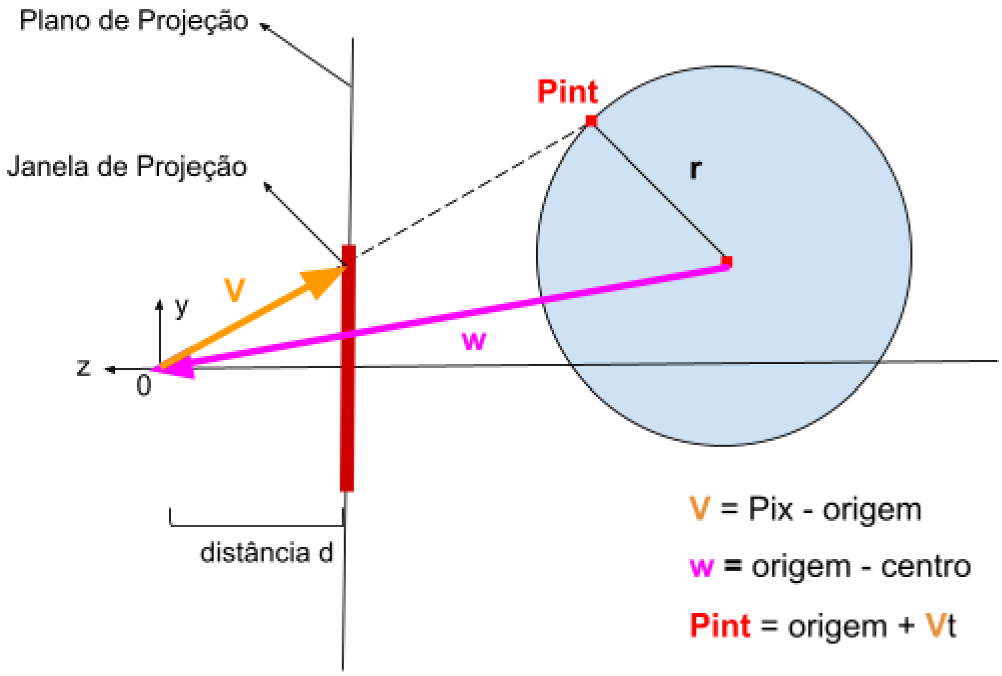
Depois disso, é necessário saber se o ponto de interseção está contido na face triangular. Para isso, há duas formas de proceder:



Se o sinal de w1 x w2 . n for o mesmo de w2 x w3 . n e o mesmo de w3 x w1 . n temos que o ponto pertence a face triangular.
Se não o ponto não pertence ao plano.



EQUAÇÕES DE INTERSECÇÃO - ESFERA



Equação para encontrar o valor de t: $at^2 + bt + c = 0$ com:

$$a = v.v$$
$$b = 2 * w.v$$
$$c = w.w - r^2$$

É só resolver a equação de segundo grau e ver se o $\Delta = b^2 - 4 * a * c \geq 0$ se sim, resolve-se a equação achando $t_1 = \frac{-b+\sqrt{\Delta}}{2*a}$ e $t_2 = \frac{-b-\sqrt{\Delta}}{2*a}$ e escolhe-se o menor valor de t como resultado, depois só substituir o t em $P_{int} = origem + v * t$ como a origem é sempre o vetor nulo temos $P_{int} = v * t$. Daí temos o P_{int} para fazer o que quiser com ele.

Importante: Após descobrir o ponto de intersecção é importante guardar também o objeto ao qual o ponto de intersecção pertence.

EQUAÇÃO PARAMÉTRICA T_T:

Fórmula a ser resolvida, suponha sem perda de generalidade que a equação paramétrica a ser resolvida é esta:

$$\frac{(x - x_c)^2}{2} + \frac{(y - y_c)^2}{2} + \frac{(z - z_c)^2}{2} - 1 = 0$$

com $P_c = \{0,\ 0,\ -10\}$. Daí o que a gente faz? A gente vai e substitui P_c na equação.

$$\frac{(x - 0)^2}{2} + \frac{(y\ -\ 0)^2}{2} + \frac{(z - (-10))^2}{2} - 1 = 0$$

$$\frac{x^2}{2} + \frac{y^2}{2} + \frac{(z + 10)^2}{2} - 1 = 0$$

Daí a gente substitui os valores das componentes x, y e z do vetor Dt da equação paramétrica do raio de interseção. Ou seja, o pixel da janela de projeção multiplicado pelo valor de t para a gente ter uma equação em função do nosso querido t. Substituindo na equação:

$$\frac{t^2 Dx^2}{2} + \frac{t^2 Dy^2}{2} + \frac{(tDz + 10)^2}{2} - 1 = 0$$

$$\frac{t^2 Dx^2}{2} + \frac{t^2 Dy^2}{2} + \frac{t^2 Dz^2 + t20Dz + 10^2}{2} - 1 = 0$$

$$\frac{t^2 Dx^2}{2} + \frac{t^2 Dy^2}{2} + \frac{t^2 Dz^2}{2} + \frac{t20Dz}{2} + \frac{10^2}{2} - 1 = 0$$

Daí surgiu a equação do segundo grau $at^2 + bt + c = 0$ com:

$$a = \frac{Dx^2}{2} + \frac{Dy^2}{2} + \frac{Dz^2}{2}$$

$$b = \frac{20Dz}{2}$$

$$c = \frac{10^2}{2} - 1$$

Como a gente já encontrou o vetor D, temos os valores de Dx, Dy e Dz. Daí é só resolver T-T.

Daí pra **encontrar a normal** do ponto de uma equação paramétrica tem que fazer o gradiente da função paramétrica. Escreve a fórmula que tu ganha a questão toda:

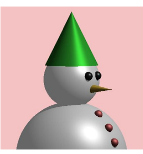
$$N = \left(\frac{\delta}{\delta x}Elipsoide,\ \frac{\delta}{\delta y}Elipsoide,\ \frac{\delta}{\delta z}Elipsoide\right)$$

só derivar ae galera

$$\frac{\delta}{\delta x}Elipsoide = \frac{d}{dx}\frac{x^2}{a^2}$$

$$\frac{\delta}{\delta y}Elipsoide = \frac{d}{dy}\frac{y^2}{b^2}$$

$$\frac{\delta}{\delta z}Elipsoide = \frac{d}{dz}\left(\frac{d}{dz}\frac{(z + 2d)^2}{c^2} - 1\right)$$



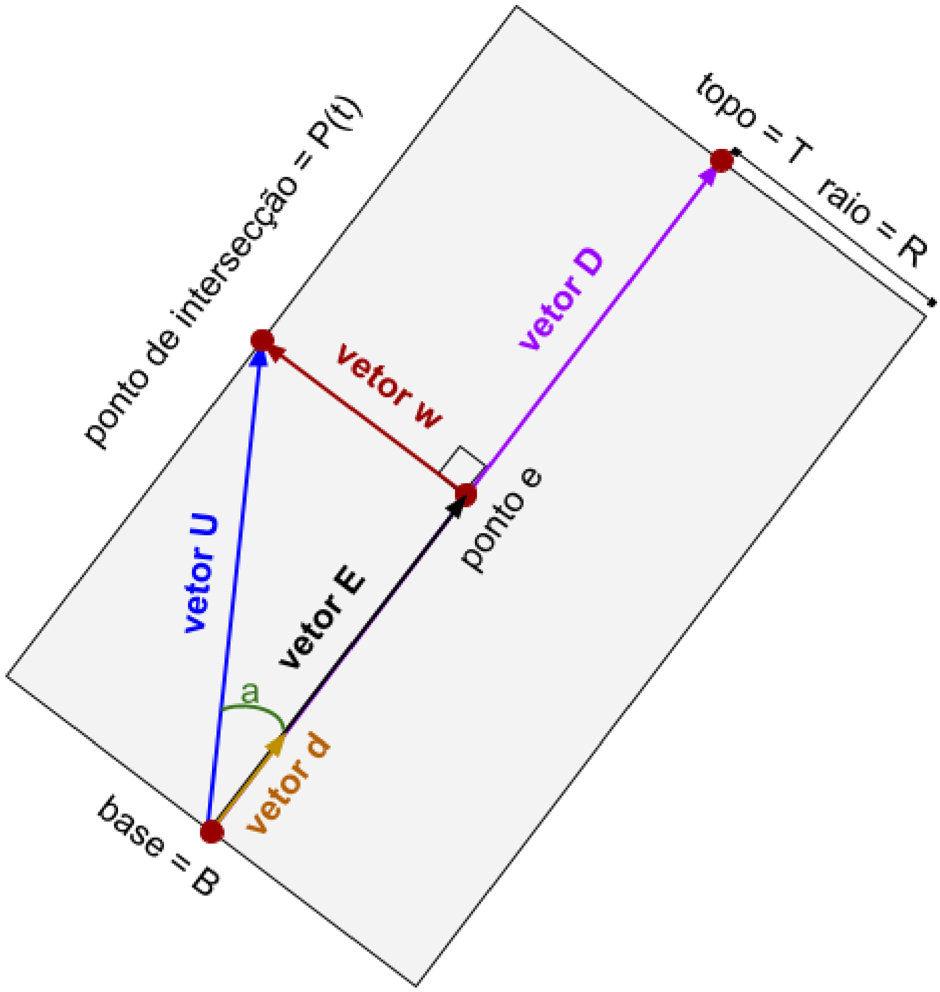
$$n = \frac{N}{||N||}$$

EQUAÇÃO DE INTERSECÇÃO - CILINDRO

Bom, o objetivo é encontrar uma equação que quando resolvida determine se uma reta atravessa a superfície de um objeto cilíndrico.
Há várias formas de definir matematicamente um cilindro, uma delas é:

> Um ponto de base
> Um ponto de altura
> Um valor de raio

Com essas informações vamos deduzir a equação



Todos os vetores apresentados são auxiliares necessários para o cálculo da intersecção.
Os valores de T, B e R são conhecidos e $P(t) = O + Vt$
Definindo matematicamente os vetores:

> $U = P(t) - B$
> $D = T - B$
> $d = D / ||D||$
> $w = U - E$

$E = ||E|| \cdot d$

Agora vamos criar uma definição mais pertinente para o vetor E. Pela trigonometria é possível observar que:

$\cos(a) = ||E|| / ||U||$ daí temos

$||E|| = ||V|| \cdot \cos(a)$

Daí é importante lembrar da relação de vetores onde $A \cdot B = ||A|| \cdot ||B|| \cdot \cos(\text{ângulo entre eles})$

Como d é normalizado, temos que $||d|| = 1$ daí temos que:

$||V|| \cdot \cos(a) = V \cdot d$

Substituindo em $||E|| = ||V|| \cdot \cos(a)$ temos

$||E|| = V \cdot d$

Substituindo $||E||$ em $E = ||E|| \cdot d$ temos

$E = (V \cdot d) \cdot d$

Substituindo o valor de E na equação de $w = U - E$ temos:

$w = U - (U \cdot d) \cdot d$

É possível observar que cada um dos componentes de w é obtido através da fórmula:

$w1 = U1 - (U \cdot d) \cdot d1$

$w2 = U2 - (U \cdot d) \cdot d2$

$w3 = U3 - (U \cdot d) \cdot d3$

Escrevendo de forma matricial obtemos:

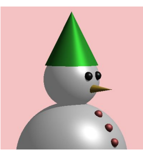
$$\underset{I}{\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}} \cdot \underset{U}{\begin{vmatrix} U1 \\ U2 \\ U3 \end{vmatrix}} - \underset{N}{\begin{vmatrix} d1 \cdot d1 & d1 \cdot d2 & d1 \cdot d3 \\ d2 \cdot d1 & d2 \cdot d2 & d2 \cdot d3 \\ d3 \cdot d1 & d3 \cdot d2 & d3 \cdot d3 \end{vmatrix}} \cdot \underset{U}{\begin{vmatrix} U1 \\ U2 \\ U3 \end{vmatrix}}$$

Isolando U temos:

$w = (I - N) \cdot U = M \cdot U$

Agora temos todos os ingredientes para fazer a dedução:

Primeiro temos que $||w|| \cdot ||w|| = R^2$. Como $w \cdot w = ||w|| \cdot ||w|| \cdot \cos(0)$ temos que $w \cdot w = ||w|| \cdot ||w||$ logo: $w \cdot w = R^2$



Substituindo os valores em $U = P(t) - B$ temos:

$$U = O + Vt - B = (O - B) + Vt$$

Substituindo o valor de U em $w = M . U$ temos:

$$w = M . ((O - B) + Vt)$$

$$w = M . (O - B) + M . Vt$$

Substituindo w em $w . w = R^2$ temos:

$$w . w - R^2 = (M . (O - B) + M . Vt) . (M . (O - B) + M . Vt) - R^2$$

$$= ((M . (O - B))^2 + 2 (M . (O - B) . M . Vt) + (M . Vt)^2) - R^2$$

$$= (M . V)^2 t^2 + 2 (M . (O - B) . M . V)t + (M . (O - B))^2 - R^2$$

Daí temos $\alpha t^2 + \beta t + \gamma = 0$ com:

$$\alpha = (M . V)^2$$

$$\beta = 2 (M . (O - B) . M . V)$$

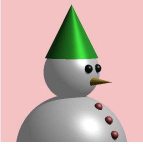
$$\gamma = (M . (O - B))^2 - R^2$$

Resolvendo utilizando báskaras temos:

Se $\Delta = 0$, intercepta a borda do cilindro,

Se $\Delta > 0$, corta o cilindro em dois pontos e

Se $\Delta < 0$, não intercepta o cilindro.



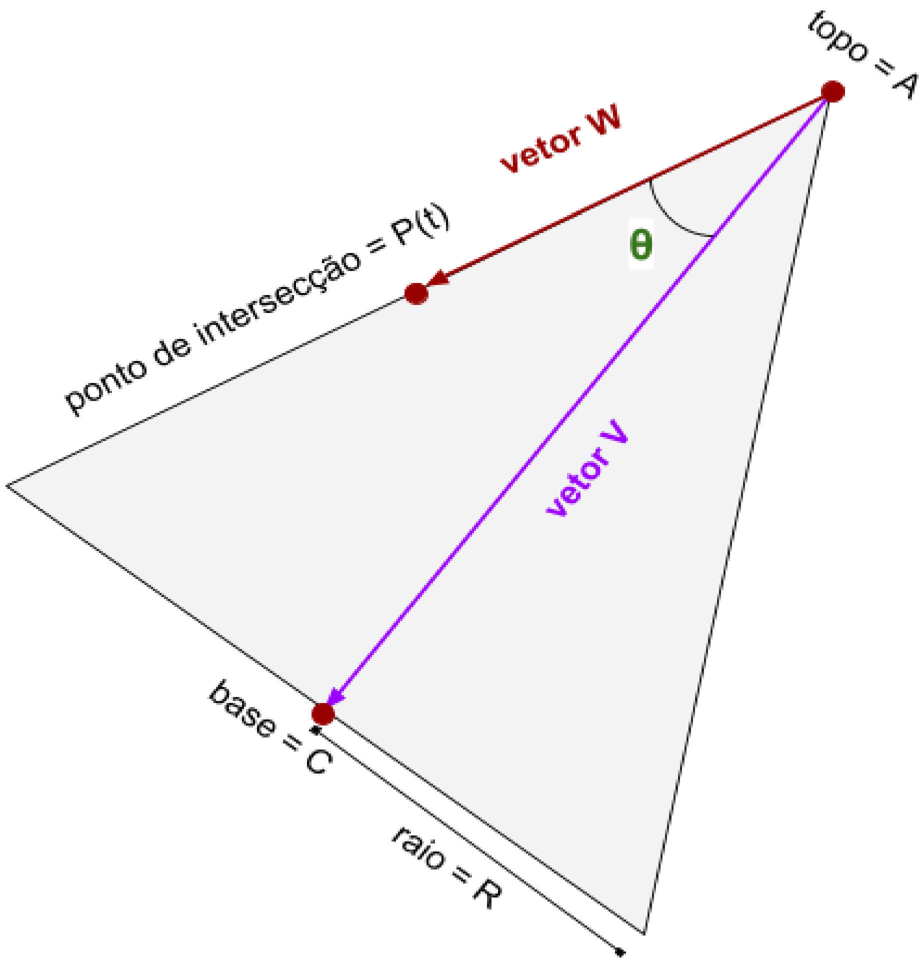
EQUAÇÃO DE INTERSECÇÃO - CONE

Bom, o objetivo é encontrar uma equação que descreva a intersecção de um cone e uma reta matematicamente definidos.

Há várias formas de definir matematicamente um cone, uma delas é:

- Um ponto de base
- Um ponto de topo
- Um valor de raio da base

Com essas informações vamos deduzir a equação



Todos os vetores apresentados são auxiliares necessários para o cálculo da intersecção.

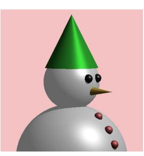
Os valores de C, A, R e θ são conhecidos e $P(t) = O + Dt$

Definindo matematicamente os vetores:

$W = P(t) - A$

$V = C - A$

$H = ||V||$



$w = W / ||W||$

$v = V / ||V||$

Como $v . w = ||v|| . ||w|| . \cos(\theta)$ e como $||v|| = ||w|| = 1$ temos:

$v . w = \cos(\theta)$

Se isso for verdade temos que o ponto de intersecção pertence ao cone.

Deduzindo a fórmula final:

$v . w = \cos(\theta)$

$v . W / ||W|| = \cos(\theta)$

$v . W / \sqrt{W . W} = \cos(\theta)$

$v . W = \cos(\theta) . \sqrt{W . W}$

Elevando ambos os lados ao quadrado

$(v . W)^2 = \cos^2(\theta) . W^2$

Substituindo $W = P - A$

$((P - A)v)^2 = \cos^2(\theta) . (P - A)^2$

$(Pv - Av)^2 = \cos^2(\theta) . (P^2 - 2PA + A^2)$

$(Pv)^2 - 2PvAv + (Av)^2 = \cos^2(\theta) . (P^2 - 2PA + A^2)$

Substituindo $P = O' + Dt$ temos:

$((O' + Dt)v)^2 - 2(O' + Dt)vAv + (Av)^2 = \cos^2(\theta) . ((O' + Dt)^2 - 2(O' + Dt)A + A^2)$

$((O' + Dt)v)^2 - 2(O' + Dt)vAv + (Av)^2 - \cos^2(\theta) . ((O' + Dt)^2 - 2(O' + Dt)A + A^2) = 0$

Simplificando temos:

$((O'v + Dvt)^2 - 2O'vAv - 2DvAvt + (Av)^2 - \cos^2(\theta) . ((O'^2 + 2O'Dt + (Dt)^2 - 2O'A - 2DAt + A^2) = 0$

$((O'v + Dvt)^2 - 2O'vAv - 2DvAvt + (Av)^2 - \cos^2(\theta) . ((O'^2 + 2O'Dt + (Dt)^2 - 2O'A - 2DAt + A^2) = 0$

$(O'v)^2 + 2O'vDvt + (Dv)^2t^2 - 2O'vAv - 2DvAvt + (Av)^2 - O'^2\cos^2(\theta) - 2O'D\cos^2(\theta)t - D^2\cos^2(\theta)t^2 + 2O'A\cos^2(\theta) + 2DA\cos^2(\theta)t - A^2\cos^2(\theta) = 0$

Daí temos $\alpha t^2 + \beta t + \gamma = 0$ com:

$\alpha = (Dv)^2 - D^2\cos^2(\theta)$

$\beta = 2O'vDv - 2DvAv - 2O'D\cos^2(\theta) + 2DA\cos^2(\theta)$

$\gamma = (O'v)^2 - 2O'vAv + (Av)^2 - O'^2\cos^2(\theta) + 2O'A\cos^2(\theta) - A^2\cos^2(\theta)$

Resolvendo utilizando báskaras temos:

Se $\Delta = 0$, intercepta a borda do cone,

Se $\Delta > 0$, corta o cone em dois pontos e

Se $\Delta < 0$, não intercepta o cone.

Além disso é importante atentar aos seguintes aspectos:

1. É necessário que ||W|| seja menor que a lateral do cone, caso contrário o ponto de intersecção não pertence ao cone especificado
2. Para evitar calcular um falso positivo por causa da similaridade dos ângulos internos dos vetores e por causa do cálculo do cosseno. Então é importante verificar o sinal do produto vetorial de V e W. Caso seja positivo, os vetores estão na mesma direção, logo o ponto pertence ao cone, caso seja negativo, os vetores estão em direções opostas portanto o vetor não pertence ao cone.

QUATÉRNIO:

Quatérnio é usado para rotacionar vetores em eixos arbitrários. Eles são formados por uma parte vetorial e uma parte escalar.

$$q = \{\overbrace{\boxed{x\ ,\ y\ ,\ z}}^{\text{VETORIAL}}, \underbrace{\boxed{w}}_{\text{ESCALAR}}\}$$

Daí, como acontece a rotação, para isso devemos ter um vetor unitário do u eixo de rotação e o grau α de rotação da figura. O quatérnio de rotação será formado por:

$$q = \{sen(\tfrac{\alpha}{2})u_x,\ sen(\tfrac{\alpha}{2})u_y,\ sen(\tfrac{\alpha}{2})u_z,\ cos(\tfrac{\alpha}{2})\}$$

Daí para aplicar o quatérnio nos pontos devemos criar as duas matrizes e lembrar de que **se o vetor u não passa pela origem** do sistema de coordenadas, devemos transladar ele para a origem do eixo de coordenadas. Assim teremos se u não passa pela origem:

$$P_f = [T_{volta}][Quat\acute{e}rnio][T_{ida}]P_i$$

Se u passa pela origem temos:

$$P_f = [Quat\acute{e}rnio]P_i$$

Com: $[T_{ida}]$ matriz que vai de algum ponto de u para a origem; $[T_{volta}]$ matriz que volta o ponto de u que foi levado até a origem para o seu lugar correto e finalmente $[Quat\acute{e}rnio]$ que é a matriz que o professor passou no slide:

$$\begin{bmatrix} (w^2 + x^2 - y^2 - z^2) & (2xy - 2wz) & 2xz + 2wy & 0 \\ (2xy + 2wz) & (w^2 - x^2 + y^2 - z^2) & (2yz - 2wx) & 0 \\ (2xz - 2wy) & (2yz + 2wx) & (w^2 - x^2 - y^2 + z^2) & 0 \\ 0 & 0 & 0 & (w^2 + x^2 + y^2 + z^2) \end{bmatrix}$$

Daí temos todos os ingredientes para a prova de terça, acho, quero, espero….

Quer mais algum conteúdo na nossa cola? Fale conosco!

SUMÁRIO inCOMPLETO