UNIDADE II

Computação Gráfica

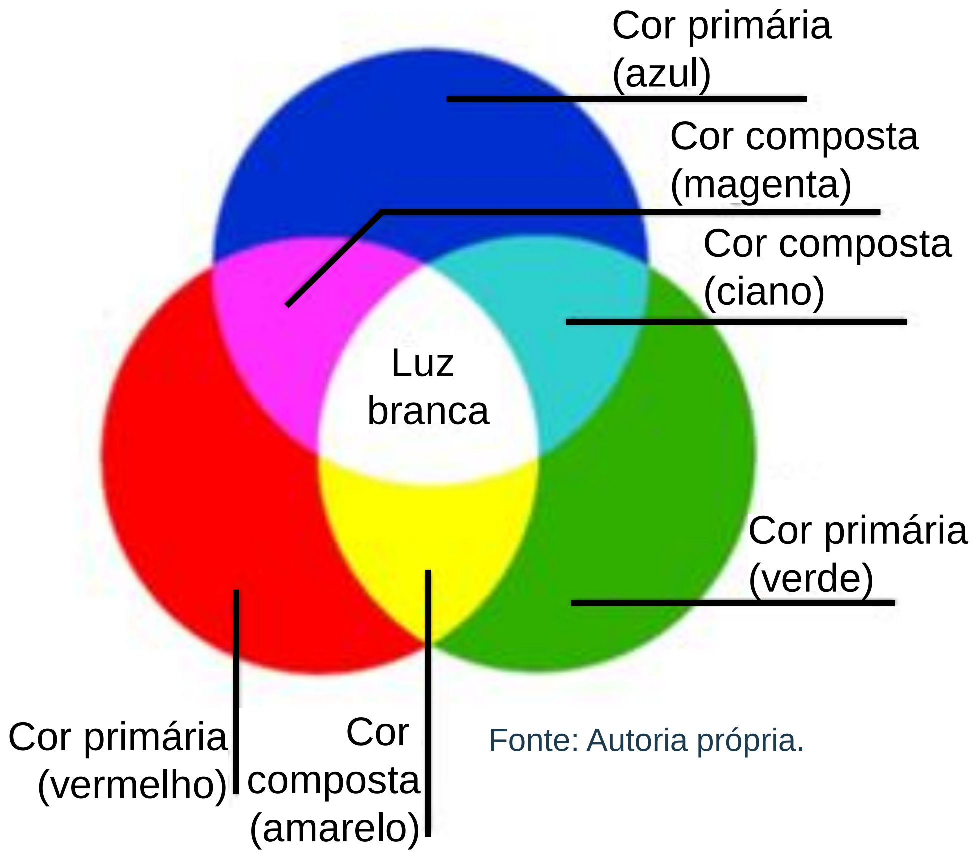
Prof. Hugo Insua

Síntese de cores

- A síntese de cores em computação gráfica é o processo de combinar diferentes cores para criar uma imagem ou cena visual. Existem dois principais métodos de síntese de cores: aditivo e subtrativo.

Síntese de cores

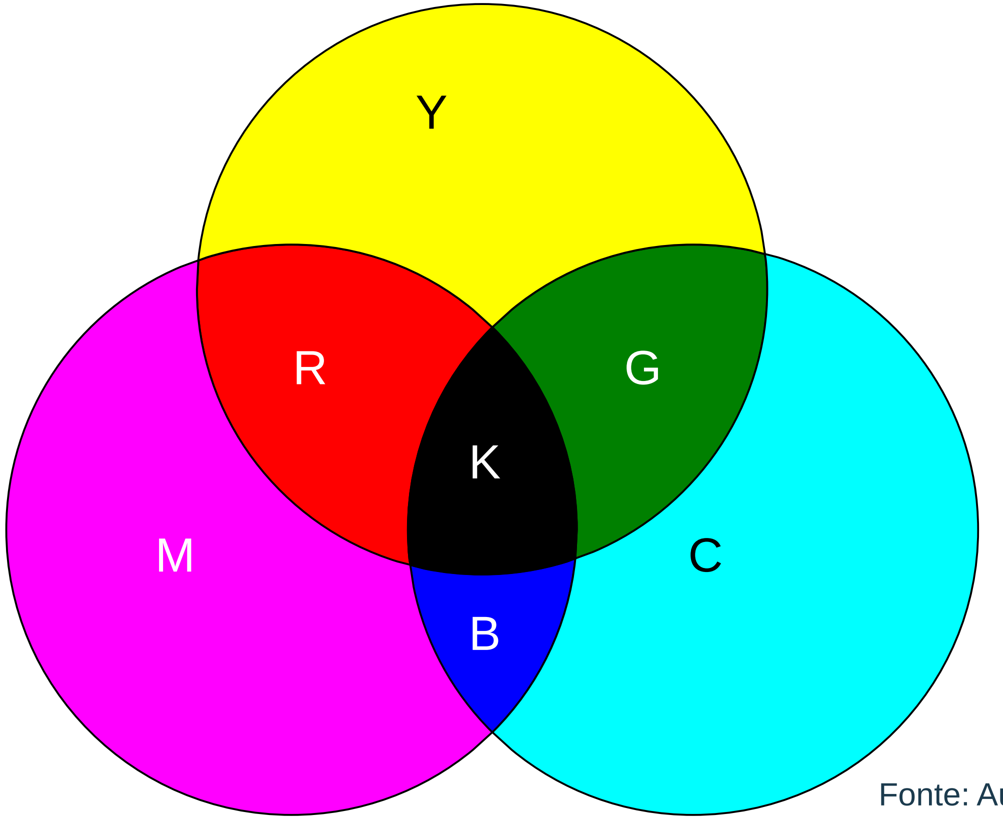
- A síntese de cores aditiva é usada em dispositivos que emitem luz, como monitores de computador, televisores e projetores. Nesse método, as cores primárias (vermelho, verde e azul) são combinadas em diferentes intensidades para criar uma ampla gama de cores. Por exemplo, se o vermelho e o verde são combinados em intensidades iguais, surge o amarelo. Se as três cores primárias são combinadas em intensidades iguais, surge o branco.



Fonte: Autoria própria.

Síntese de cores

- Por outro lado, a síntese de cores subtrativa é usada em dispositivos que refletem a luz, como impressoras e tintas. Nesse método, as cores primárias (ciano, magenta e amarelo) são combinadas em diferentes quantidades para criar uma ampla gama de cores. Por exemplo, se o ciano e o magenta são combinados em intensidades iguais, eles criam a cor azul. Se todas as três cores primárias são combinadas em quantidades iguais, elas formam o preto.



Fonte: Autoria própria.

Síntese de cores

Luz e cores

- O modelo de cores mais comum é o RGB (Red, Green, Blue), que utiliza as cores primárias de luz para representar todas as outras cores. Cada *pixel* em uma imagem é composto por uma combinação de valores de intensidade dessas três cores, que variam de 0 a 255. Por exemplo, um *pixel* com valor de intensidade (255,0,0) representa a cor vermelha pura, enquanto um *pixel* com valor de intensidade (0,255,0) representa a cor verde pura.

Síntese de cores

Luz e cores

- Outro modelo de cores comum é o CMY (Ciano, Magenta, Amarelo), que é utilizado principalmente para impressão. Nesse modelo, as cores primárias são as cores secundárias do modelo RGB.

Síntese de cores

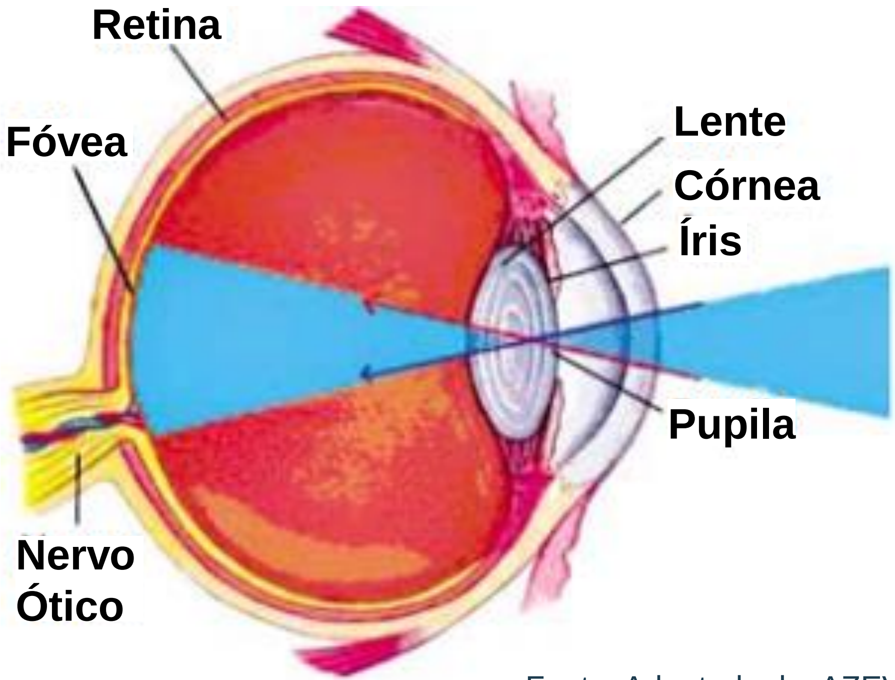
Luz e cores

- O sistema de iluminação mais utilizado em computação gráfica é o modelo de iluminação de Phong, que simula a iluminação de objetos em uma cena 3D. Este modelo leva em consideração três componentes principais de luz: luz ambiente, luz difusa e luz especular. A luz ambiente representa a luz que é refletida em todas as direções e é geralmente usada para simular a iluminação de fundo em uma cena. A luz difusa representa a luz que é refletida em todas as direções de maneira uniforme, e a luz especular representa a luz que é refletida de maneira mais concentrada, criando reflexos brilhantes em superfícies lisas.

Síntese de cores

Sistema visual humano

- O olho humano funciona como uma lente que capta luz por meio da córnea e a transforma em sinais elétricos, que são interpretados pelo cérebro.
- A córnea é responsável por focar a luz na retina, que é a camada sensível à luz na parte de trás do olho. A retina contém células fotossensíveis chamadas de cones e bastonetes, que são responsáveis por detectar a luz e transmitir sinais elétricos para o cérebro por meio do nervo óptico.

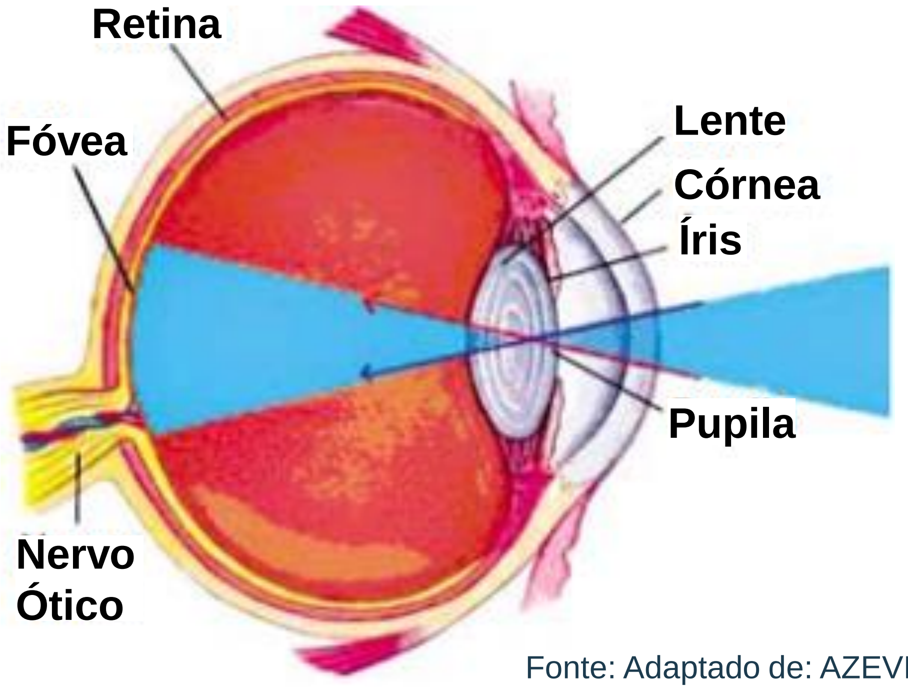


Fonte: Adaptado de: AZEVEDO (2003).

Síntese de cores

Sistema visual humano

- Os cones são responsáveis por detectar as cores, enquanto os bastonetes são responsáveis por detectar a luz em condições de baixa luminosidade. Existem três tipos de cones no olho humano, que são sensíveis às cores vermelho, verde e azul, respectivamente. A combinação de sinais dos cones permite que o cérebro interprete uma ampla gama de cores. O olho humano funciona como uma lente que capta luz por meio da córnea e a transforma em sinais elétricos, que são interpretados pelo cérebro.

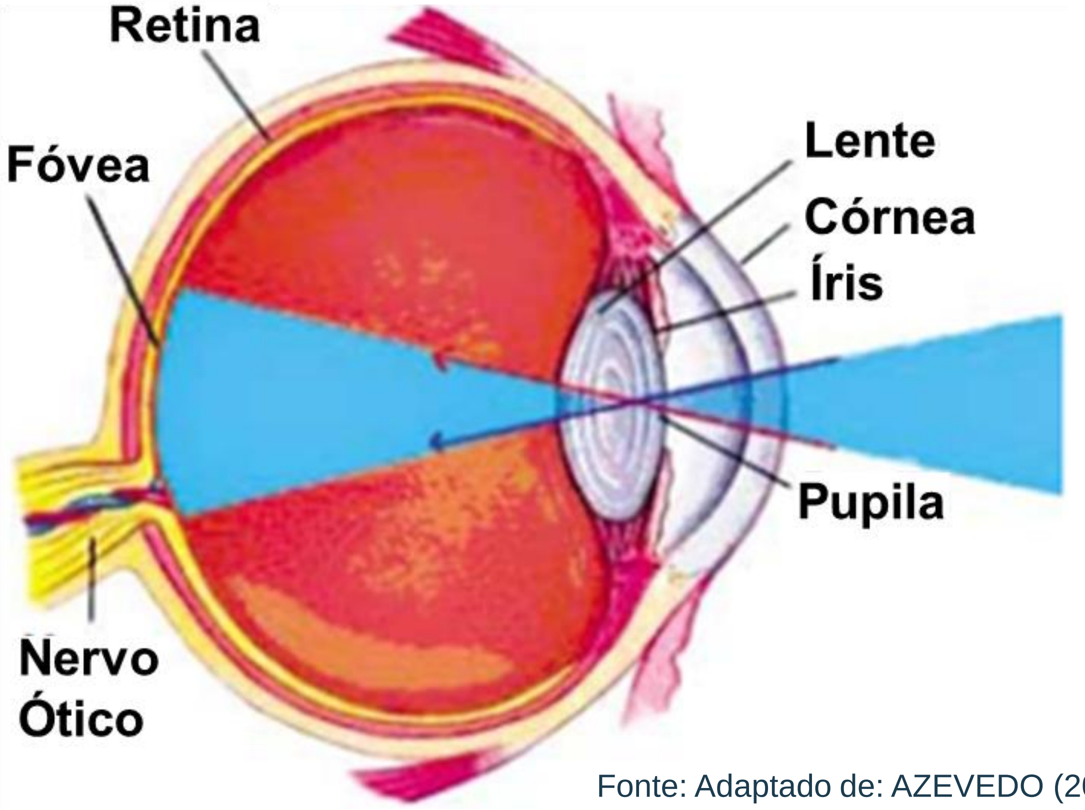


Fonte: Adaptado de: AZEVEDO (2003).

Síntese de cores

Sistema visual humano

- Os cones são responsáveis por detectar as cores. Existem três tipos de cones no olho humano, que são sensíveis às cores vermelho, verde e azul, respectivamente.
- Os bastonetes são responsáveis por detectar a luz em condições de alta ou baixa luminosidade e também o contraste.



Fonte: Adaptado de: AZEVEDO (2003).

Síntese de cores

Aspectos técnicos sobre cor e luz

- No modelo RGB, as cores são representadas por um conjunto de três valores (canais) que indicam as intensidades das componentes de luz vermelha, verde e azul em um determinado *pixel* de uma imagem.

- Para criar cores adicionais, é possível combinar as intensidades das três cores primárias (vermelho, verde e azul) em diferentes proporções. Por exemplo, a cor amarela pode ser criada com uma combinação de intensidades de vermelho e verde, enquanto a cor roxa pode ser criada com uma combinação de intensidades de vermelho e azul.

Síntese de cores

Aspectos técnicos sobre cor e luz

- Outro modelo matemático comumente utilizado em computação gráfica é o modelo de cores HSL (Hue, Saturation, Lightness), que separa as cores em três componentes diferentes: matiz, saturação e luminosidade. A matiz (hue) é a característica que define a cor em si, enquanto a saturação indica a pureza da cor e a luminosidade (lightness) indica a quantidade de luz presente na cor.

Síntese de cores

Aspectos técnicos sobre cor e luz

- Em relação à representação da luz, um modelo matemático comum é o modelo de iluminação de Phong, que leva em consideração a luz ambiente, a luz difusa e a luz especular para simular a iluminação de objetos em uma cena 3D. Este modelo utiliza equações matemáticas para calcular as intensidades de luz que são refletidas pelos objetos com base em suas propriedades físicas, como reflexão e absorção de luz.

Síntese de cores

Profundidade de cor (*bits* por *pixel*)

- As imagens digitais são representadas na forma de matrizes de *pixels*, em que, a cada *pixel*, está associado a uma cor.

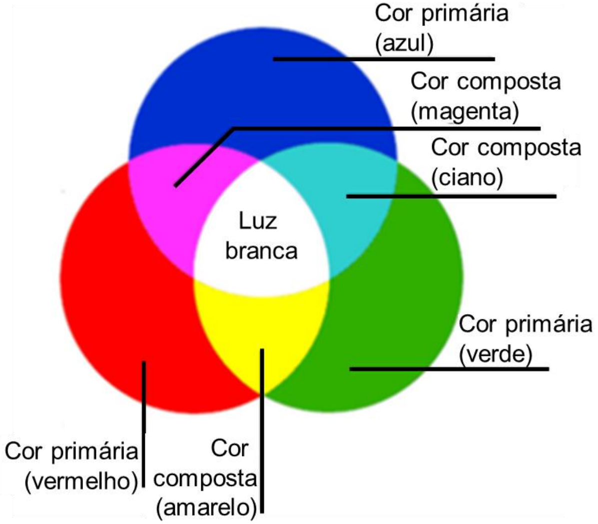
- Denomina-se profundidade de cor (color depth) a quantidade de *bits* necessários para se representar um certo conjunto de cores. A profundidade de cor é, normalmente, estabelecida em *bits* por *pixels* (bpp).

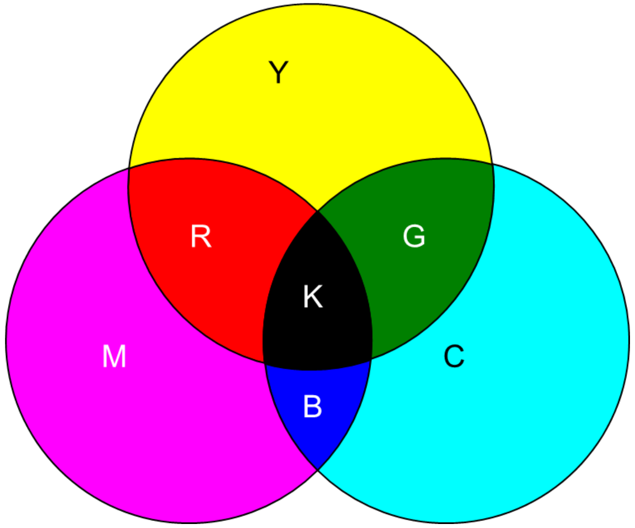
- Por exemplo, para representarmos uma imagem em branco e preto (duas cores), é necessário apenas um *bit* para cada *pixel* ($2^1 = 2$): *bit* = 0 => *pixel* apagado ou *bit* = 1 => *pixel* aceso. Assim, cada *byte* (8 *bits*) de memória pode armazenar 8 *pixels*. Considerando uma imagem de 320 x 200 *pixels* nesse sistema, vemos que a memória ocupada será de 320 x 200 x 1 = 64000 *bits* = 8000 *bytes* (8 *kbytes*?).

Síntese de cores

Profundidade de cor (*bits* por *pixel*)

- Por outro lado, com um único *byte*, podemos representar $2^8 = 256$ cores. Neste caso, as cores que serão representadas dependerão do *hardware* ou de uma paleta de cores armazenada com o arquivo. Imagens assim são denominadas de indexadas.
- Um outro sistema de estabelecer cores para cada *pixel* é o da síntese de cores a partir de um conjunto de cores primárias.
- Os dois sistemas de cores primárias mais empregados em CG são os sistemas ADITIVOS (RGB: R = Red, G = Green, B = Blue) e SUBTRATIVOS (CMY: C = Cyan, M = Magenta, Y = Yellow), cada um com 3 canais de cores primárias (tricromia).

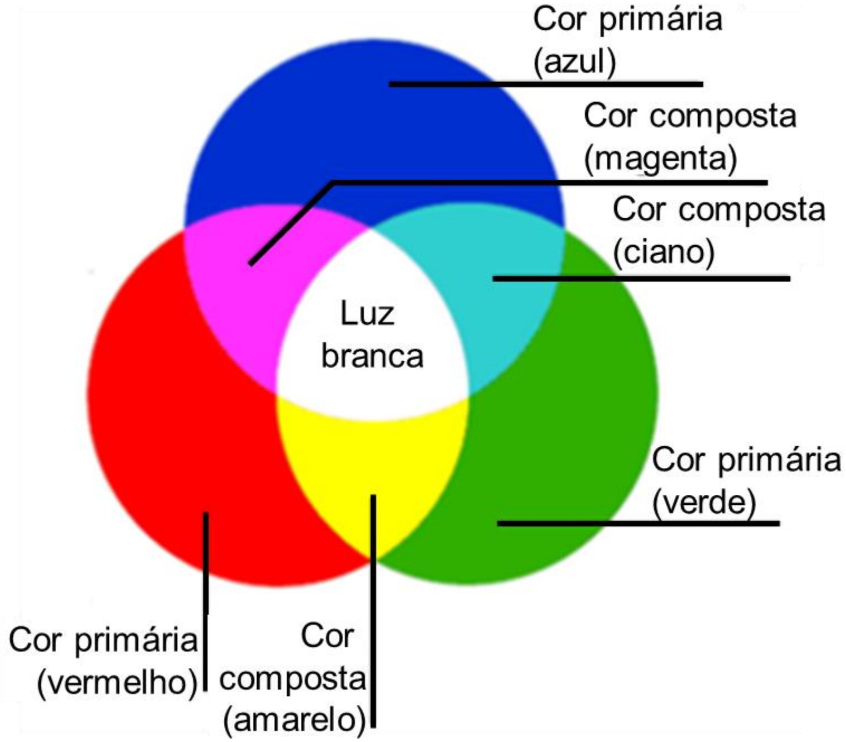


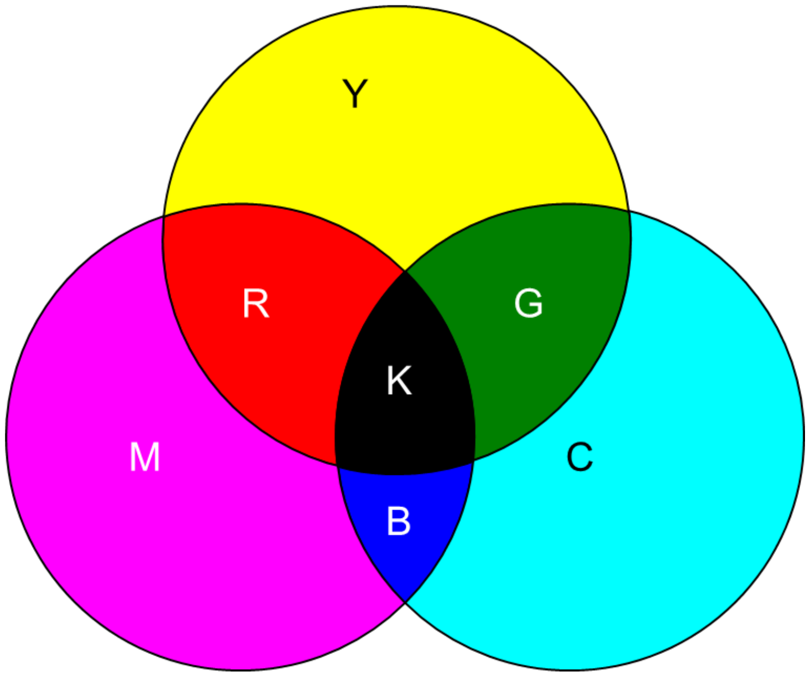


Síntese de cores

Profundidade de cor (*bits* por *pixel*)

- Aqui temos as profundidades de 8 bpp (grayscale), que emprega apenas um canal, totalizando 256 tons de cinzas, 16 bpp (highcolor), que permite a representação de 65536 cores por meio de uma síntese que utiliza 5 *bits* para o canal da cor vermelha (R), 6 *bits* para o da cor verde (G) e 5 *bits* para representar o azul (B) e o sistema de 24 *bits* por *pixel* (truecolor), que emprega um *byte* para cada canal, totalizando $2^8 \times 2^8 \times 2^8 = 2^{24} =$ 16.777.216 cores.





Síntese de cores

Formatos de arquivos de imagens

- Uma vez atribuída a propriedade cor a cada um dos *pixels* da matriz de *pixels* que forma uma imagem digital, esta é armazenada em formatos específicos, identificados por suas extensões. Por exemplo: BMP, JPG, TIF, PNG, PCX, PIX, DDS, FIT, GIF, PCD, TGA, RAW, RLE, WMF, CDR, PDF, PS, SVG, AVI...

- Cada formato possui uma utilização específica, seja ela própria de um aplicativo (formatos proprietários), como o PSD, CDR, DDS, de um ambiente operacional, como BMP, GIF, JPG, ou de uma mídia, como o PCD, AVI, MPG, WMV...

- Os formatos que são próprios de um certo ambiente acabam por assumir características de uso geral, e alguns tornam-se muito populares, como o GIF e o JPG, que se tornaram padrão na internet.

Síntese de cores

- Alguns armazenam dados de imagens na forma matricial (*pixels*), como o BMP, GIF, TIF, JPG, e outros na forma de vetores (vértices, arestas e faces), como o WMF, CDR e SVG e outros na forma de streamers (sequências de imagens), como o AVI, MPG, MOV etc.

- Outra característica muito importante a ser destacada é a forma de armazenagem, profundidade da informação (cores, luminosidade) em *bits*, por exemplo, 2, 8, 16, 24, 32, 48, 64, 98 *bits*..., e a compressão, com perda de informação (lossy) ou sem perda de informação (lossless).

- Por exemplo, arquivos BMP (bitmap: mapa de *bits*), em geral, são arquivos de imagens diretas, mapas de *bits* na memória, armazenadas sem qualquer perda de qualidade (resolução) de cores ou da distribuição espacial da imagem. Já os arquivos de tipo GIF e JPEG são formatos compactos obtidos a partir de transformações que reduzem a quantidade de informação em favor de um ARQUIVO menor.

Síntese de cores

- No caso dos arquivos GIF, as cores são INDEXADAS, representadas na forma de uma tabela de algumas poucas cores (PALETA), em que cada *pixel* guarda referência a uma cor da tabela. Trata-se de um arquivo compacto com perda da qualidade de cor.

- Os arquivos JPEG, por outro lado, preservam a profundidade de cor de uma certa maneira, mas seu algoritmo de compressão por blocos, usando transformadas discretas de cossenos (DCT: Discrete Cosine Transform), substitui a representação por *pixels* por uma de coeficientes de uma série matemática truncada. Esse truncamento produz uma distorção na imagem que a faz perder qualidade espacial.

Síntese de cores

- Apesar de perda de qualidade, tanto GIF e JPEG são preferíveis para uso na internet por conta do tamanho reduzido de seu arquivos, que economiza em transmissão e armazenamento em disco. Entretanto, ao representarmos as imagens que armazenam no dispositivo gráfico de saída, elas são decodificadas (decode) para o formato de mapa de memória, ou Bitmap, cujo tamanho depende da resolução espacial da matriz de *pixel* e da profundidade de *pixel*. Isso também vale para o caso de o arquivo de imagem ser vetorial, já que a saída gráfica mais comum é feita em um dispositivo raster (monitor, impressora matricial...).

Síntese de cores

- Como dissemos anteriormente, a imagem digital colorida é codificada em um certo número de canais (cores primárias ou filtros) de acordo com especificações próprias de cada modelo. Os modelos de cores mais comuns são: RGB, CMY, XYZ, HSB, OSA, HVC, HSV, HSL, YIQ, OPP... Mas existem variações desse modelos (LRGB e CMYK) e outros modelos com aplicações específicas (modelos de cores falsas).

- Basicamente, os modelos dividem-se em dois tipos sistemas: ADITIVOS e SUBTRATIVOS.

Síntese de cores

- O principal sistema aditivo é o modelo RGB, usado nos monitores de vídeo, cujas cores primárias são o VERMELHO (R), VERDE (G) e o AZUL (B). Como já visto, este é o mesmo modelo usado pelo olho humano para enviar sinais ao cérebro, que sintetizará a imagem cromática por meio das células da retina denominadas cones (cones R, G e B). Nos sistemas digitais, ao contrário do olho, que é analógico, cada canal (R, G e B) possui 256 níveis de intensidade (8 *bits*). Na verdade, o olho humano apresenta mais um canal, que podemos codificar como L (LUMINÂNCIA), que é processado por células da retina denominadas bastonetes.

- A LUMINÂNCIA, por sua vez, também é um canal de um modelo de síntese de cores denominado HLS (Hue, Lightness, Saturation), ou seja, matiz, luminância e saturação.

Síntese de cores

- Já o sistema subtrativo mais popular é o CMY, usado nas impressoras, e se baseia nas cores primárias que FILTRAM certos matizes da luz branca, refletindo as cores CIANO (C), MAGENTA (M) e AMARELA (Y).

- Não devemos confundir as cores primárias subtrativas do modelo CMY com aquele que aprendemos nas aulas de Arte e Desenho do Ensino Fundamental e Médio, o modelo de cores primárias vermelho, amarelo e azul, simplificado do CMY (note que o ciano não é azul e nem o magenta é vermelho!)

- O universo de cores que um dado modelo é capaz de representar é denominado ESPAÇO DE CORES, e não são completamente equivalentes. Ou seja, não é possível representar o espaço de cores do RGB no modelo CMY, mas o inverso sim, pois o CMY é um subespaço de RGB.

Síntese de cores

- As especificações técnicas do processamento de cores é um dos assuntos mais complexos em CG e será visto com maiores detalhes na disciplina de Processamento de Imagens. Por agora, usaremos apenas os modelos RGB e CMY.

A síntese de cores por esses modelos teóricos pode ser obtida com expressões do tipo:

Cor = r.R + g.G + b.B, em que r, g e b são, respectivamente, as quantidades (percentuais) de R, G e B. Considerando valores integrais (0 ou 1), temos:

- 1.R + 1.G +1.B = Branca
- 0.R + 0.G + 0.B = Preta

Síntese de cores

- Veja na tabela abaixo as possíveis combinações (teóricas) que transformam um modelo em outro...

Cor	R	G	B	= >	C	M	Y
PRETO	0	0	0	= >	1	1	1
VERMELHO	1	0	0	= >	0	1	1
VERDE	0	1	0	= >	1	0	1
AZUL	0	0	1	= >	1	1	0
AMARELO	1	1	0	= >	0	0	1
MAGENTA	1	0	1	= >	0	1	0
CIANO	0	1	1	= >	1	0	0
BRANCO	1	1	1	= >	0	0	0

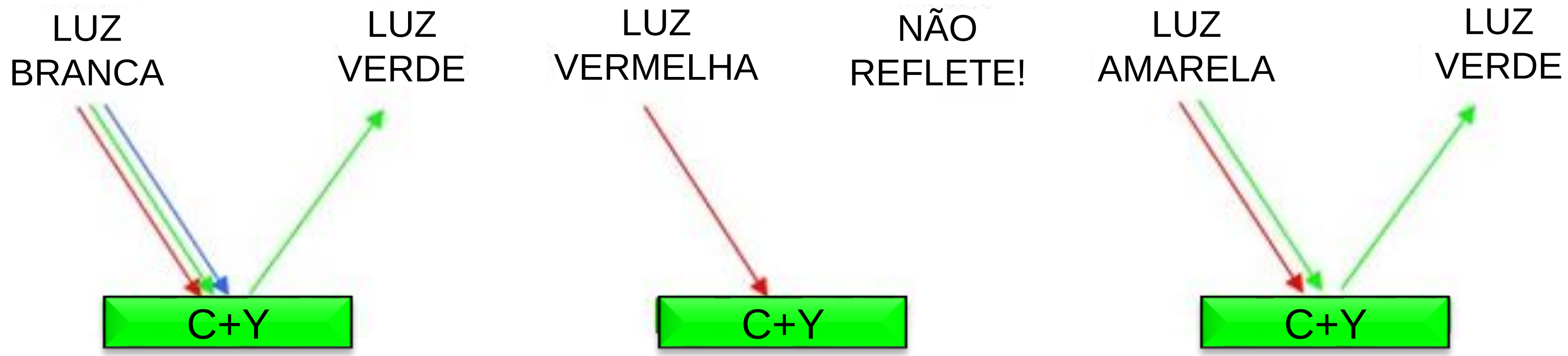
Síntese de cores

Um exercício fundamental é descobrir qual é a cor da LUZ REFLETIDA por um determinado objeto quando iluminado por uma LUZ INCIDENTE de dada cor e demais combinações (e.g. Qual é a cor do objeto? E qual é a cor da luz incidente?). O raciocínio simplista é:

- LUZ INCIDENTE (modelo aditivo).
- OBJETO PIGMENTADO (modelo subtrativo).
- LUZ REFLETIDA (modelo aditivo).

Síntese de cores

- Um objeto para ser VERDE, por exemplo, deve ser pintado com a mistura dos pigmentos CIANO e AMARELO (G = C + Y). Enquanto que o pigmento ciano absorverá o vermelho da luz incidente, o amarelo absorverá a componente azul, restando à luz refletida o canal verde! Só que, quando iluminado por uma luz, digamos, de cor vermelha, azul ou magenta, sua coloração aparecerá preta, pois não refletirá cor alguma (lembre-se de que o magenta absorve o verde e que as cores primárias vermelha e azul não possuem o verde). Por outro lado, se a luz for verde, amarela, ciano, a cor aparente será verde.



Lembrando:
No modelo subtrativo CMY:

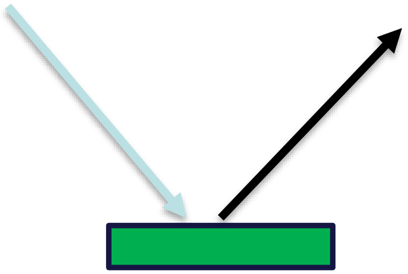
- **C**iano absorve a componente VERMELHA(R) da luz BRANCA
- **M**agenta absorve a componente VERDE(G) da luz branca
- **Y**ellow absorve a componente AZUL(B)

Síntese de cores

Um objeto de cor VERDE iluminado por uma luz AZUL refletirá a cor:

(a) Azul. (b) Vermelha. (c) Amarela. (d) Preta. (e) Branca.

Resolução



VERDE	0	1	0
AZUL	0	0	1

Verde = C + Y

Luz incidente $*$ Cor do objeto = Luz refletida

Montando uma matriz com a tabela de cores.

$$\begin{pmatrix}0\\0\\1\end{pmatrix} * \begin{pmatrix}0\\1\\0\end{pmatrix} = \begin{pmatrix}0*0\\0*1\\1*0\end{pmatrix} = \begin{pmatrix}0\\0\\0\end{pmatrix} \Rightarrow preto$$

Síntese de cores

Qual é o número de *bits* que permite, num terminal de vídeo colorido, representar uma opção de 1024 cores:

Solução:

- 2^{n^o} de *bits* = quantidade de cores
- $2^x = 1024$
- $2^x = 2^{10}$
- x = 10 *bits*

Síntese de cores

Qual é a quantidade de memória necessária (em *bytes*) para representar uma imagem com 65.536 cores por *pixel* (High Color) em resolução de 1280 x 1024 (SXGA)?

Solução:

- O sistema de 65.536 cores usa 16 *bits* por *pixel* (2^{16} = 65.536 cores) e sabendo que 16 *bits* = 2 *bytes*, então esse sistema usa 2 *bytes* por *pixel*. Considerando que o dispositivo de saída tem 1280 x 1024 = 1.310.720 *pixels* de resolução, então:

1.310.720 x 2 = 2.621.440 *bytes*

Síntese de cores

Highcolor, Truecolor, ... são padrões de diferentes profundidades de cores (*bits*/*pixel*). O padrão highcolor é capaz de representar 65536 cores enquanto que o truecolor representa quase 17 milhões de cores (16.777.216 cores), embora seja mais comum referi-lo como um padrão de 16 milhões de cores. O olho humano é capaz de distinguir até 10 milhões de cores. Qual é a profundidade de cor do padrão truecolor?

Solução:

- 2^x =16.77.216
- $2^x = 2^{24}$
- x = 24 *bits*

Síntese de cores

Um operador de luz de teatro teve um de seus três holofotes de luz primária (RGB) danificado (queimado). Ao iluminar um fundo de cor branca, ele só conseguia fazê-lo refletir duas cores primárias e o ciano. De que cor era a luz do holofote danificado?

Lembrando:

- Luz incidente (modelo aditivo).
- Objeto pigmentado (modelo subtrativo).
- Luz refletida (modelo aditivo).

Solução:

- Já que um dos holofotes está queimado, o funcionário só conseguirá refletir 2 das 3 cores primárias do RGB.

Considerando a mistura de cores:

- R + G = Y
- R + B = M
- G + B = C
- Como o funcionário refletiu o ciano (G + B), logo o holofote queimado é da cor vermelha (R).

Interatividade

A quantidade de memória necessária (em *megabytes*) para representar uma imagem com 16.777.216 cores por *pixel* (Truecolor) em resolução de 1280 x 1024 (SXGA) é de **APROXIMADAMENTE**:

a) 4.
b) 3.
c) 400.
d) 300.
e) 390.

Resposta

A quantidade de memória necessária (em *megabytes*) para representar uma imagem com 16.777.216 cores por *pixel* (Truecolor) em resolução de 1280 x 1024 (SXGA) é de **APROXIMADAMENTE**:

a) 4.
b) 3.
c) 400.
d) 300.
e) 390.

Transformações geométricas em 2D e 3D

- Quando se trata de gerar imagens em duas ou três dimensões, apenas a criação de primitivas e aplicação de atributos não é suficiente. É absolutamente necessário que sejam feitas certas transformações. Por um lado, as transformações de primitivas facilitam a geração de imagens porque os objetos gráficos podem ser gerados (calculados) em sua forma mais simples, e depois alterados para a imagem final que se deseja. Por outro lado, a aplicação de transformações permite que se crie uma “linguagem comum” (como será visto a seguir) que unifica e generaliza as transformações.

Transformações geométricas em 2D e 3D

Na disciplina de Álgebra Linear aprendemos o que são transformações e como podemos distinguir as transformações lineares e não lineares por meio de operações algébricas de soma vetorial e produto de vetor por escalar. As transformações se aplicam igualmente a funções matrizes e vetores. Em Computação Gráfica, preocupamo-nos, sobretudo, com as transformações geométricas aplicadas a vetores e espaços vetoriais. As transformações geométricas básicas são:

1. Translação (T)
2. Reflexão (M)
3. Escala (S)
4. Rotação (R)
5. Cisalhamento (C)

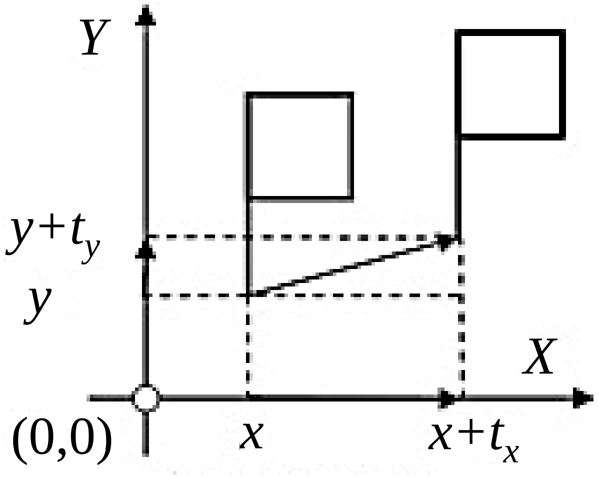
Transformações geométricas em 2D e 3D

- Dessas cinco transformações geométricas básicas, apenas a transformação de translação é não linear, todas as outras são lineares, e, adiante, veremos a implicação desse fato.

- Por ora, considere a imagem de uma bandeira cujos vértices V, de coordenada (x, y), transformam-se em V', de coordenadas (x', y'), como passamos a descrever em cada caso, com as respectivas matrizes de transformação.

Translação

- A translação é o ato de levar um objeto de um ponto para outro, num sistema de referência. O objetivo de uma translação é bem simples: ao calcularmos os pontos de um objeto, devemos escolher a posição mais simples para sua geração – por exemplo, para um círculo fazer coincidir o centro com a origem dos eixos – e depois transferir a figura para a posição final.



$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases} \rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

$$\downarrow$$

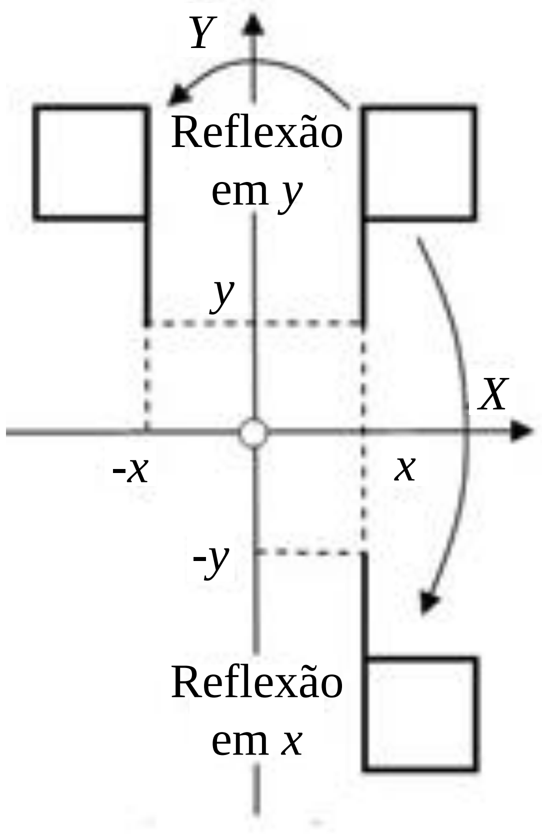
$V' = V + T(t_x, t_y) \leftarrow$ Equação matricial p/ a transformação

Fonte: Autoria própria.

Transformações geométricas em 2D e 3D

Reflexão

- A translação é o ato de “girar” um objeto em torno dos eixos ou da origem de um sistema de referência.



$$\begin{cases} x' = -x \\ y' = y \end{cases} \rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{(reflexão em y)}$$

$$\downarrow$$

$V' = M(-1,+1)V \leftarrow$ Equação matricial p/ a transformação

$$\begin{cases} x' = x \\ y' = -y \end{cases} \rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{(reflexão em x)}$$

$$\downarrow$$

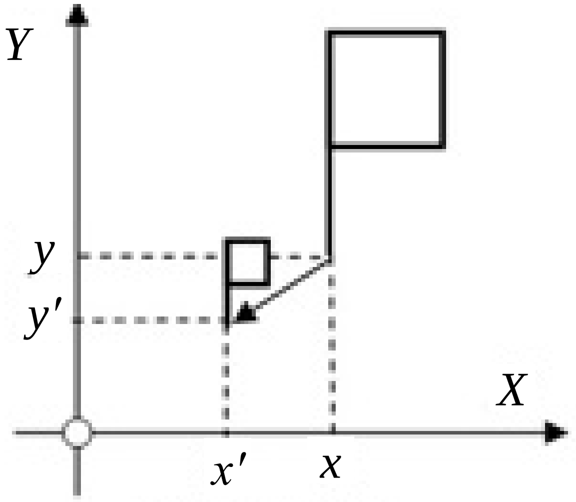
$V' = M(+1,-1)V \leftarrow$ Equação matricial p/ a transformação

Fonte: Autoria própria.

Transformações geométricas em 2D e 3D

Escala ou dilatação

- Quando se aplica uma transformação de escala a um objeto, o resultado é um novo objeto semelhante ao original, mas “esticado” ou “encolhido”.



$$\begin{cases} x' = s_x x \\ y' = s_y y \end{cases} \rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\downarrow$$

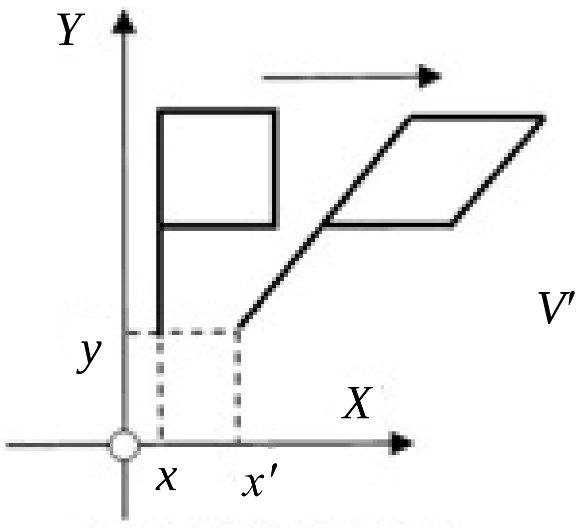
$V' = S(s_x, s_y)V \leftarrow$ Equação matricial p/ a transformação

Fonte: Autoria própria.

Transformações geométricas em 2D e 3D

Cisalhamento

- Quando se aplica uma transformação de cisalhamento a um objeto, o resultado é um novo objeto inclinado na direção do eixo x ou do eixo y.



$$\begin{cases} x' = x + c_x y \\ y' = c_y x + y \end{cases} \rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & c_x \\ c_y & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

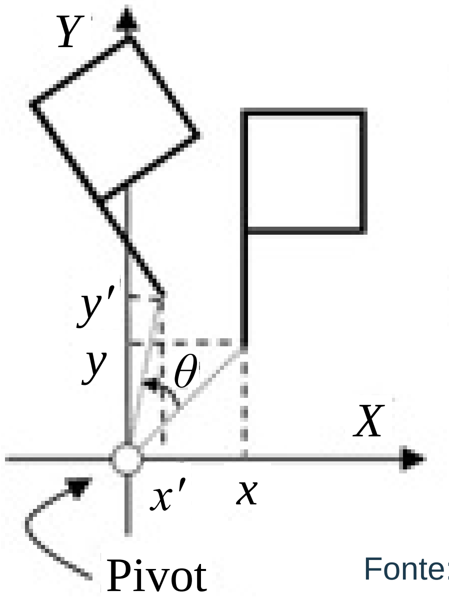
$$\downarrow$$

$V' = C(c_x, c_y)V \leftarrow$ Equação matricial p/ a transformação

Fonte: Autoria própria.

Rotação

- É o ato de girar um objeto de um ângulo, num sistema de referência. Como no caso da translação, objetivo de uma rotação é bem simples: ao modelar-se os pontos de um objeto, deve-se escolher o ângulo mais simples para sua geração, e depois girar a figura para a posição final. Um rotação em duas dimensões é aplicada pelo reposicionamento de cada ponto original (calculado) do objeto ao longo de um círculo imaginário no plano xy. Logo, deve ser especificado um ângulo de rotação θ, que especifica o arco que um dado ponto deve ser deslocado em torno da origem do sistema de referência (0, 0).



$$\begin{cases} x' = x\cos\theta - y\sin\theta \\ y' = x\sin\theta + y\cos\theta \end{cases} \rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\downarrow$$

$$V' = R(\theta)V \leftarrow \text{Equação matricial p/ a transformação}$$

Fonte: Autoria própria.

Interatividade

Assinale a alternativa que contém os valores obtidos a partir da transformação de escala do ponto (3,2), com os fatores de escala $E_x = -2$ e $E_y = 3$.

a) (5,4)
b) (–6,6)
c) (6,–6)
d) (1,5)
e) (0,7)

Resposta

Assinale a alternativa que contém os valores obtidos a partir da transformação de escala do ponto (3,2), com os fatores de escala Ex = –2 e Ey = 3.

a) (5,4)
b) (–6,6)
c) (6,–6)
d) (1,5)
e) (0,7)

Concatenação de transformações

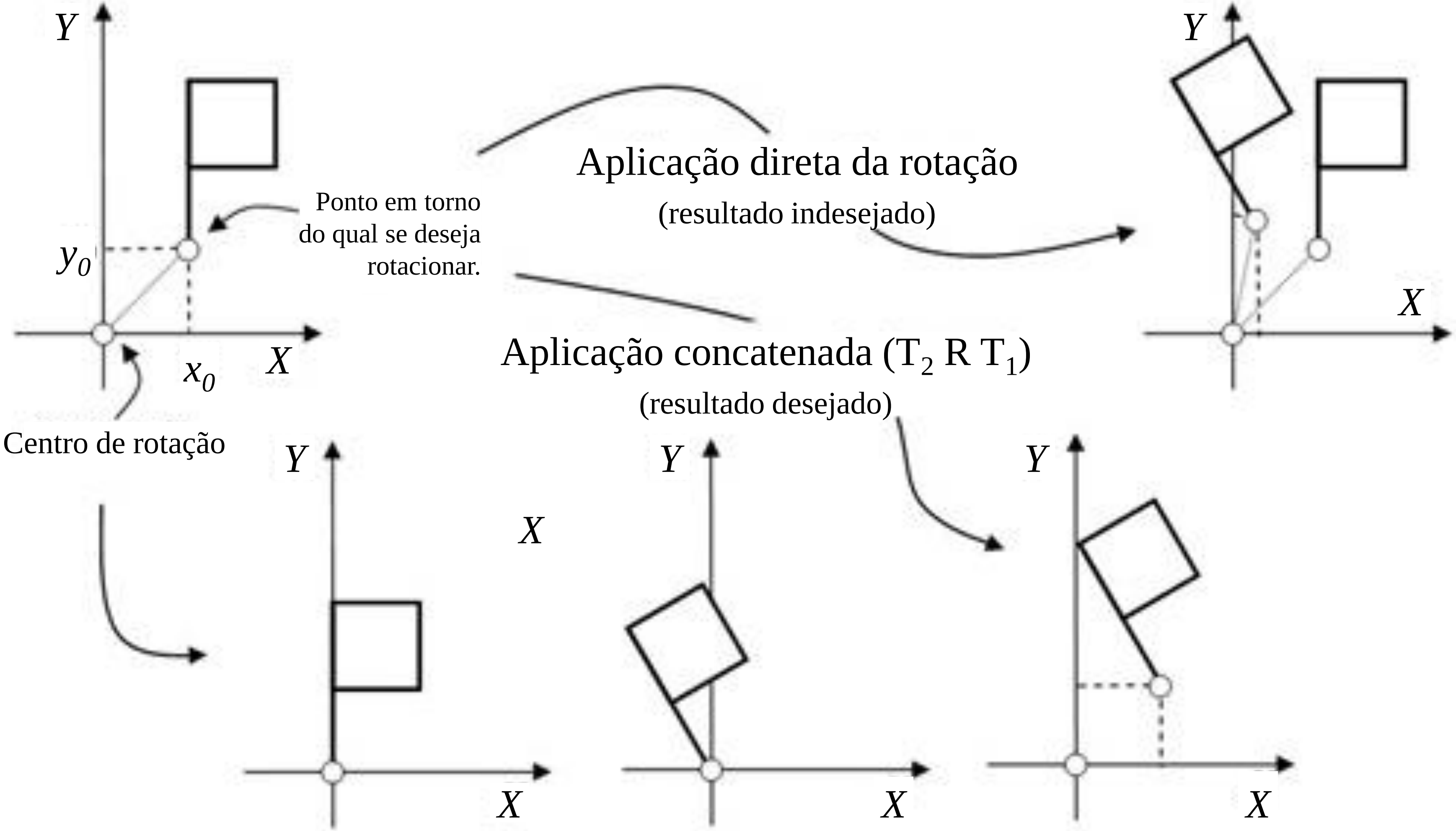
- As transformações apresentadas anteriormente são elementares, isto é, são aplicadas independentemente umas das outras. Entretanto, no caso geral, as transformações devem ser consideradas como sendo aplicadas em conjunto. Por exemplo, quando se deseja rotacionar um objeto deslocado da origem em torno de um ponto arbitrário (x_0,y_0), que chamaremos de PIVOT, se aplicarmos as expressões apresentadas, o mesmo sofrerá um aparente deslocamento ao longo do círculo imaginário de rotação centrado na origem. A sequência que deve ser aplicada é 1) deslocar a figura de modo que o ponto PIVOT coincida com a origem das coordenadas; 2) aplicar a rotação no objeto deslocado; e 3) transladar a figura de volta para o local original.

Concatenação de transformações

- Concatenar transformações requer estratégia (lógica) e uma boa dose de visão espacial para se obter o resultado desejado. A sequência correta de transformações não é única e uma má escolha pode diminuir a eficiência do processo, tornando-o mais lento ou impreciso.

- Uma vez decidida qual é a sequência adequada, devemos transformar os vértices do modelo, armazenados no vetor (ou matriz de vetores) V, em vértices transformados V'. Isso pode se feito de duas formas: 1) aplicando as transformações elementares uma a uma, o que é pouco eficiente, pois aumenta muito a quantidade de cálculos, diminuindo a precisão e aumentando o tempo de execução, ou 2) concatenando as matrizes das transformações elementares em uma única matriz de transformação, o que parece mais complicado, mas é muito mais eficiente.

Concatenação de transformações

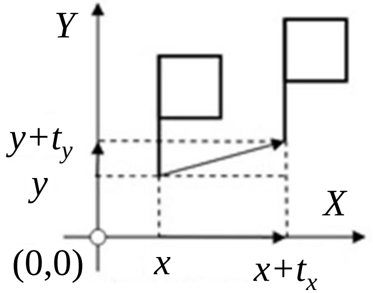


Fonte: Autoria própria.

Concatenação de transformações

Coordenadas homogêneas

Considere a transformação de translação mostrada nos slides anteriores:



$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases} \rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

$$\downarrow$$

$V' = V + T(t_x,t_y) \leftarrow$ Equação matricial p/ a transformação

Fonte: Autoria própria.

- Introduzindo um coordenada extra, que chamaremos de z, podemos escrever a equação matricial da transformação da seguinte maneira:

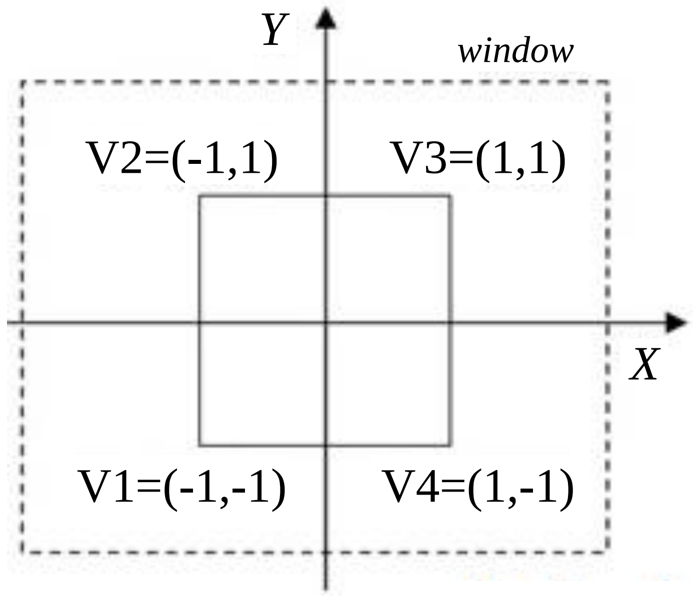
$$\begin{cases} x' = x + t_x \\ y' = y + t_y \\ z' = z \end{cases} \rightarrow \begin{cases} x' = 1.x + 0.y + t_x.z \\ y' = 0.x + 1.y + t_y.z \\ z' = 0.x + 0.y + 1.z \end{cases} \rightarrow \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

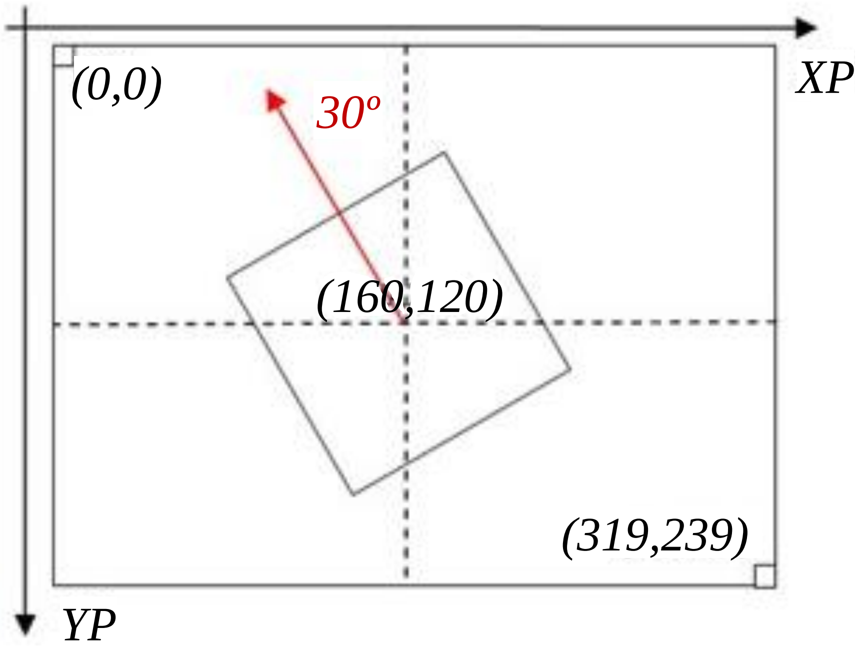
- Desta maneira, podemos transformar de uma vez por todas os vértices usando a equação matricial em coordenadas homogêneas.

Concatenação de transformações

Coordenadas homogêneas

- Sabemos como desenhar uma figura inscrita em uma janela (*window*) do SRU em uma janela de visualização (*viewport*) do dispositivo gráfico de saída (SRD). Para tanto, usamos as relações de proporcionalidade, ou seja, uma regra de três simples. No exemplo a seguir faremos o mesmo trabalho usando os conceitos desta aula. Vamos desenhar um quadrado de lado 2, centrado na origem do SRU (ou SRO), rotacionado de 30° centrado no meio da tela de um dispositivo de 320 x 240 pixels.





Fonte: Autoria própria.

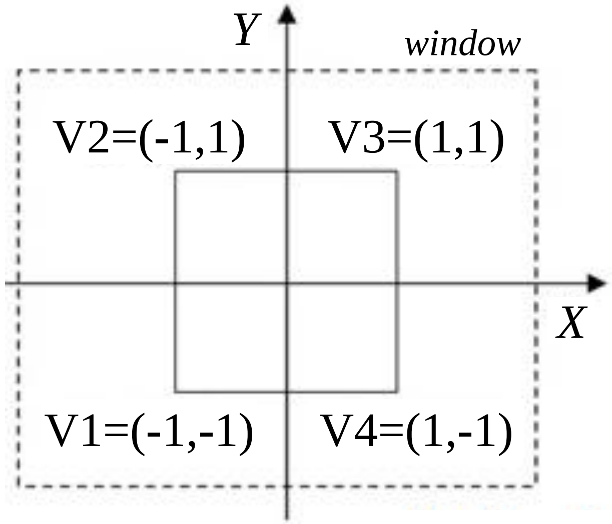
Concatenação de transformações

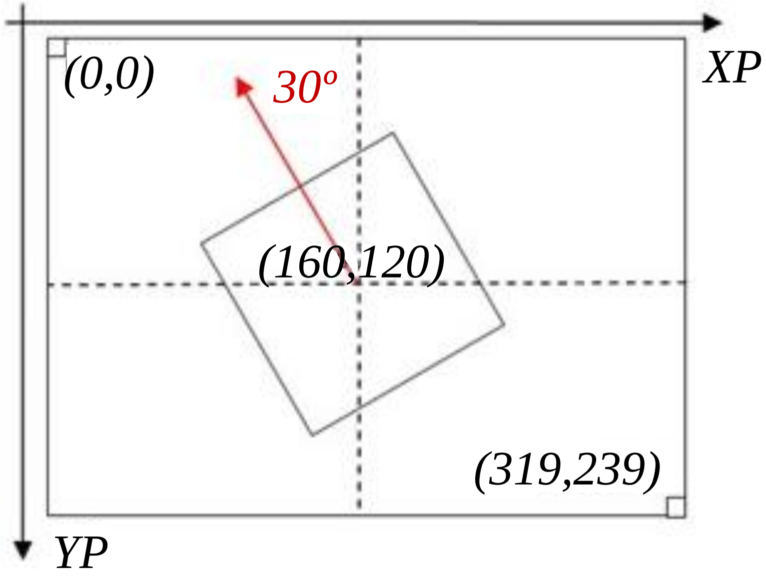
Coordenadas homogêneas

- Os vértices do quadrado no SRU (ou SRO) possuem coordenadas: V = {V1, V2, V3, V4 } e as arestas A = {(V1,V2), (V2,V3), (V3,V4), (V4,V1)}

Podemos escrever as matrizes de vértices e arestas como:

$$\boldsymbol{V} = \begin{pmatrix} -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \qquad \boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$$





Fonte: Autoria própria.

Concatenação de transformações

Coordenadas homogêneas

- As etapas da transformação são as seguintes: Rotação, Escala, Reflexão e Translação. Estas transformações deverão ser concatenadas e aplicadas a cada um dos vértices.
- Rotação: Primeiro iremos girar o objeto conforme desejamos usando a matriz homogênea de rotação em 2D:

$$R(30^o) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos 30^o & -\sin 30^o & 0 \\ \sin 30^o & \cos 30^o & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0{,}86603 & -0{,}5 & 0 \\ 0{,}5 & 0{,}86603 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Concatenação de transformações

Coordenadas homogêneas

- Escala: A escala depende das dimensões das janelas de visualização no espaço real e no espaço do dispositivo gráfico. Vamos considerar que o quadrado esteja limitado por uma *window* de dimensão 4 x 4, ou seja de (–2, –2) a (2, 2). A janela de visualização do dispositivo gráfico será a janela toda, ou seja, 320 x 240. Dessa forma, para conter uma janela de dimensões 4 x 4 em outra de dimensões 320 x 240, os fatores de escala em X e Y serão (a rigor sx = 319/4 = 79.75 sy = 239/4 = 59.75):
- Fator de Escala em X: 320/4 = 80
- Fator de Escala em Y: 240/4 = 60

Assim, a matriz que representa o escalonamento do objeto dados será:

$$S(80.60) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 80 & 0 & 0 \\ 0 & 60 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Concatenação de transformações

Coordenadas homogêneas

Reflexão: Como as ordenadas reais do SRU são crescentes de baixo para cima e as coordenadas do dispositivo gráfico, SRD, são crescentes de cima para baixo, devemos refletir verticalmente o objeto. Ou seja, a matriz que representa a reflexão em relação ao eixo X é:

$$\boldsymbol{M(1,-1)} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

Concatenação de transformações

Coordenadas homogêneas

- Translação: Como a origem do dispositivo gráfico encontra-se no canto superior esquerdo, precisamos deslocar todo o objeto para o centro da janela de visualização, como queremos, cujas coordenadas são (160, 120). A matriz de transformação de translação em coordenadas homogêneas para este exemplo é:

$$\boldsymbol{T(160,120)} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{160} \\ \mathbf{0} & \mathbf{1} & \mathbf{120} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

Concatenação de transformações

Coordenadas homogêneas

- <u>Finalmente</u>: Cada um dos vértices V = { V1, V2, V3, V4 } da figura original deverá ser transformado em vértices VP = { VP1, VP2, VP3, VP4 } do dispositivo gráfico pela relação:

$$VP = T(160.120)M(1,\text{-}1)S(80,60)R(30°)\,V =$$

$$= \begin{pmatrix} 1 & 0 & 160 \\ 0 & 1 & 120 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 80 & 0 & 0 \\ 0 & 60 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0{,}86603 & -0{,}5 & 0 \\ 0{,}5 & 0{,}86603 & 0 \\ 0 & 0 & 1 \end{pmatrix} V =$$

$$= \begin{pmatrix} 69{,}282 & -40 & 160 \\ -30 & -51{,}962 & 120 \\ 0 & 0 & 1 \end{pmatrix} V$$

$$VP = \underbrace{\begin{pmatrix} 69{,}282 & -40 & 160 \\ -30 & -51{,}962 & 120 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Matriz homogênea de transformação}} \underbrace{\begin{pmatrix} -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix}}_{\text{Vértices em coordenadas homogêneas}} = \begin{pmatrix} 130{,}7 & 50{,}72 & 189{,}3 & 269{,}3 \\ 202{,}0 & 98{,}04 & 38{,}04 & 142{,}0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Concatenação de transformações

Coordenadas homogêneas

- Finalmente: Cada um dos vértices V = { V1, V2, V3, V4 } da figura original deverá ser transformado em vértices VP = { VP1, VP2, VP3, VP4 } do dispositivo gráfico pela relação:

$$VP = T(160.120)M(1,\text{-}1)S(80,60)R(30°)\,V =$$

$$= \begin{pmatrix} 1 & 0 & 160 \\ 0 & 1 & 120 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 80 & 0 & 0 \\ 0 & 60 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0{,}86603 & -0{,}5 & 0 \\ 0{,}5 & 0{,}86603 & 0 \\ 0 & 0 & 1 \end{pmatrix} V =$$

$$= \begin{pmatrix} 69{,}282 & -40 & 160 \\ -30 & -51{,}962 & 120 \\ 0 & 0 & 1 \end{pmatrix} V$$

$$VP = \underbrace{\begin{pmatrix} 69{,}282 & -40 & 160 \\ -30 & -51{,}962 & 120 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Matriz homogênea de transformação}} \underbrace{\begin{pmatrix} -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix}}_{\text{Vértices em coordenadas homogêneas}} = \begin{pmatrix} 130{,}7 & 50{,}72 & 189{,}3 & 269{,}3 \\ 202{,}0 & 98{,}04 & 38{,}04 & 142{,}0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Concatenação de transformações

Coordenadas homogêneas

Observe que a matriz VP deve conter valores inteiros, pois são valores e *pixels*, então:

$$VP = \begin{pmatrix} \mathbf{131} & \mathbf{51} & \mathbf{189} & \mathbf{269} \\ \mathbf{202} & \mathbf{98} & \mathbf{38} & \mathbf{142} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix}$$

- Assim, os vértices V foram transformados em VP.

Vértice	V	VP
1:	(-1,-1)	(131, 202)
2:	(-1, 1)	(51, 98)
3:	(1, 1)	(189, 38)
4:	(1, -1)	(269, 142)

Interatividade

São transformações geométricas em 2D:

a) Rotação, translação e precessão.
b) Rotação, revolução e convolução.
c) Rotação, precessão e mutação.
d) Rotação, translação e revolução.
e) Rotação, translação e escala.

Resposta

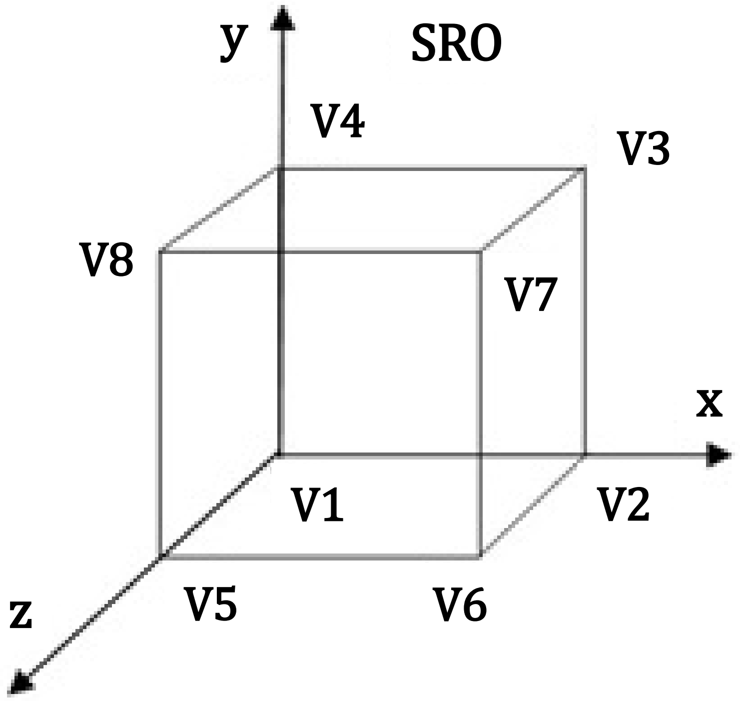
São transformações geométricas em 2D:

a) Rotação, translação e precessão.
b) Rotação, revolução e convolução.
c) Rotação, precessão e mutação.
d) Rotação, translação e revolução.
e) Rotação, translação e escala.

Representação e modelagem de objetos 2D e 3D

Um cubo de lado 1 com um dos vértices apoiado na origem do plano cartesiano será representado pela lista de vértices no SRO (ou no SRN) e aresta como na figura abaixo.

- Devemos saber representar os objetos, imagens, figuras, em termos das coordenadas de seus vértices e arestas.



Fonte: Autoria própria.

Representação e modelagem de objetos 2D e 3D

Neste caso podemos utilizar a notação de vetores (*arrays*). Um conjunto de vetores receberá a lista dos vértices armazenado em uma matriz 3 x N (ou 2 x N no caso de 2D), onde N é o número de vértices ordenados de 1 até N colunas e outro receberá a lista de arestas. Assim:

Vértices	V1	V2	V3	V4	V5	V6	V7	V8
Coordenadas	(0,0,0)	(1,0,0)	(1,1,0)	(0,1,0)	(0,0,1)	(1,0,1)	(1,1,1)	(0,1,1)

$$\boldsymbol{VÉRTICES_{CUBO}} = \begin{pmatrix} \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix}$$

- Já os vetores de arestas serão armazenados em uma matriz 2 x M, independentemente se o objeto é 2D ou 3D. As arestas são ordenadas em colunas de 1 até M. A primeira linha contém o índice do vértice inicial e a segunda o índice do vértice final.

Representação e modelagem de objetos 2D e 3D

- Já os vetores de arestas serão armazenados em uma matriz 2 x M, independentemente se o objeto é 2D ou 3D. As arestas são ordenadas em colunas de 1 até M. A primeira linha contém o índice do vértice inicial e a segunda o índice do vértice final.

Arestas	V1-V2	V2-V3	V3-V4	V4-V1	V1-V5	V2-V6	V3-V7	V4-V8	V5-V6	V6-V7	V7-V8	V8-V5

$$ARESTAS_{CUBO} = \begin{pmatrix} 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 1 & 5 & 6 & 7 & 8 & 6 & 7 & 8 & 5 \end{pmatrix}$$

Representação e modelagem de objetos 2D e 3D

- Quando desenhamos na tela do computador um objeto modelado por seus vértices V = {V1, V2, V3, ..., Vn}, devemos primeiro transformar as coordenadas de cada VÉRTICE conforme a necessidade da representação (isso implica em transformações geométricas descritas no módulo 7 e transformações dos vértices resultantes do SRO à SRU e à SRD e, em seguida, desenhar as ARESTAS ligando as coordenadas X e Y, em *pixel*, de cada um de seus vértices extremos.

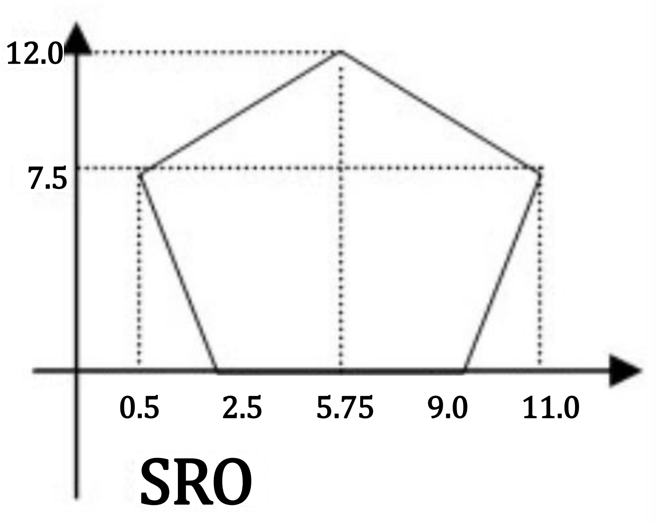
Representação e modelagem de objetos 2D e 3D

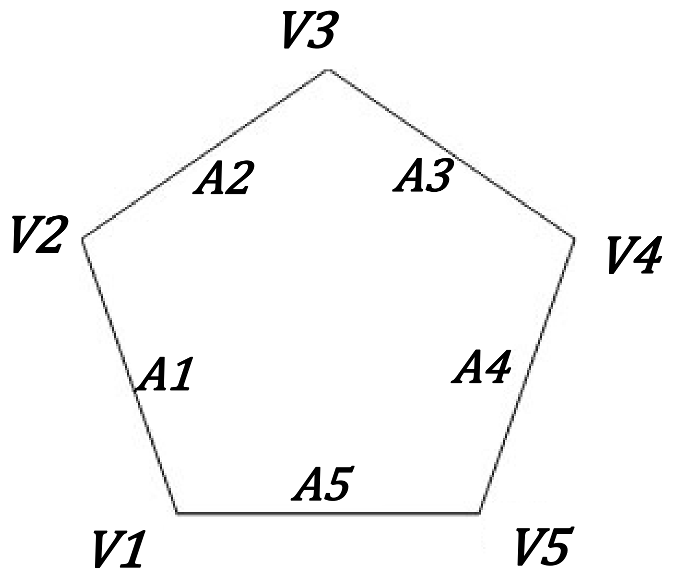
- Por exemplo: Considere a aresta A1 que liga os vértices V1 e V2. Após efetuarmos uma série de cálculos para posicionar o objeto adequadamente no SRU e passá-lo para o SRD. Para desenhar o objeto em uma determinada *viewport* do dispositivo gráfico de saída, tomamos apenas os valores transformado das coordenadas X*pixel* e Y*pixel* dos vértices V1 e V2 e os ligamos com um algoritmo de linhas (Bresenham).

- Nesta aula, contudo, iremos nos preocupar somente em como podemos criar uma estrutura de vértices e arestas para serem usadas oportunamente.

Representação e modelagem de objetos 2D e 3D

- Considere um pentágono, representado no SRO, ou no SRU, escreva as matrizes dos vértices e arestas para ele.





$$VÉRTICES = \begin{pmatrix} 2.5 & 0.5 & 5.75 & 11 & 9 \\ 0 & 7.5 & 12 & 7.5 & 0 \end{pmatrix}$$

$$ARESTAS = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \end{pmatrix}$$

Interatividade

Considere um quadrado Q de lado unitário, com os quatro vértices nomeados como A, B, C e D no sentido horário. Se as coordenadas (x, y) do vértice A são x = 0, y = 0, ou seja, (0, 0), a matriz de vértices resultante para este quadrado será:

a) $V_Q = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$

b) $V_Q = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$

c) $V_Q = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$

d) $V_Q = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$

e) $V_Q = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$

Resposta

Considere um quadrado Q de lado unitário, com os quatro vértices nomeados como A, B, C e D no sentido horário. Se as coordenadas (x, y) do vértice A são x = 0, y = 0, ou seja, (0, 0), a matriz de vértices resultante para este quadrado será:

a) $V_Q = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$

b) $V_Q = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$

c) $V_Q = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$

d) $V_Q = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$

e) $V_Q = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$

ATÉ A PRÓXIMA!