Computação Gráfica

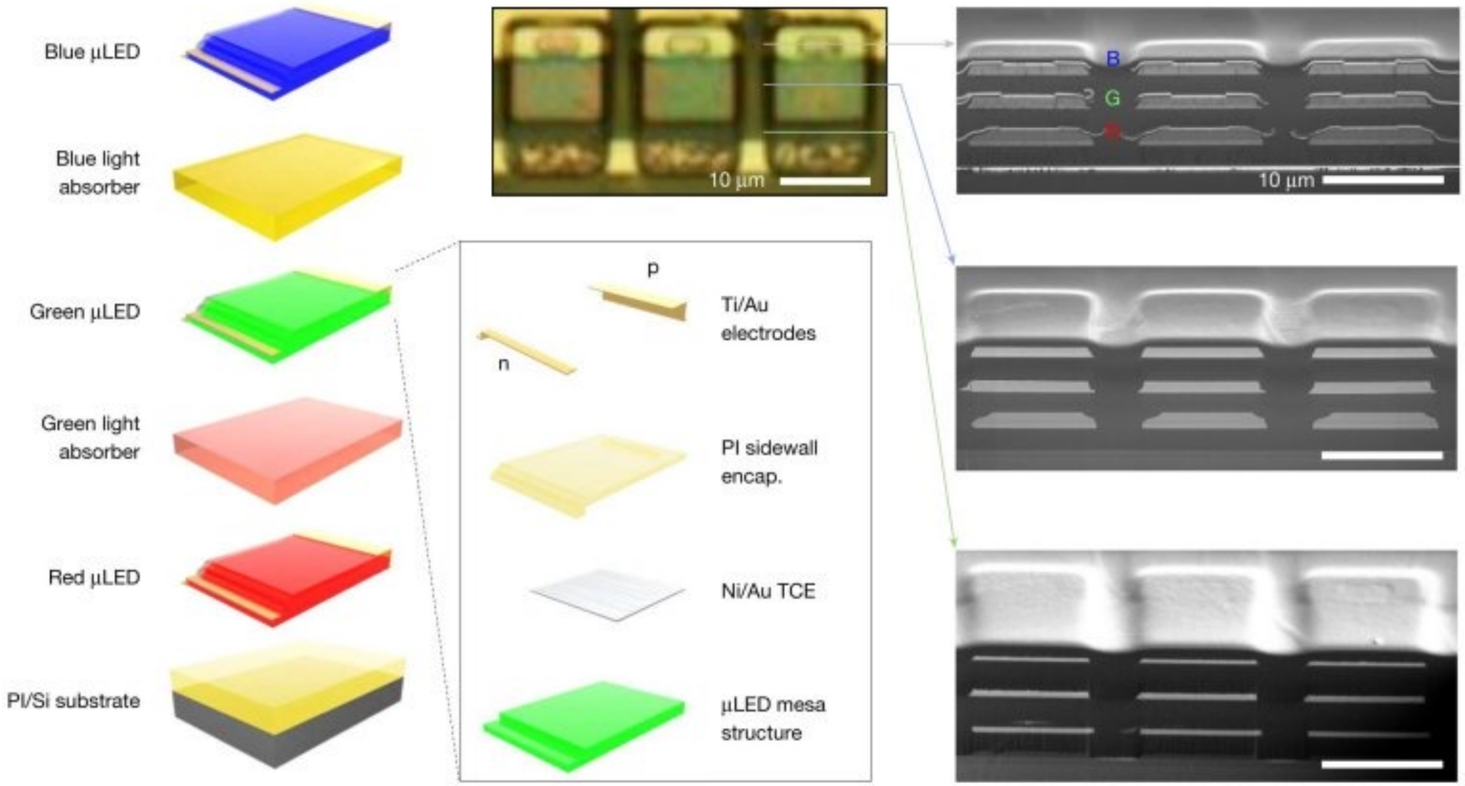
Aula 3: Renderização e Supersampling

Kahoot!

https://create.kahoot.it/

Novidade

Leds organizados na vertical.



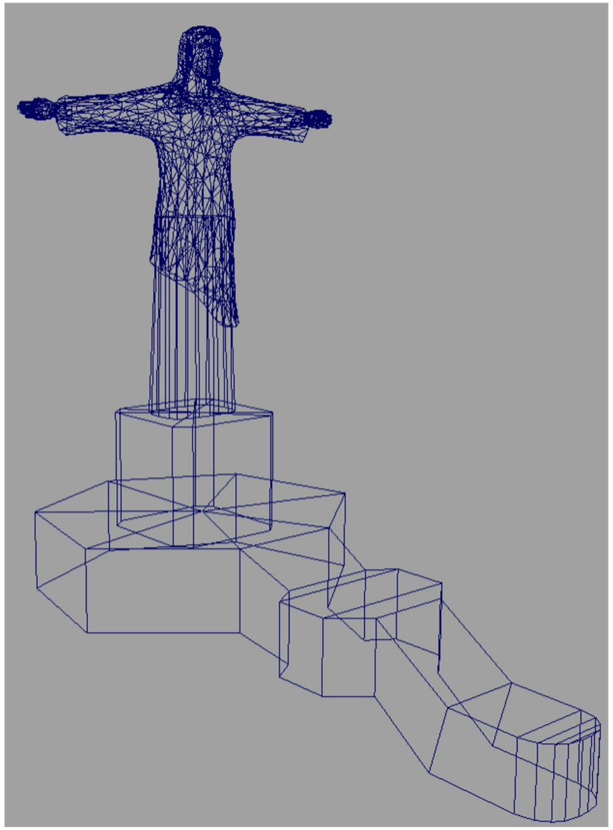
Referência: *Vertical full-colour micro-LEDs via 2D materials-based layer transfer;* Jiho Shin, Hyunseok Kim, Suresh Sundaram, Junseok Jeong, Bo-In Park, Celesta S. Chang, Joonghoon Choi, Taemin Kim, Mayuran Saravanapavanantham, Kuangye Lu, Sungkyu Kim, Jun Min Suh, Ki Seok Kim, Min-Kyu Song, Yunpeng Liu, Kuan Qiao, Jae Hwan Kim, Yeongin Kim, Ji-Hoon Kang, Jekyung Kim, Doeon Lee, Jaeyong Lee, Justin S. Kim, Han Eol Lee, Hanwool Yeon, Hyun S. Kum, Sang-Hoon Bae, Vladimir Bulovic, Ki Jun Yu, Kyusang Lee, Kwanghun Chung, Young Joon Hong, Abdallah Ougazzaden, Jeehwan Kim; Nature; Vol.: 614, pages 81-87; DOI: 10.1038/s41586-022-05612-1

Renderização

Processo de transformar os dados dos modelos 3D em imagens.

Insper

Na prática



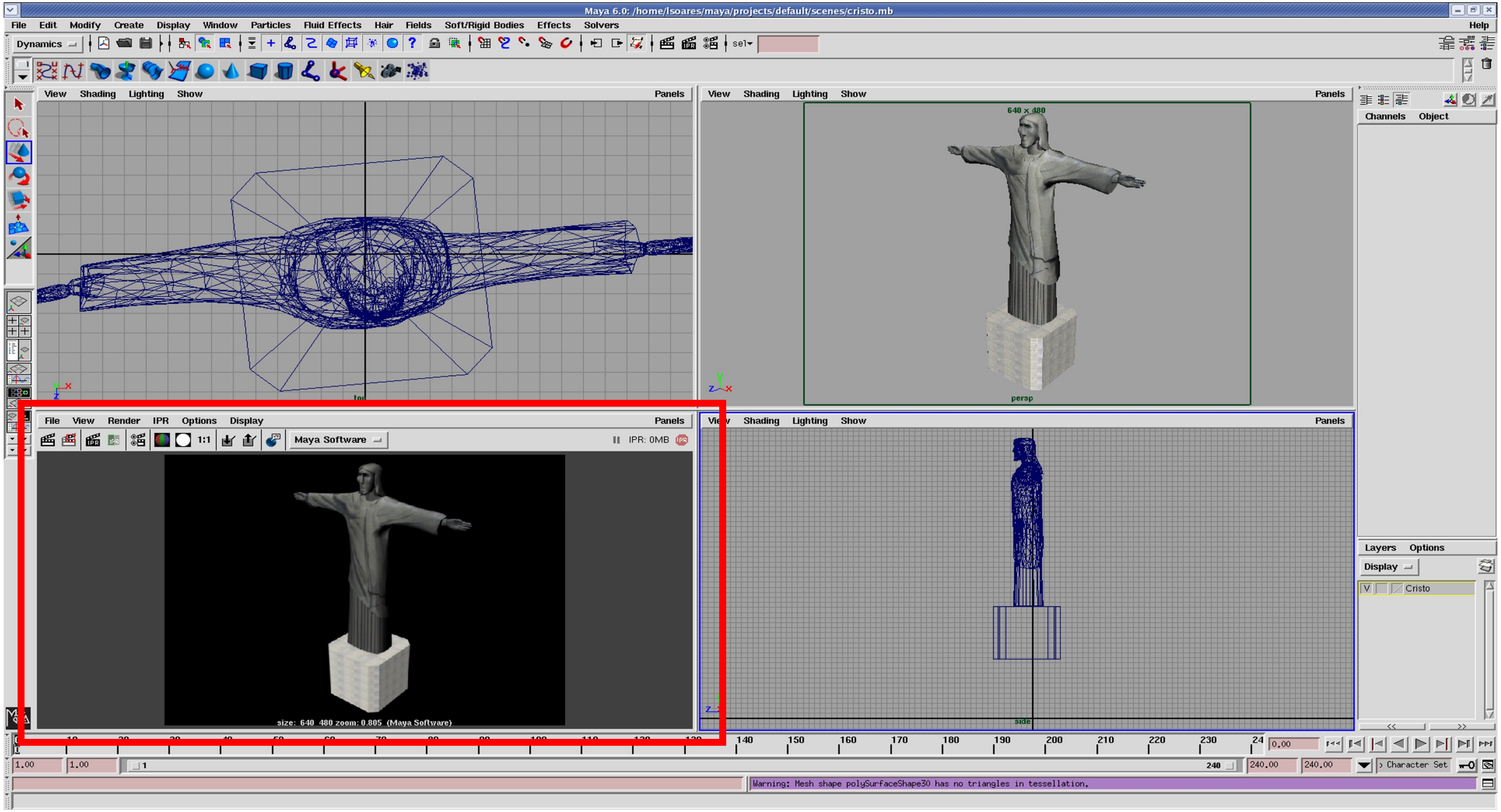
Várias estratégias podem ser usadas

Non-photorealistic

Photorealistic

Insper

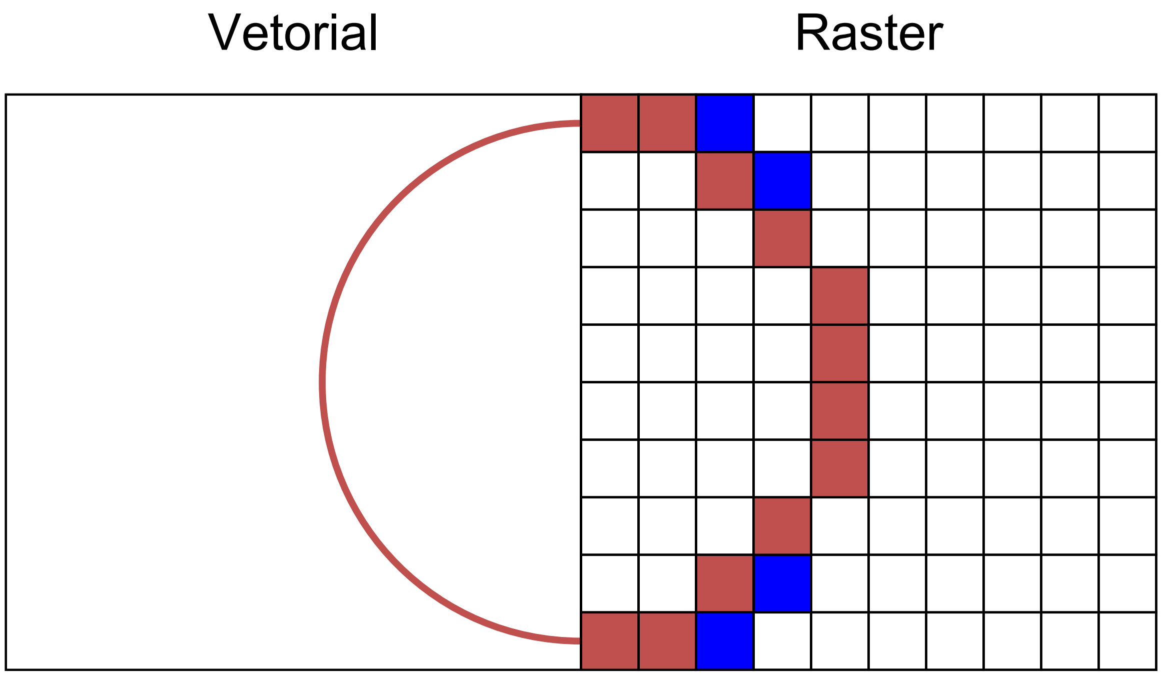
Rendering no Maya



Insper

Conceito de Imagem Raster x Vetorial

A imagem raster é baseada em pontos (pixels) distribuídos em duas dimensões (2D). Já imagens vetoriais são dados geométricos usados para desenhar uma imagem.



Técnicas de Renderização

- Scanline (Rasterização)
- Ray Casting
- Ray Tracing
- Photon Mapping
- Radiosity
- Monte Carlo
- Caustics
- A-Buffer (Maya)

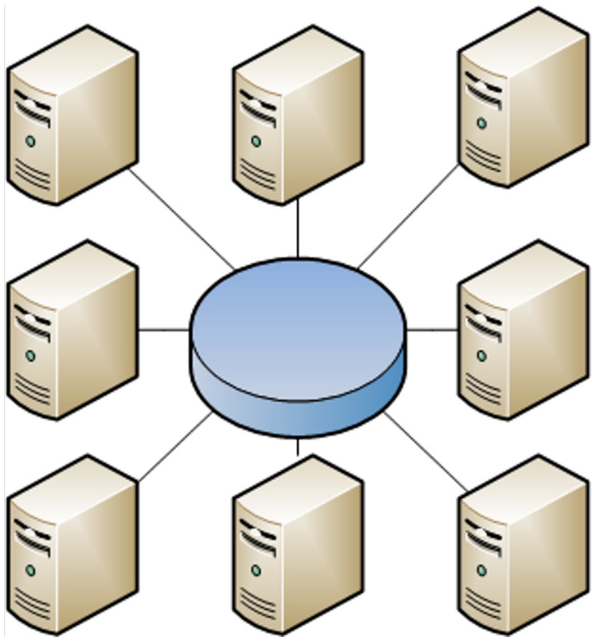
Sistemas de Renderização

Nome	Preço	Hardware	SO	Plugin
3Delight	Free/$720	CPU	Windows, macOS, Linux	3ds Max, Katana, Maya
Arnold	$360/year	CPU	Windows, macOS, Linux	Maya, Houdini, Cinema 4D, 3ds Max, Katana and Softimage
Artlantis	$705	CPU	Windows, macOS	Standalone
Blender	Free	CPU/GPU	Windows, macOS, Linux	Amarath, Ragdoll Tools, Magic UV, Ragdoll Tools
Clarisse	Free/$999	CPU/GPU	Windows, macOS, Linux	Standalone
Corona	$30/month	CPU	Windows	3ds Max, Cinema4D
Enscape	$469/year	GPU	Windows	Revit, SketchUp, Rhino, ArchiCAD
FelixRender	$160/month	N/A	Windows	3ds Max, AutoCAD, Rhinoceros
FluidRay	$14.99/month	GPU	Windows, macOS	Standalone / SketchUp, Modo, Shade 3D, Rhino
FurryBall	$110	GPU	Windows	Standalone / Maya, Cinema 4D, 3ds Max
Guerilla Render	Free/$2,000	CPU	Windows, Linux	Maya
Indigo Render	$835	GPU	Windows, macOS, Linux	3ds Max, Blender, Cinema 4D, iClone, Maya, Revit, SketchUp
Iray	$295/year	GPU	Windows, macOS	3ds Max, Cinema 4D, Maya, Rhinoceros
KeyShot	$995	GPU	Windows, macOS	Standalone
Lumion	$1,630	GPU	Windows	SketchUp, Autodesk Revit, ArchiCAD, Bentley, Allplan, Vectorworks, Rhinoceros, 3ds Max
LuxCoreRender	Free	GPU	Windows, macOS, Linux	3ds Max, Blender, Carrara, Cinema 4D, DAZ Studio, Maya, Poser, SketchUp, XSI
Marmoset Toolbag	$189	GPU	Windows, macOS	Standalone
Maxwell Render	$520	CPU/GPU	Windows, macOS, Linux	3ds Max, ArchiCAD, Cinema 4D, formZ, Maya, Modo, Revit, Rhinoceros, SketchUp, SolidWorks
Octane Render	$699/year	GPU	Windows, macOS, Linux	3ds Max, ArchiCAD, AutoCAD, Blender, Carrara, Cinema 4D, DAZ Studio, Houdini, Inventor, Lightwave, Maya, Modo, Nuke, Poser, Revit, Rhinoceros, SketchUp, Softimage
Redshift	$500	GPU	Windows, macOS, Linux	3ds Max, Cinema 4D, Houdini, Maya, Softimage
RenderMan	$595	CPU	Windows, macOS, Linux	Blender, Houdini, Katana, Maya
Solidworks Visualize	$1,830	CPU/GPU	Windows	Standalone
Thea Render	$270	CPU/GPU	Windows, macOS, Linux	SketchUp, Cinema 4D, Rhino
V-Ray	$350/year	CPU/GPU	Windows, macOS, Linux	3ds Max, Blender, Cinema 4D, Maya, Modo, Nuke, Revit, Rhinoceros, SketchUp, Unreal

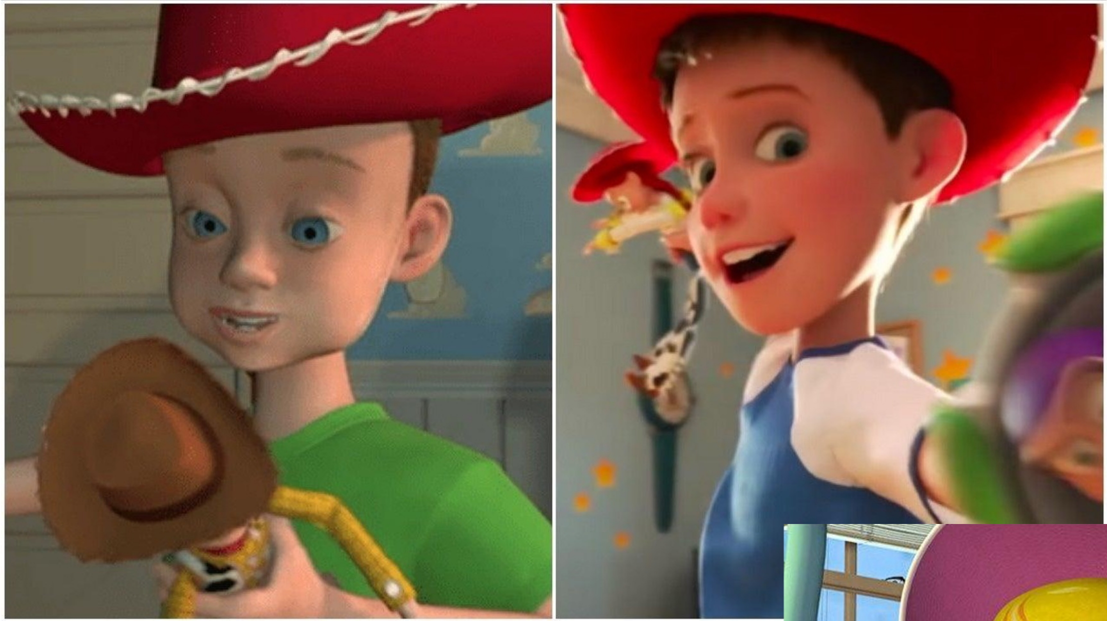
https://all3dp.com/1/best-3d-rendering-software/

Batch Rendering e Rendering Farm

Distribui a tarefa em diversos computadores (cluster), usualmente o ganho é linear sendo que cada computador gera um quadro inteiro.



Toy Story 1 e 4



Pixar

In order to render "Toy Story," the animators had **117 computers running 24 hours a day**. **Each individual frame could take from 45 minutes to 30 hours to render**, depending on how complex. There were a total of 114,240 frames to render. Throughout the movie, there are over 77 minutes of animation spread across 1,561 shots. They had to invent a new software, called Renderman, to handle all this footage.

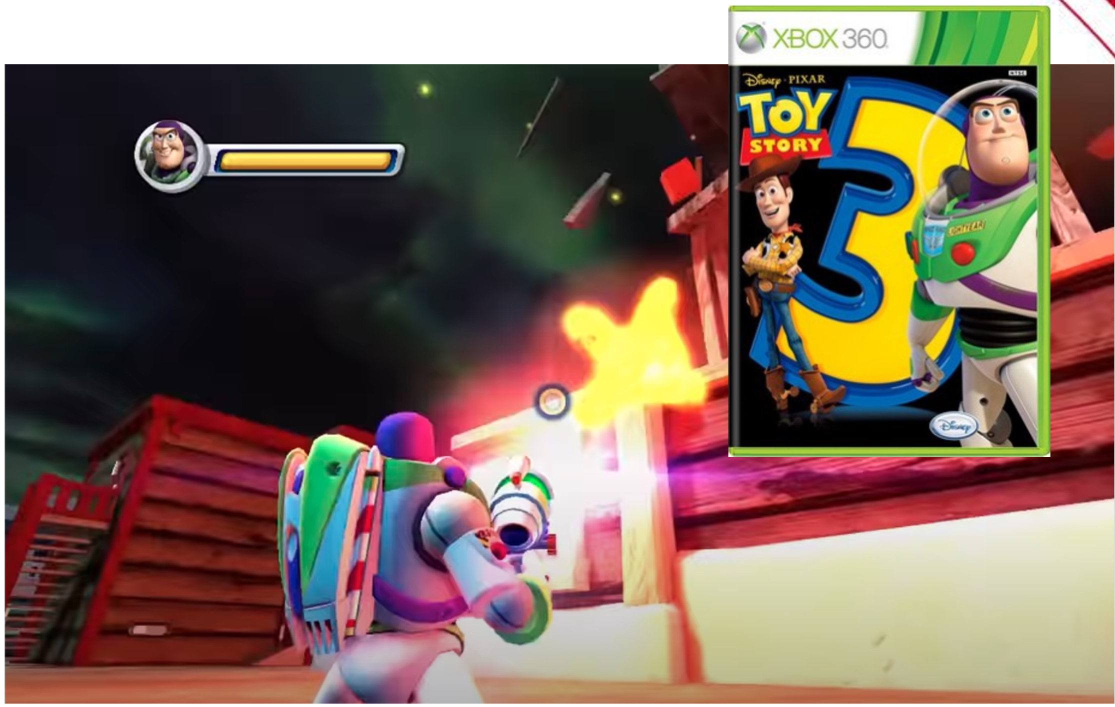
According to producer Jonas Rivera, if they had to do that today, they could render "Toy Story" faster than you could watch the entire movie. However, **the complexity of "Toy Story 4" means it can take 60 to 160 hours to render one frame**.

https://www.insider.com/pixars-animation-evolved-toy-story-2019-6

Toy Story 3: The Game

Taxa de quadros/segundo?
FPS (Frames per second)

- abaixo de 20: Impraticável.
- 20-30: percepção de descontinuidade.
- 30-45: aceitável, mas ainda perceptível.
- 45-60: suave e aceito como ideal
- +60: animações parecem naturais.



Quanto tempo para produzir o quadro?

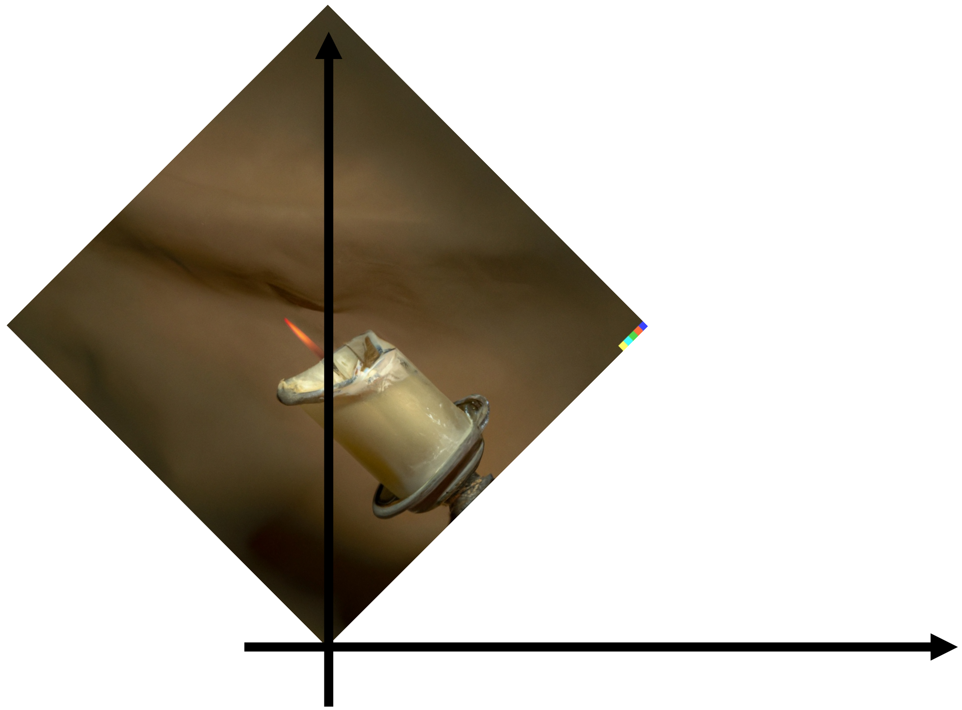
$$T = \frac{1}{f}$$

- 20: 0,05 segundos
- 40: 0,025 segundo
- 60: ~0,016 segundos
- 120: ~0,0083 segundos

* sem contar outros atrasos que acontecem em qualquer sistema.

Transformações

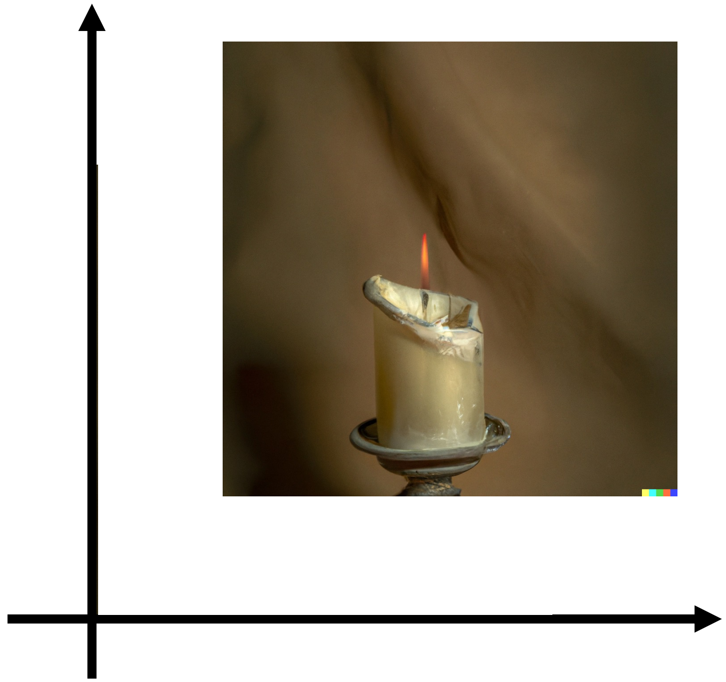
Rotação



Rotação de 45º

Transformações

Translação



Translação (1,1)

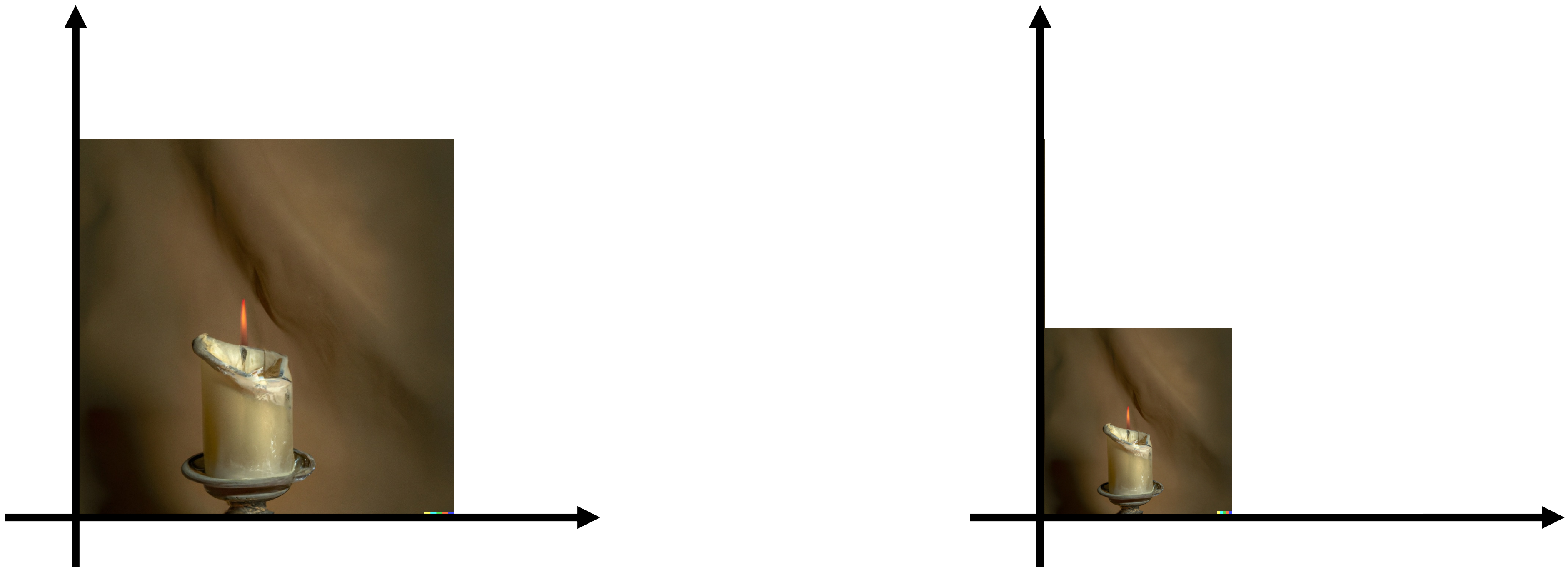
Transformações

Reflexão

Reflexão em X

Transformações

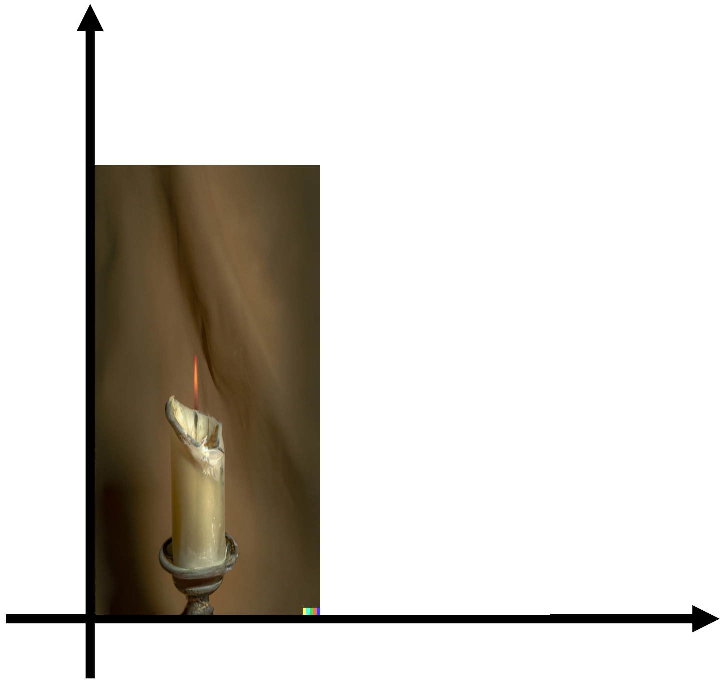
Escala Uniforme



Escala (0.5, 0.5)

Transformações

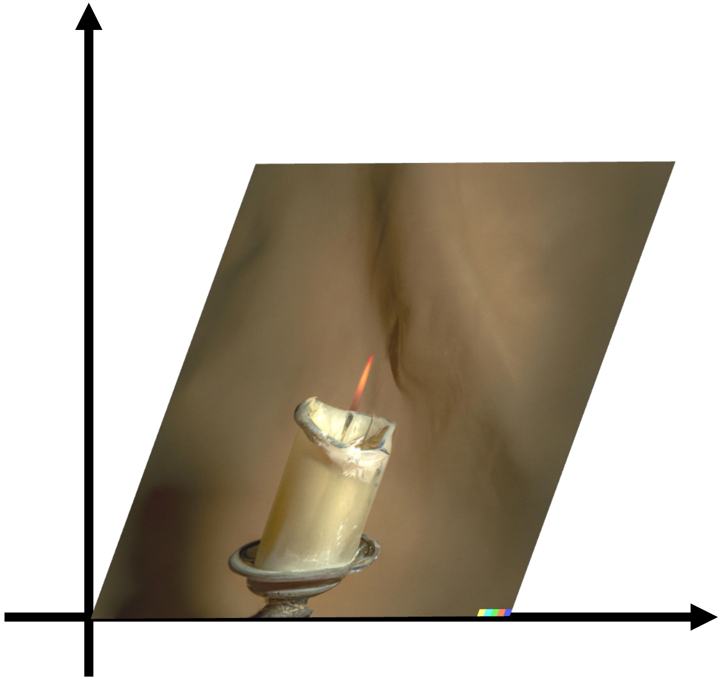
Escala Assimétrica



Escala (0.5, 1.0)

Transformações

Cisalhamento

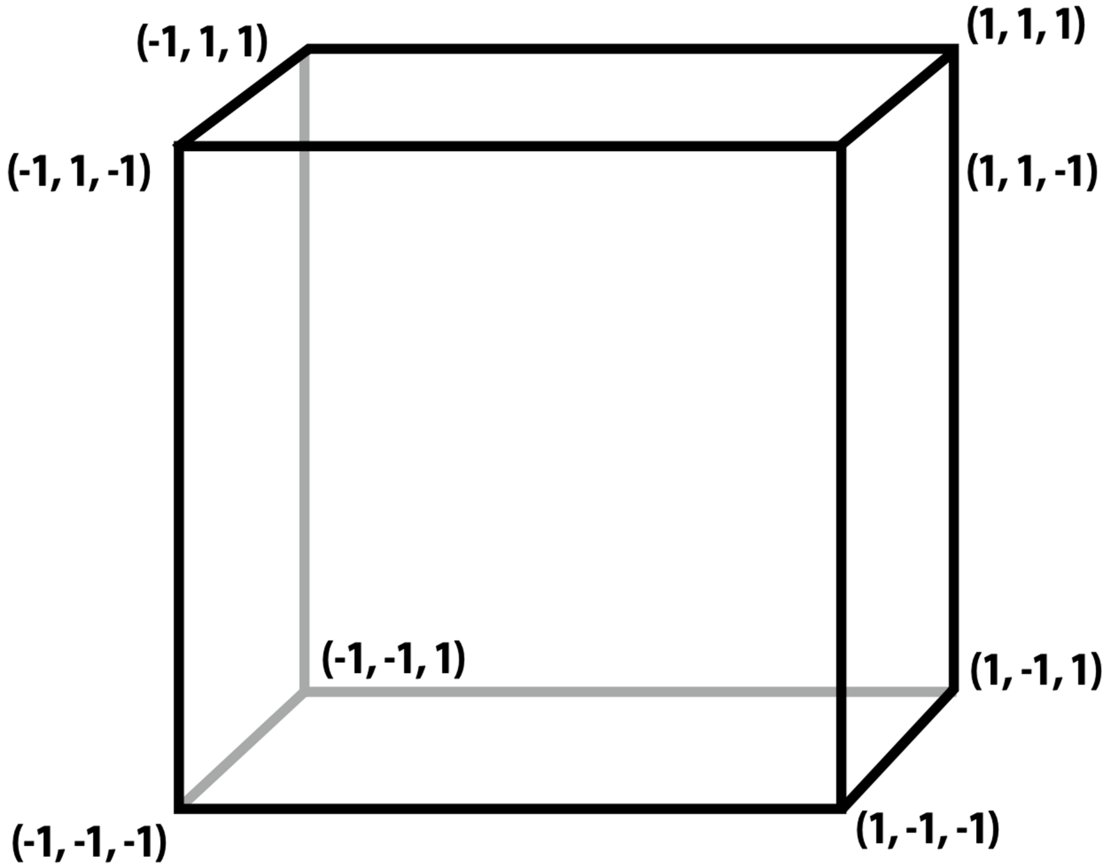


Escala (0.5, 1.0)

Insper

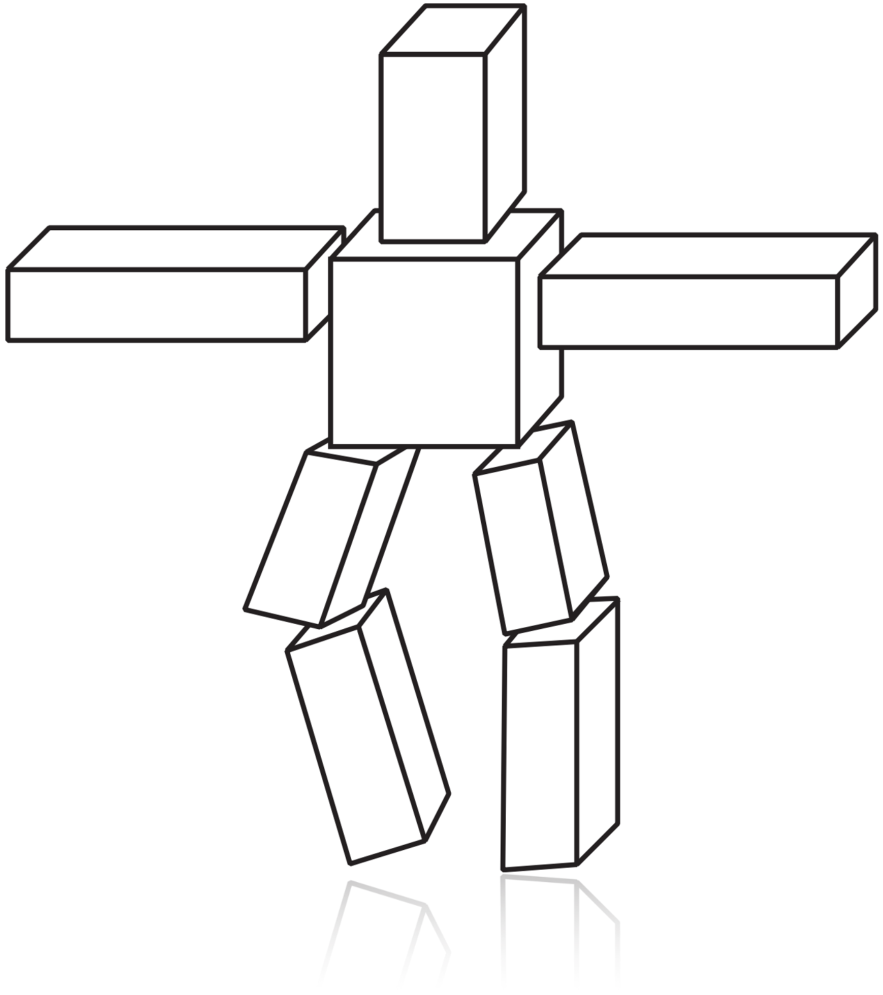
Cubo

Podemos definir e manipular um cubo pelos seus vértices.

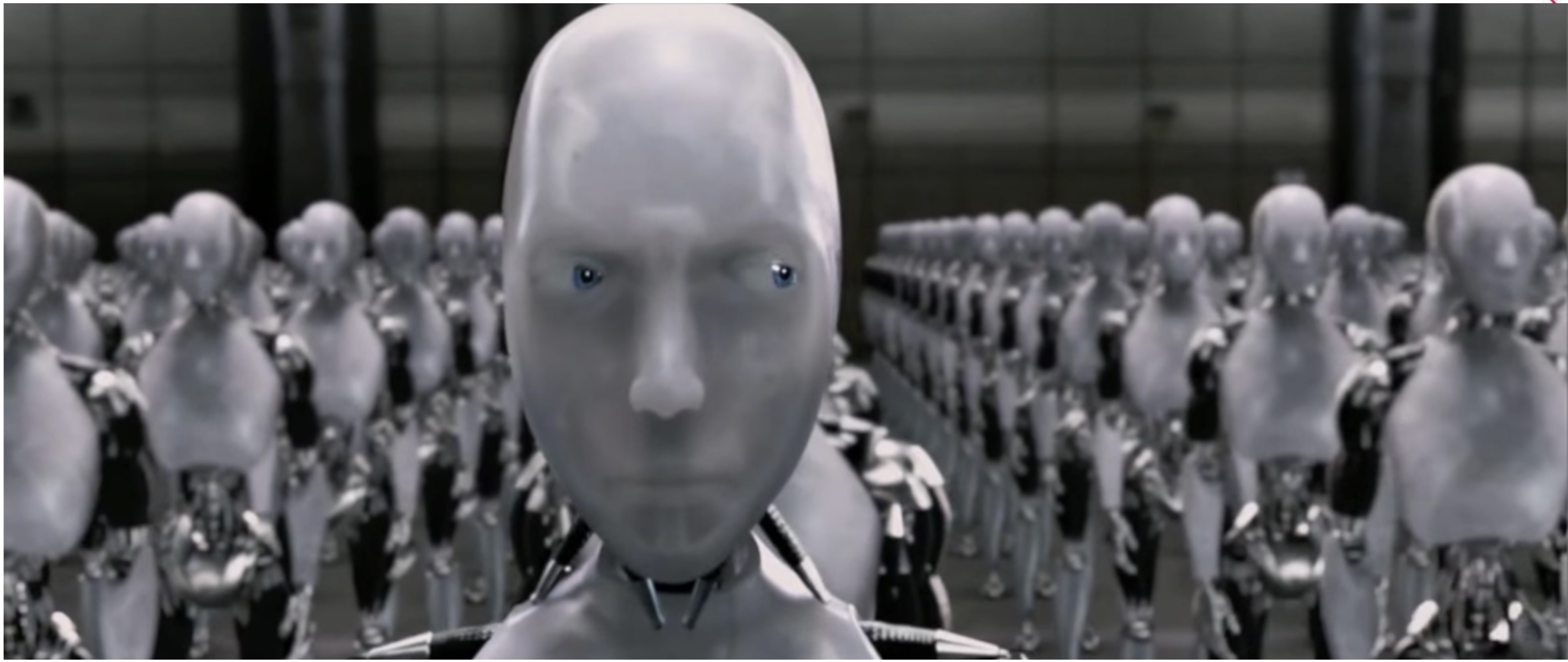


Desenhar um pessoa por cubos

Podemos esboçar um personagem humanóide por cubos.



Criando um exército de robôs



Filme: Eu Robô

Transformações descrevem a posição de cada instância.

A pose de um personagem

Filme: Eu Robô

As transformações podem descrever as posições relativas de cada parte do corpo do personagem sendo animado.

Por que estudar Transformações ?

Modelagem

- Definir formas geométricas em coordenadas convenientes
- Possibilitar várias cópias do mesmo objeto
- Representar com eficiência hierarquia de objetos

Visualização

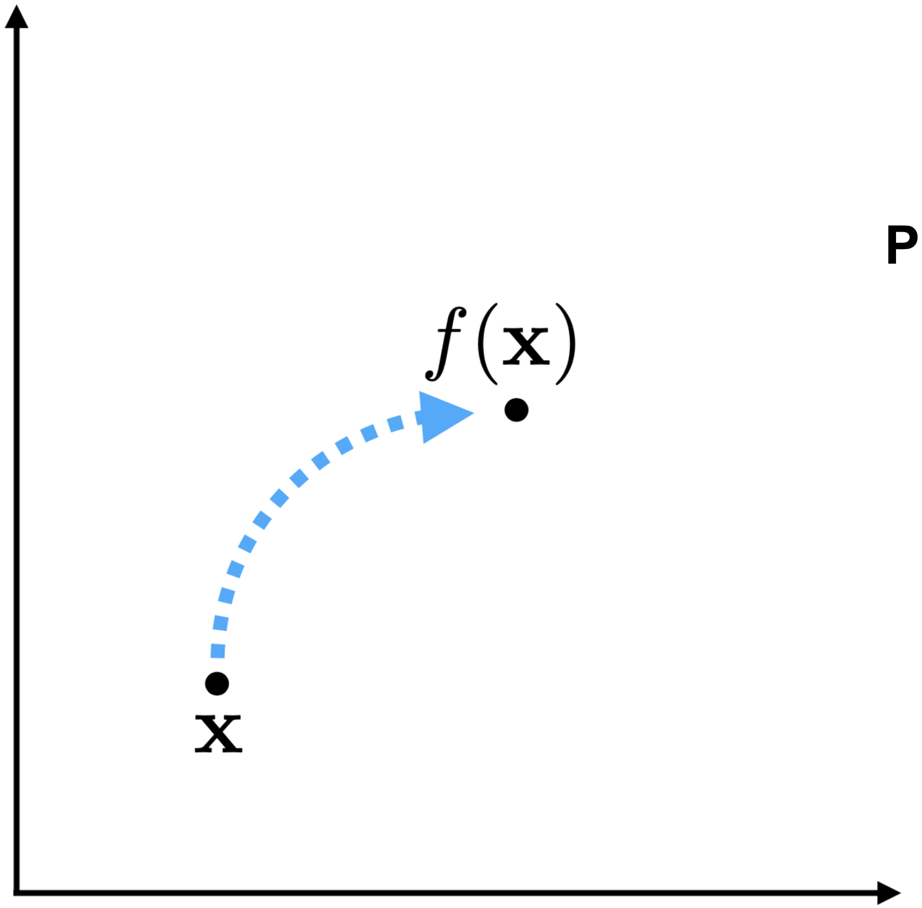
- Levar coordenadas do mundo para as coordenadas da câmera
- Projeções ortográficas e perspectivas

Um pouco de teoria:

relembrando Álgebra Linear

Ideia Básica das Transformações

Funções agindo em pontos



Por exemplo, pontos 2D

(x’,y’) = f(x,y)

Insper

Transformações Lineares

O que esse Linear significa?

$$f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$$

$$f(a\mathbf{x}) = af(\mathbf{x})$$

Cálculos são simples de serem realizados no computador

A composição das transformações lineares também é linear

- Leva a uma representação uniforme de transformações

Transformações Lineares

Caso u e v sejam vetores e desejarmos fazer uma operação.

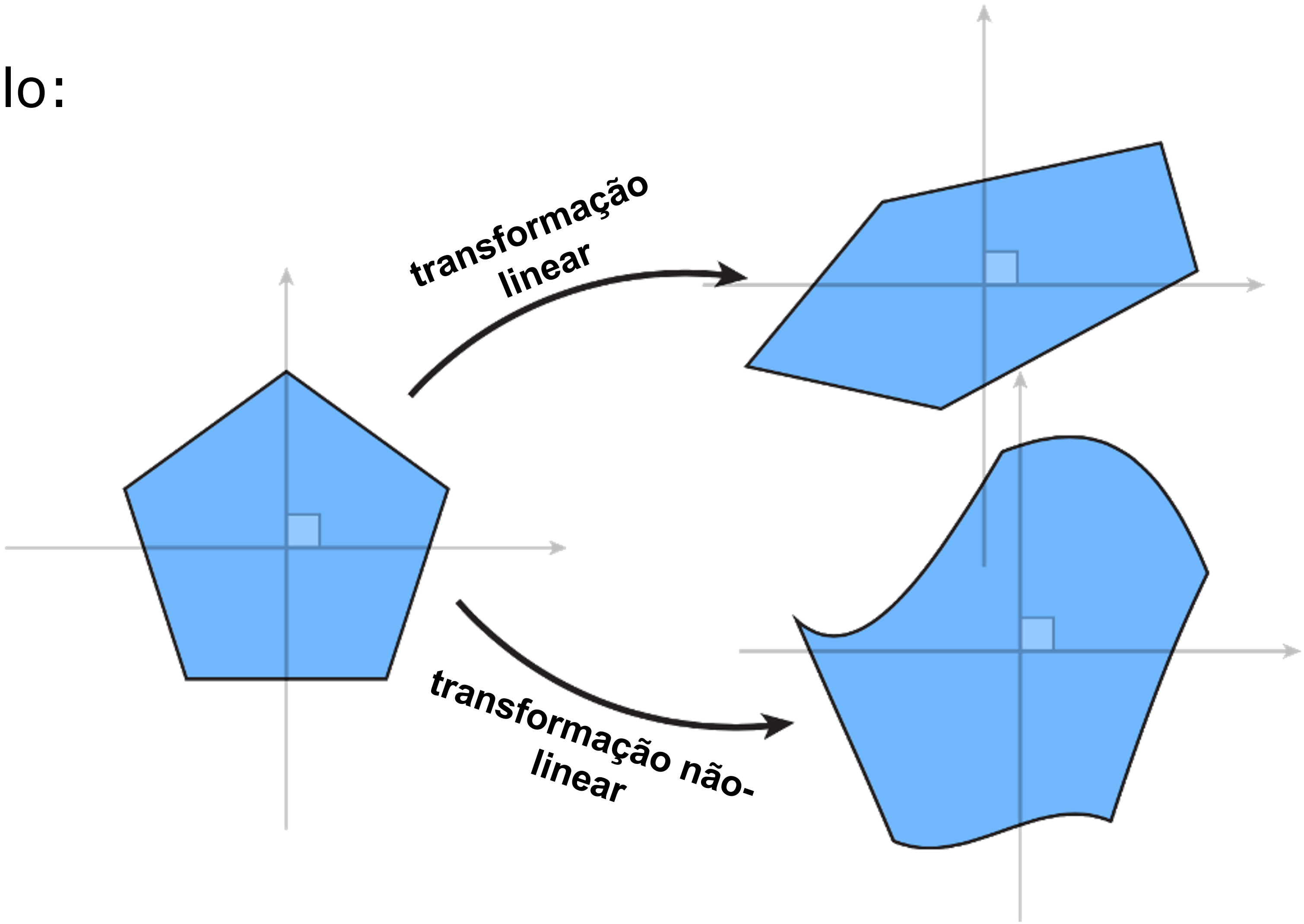
$$f(\mathbf{u}+\mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$$
$$f(a\mathbf{u}) = af(\mathbf{u})$$

Isso simplifica muito algumas estratégias que serão usadas.

$$\begin{tikzcd}
\mathbf{x},\mathbf{y} \arrow[r] \arrow[dd] & \mathbf{x}+\mathbf{y} \arrow[d] \\
 & f(\mathbf{x})+f(\mathbf{y}) \arrow[d, phantom, "="] \\
f(\mathbf{x}),f(\mathbf{y}) \arrow[r] & f(\mathbf{x}+\mathbf{y})
\end{tikzcd}$$

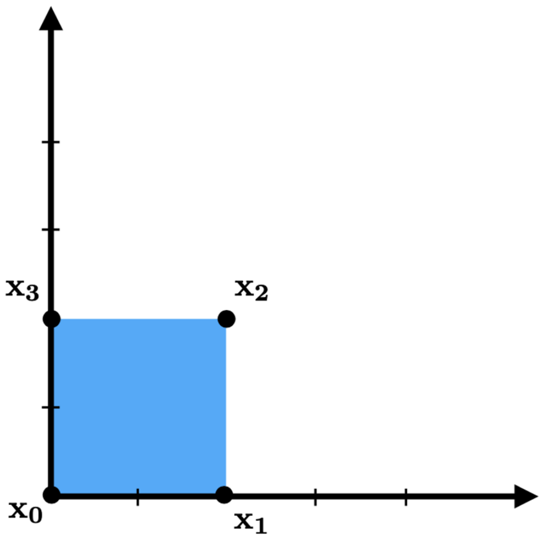
Transformações Lineares (map)

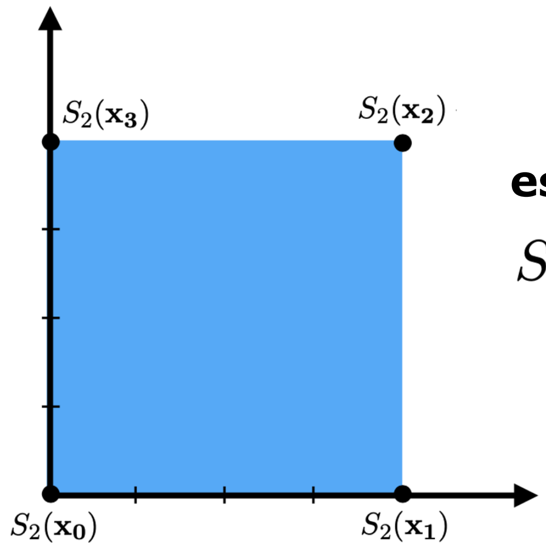
Exemplo:



Ideia básica: transformações lineares levam linhas a linhas

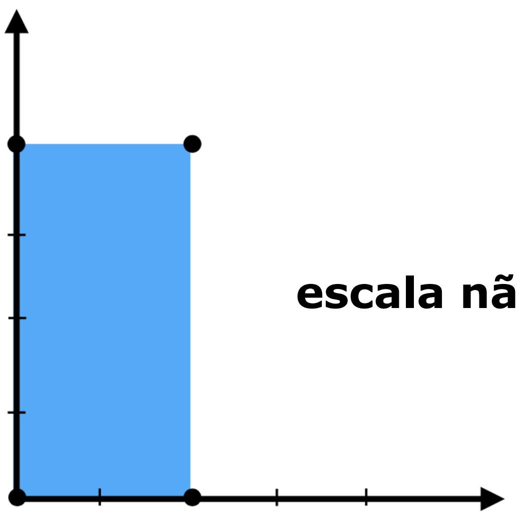
Escala





escala uniforme

$$S_a(\mathbf{x}) = a\mathbf{x}$$



escala não uniforme

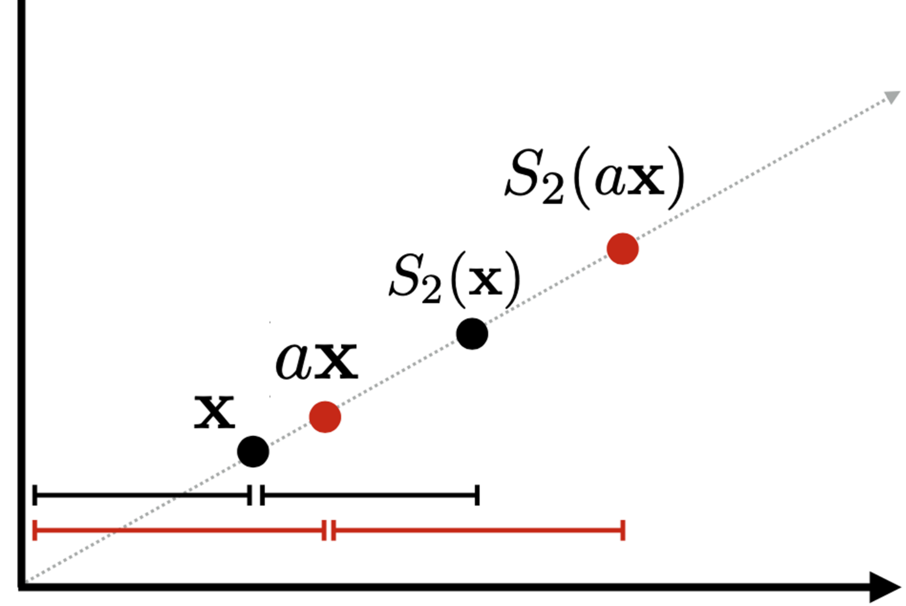
A escala é uma transformação linear ?

$$S_2(\mathbf{x}) = 2\mathbf{x}$$

$$aS_2(\mathbf{x}) = 2a\mathbf{x}$$

$$S_2(a\mathbf{x}) = 2a\mathbf{x}$$

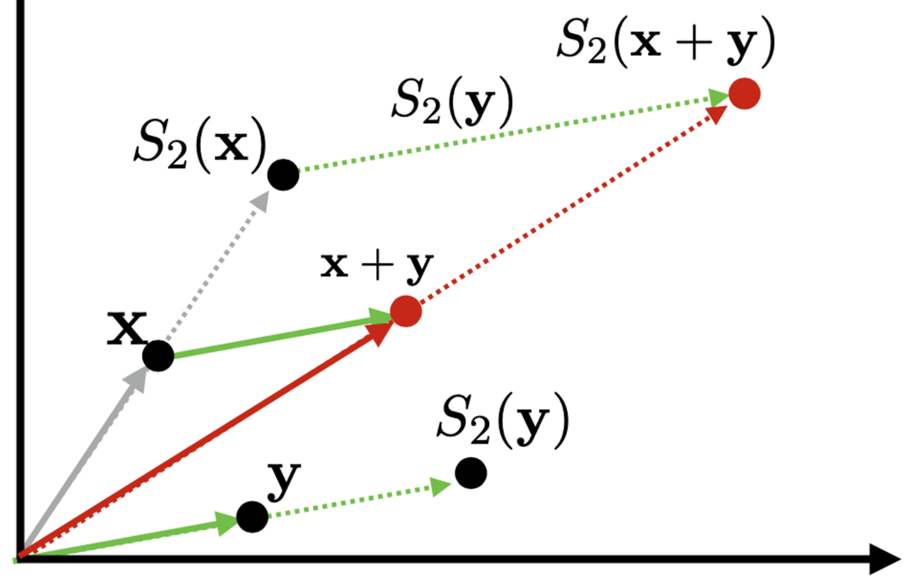
$$S_2(a\mathbf{x}) = aS_2(\mathbf{x})$$



$$S_2(\mathbf{x}+\mathbf{y}) = 2(\mathbf{x}+\mathbf{y})$$

$$S_2(\mathbf{x}) + S_2(\mathbf{y}) = 2\mathbf{x} + 2\mathbf{y}$$

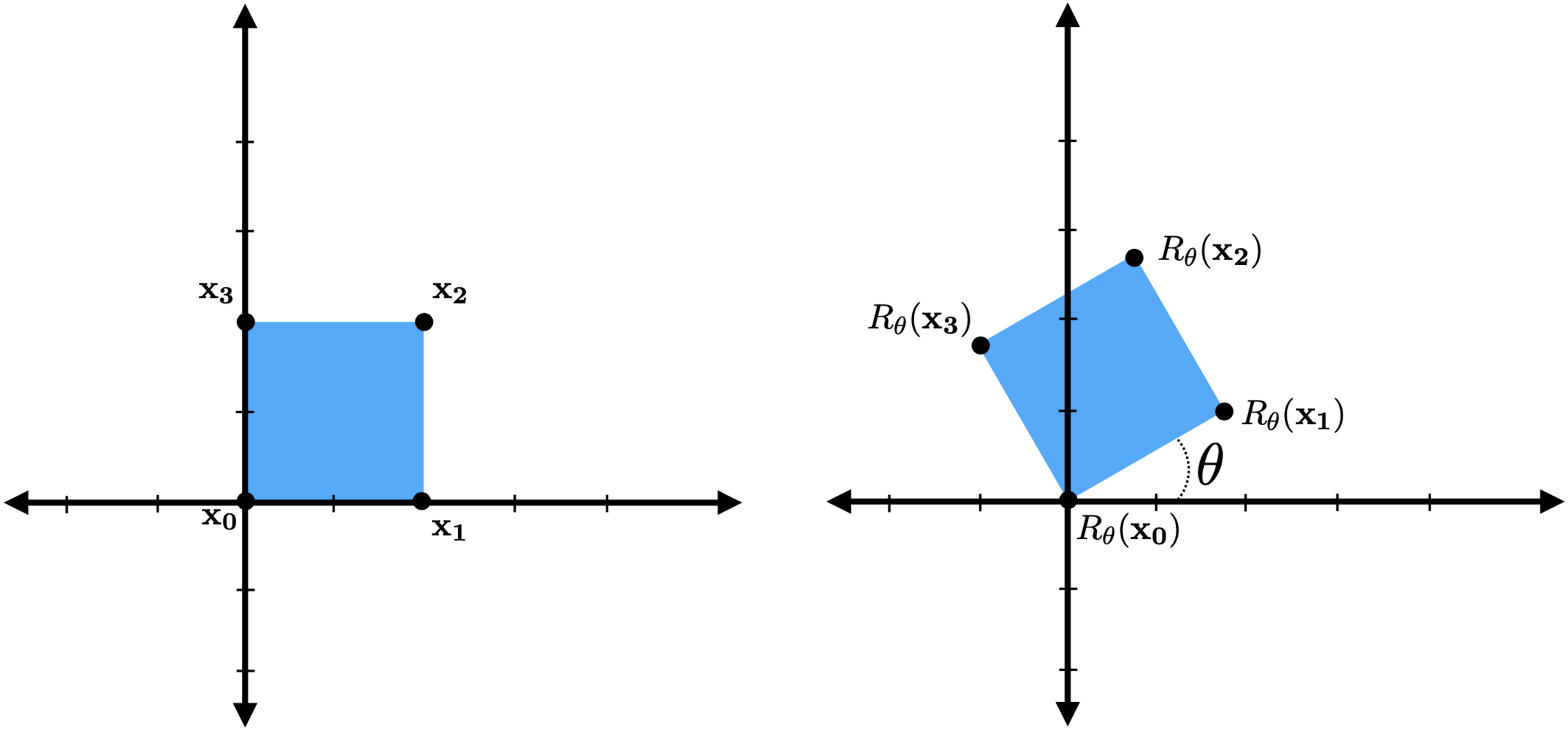
$$S_2(\mathbf{x}+\mathbf{y}) = S_2(\mathbf{x}) + S_2(\mathbf{y})$$



a resposta é: **SIM**

Insper

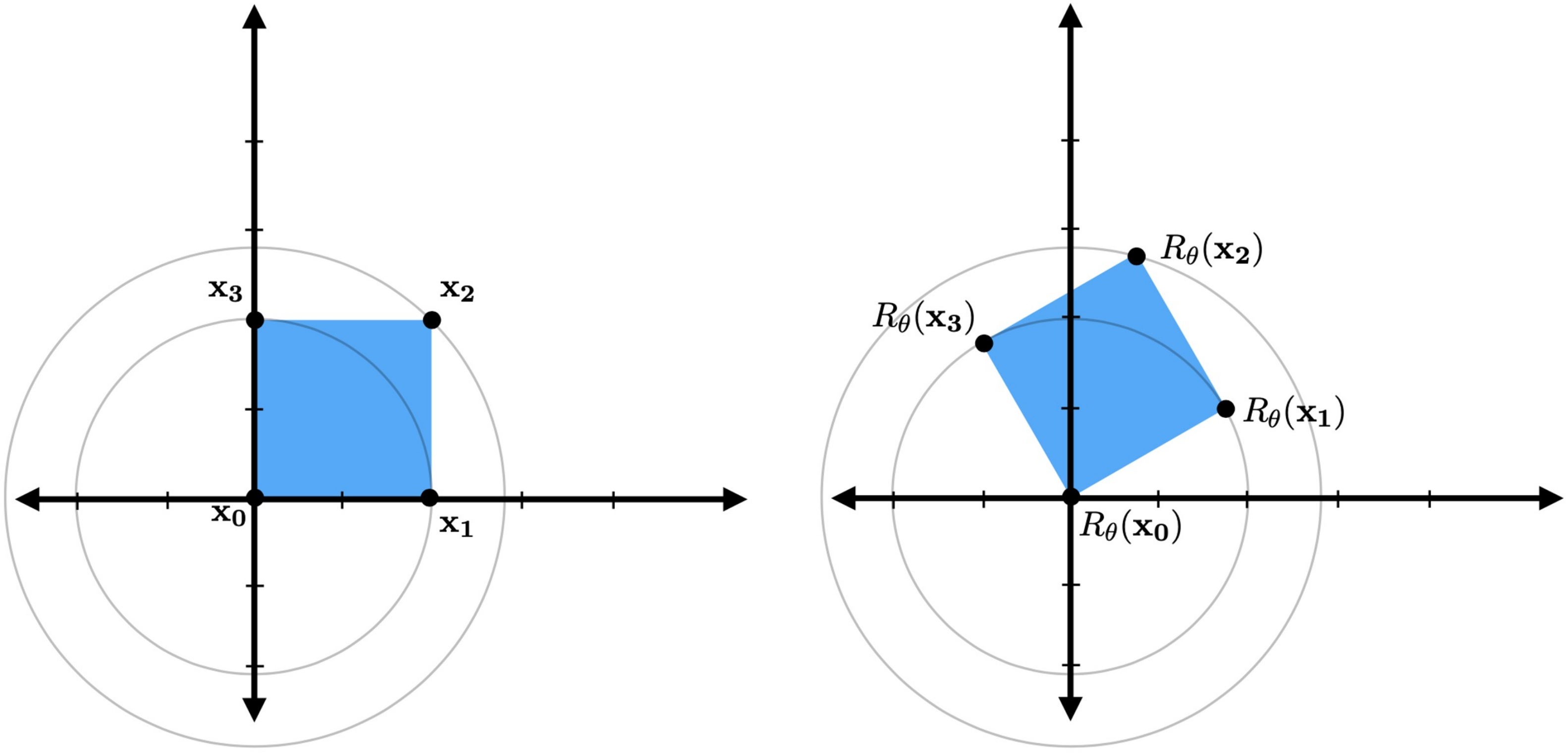
Rotação



R_θ = rotaciona θ no sentido anti-horário

Insper

Rotação como um movimento circular



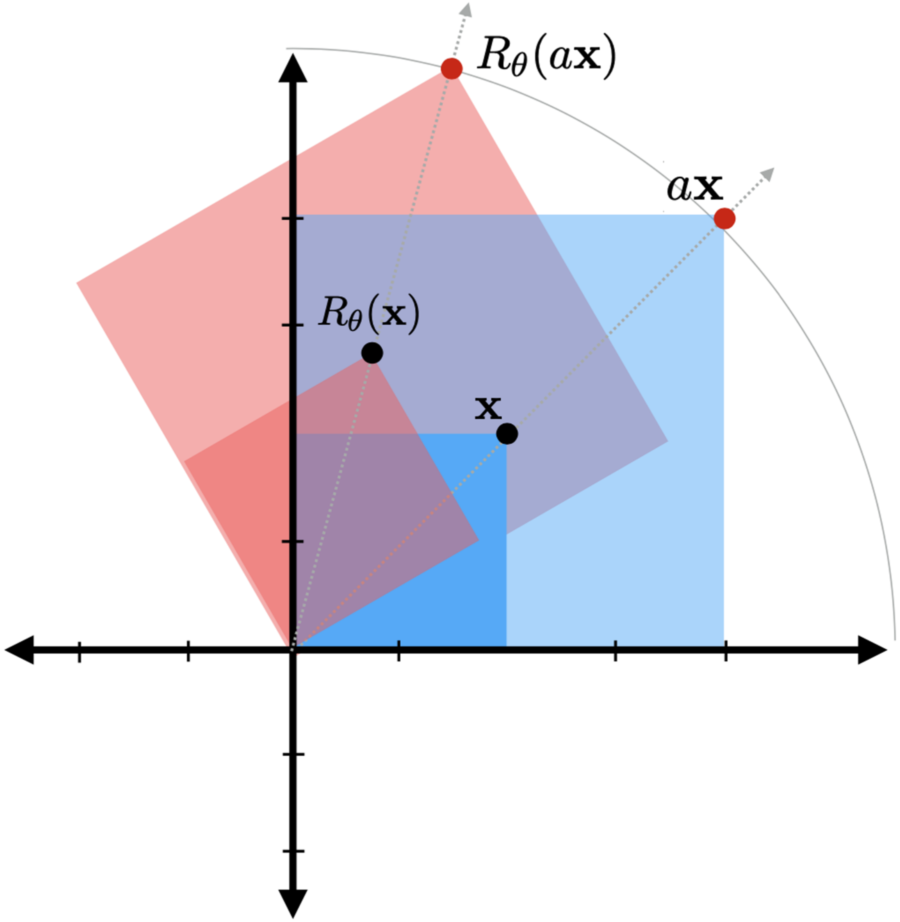
R_θ = rotaciona θ no sentido anti-horário

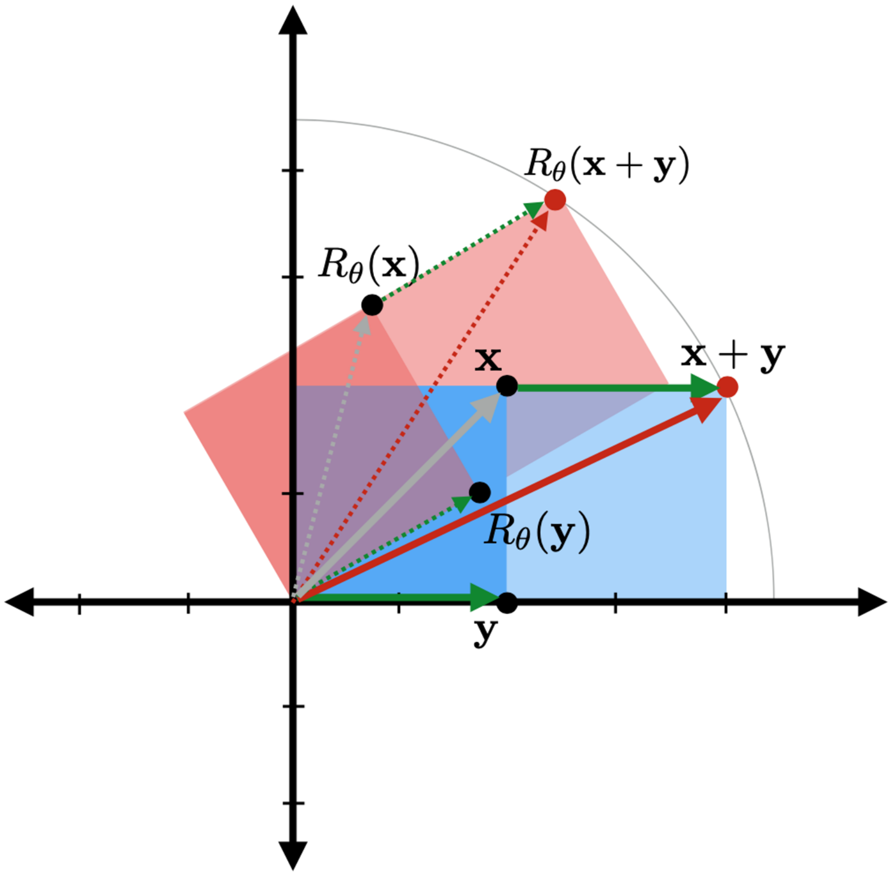
Conforme o ângulo muda, os pontos se movem ao longo de uma trajetória circular.

Portanto, as rotações preservam o comprimento dos vetores: $|R_\theta(\mathbf{x})| = |\mathbf{x}|$

Insper

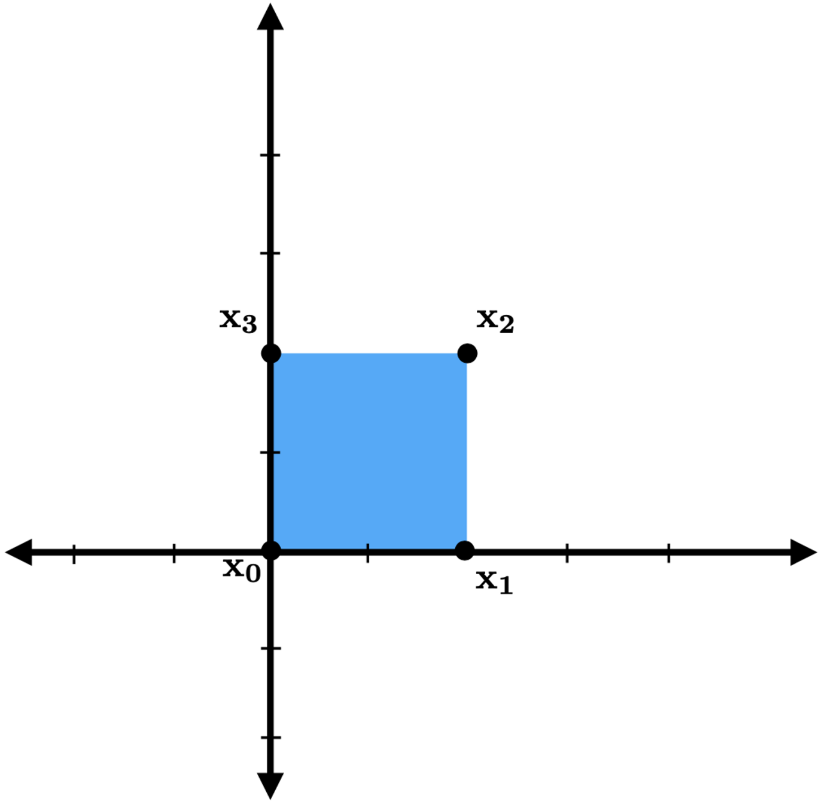
A transformação de rotação é linear ?

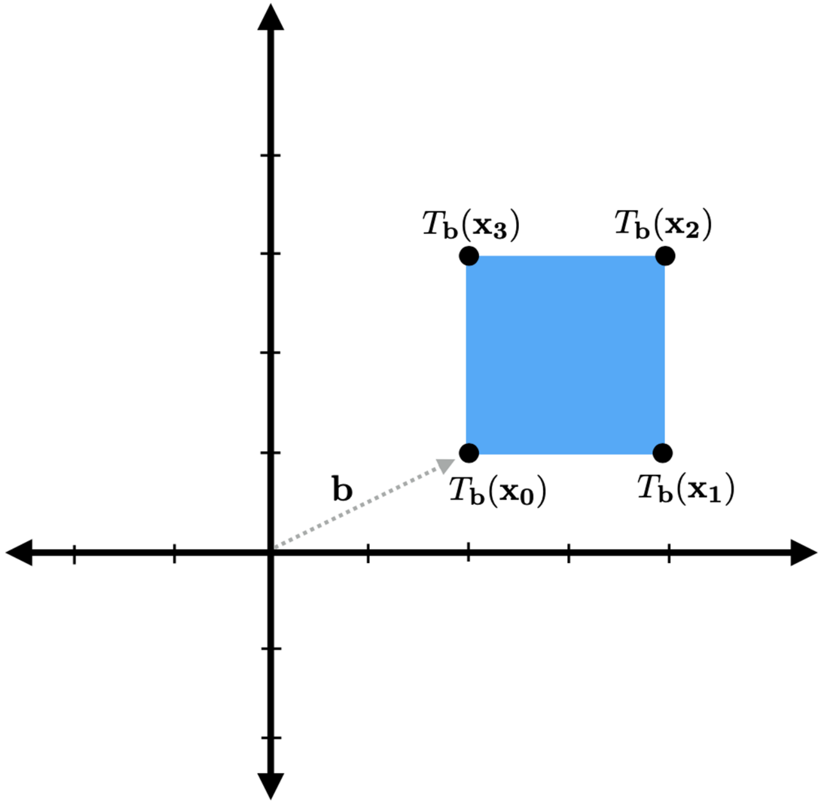




SIM

Translação



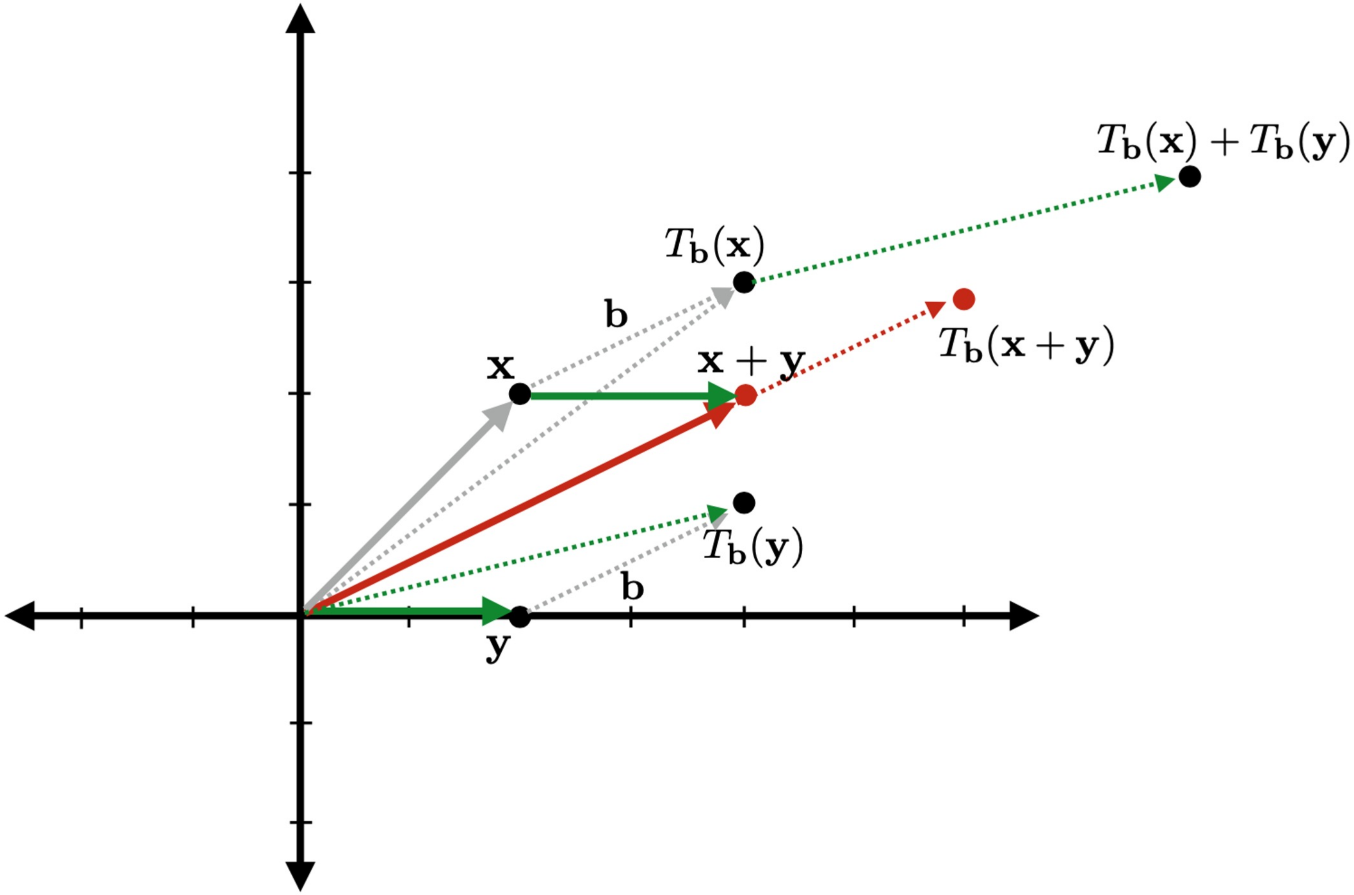


T_b — "translade (mova) por b"

$T_b(x) = x + b$

Insper

A transformação de translação é linear?



NÃO

Insper

Transformação Afim

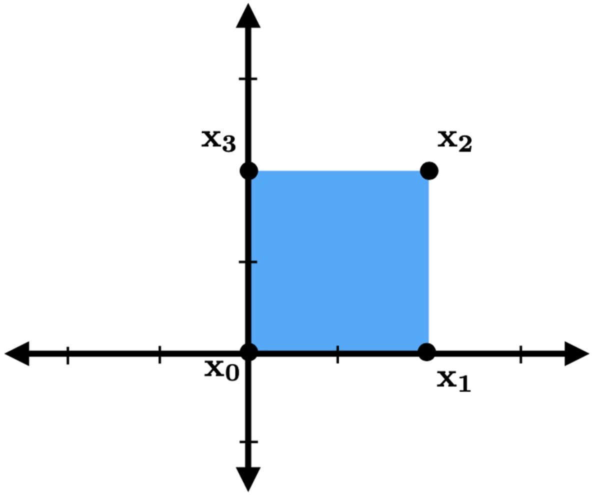
Dizemos que a translação é uma transformação **afim**

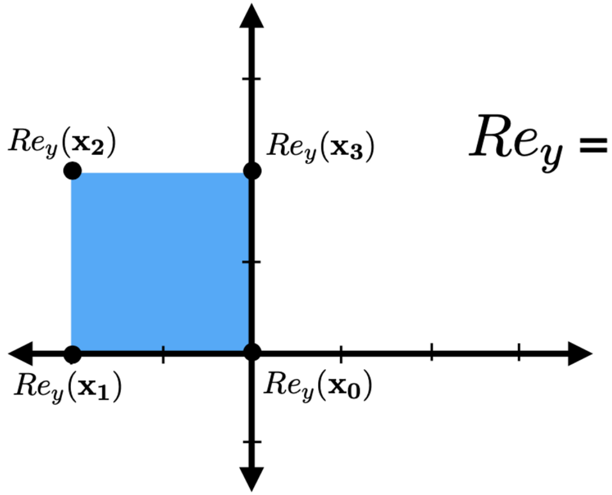
Na prática:

Composição de uma transformação linear + translação

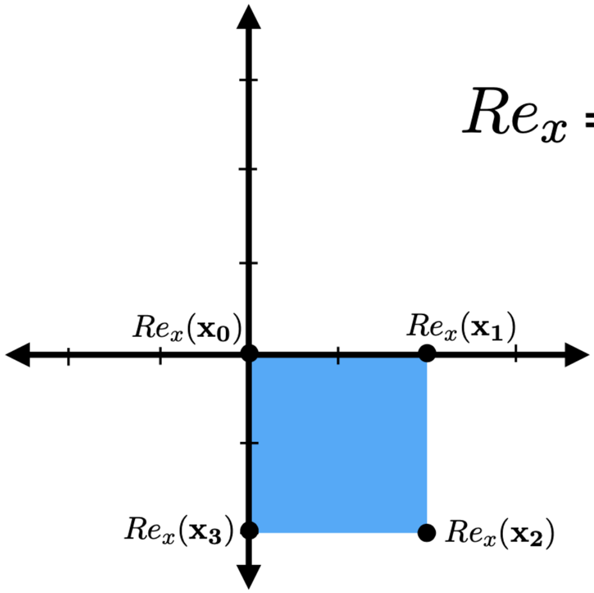
$$f(\mathbf{x}) = g(\mathbf{x}) + \mathbf{b}$$

Reflexão



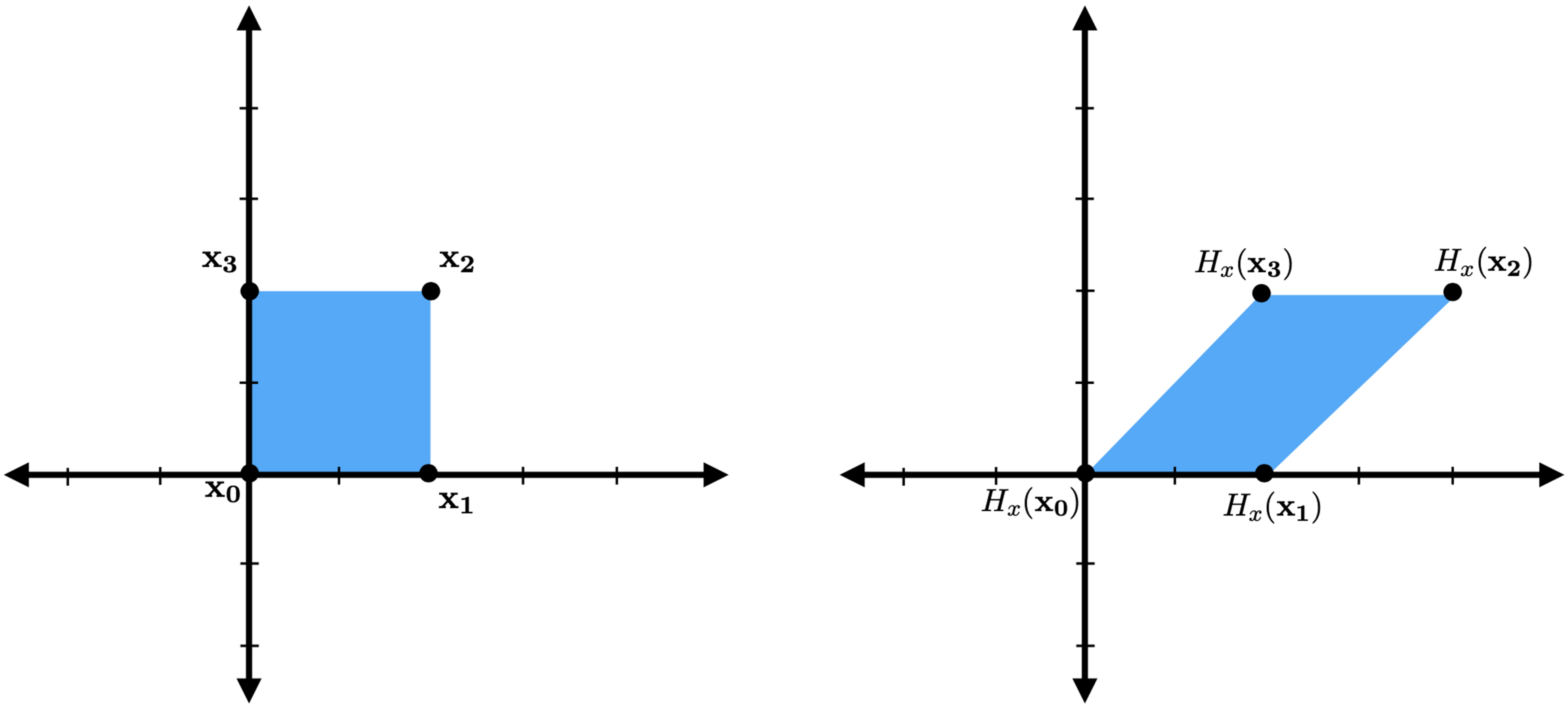


Re_y = **reflexão em torno do y**



Re_x = **reflexão em torno do x**

Cisalhamento (na direção de x)

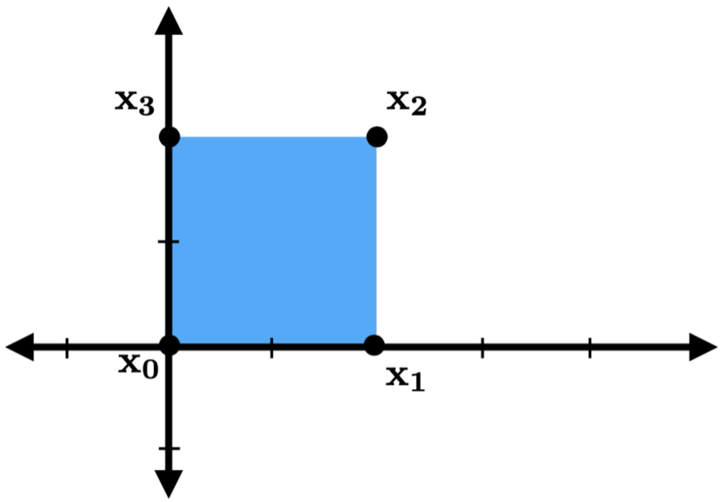


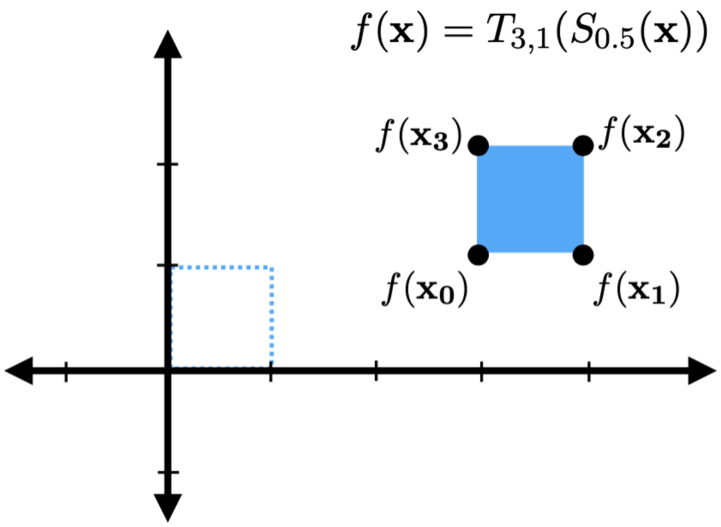
Insper

A reflexão e cisalhamento são lineares ?

SIM, ambas são lineares

Composição de Transformações

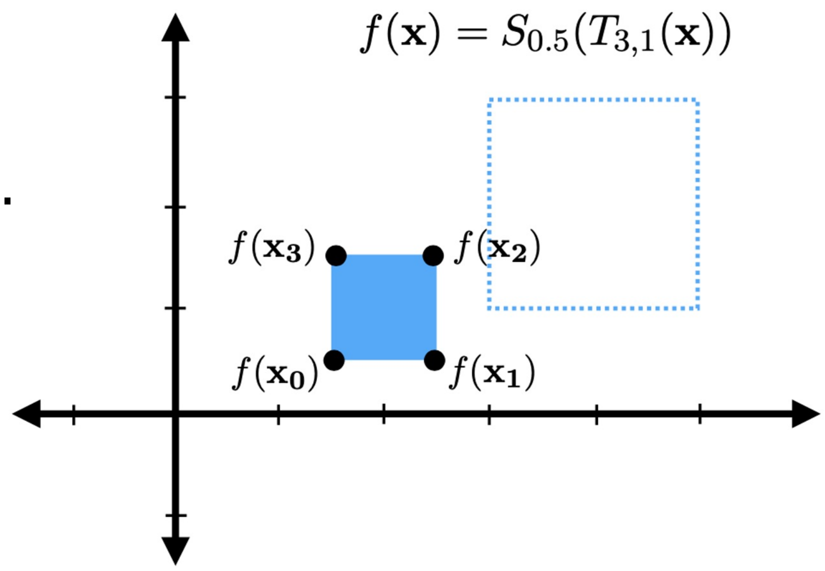




Se pode fazer a composição de transformações mais básicas para se conseguir transformações mais complexas.

Observação: a ordem importa

- superior-direito: escala e translada
- inferior-direito: translada e escala



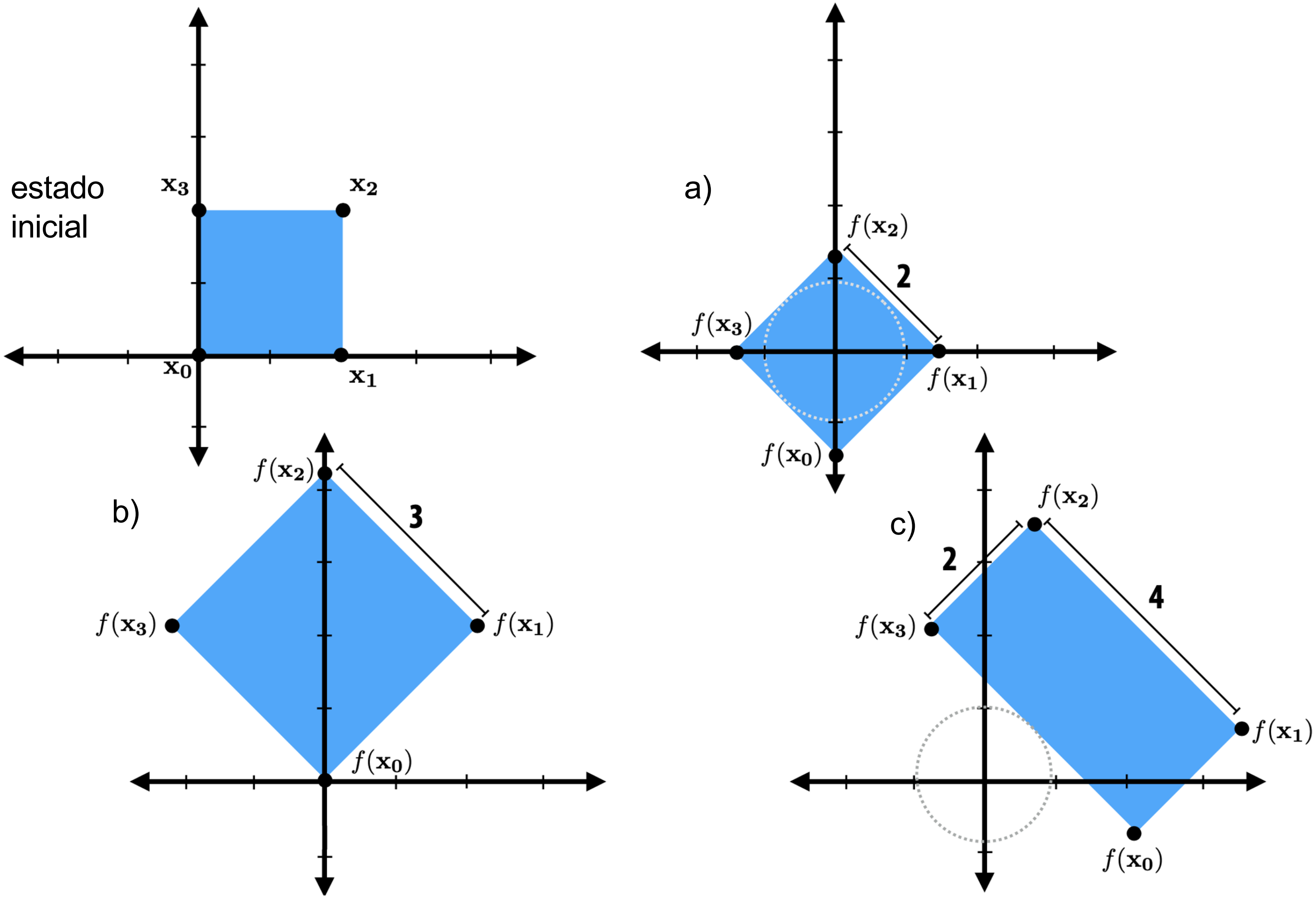
ATIVIDADE 1: Transformações Geométricas

Acesse o documento

Realize todos os exercícios.

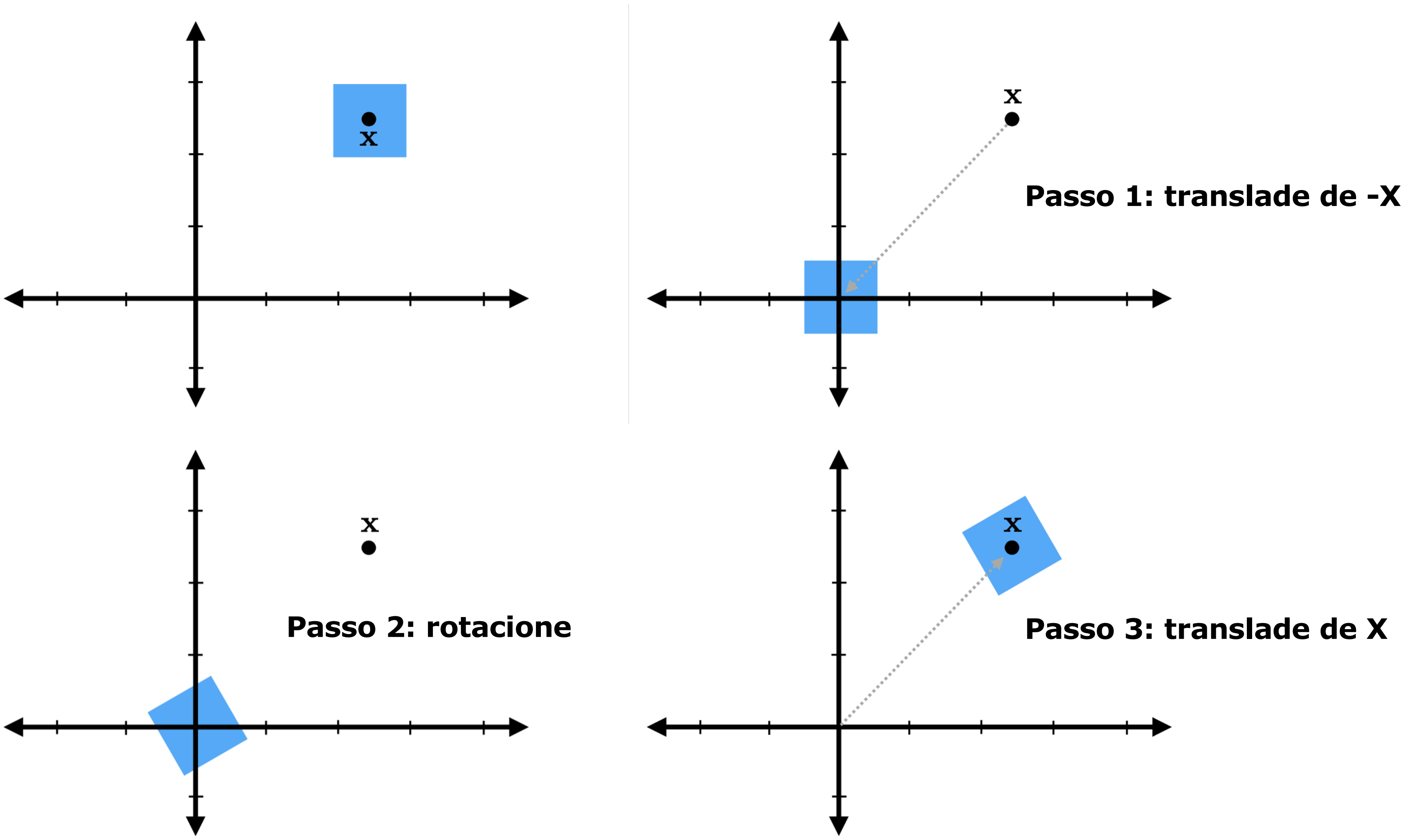
Voltamos em 15 minutos?

Como você faria essas transformações?



Usualmente existe mais de uma maneira de se fazer.

Rotacionar sobre um ponto específico



Sumário de Transformações Geométricas Básicas

Lineares:

$$f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$$
$$f(a\mathbf{x}) = af(\mathbf{x})$$

- Escala
- Rotação
- Reflexão
- Cisalhamento

Não Lineares

- Translação

Transformações Afim:

Composição de uma transformação linear + translação

$$f(\mathbf{x}) = g(\mathbf{x}) + \mathbf{b}$$

Isometria

Preserva a distância entre os pontos (comprimento)

$$|f(\mathbf{x}) - f(\mathbf{y})| = |\mathbf{x} - \mathbf{y}|$$

- Translação
- Rotação
- Reflexão

Revisão: Multiplicação de Matrizes

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

Multiplicação de Matrizes (Revisão)

$$x' = a\,x + b\,y$$
$$y' = c\,x + d\,y$$

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

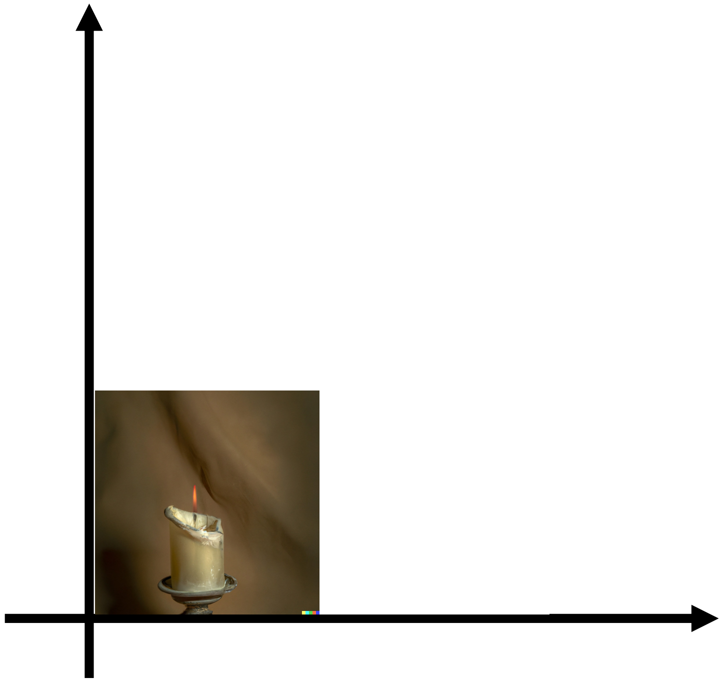
$$\mathbf{x}' = \mathbf{M}\,\mathbf{x}$$

Multiplicação de matrizes é uma combinação linear de colunas.

Server para armazenar as transformações discutidas

Insper

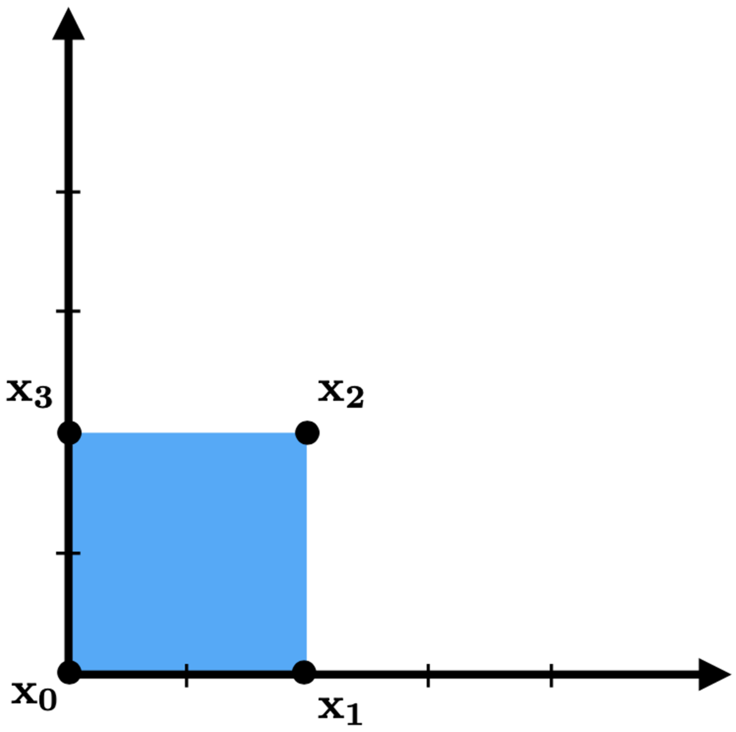
Matriz de Escala

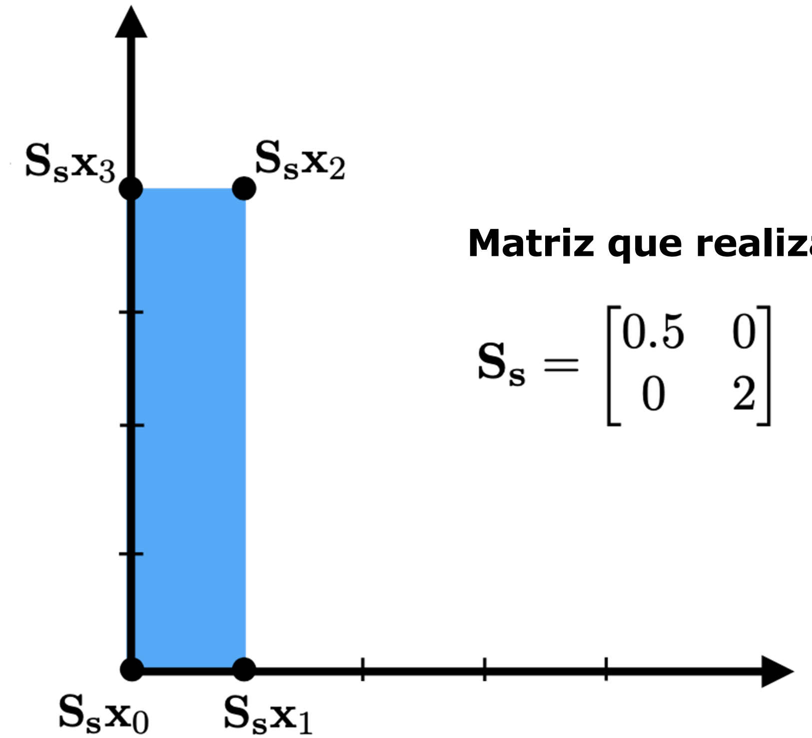


$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Transformações Lineares em 2D

Considere uma escala não uniforme: $\mathbf{S_s} = \begin{bmatrix} \mathbf{s}_x & 0 \\ 0 & \mathbf{s}_y \end{bmatrix}$



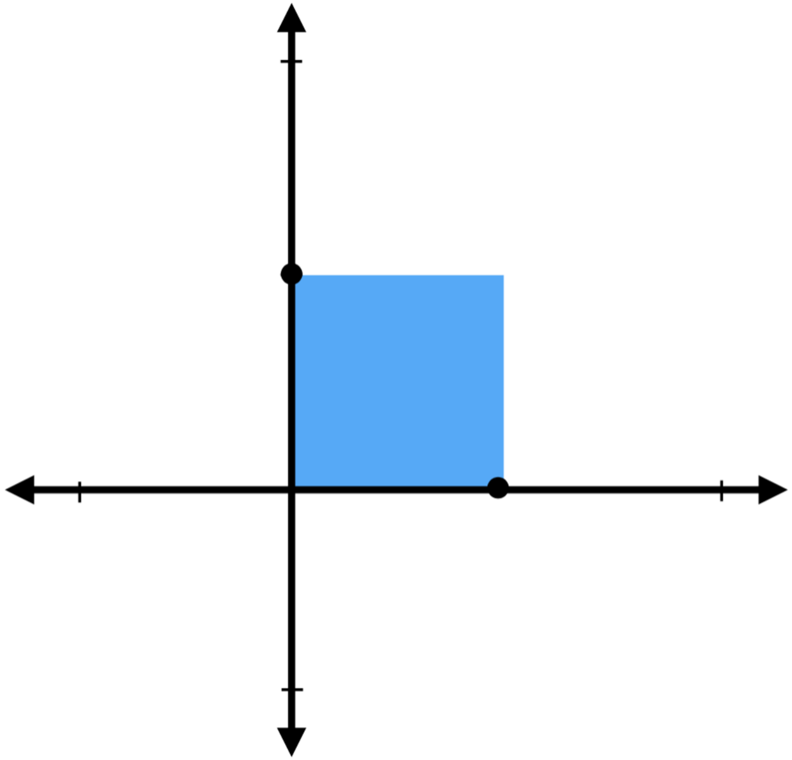


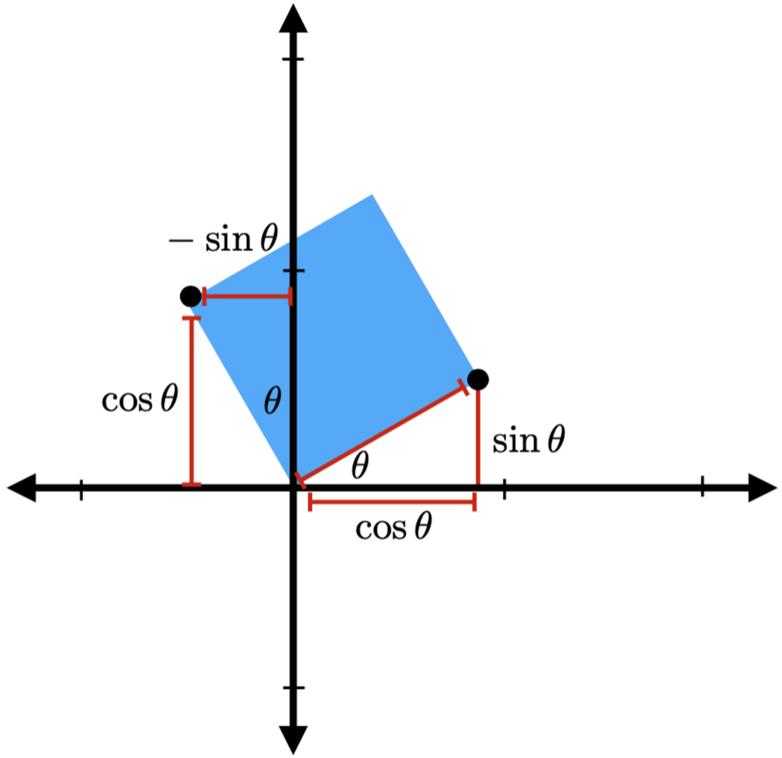
Matriz que realiza a escala

$$\mathbf{S_s} = \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix}$$

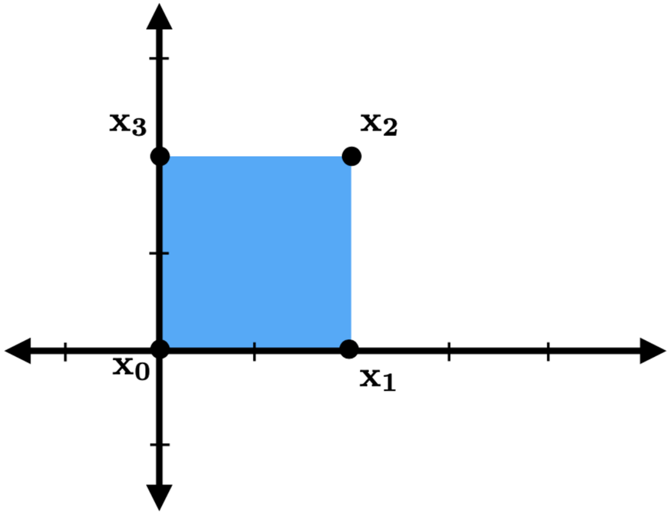
Matriz de Rotação (2D)

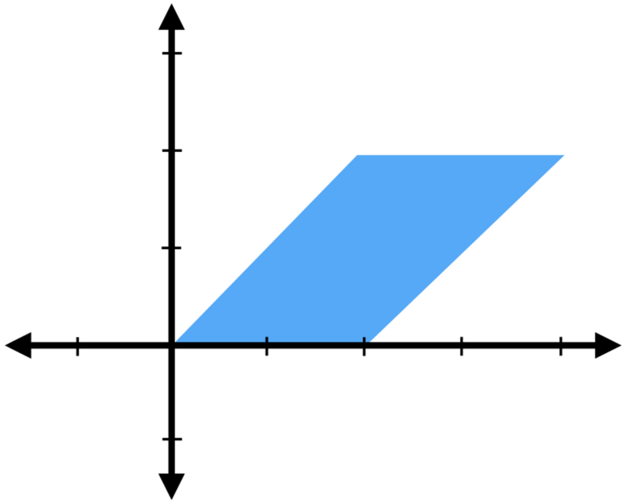
$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$





Cisalhamento



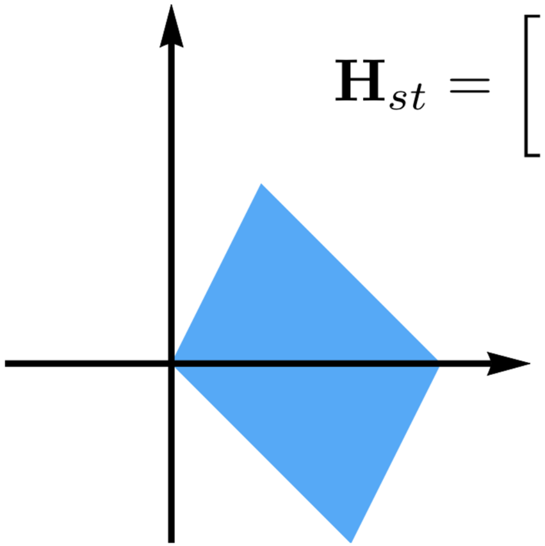


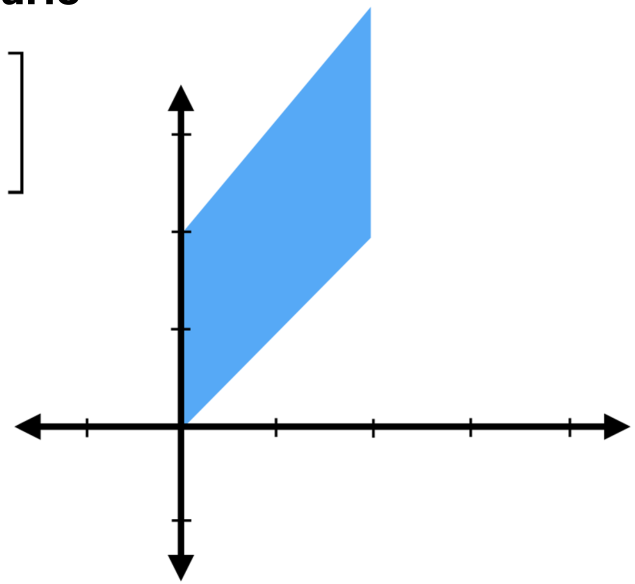
Cisalhamento em x:

$$\mathbf{H}_{xs} = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix}$$

Cisalhamento Arbitrário

$$\mathbf{H}_{st} = \begin{bmatrix} 1 & s \\ t & 1 \end{bmatrix}$$

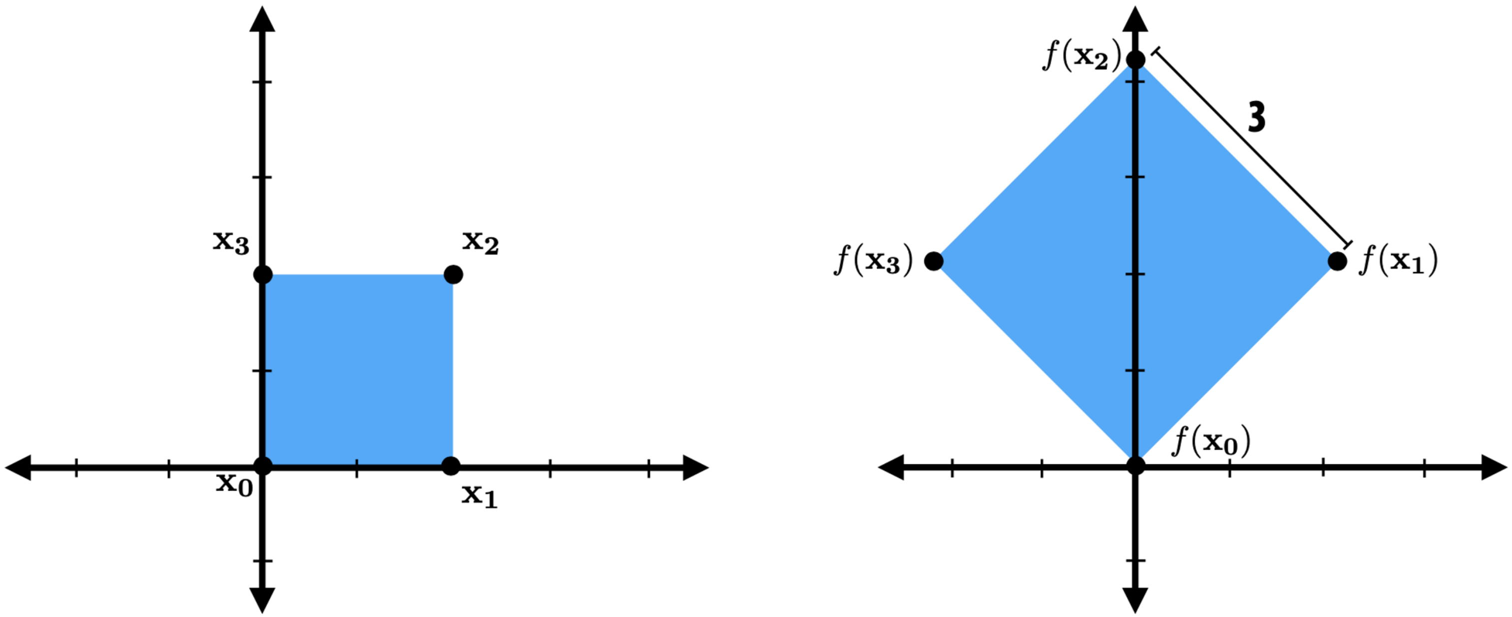




Cisalhamento em y:

$$\mathbf{H}_{ys} = \begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix}$$

Como compor matrizes de transformações lineares ?



Componha transformações lineares via multiplicação de matrizes.

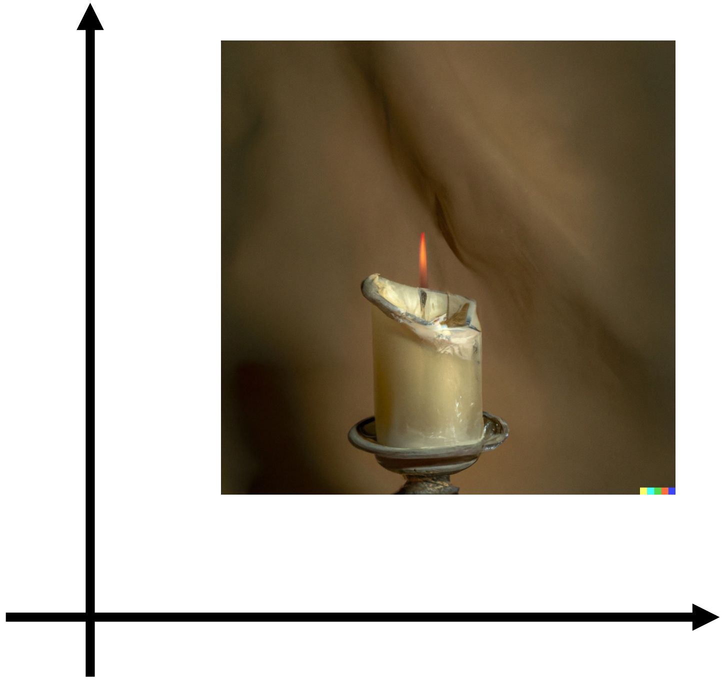
No exemplo: Uma escala uniforme seguido de uma rotação

$$f(\mathbf{x}) = R_{\pi/4}\mathbf{S}_{[1.5,1.5]}\mathbf{x}$$

Permite uma implementação simples e eficiente: reduz a cadeia complexa de transformações a uma única multiplicação de matriz.

Insper

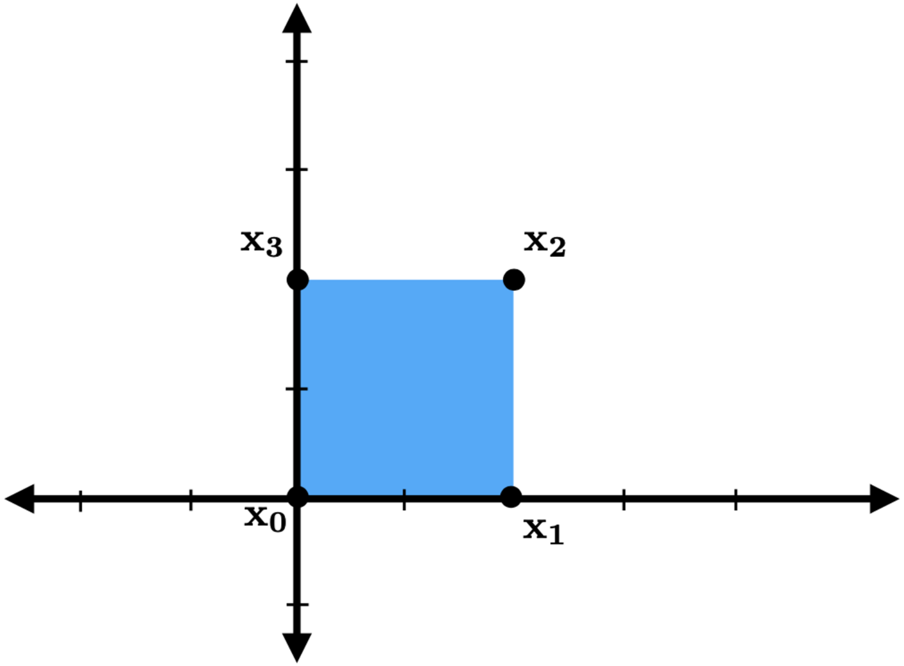
Translação - Como lidar com essa transformação?

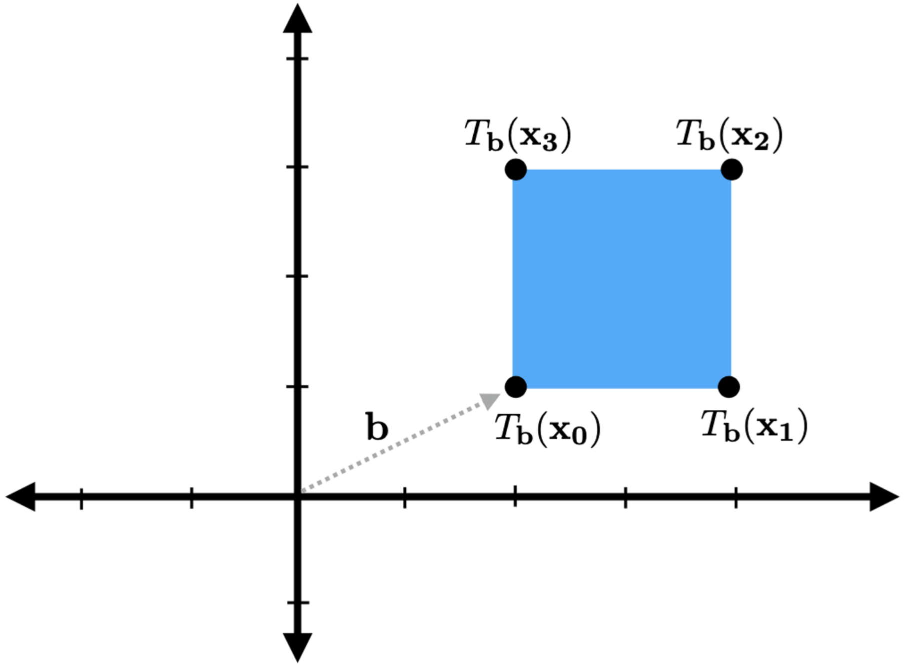


$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad ????$$

E como tratar translações (não linear) ?

$$T_{\mathbf{b}}(\mathbf{x}) = \mathbf{x} + \mathbf{b}$$





- A operação de translação não pode ser representada por uma matriz 2x2

$$x' = x + t_x$$
$$y' = y + t_y$$

Coordenadas Homogêneas (em 2D)

Adicionar mais uma coordenada (coordenada w)

Ponto 2D : $(\mathbf{x}, \mathbf{y}, \mathbf{1})^T$

Vetor 2D : $(\mathbf{x}, \mathbf{y}, \mathbf{0})^T$

A matriz para translação então fica:

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix}$$

Coordenadas Homogêneas (2D)

Ideia: representa pontos 2D com 3 valores (coordenadas homogêneas)

Podemos usar a descrição 2D-H para coordenadas homogêneas em 2D

Logo as transformações são representadas por matrizes 3x3

Para se recuperar as coordenadas 2D de um ponto, basta dividir por w

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow \begin{bmatrix} x/w \\ y/w \end{bmatrix}$$

Insper

Propriedades das Coordenadas Homogêneas

Operações em coordenadas homogêneas:
(válidas se a coordenada w do resultado for 1 ou 0)

vetor	**+ vetor**	**= vetor**
ponto	**– ponto**	**= vetor**
ponto	**+ vetor**	**= ponto**
ponto	**+ ponto**	**= ??**

Transformações Afim em Coordenadas Homogêneas

transformação afim = transformação linear + translação

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

usando coordenadas homogêneas:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Exemplos com Matrizes Homogêneas

Para transformações que já eram lineares, não muda muito:

$$\mathbf{S_s} = \begin{bmatrix} \mathbf{S}_x & 0 & 0 \\ 0 & \mathbf{S}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note nestes casos que a terceira coluna e linha, não fazem nada

$$\mathbf{S_s} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{S}_x x \\ \mathbf{S}_y y \\ 1 \end{bmatrix}$$

Se divide pela última coordenada, para ter as coordenadas finais

$$\begin{bmatrix} \mathbf{S}_x x \\ \mathbf{S}_y y \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{S}_x x/1 \\ \mathbf{S}_y y/1 \end{bmatrix} = \begin{bmatrix} \mathbf{S}_x x \\ \mathbf{S}_y y \end{bmatrix}$$

Translação em Coordenadas Homogêneas

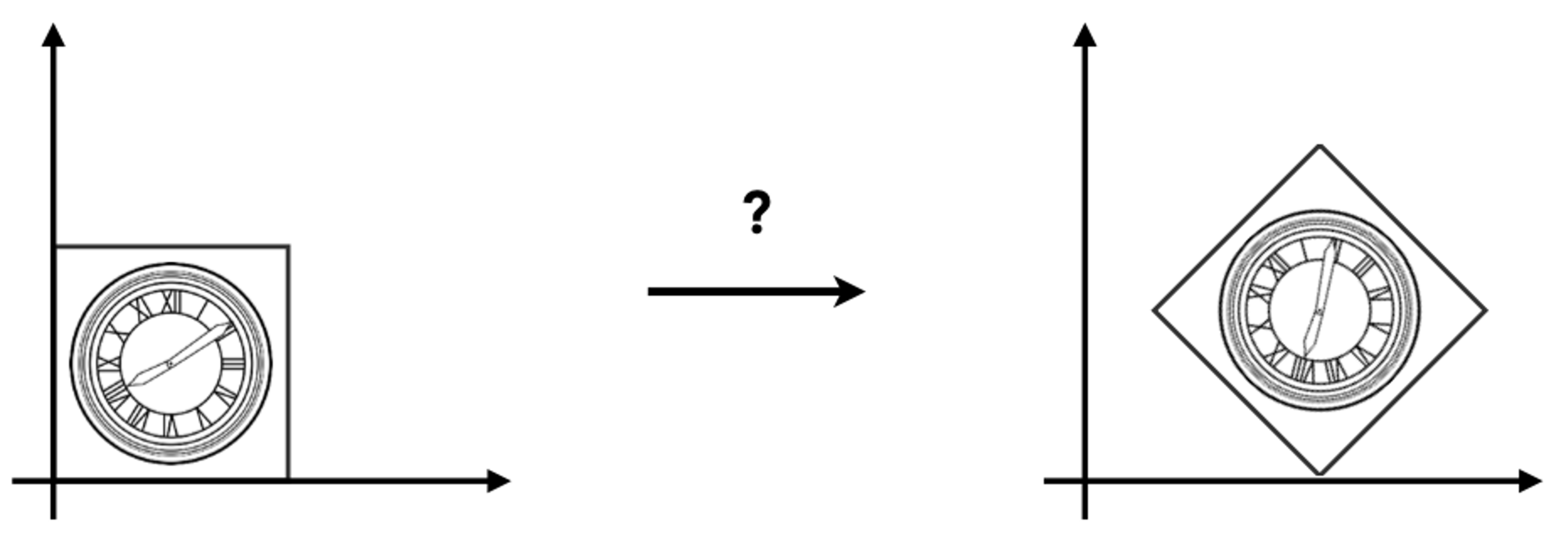
Translação representada por uma matriz 3x3

$$\mathbf{T_b} = \begin{bmatrix} 1 & 0 & \mathbf{b}_x \\ 0 & 1 & \mathbf{b}_y \\ 0 & 0 & 1 \end{bmatrix}$$

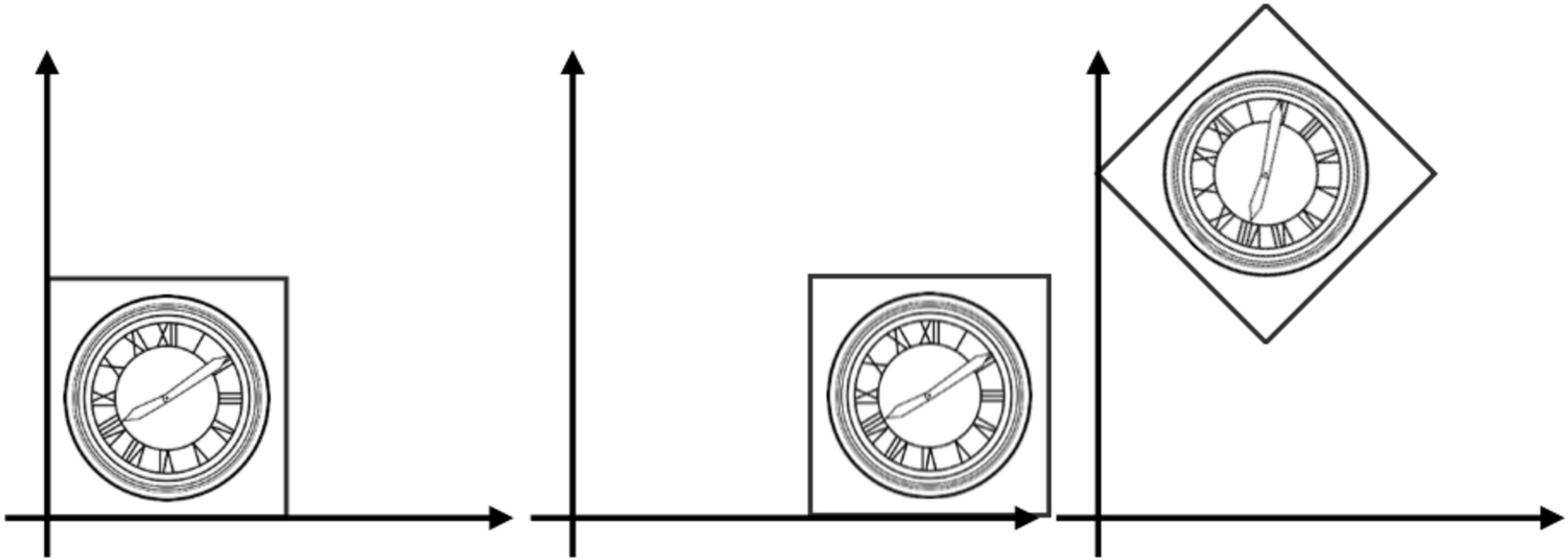
$$\mathbf{T_b x} = \begin{bmatrix} 1 & 0 & \mathbf{b}_x \\ 0 & 1 & \mathbf{b}_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_x \\ \mathbf{x}_y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_x + \mathbf{b}_x \\ \mathbf{x}_y + \mathbf{b}_y \\ 1 \end{bmatrix}$$

Coordenadas homogêneas nos permitem codificar as translações como transformações lineares!

Composição de Transformações



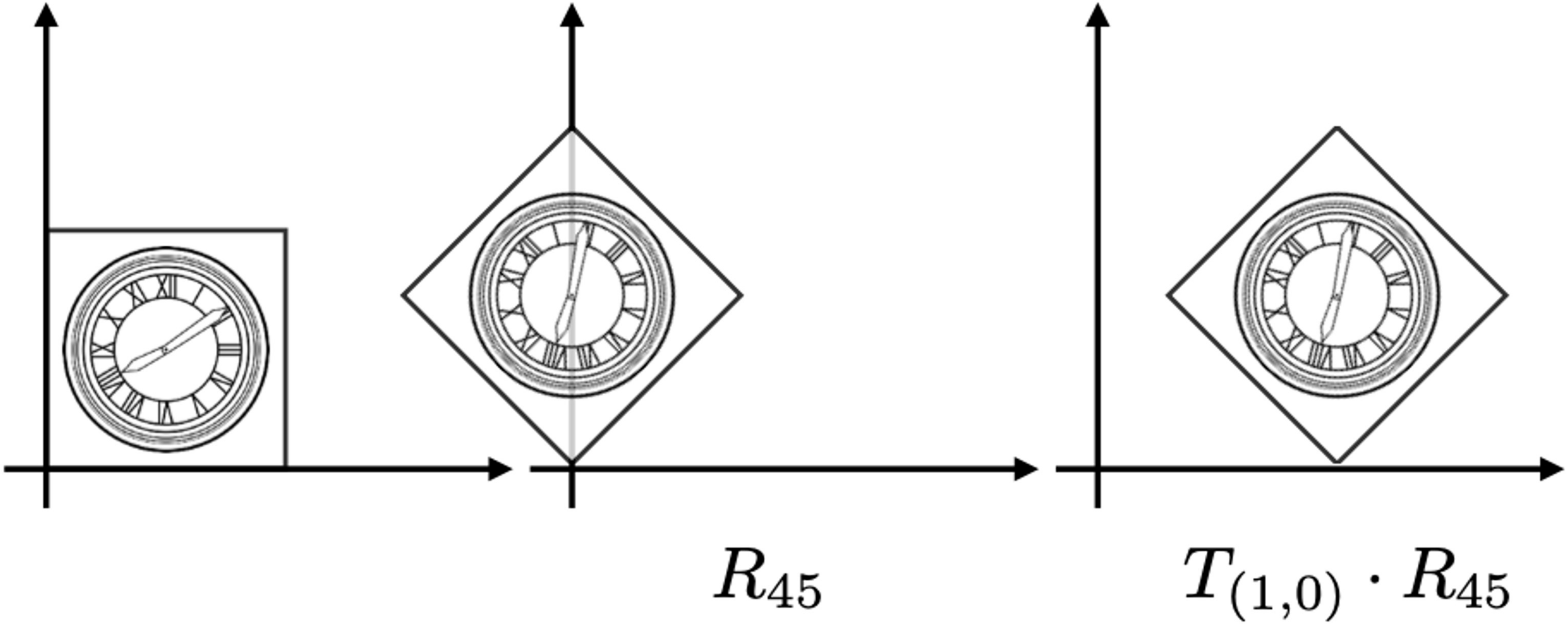
Translade e Rotacione



$$T_{(1,0)} \qquad R_{45} \cdot T_{(1,0)}$$

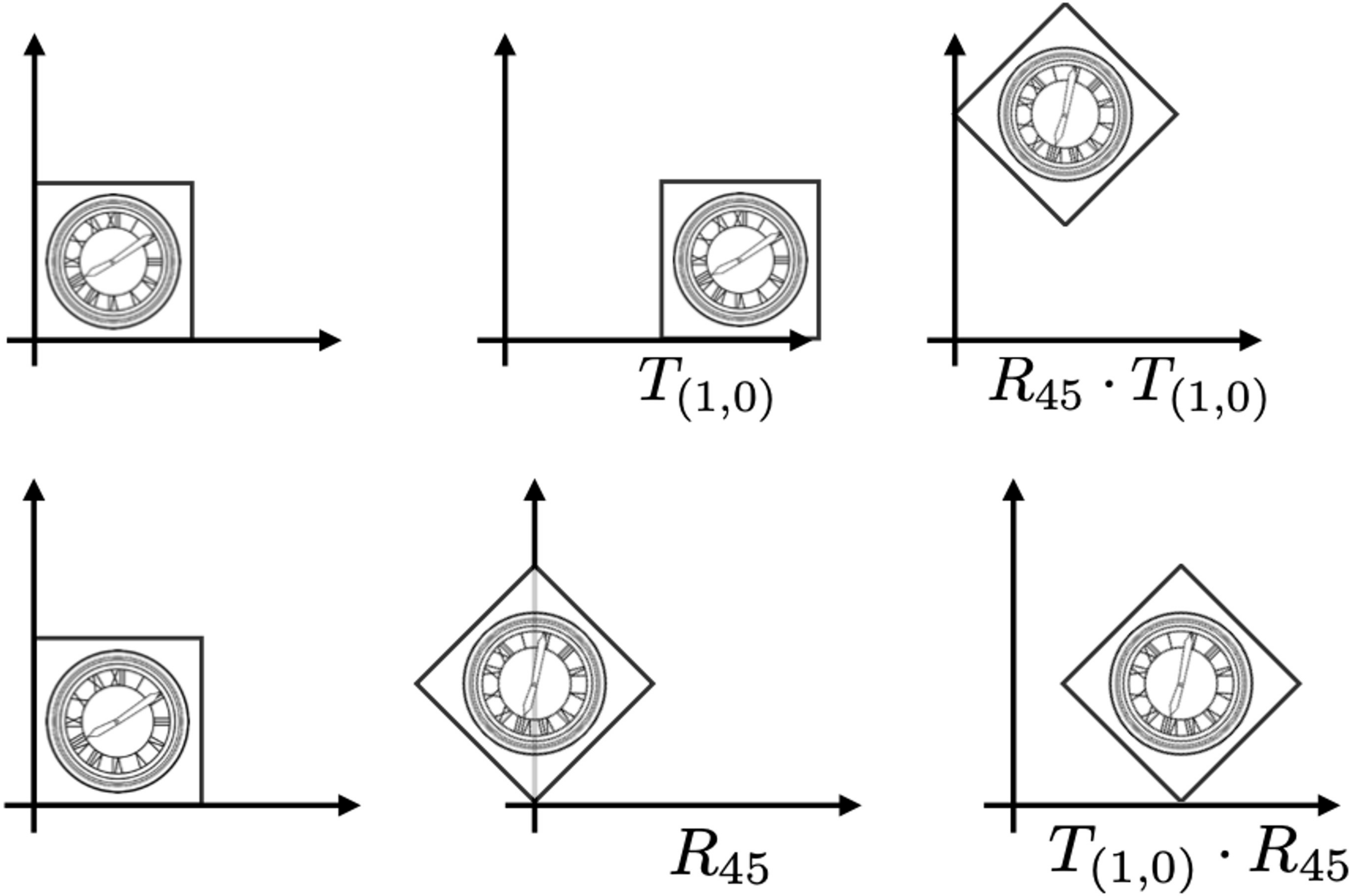
Insper

Rotacione e Translade



Insper

A ordem importa!



Insper

Ordem das Transformações

Multiplicação de matrizes não é comutativa.

$$R_{45} \cdot T_{(1,0)} \neq T_{(1,0)} \cdot R_{45}$$

Ou seja:

$$\begin{bmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \neq \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Perceba que as matrizes são aplicadas da direita para a esquerda

$$T_{(1,0)} \cdot R_{45} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Compondo as Transformações

Sequência de transformações afins A1, A2, A3, ...

Compostas por multiplicações de matrizes

Importante para melhorar desempenho na execução

$$A_n(\ldots A_2(A_1(\mathbf{x}))) = \underbrace{\mathbf{A}_n \cdots \mathbf{A}_2 \cdot \mathbf{A}_1}_{} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

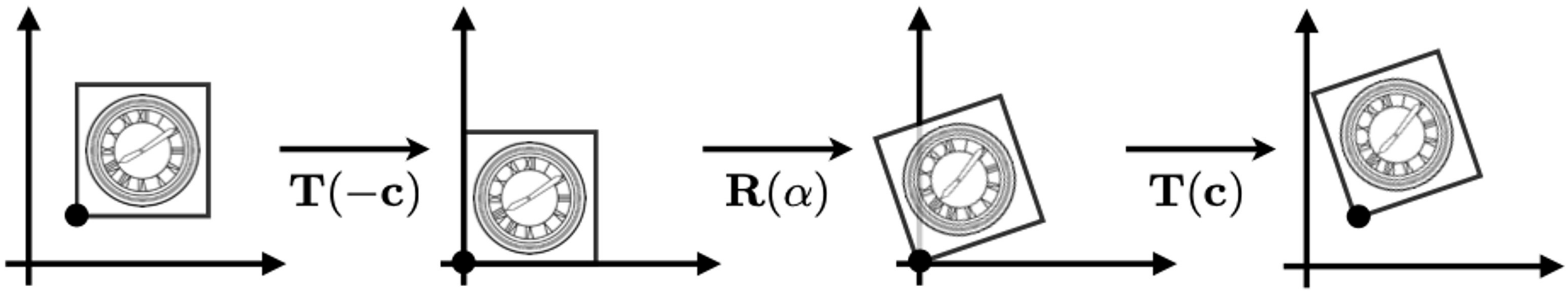
Pré-multiplique as n matrizes para obter uma matriz única que representa as transformações combinadas

Insper

Decompondo Transformações Complexas

Como girar em torno de um determinado ponto c?

1. Transladar o centro para origem
2. Realizar a Rotação
3. Transladar de volta



Representação Matricial:

$$\mathbf{T}(\mathbf{c}) \cdot \mathbf{R}(\alpha) \cdot \mathbf{T}(-\mathbf{c})$$

ATIVIDADE 1 e 2: Transformações Geométricas

Acesse o documento

Realize todos os exercícios.

Voltamos em 30 minutos?

Vídeo/Leitura para realizar antes da próxima aula

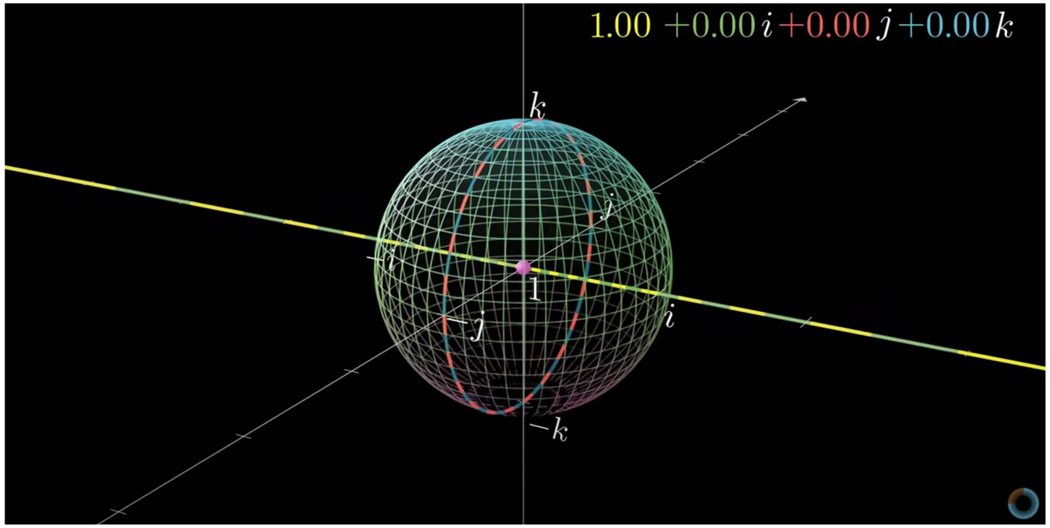
Ver os vídeos:

https://www.youtube.com/watch?v=d4EgbgTm0Bg

[Quaternions and 3d rotation, explained interactively](https://www.youtube.com/watch?v=d4EgbgTm0Bg)

Ler/Ver/Estudar:

https://eater.net/quaternions



Insper

Projeto 1

Estúdio, aproveite esse momento para implementar o seu projeto.

Computação Gráfica

Luciano Soares
<lpsoares@insper.edu.br>