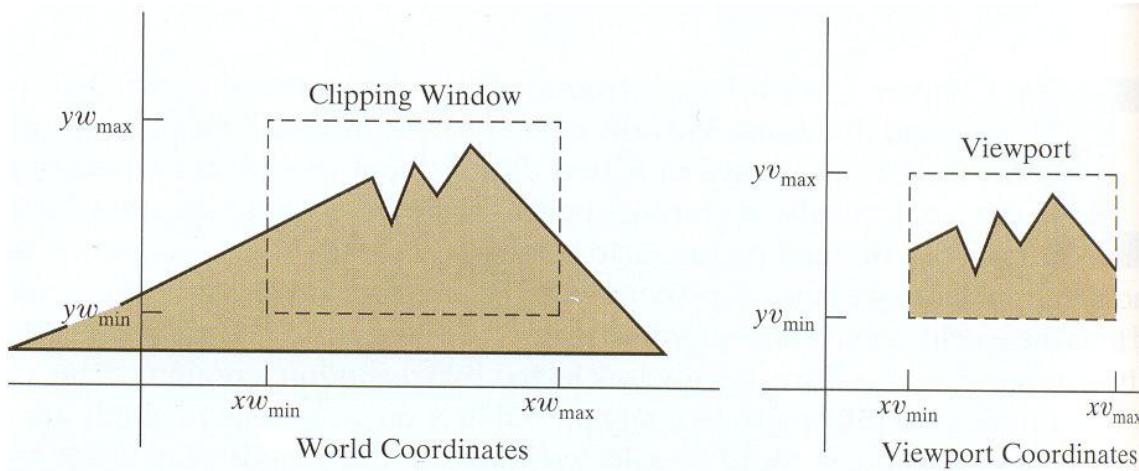


## RESUMO CV

Envolvem também procedimentos para remover as partes da cena que não foram seleccionadas – *clipping*



- Para transformar os **pontos** da janela em pontos do *viewport* mantemos as **proporções**:

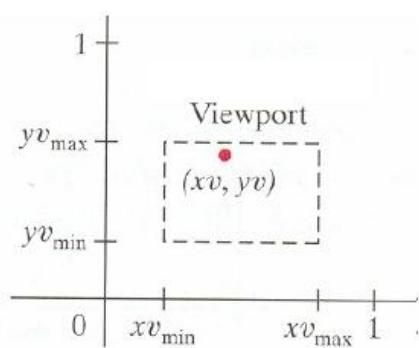
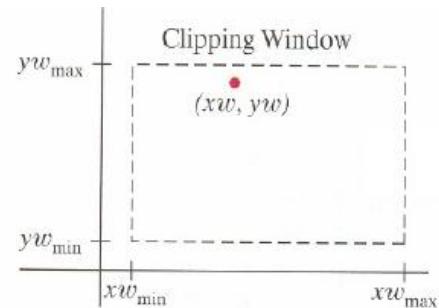
$$\frac{xv - xv_{\min}}{xv_{\max} - xv_{\min}} = \frac{xw - xw_{\min}}{xw_{\max} - xw_{\min}}$$

$$\frac{yv - yv_{\min}}{yv_{\max} - yv_{\min}} = \frac{yw - yw_{\min}}{yw_{\max} - yw_{\min}}$$

- Resolvendo estas expressões, temos:

$$xv = s_x xw + t_x$$

$$yv = s_y yw + t_y$$



- Os factores de escala são:

$$sx = \frac{xv_{\max} - xv_{\min}}{xw_{\max} - xw_{\min}}$$

$$sy = \frac{yv_{\max} - yv_{\min}}{yw_{\max} - yw_{\min}}$$

- E os deslocamentos são:

$$tx = \frac{xw_{\max}xv_{\min} - xw_{\min}xv_{\max}}{xw_{\max} - xw_{\min}}$$

$$ty = \frac{yw_{\max}yv_{\min} - yw_{\min}yv_{\max}}{yw_{\max} - yw_{\min}}$$

## Transformações 2D básicas

$p = (x, y) \rightarrow$  ponto original

$$\mathbf{P} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$p' = (x', y') \rightarrow$  ponto transformado

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

- As transformações básicas são:

- Translação

Nota: alguns livros e packages gráficos mais antigos usam uma notação em que cada ponto é um vector linha e não um vector coluna:  $\mathbf{P} = [x \ y]$

- Mudança de escala

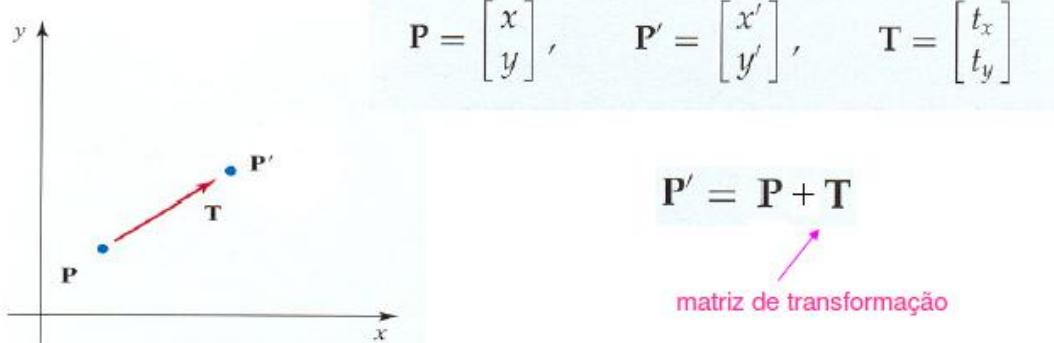
- Rotação

Notação vectorial

## Translação

- Para fazer uma translação é necessário especificar deslocamentos em  $x$  e  $y$

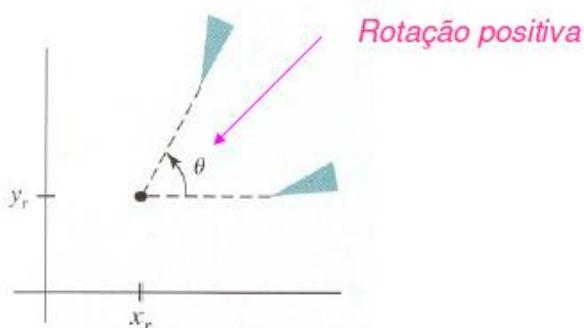
$$x' = x + t_x, \quad y' = y + t_y$$



Para aplicar uma translação a um polígono, transformam-se os vértices

## Rotação

- Para efectuar uma rotação é necessário especificar:
  - um ponto em torno do qual se vai efectuar a rotação  $(x_r, y_r)$  ( intersecção entre o eixo de rotação e o plano  $xy$  )
  - um ângulo de rotação  $\theta$  (positivo no sentido anti-horário)



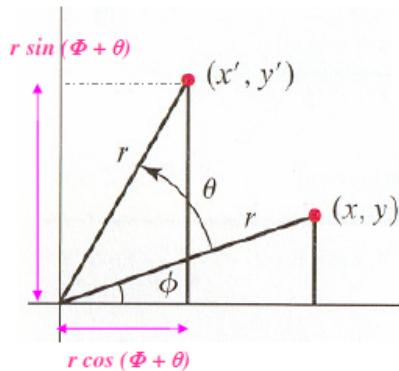
## Rotação em torno da origem

- Por facilidade, calculemos em 1º lugar a matriz de transformação correspondente a uma rotação em torno da origem:

$$x' = r \cos(\phi + \theta) = r \cos \phi \cos \theta - r \sin \phi \sin \theta$$

$$y' = r \sin(\phi + \theta) = r \cos \phi \sin \theta + r \sin \phi \cos \theta$$

As coordenadas originais do ponto em coordenadas polares são:



$$x = r \cos \phi$$

$$y = r \sin \phi$$

Substituindo nas expressões anteriores temos:

$$x' = x \cos \theta - y \sin \theta$$

TPC: verificar

$$y' = x \sin \theta + y \cos \theta$$

## Rotação em torno da origem

$$x' = r \cos(\phi + \theta) = r \cos \phi \cos \theta - r \sin \phi \sin \theta$$

$$y' = r \sin(\phi + \theta) = r \cos \phi \sin \theta + r \sin \phi \cos \theta$$

$$x = r \cos \phi, \quad y = r \sin \phi$$

Nota: quando se usa a notação em que cada ponto é uma matriz linha, altera-se a ordem da multiplicação de matrizes e as matrizes de transformação são as transpostas:  $P' = P \cdot R^T$

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

$$P' = R \cdot P$$

com

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

TPC: verificar

↑  
Matriz de rotação de um ângulo  $\theta$  em torno da origem

## Rotação em torno de um ponto arbitrário

- Usando a relação trigonométrica indicada pelos dois triângulos da figura podemos generalizar a expressão das coordenadas  $x', y'$

$$x' = x_r + (x - x_r) \cos \theta - (y - y_r) \sin \theta$$

$$y' = y_r + (x - x_r) \sin \theta - (y - y_r) \cos \theta$$

Nota: um método alternativo consiste em considerar esta transformação como composta de transformações elementares (ver mais à frente)

Nota: Obtenção do valor de  $x'$

$$x' = r \cos(\theta + \phi) + x_r$$

$$y' = r \sin(\theta + \phi) + y_r$$

$$x = r \cos \phi + x_r$$

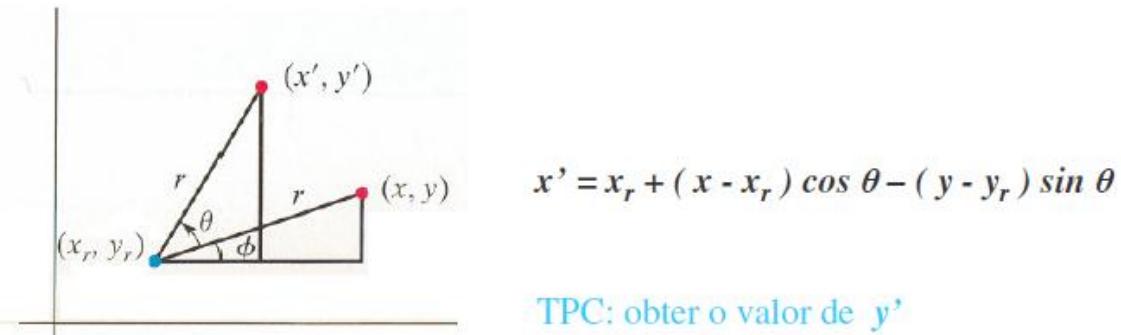
$$y = r \sin \phi + y_r$$

$$x' = r \cos \theta \cos \phi - r \sin \theta \sin \phi + x_r$$

$$y' = r \cos \theta \sin \phi + r \sin \theta \cos \phi + y_r$$

exprimindo em função de  $x$  e  $y$ :

$$x' = (x - x_r) \cos \theta - (y - y_r) \sin \theta + x_r$$



## Mudança de escala em torno da origem

- Para modificar o tamanho de um objecto aplica-se uma mudança de escala, sendo  $s_x$  e  $s_y$  factores de escala:

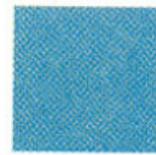
$$x' = x \cdot s_x$$

$$y' = y \cdot s_y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

matriz de transformação

$$P' = S \cdot P$$



Transformação de um quadrado num quadrado maior através de uma mudança de escala,  $s_x=2, s_y=2$

## Mudança de escala

- Podem-se usar valores de  $s > 0$

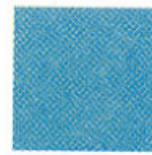
$$x' = x \cdot s_x$$

$$y' = y \cdot s_y$$

$s_x = s_y \rightarrow$  mudança de escala uniforme

$s_x \neq s_y \rightarrow$  mudança de escala não uniforme

Transformação de um quadrado num quadrado maior através de uma mudança de escala,  $s_x=2, s_y=2$

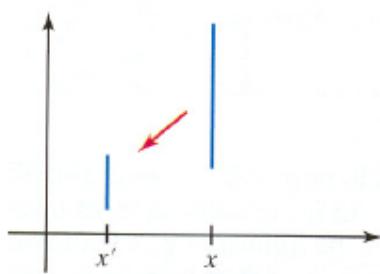


Transformação de um quadrado num rectângulo através de uma mudança de escala,  $s_x=2, s_y=1$

- Os objectos transformados pela transformação S são também reposicionados:

$s < 1 \rightarrow$  objecto **aproxima-se** da origem

$s > 1 \rightarrow$  objecto **afasta-se** da origem



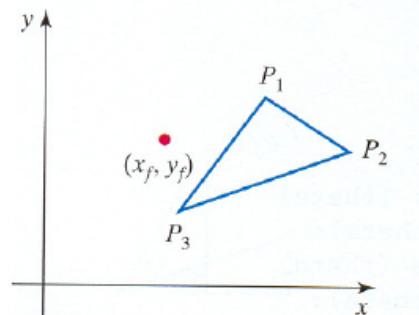
Um segmento de recta diminui em tamanho e aproxima-se da origem quando se aplica uma mudança de escala  $s_x = s_y = 0,5$

### Mudança de escala com um ponto fixo

- Mas podemos controlar a posição de um objecto, escolhendo um **ponto fixo**,  $(x_f, y_f)$  que permanece inalterado:

$$x' - x_f = (x - x_f) \cdot s_x$$

$$y' - y_f = (y - y_f) \cdot s_y$$



Mudança de escala relativa a um ponto fixo  $(x_f, y_f)$

$$x' = x \cdot s_x + x_f(1 - s_x)$$

$$y' = y \cdot s_y + y_f(1 - s_y)$$

Const. para todos os pontos do objecto

Nota: é possível realizar esta transformação e a rotação em torno de um ponto arbitrário na forma matricial multiplicando cada ponto apenas por uma matriz como veremos mais à frente

- Vimos que as três transformações básicas 2D podem ser expressas na forma geral:

$$\mathbf{P}' = \mathbf{M}_1 \cdot \mathbf{P} + \mathbf{M}_2$$

$\mathbf{M}_1$  é uma matriz  $2 \times 2$  contendo factores multiplicativos

$\mathbf{M}_2$  é uma matriz coluna com dois elementos: os dois termos de translação

- Uma abordagem mais eficiente consiste em ter uma única matriz que:
  - represente todas as transformações de uma sequência
  - possa ser aplicada uma única vez a cada ponto
- Isto consegue-se usando **coordenadas homogéneas**
- É possível combinar numa única matriz os termos multiplicativos e os aditivos se utilizarmos matrizes de  $3 \times 3$
- Todas as transformações podem ser representadas por multiplicações de matrizes
- Podemos usar a terceira coluna para os termos aditivos
- É necessário passar a representar cada ponto por três coordenadas:

$$(x, y) \rightarrow (x_h, y_h, h), \quad h \neq 0$$

$$x = x_h / h \quad y = y_h / h$$

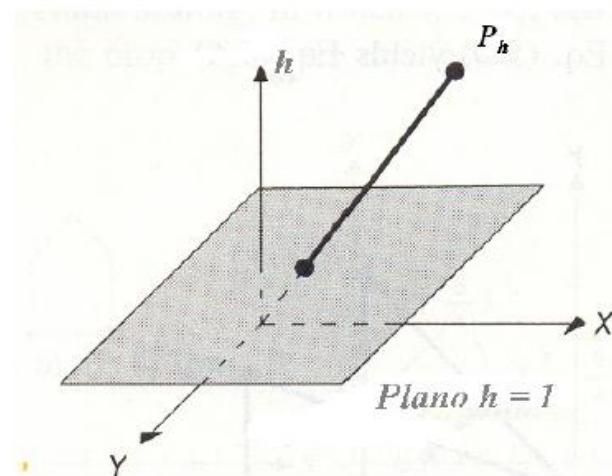
$$(x.h, y.h, h)$$

- Uma escolha que facilita é:

*plano  $h = 1$*

- O que resulta em:

$$(x, y) \rightarrow (x, y, 1)$$



- Existe uma infinidade de pontos  $P_h$  no espaço tridimensional das coordenadas homogéneas que correspondem a um único ponto  $(x, y)$

## Transformações 2D em coordenadas homogéneas

- Quando se usam coordenadas homogéneas podemos realizar todas as transformações geométricas por multiplicação de matrizes
- Translação 2D:

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

$$\mathbf{P}' = T(t_x, t_y) \cdot \mathbf{P}$$

TPC: verificar

- Rotação 2D:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{R}(\theta) \cdot \mathbf{P}$$

- Mudança de escala 2D:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{S}(s_x, s_y) \cdot \mathbf{P}$$

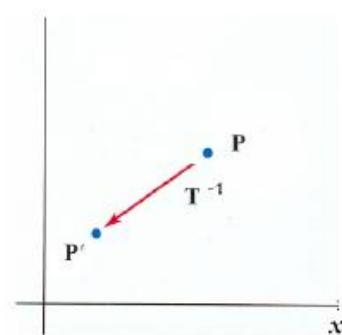
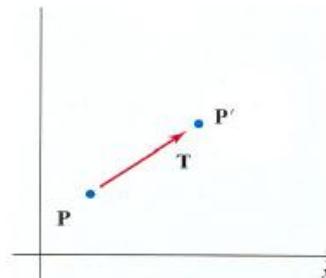
## Transformações inversas

- A transformação inversa de uma translação:

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

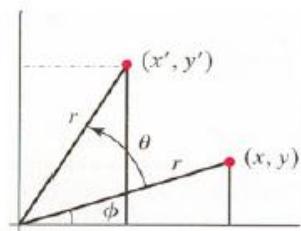
é uma translação com parâmetros simétricos em  $x$  e  $y$ :

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix}$$



- Uma rotação inversa obtém-se substituindo o ângulo de rotação pelo seu simétrico:

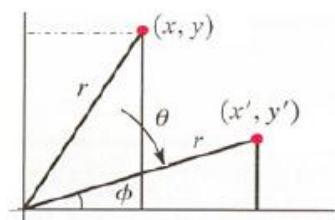
$$\mathbf{R}^{-1} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- Podemos calcular a inversa de qualquer matriz de rotação calculando a transposta;

$$(\mathbf{R}^{-1} = \mathbf{R}^T)$$

- Isto deve-se ao facto de apenas o seno ser afectado pela mudança de sinal do ângulo



TPC: verificar

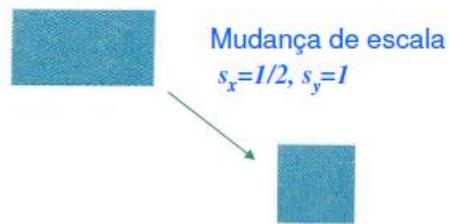
- Uma mudança de escala inversa obtém-se substituindo os factores de escala pelos seus recíprocos:

$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- O produto de uma **qualquer matriz M** correspondente a uma transformação pela matriz correspondente à sua transformação inversa produz uma matriz identidade:

$$\mathbf{M} \cdot \mathbf{M}^{-1} = \mathbf{I}$$



## Transformações compostas

- Usando a representação matricial podemos gerar uma matriz correspondente a uma sequência de transformações calculando o produto das transformações individuais
- O produto de matrizes de transformação chama-se **concatenação** ou **composição de matrizes**
- A concatenação de duas matrizes calcula-se como:

$$P' = M_2 \cdot M_1 \cdot P$$

Segunda transformação a ser aplicada      Primeira transformação a ser aplicada

Nota: A aplicação das matrizes de transformação faz-se multiplicando da direita para a esquerda (nesta notação)

- As coordenadas do ponto transformado  $P'$  são calculadas através da multiplicação de uma única matriz

## Composição de duas translações sucessivas

$$\begin{aligned} P' &= T(t_{2x}, t_{2y}) \cdot \{T(t_{1x}, t_{1y}) \cdot P\} \\ &= \{T(t_{2x}, t_{2y}) \cdot T(t_{1x}, t_{1y})\} \cdot P \end{aligned}$$

$$\begin{bmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{1x} + t_{2x} \\ 0 & 1 & t_{1y} + t_{2y} \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(t_{2x}, t_{2y}) \cdot T(t_{1x}, t_{1y}) = T(t_{1x} + t_{2x}, t_{1y} + t_{2y})$$

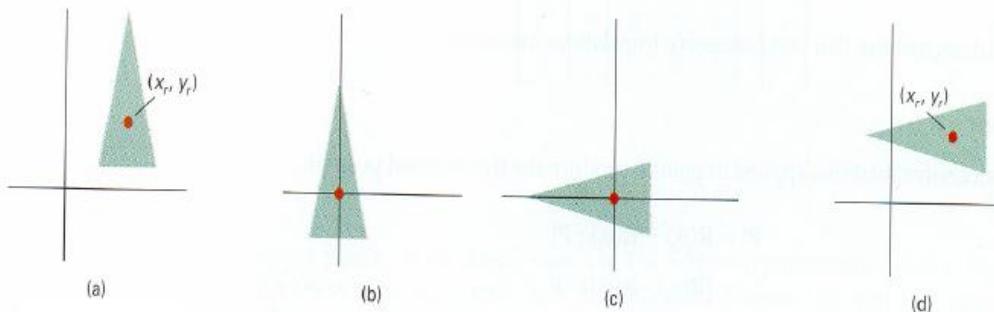
TPC: verificar

## Composição de duas mudanças de escala sucessivas

$$\begin{bmatrix} s_{2x} & 0 & 0 \\ 0 & s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{1x} & 0 & 0 \\ 0 & s_{1y} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{1x} \cdot s_{2x} & 0 & 0 \\ 0 & s_{1y} \cdot s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}(s_{2x}, s_{2y}) \cdot \mathbf{S}(s_{1x}, s_{1y}) = \mathbf{S}(s_{1x} \cdot s_{2x}, s_{1y} \cdot s_{2y})$$

Rotação em torno de um ponto arbitrário  $(x_r, y_r)$



Posição original  
do objecto e do  
ponto  $(x_r, y_r)$

1- Aplicar a  
translação que  
leva  $(x_r, y_r)$  para  
a origem

2- Rodar em  
torno da origem

3- Aplicar translação por  
forma a levar o ponto  
arbitrário a  $(x_r, y_r)$

## Rotação em torno de um ponto arbitrário

- É necessário aplicar uma:

1 - uma translação por forma a que o ponto arbitrário coincida com a origem

2 - uma rotação em torno da origem

3 - uma translação por forma a levar o ponto arbitrário para a posição original

3- Translação do ponto arbitrário para a posição original

$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix}$$

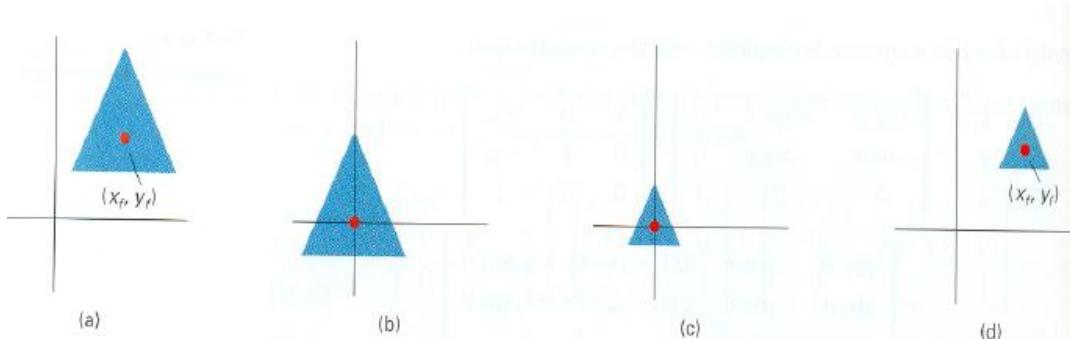
$$= \begin{bmatrix} \cos \theta & -\sin \theta & x_r(1 - \cos \theta) + y_r \sin \theta \\ \sin \theta & \cos \theta & y_r(1 - \cos \theta) - x_r \sin \theta \\ 0 & 0 & 1 \end{bmatrix}$$

1- Ponto arbitrário para a origem

TPC: verificar

$$\mathbf{T}(x_r, y_r) \cdot \mathbf{R}(\theta) \cdot \mathbf{T}(-x_r, -y_r) = \mathbf{R}(x_r, y_r, \theta)$$

Mudança de escala mantendo um ponto fixo ( $x_f, y_f$ )



Posição original do objecto e do ponto ( $x_f, y_f$ )

1- Aplicar a translação que leva ( $x_f, y_f$ ) para a origem

2- Aplicar a mudança de escala

3- Aplicar translação por forma a levar o ponto fixo a ( $x_f, y_f$ )

## Propriedades de concatenação de matrizes

- A multiplicação de matrizes é **associativa**
- Dadas quaisquer três matrizes  $M_1, M_2, M_3$ , o seu produto pode ser obtido multiplicando em primeiro lugar  $M_3$  por  $M_2$ , ou multiplicando primeiro  $M_2$  e  $M_1$

$$M_3 \cdot M_2 \cdot M_1 = (M_3 \cdot M_2) \cdot M_1 = M_3 \cdot (M_2 \cdot M_1)$$

- Mas os produtos de transformações em geral **não são comutativos**:

$$M_2 \cdot M_1 \neq M_1 \cdot M_2$$

- Por exemplo, quando se pretender aplicar uma rotação e uma translação a um objecto, é necessário ter cuidado com a ordem em que se aplicam as transformações

## Concatenação de transformações e eficiência computacional

- Uma transformação bidimensional representando uma combinação de transformações (translações, rotações, mudanças de escala) pode ser representada como:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} rs_{xx} & rs_{xy} & trs_x \\ rs_{yx} & rs_{yy} & trs_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

*termos multiplicativos correspondentes às rotações e mudanças de escala*

*termos correspondentes a distâncias de translação e centro de rotação ou ponto fixo na mudança de escala*

- Por exemplo: se um objecto tiver que sofrer uma mudança de escala e uma rotação em torno do seu centróide ( $x_c, y_c$ ) e depois sofrer uma translação, temos:

$$\mathbf{T}(t_x, t_y) \cdot \mathbf{R}(x_c, y_c, \theta) \cdot \mathbf{S}(x_c, y_c, s_x, s_y)$$

$$= \begin{bmatrix} s_x \cos \theta & -s_y \sin \theta & x_c(1 - s_x \cos \theta) + y_c s_y \sin \theta + t_x \\ s_x \sin \theta & s_y \cos \theta & y_c(1 - s_y \cos \theta) - x_c s_x \sin \theta + t_y \\ 0 & 0 & 1 \end{bmatrix}$$

- As coordenadas transformadas são dadas por:

$$x' = x \cdot r s_{xx} + y \cdot r s_{xy} + t_r s_x, \quad y' = x \cdot r s_{yx} + y \cdot r s_{yy} + t_r s_y$$

- Assim, para qualquer sequência de transformações, depois de concatenadas numa única matriz, são necessárias apenas:
  - 4 multiplicações
  - 4 adições
- Se não se calcular uma única matriz, cada uma das transformações tem que ser aplicada individualmente, sendo o número de operações muito superior
- Deve-se, portanto, usar uma **única matriz de transformação** resultante das várias transformações a aplicar
- Note-se que os coeficientes dessa matriz apenas têm que ser calculados uma única vez

- Os problemas de eficiência são importantes nas rotações pois estas utilizam transformações trigonométricas e várias multiplicações para transformar cada ponto.
- Em animação e outras aplicações que envolvem muitas rotações de ângulo pequeno, podem usar-se **aproximações para aumentar a eficiência** computacional
- Para ângulos  $< 10^\circ$ ,
$$\cos \theta \sim 1$$

$$\sin \theta \sim \theta \text{ (em rad)}$$
- Podemos calcular  $(x', y')$ 

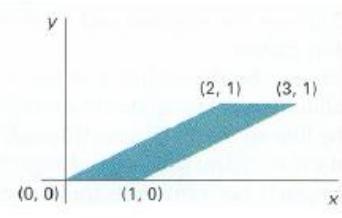
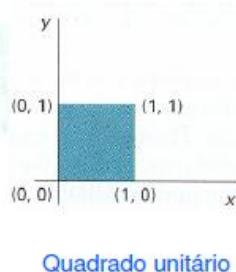
$$x' = x - y \sin \theta, \quad y' = x \sin \theta + y$$
- Mas o **erro acumulado** pode tornar-se muito grande

## Outras transformações

- Além das transformações básicas (translação, rotação e mudança de escala) alguns *packages* gráficos fornecem outras transformações que podem ser úteis, como:
  - reflexão
  - *shearing*
- *Shearing* na direcção  $x$ :

$$x' = x + sh_x \cdot y, \quad y' = y$$

$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



## Reflexão

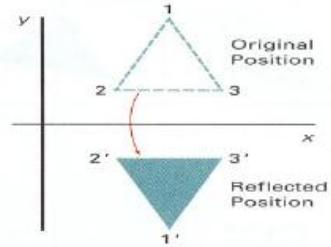
- Uma reflexão é uma transformação que produz “uma imagem num espelho”
- Uma reflexão em torno do eixo dos  $xx$  ( $y=0$ ) obtém-se multiplicando por -1 as coordenadas  $y$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

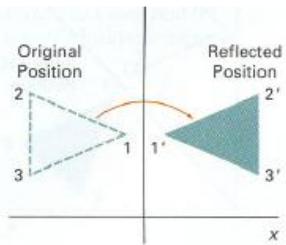
- Uma reflexão em torno do eixo dos  $yy$  ( $x=0$ ) obtém-se multiplicando por -1 as coordenadas  $x$

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

TPC: verificar



Reflexão em torno do eixo dos  $xx$



Reflexão em torno do eixo dos  $yy$

## Transformações Geométricas 3D

- Um ponto definido num espaço tridimensional é representado, em coordenadas homogéneas, por um vector com 4 elementos

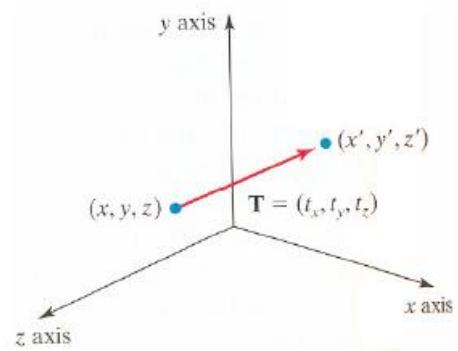
$$P = (x, y, z)$$

$$P_h = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



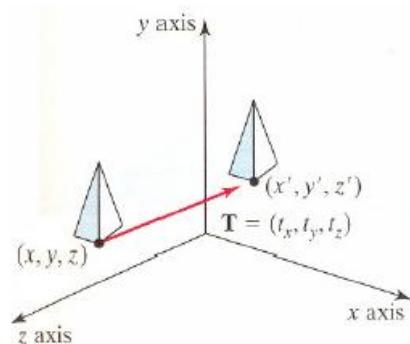
## Translação

$$x' = x + t_x \quad y' = y + t_y \quad z' = z + t_z$$



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{P}$$



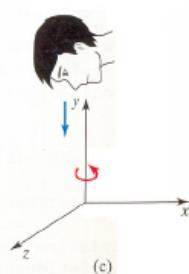
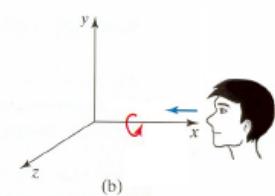
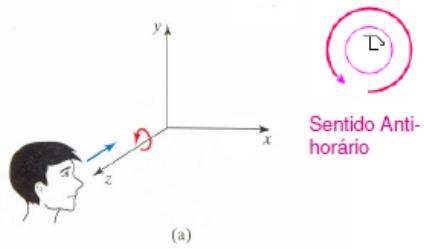
## Rotação

- Podemos rodar um objecto em torno de um eixo qualquer no espaço, mas as rotações em torno de um dos eixos coordenados são mais simples

- Por convenção:

ângulos de rotação  $> 0 \rightarrow$  rotações anti-horárias

- Isto está de acordo com a convenção estabelecida para 2D (no plano  $XOY$ , em torno da origem)



## Rotações em torno dos eixos coordenados

- A transformação correspondente à rotação 2D no plano  $XOY$ , em torno da origem, é facilmente generalizável a 3D

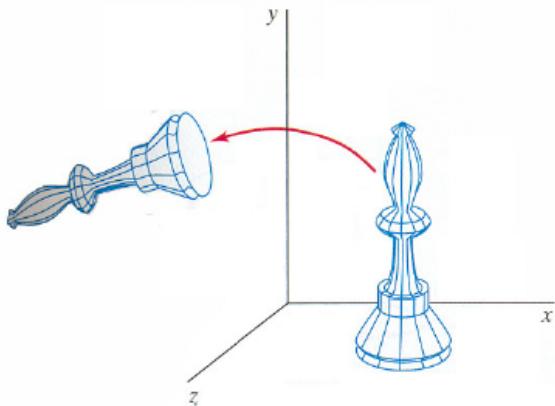
$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

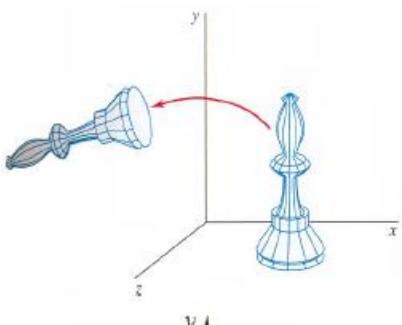
$$z' = z$$

- O parâmetro  $\theta$  especifica o ângulo de rotação em torno do eixo  $ZZ'$

- Para qualquer ponto, a coordenada  $z$  permanece inalterada



## Rotações em torno do eixo $ZZ'$



- As equações correspondentes à rotação 2D no plano  $XOY$ , em torno da origem, são facilmente generalizáveis a 3D:

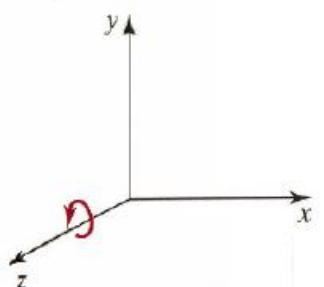
$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

$$z' = z$$

- Obtemos uma rotação de  $\theta$  em torno do eixo  $ZZ'$ :

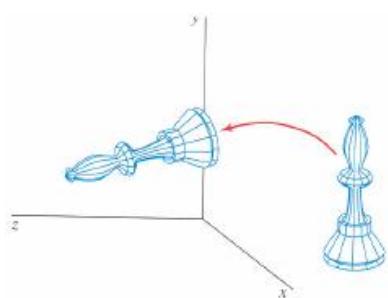
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



Nesta rotação as coordenadas  $z$  não se alteram

$$\mathbf{P}' = \mathbf{R}_z(\theta) \cdot \mathbf{P}$$

## Rotações em torno do eixo $XX'$



- As rotações em torno dos outros eixos podem-se obter através de uma permutação cíclica de  $x, y$  e  $z$ :

$$x \rightarrow y \rightarrow z \rightarrow x$$

- Assim, obtemos uma rotação de  $\theta$  em torno do eixo  $XX'$ :

$$y' = y \cos \theta - z \sin \theta$$

$$z' = y \sin \theta + z \cos \theta$$

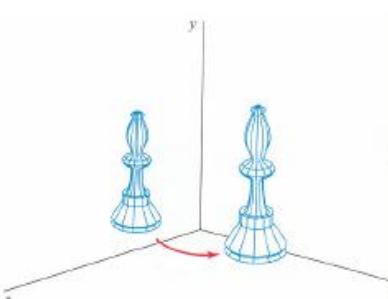
$$x' = x$$

Nesta rotação as coordenadas  $x$  Não se alteram

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

7

## Rotações em torno do eixo $YY'$



- Fazendo a mesma permutação, obtemos uma rotação de  $\theta$  em torno do eixo  $YY'$ :

$$z' = z \cos \theta - x \sin \theta$$

$$x' = z \sin \theta + x \cos \theta$$

$$y' = y$$

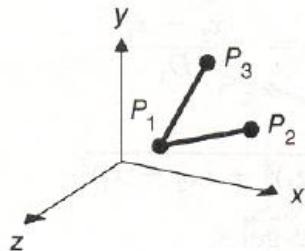
Nesta rotação as coordenadas  $y$  não se alteram

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

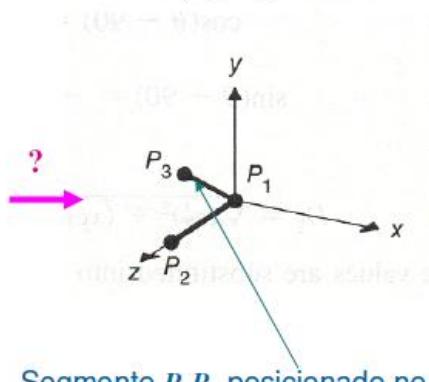
## Exemplo de uma transformação composta

- Considere-se a transformação que permite levar os segmentos de recta da posição a) à posição b)

a) posição inicial

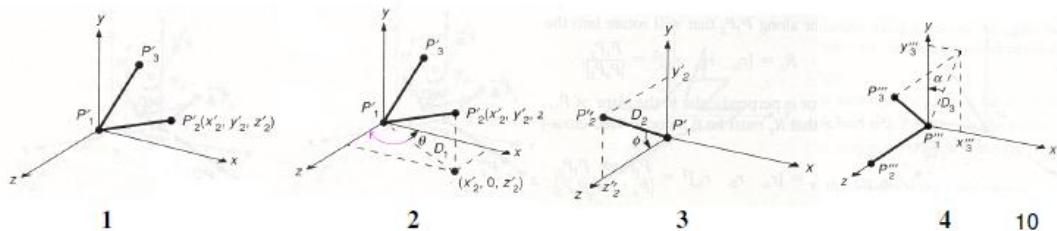
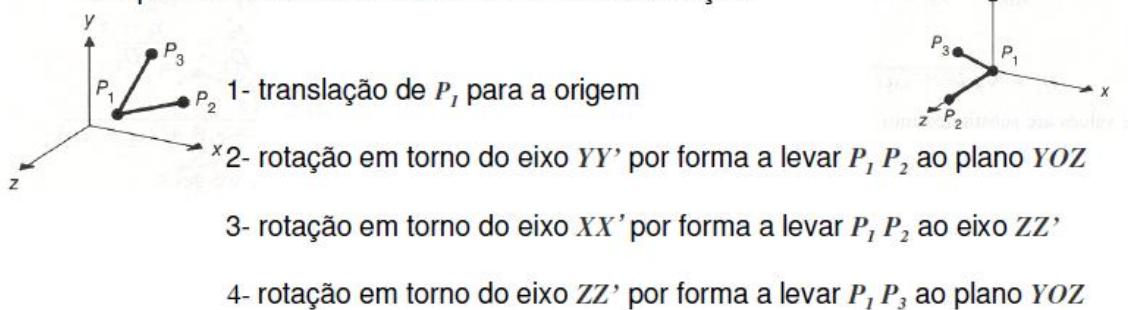


b) posição final



Segmento  $P_1P_3$  posicionado no plano YOZ

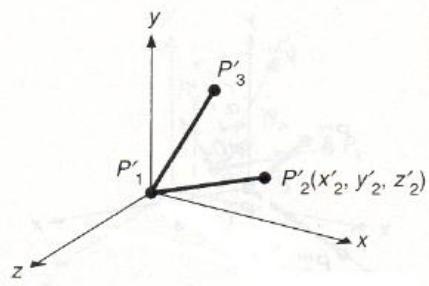
- Para calcular a matriz correspondente a esta transformação composta pode-se decompor este problema em sub-problemas mais simples para os quais se conhecem as matrizes de transformação:



1- translação de  $P_1$  para a origem:

$$T(-x_1, -y_1, -z_1) = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P'_1 = T(-x_1, -y_1, -z_1) \cdot P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



$$P'_2 = T(-x_1, -y_1, -z_1) \cdot P_2 = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \\ 1 \end{bmatrix}$$

$$P'_3 = T(-x_1, -y_1, -z_1) \cdot P_3 = \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \\ 1 \end{bmatrix}$$

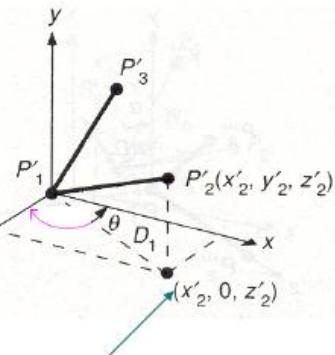
2- rotação em torno do eixo YY' por forma a levar  $P_1 P_2$  ao plano YOZ

$\hat{\text{Angulo de rotação}}: -(90 - \theta) = \theta - 90$

$$\cos(\theta - 90) = \sin\theta = \frac{z'_2}{D_1} = \frac{z_2 - z_1}{D_1}$$

$$\sin(\theta - 90) = -\cos\theta = -\frac{x'_2}{D_1} = -\frac{x_2 - x_1}{D_1}$$

$$D_1 = \sqrt{(z'_2)^2 + (x'_2)^2} = \sqrt{(z_2 - z_1)^2 + (x_2 - x_1)^2}$$

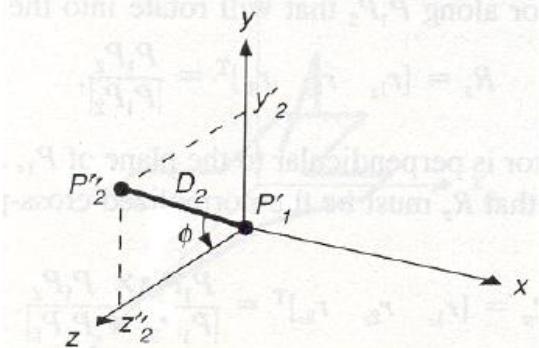


$$P''_2 = R_y(\theta - 90) \cdot P'_2 = [0 \quad y_2 - y_1 \quad D_1 \quad 1]^T$$

3- rotação em torno do eixo  $XX'$  por forma a levar  $P_1 P_2$  ao eixo  $ZZ'$

$$\cos\phi = \frac{z''_2}{D_2}, \sin\phi = \frac{y''_2}{D_2},$$

$$D_2 = |P''_1 P''_2|$$



$$D_2 = |P''_1 P''_2| = |P_1 P_2| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$\begin{aligned} P'''_2 &= R_x(\phi) \cdot P''_2 = R_x(\phi) \cdot R_y(\theta - 90) \cdot P'_2 \\ &= R_x(\phi) \cdot R_y(\theta - 90) \cdot T \cdot P_2 = [0 \quad 0 \quad |P_1 P_2| \quad 1]^T \end{aligned}$$

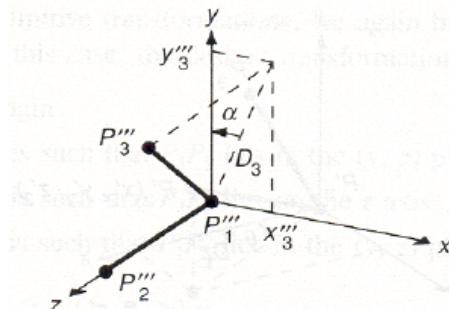
4- rotação em torno do eixo  $ZZ'$  por forma a levar  $P_1 P_3$  ao plano  $YOZ$

$$\cos\alpha = y'''_3/D_3$$

$$\sin\alpha = x'''_3/D_3$$

$$D_3 = \sqrt{(x'''_3)^2 + (y'''_3)^2}$$

A transformação composta será:



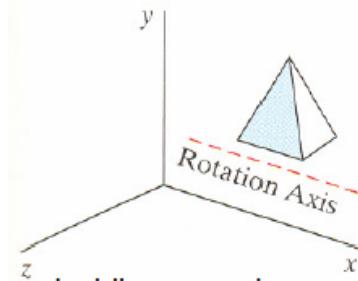
$$R_z(\alpha) \cdot R_x(\phi) \cdot R_y(\theta - 90) \cdot T(-x_1, -y_1, -z_1) = R \cdot T$$

## Rotações 3D gerais

Rotação em torno de um eixo paralelo a um eixo coordenado:

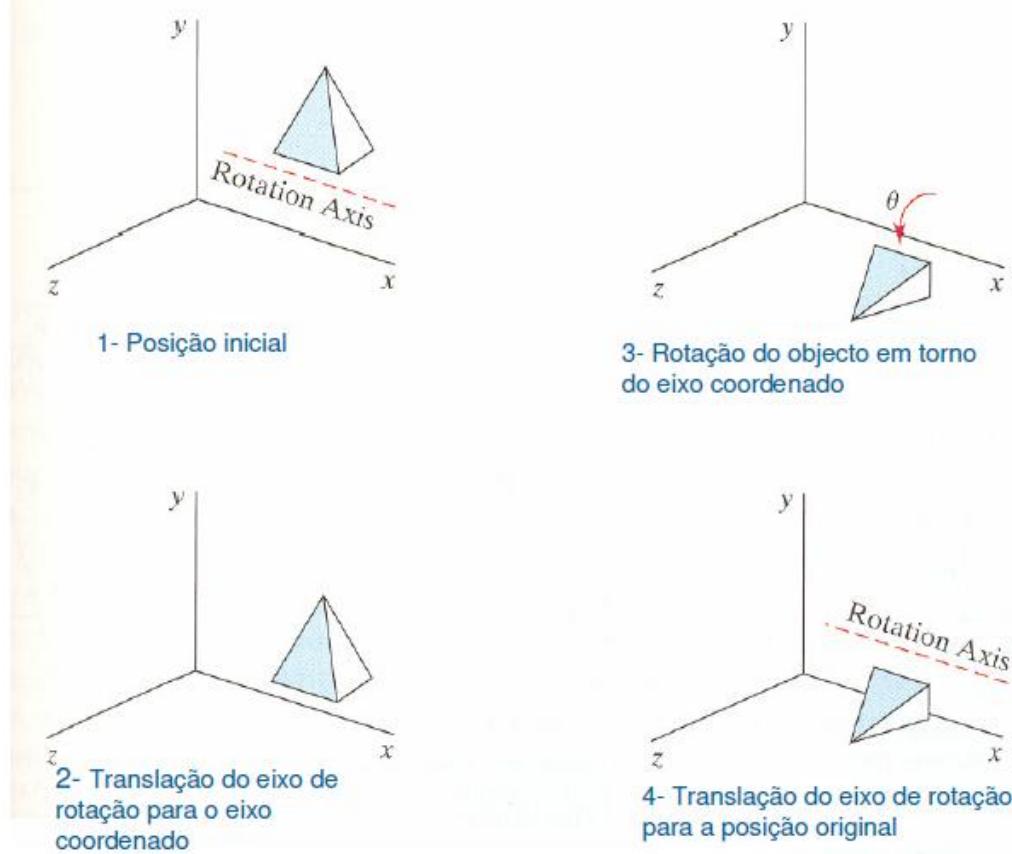
- A matriz correspondente a uma rotação em **torno de um eixo paralelo** a um eixo coordenado pode ser determinada como uma transformação resultante de:

1. Translação que leve o eixo de rotação a coincidir com o eixo coordenado
2. Rotação em torno do eixo
3. Translação inversa por forma a que o eixo de rotação volte à posição original

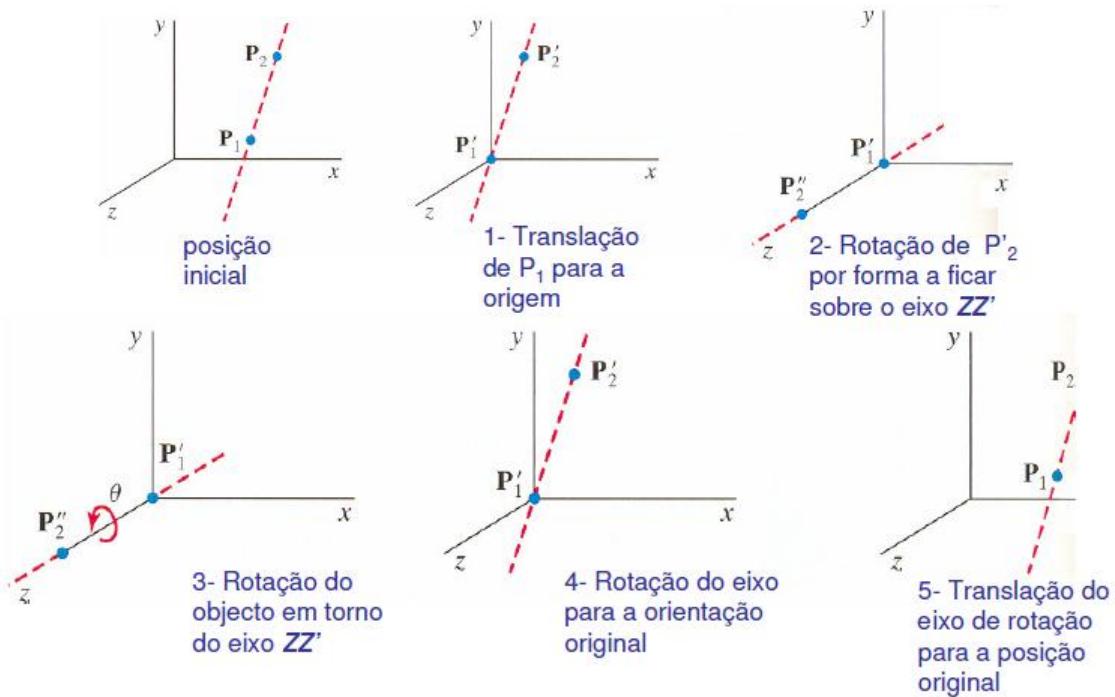


$$\mathbf{P}' = \mathbf{T}^{-1} \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T} \cdot \mathbf{P}$$

Rotação em torno de um eixo paralelo a um eixo coordenado



## Rotação em torno de um eixo arbitrário: obtenção da transformação composta



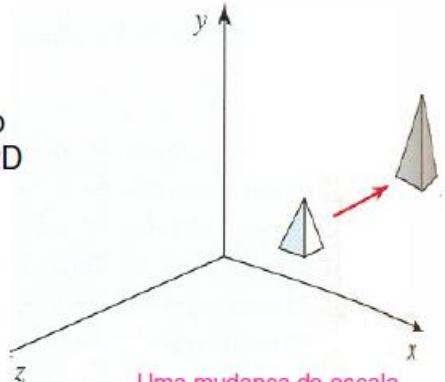
## Mudança de escala

- Uma mudança de escala tridimensional corresponde a uma matriz de transformação que se obtém por generalização da matriz 2D

$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = S \cdot P$$

$$x' = s_x \cdot x, \quad y' = s_y \cdot y, \quad z' = s_z \cdot z$$

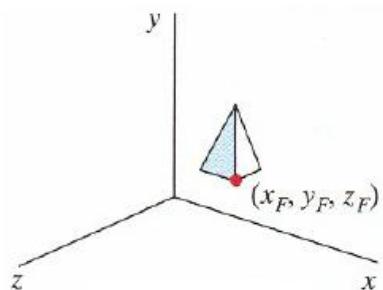


Uma mudança de escala tridimensional altera também a posição do objecto em relação à origem

## Mudança de escala em relação a um ponto fixo

- Uma mudança de escala tridimensional em relação a um ponto fixo é uma transformação composta pela seguinte sequência de transformações:

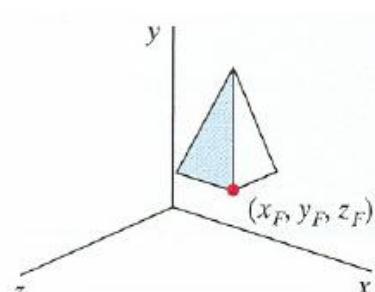
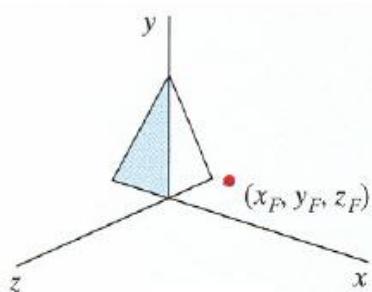
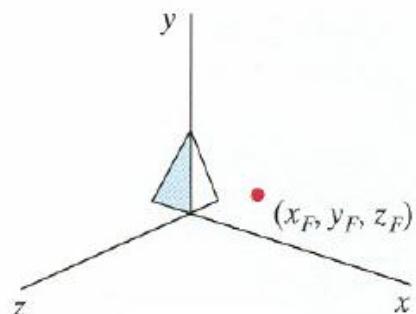
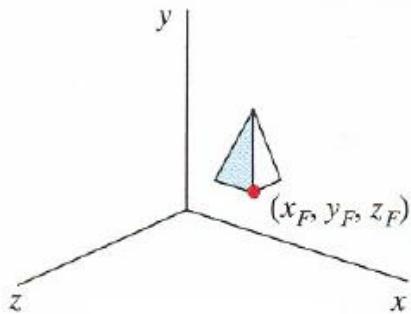
- Translação do ponto fixo para a origem
- Mudança de escala
- Translação do ponto fixo para a sua posição original



Mudança de escala tridimensional mantendo um ponto fixo  $(x_p, y_p, z_p)$

$$\mathbf{T}(x_f, y_f, z_f) \cdot \mathbf{S}(s_x, s_y, s_z) \cdot \mathbf{T}(-x_f, -y_f, -z_f) = \begin{bmatrix} s_x & 0 & 0 & (1-s_x)x_f \\ 0 & s_y & 0 & (1-s_y)y_f \\ 0 & 0 & s_z & (1-s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

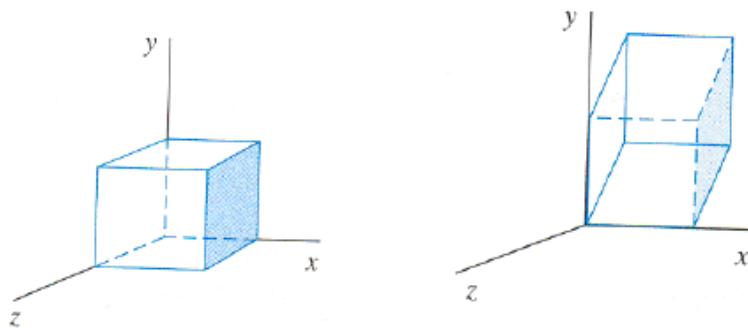
## Mudança de escala em relação a um ponto fixo



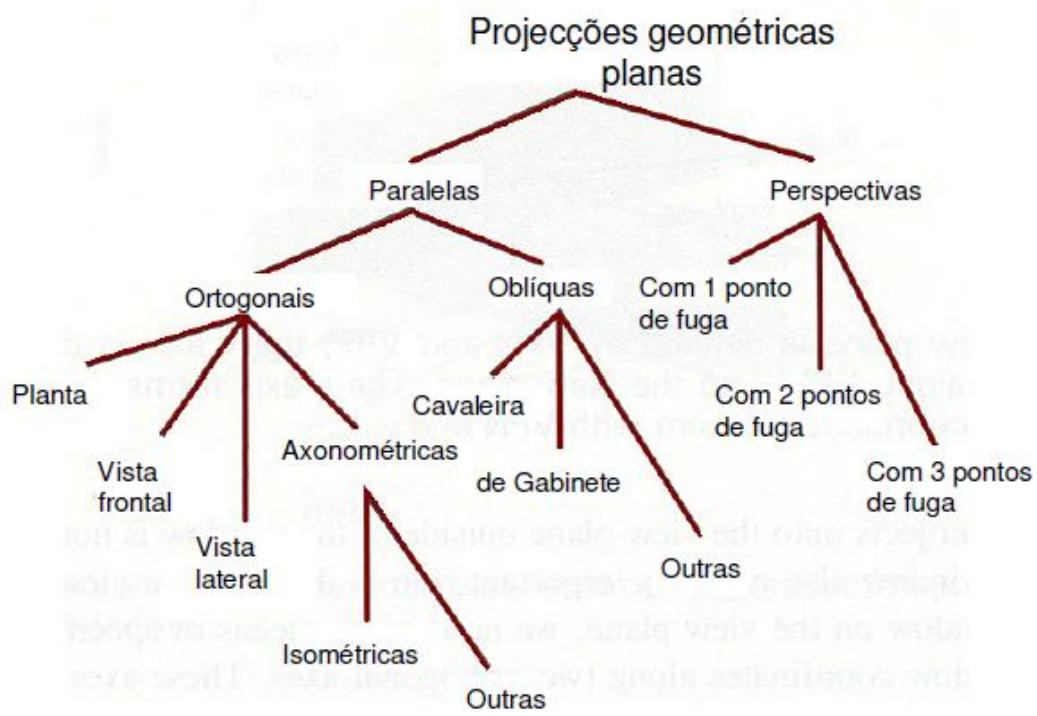
## Outras transformações 3D

- Há outras transformações tridimensionais que são úteis:
  - *Shears*
  - Reflexões

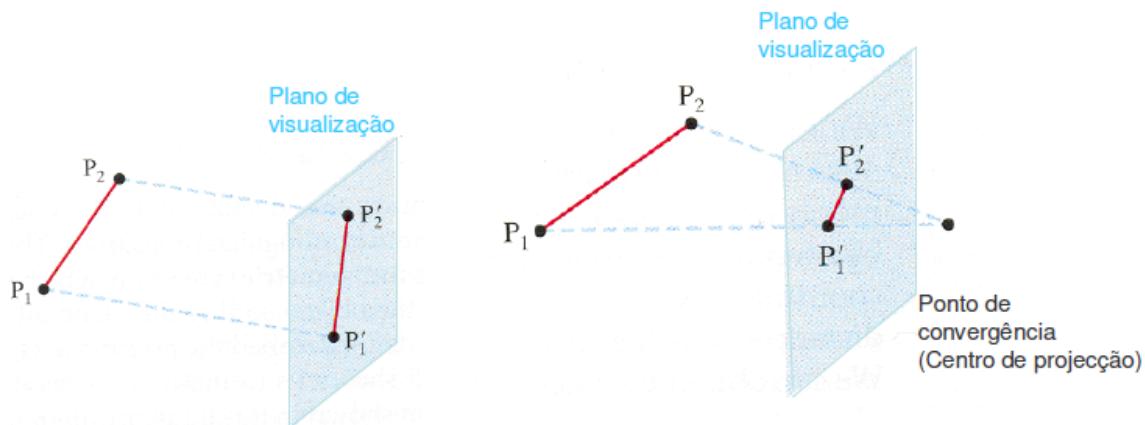
Shear em ZZ'



- Existem duas classes de projecções geométricas planas:
  - Projecções paralelas
  - Projecções perspectivas
- As projecções perspectivas permitem gerar imagens com mais realismo
- Mas envolvem cálculos mais complexos, e nem sempre são a melhor opção !



## Projeções paralelas e perspectivas



Nas projeções paralelas as rectas projectantes são paralelas, i.e., convergem num ponto a uma distância infinita.

Nas projeções perspectivas as rectas projectantes convergem no centro de projecção.

## Projecções paralelas ortogonais (ou ortográficas)

- São projecções paralelas obtidas por projectores perpendiculares ao plano de projecção
- Têm a desvantagem de não dar facilmente uma ideia da estrutura tridimensional dos objectos

### Projecção cavaleira

- Nesta projecção a profundidade ( $L_1$ ) do cubo é representada com uma grandeza igual à largura e à altura
- Tem a desvantagem de não parecer muito realista
- O ângulo  $\Phi$  é geralmente:
  - $\Phi = 30^\circ$
  - $\Phi = 45^\circ$

### Projecção de gabinete

- Nesta projecção a profundidade ( $L_1$ ) do cubo é representada com uma grandeza igual a metade da largura e da altura
- Tem a vantagem de parecer mais realista que a projecção cavaleira

- As projecções podem aplicar-se através de **multiplicação de matrizes** de projecção 4x4 (em coordenadas homogéneas)
- Vamos considerar **casos simples** de projecções perspectivas e ortogonais com plano de visualização (ou projecção) paralelos ao plano **xoy**

Projecção perspectiva com plano de projecção  $z=d$

$P(x, y, z)$  - ponto original

$P_p(x_p, y_p, z_p)$  - ponto projectado

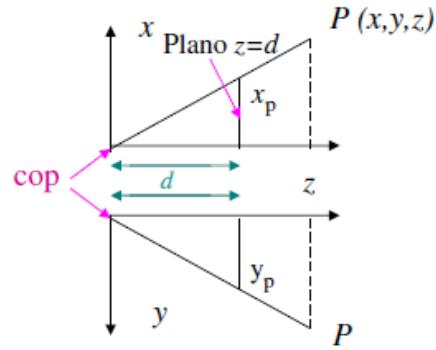
Por semelhança de triângulos:

$$x_p/d = x/z \quad y_p/d = y/z$$

Multiplicando por  $d$ :

$$x_p = \frac{d \cdot x}{z} = \frac{x}{z/d}$$

$$y_p = \frac{d \cdot y}{z} = \frac{y}{z/d}$$



A divisão por  $d$  implica que a projecção de objectos mais distantes seja menor

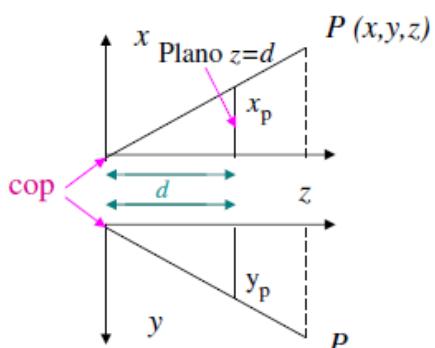
Projecção perspectiva com plano de projecção  $z=d$  e centro de projecção na origem

$P(x, y, z)$  - ponto original

$P_p(x_p, y_p, z_p)$  - ponto projectado

Todos os valores de  $z$  são possíveis excepto  $z=0$

A projecção pode ser representada como:



$$M_{pers} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \longrightarrow P_p = M_{pers} \cdot P$$

Multiplicando  $P$  por  $M_{pers}$  obtém-se o ponto projectado – FAZER na AULA !

Projecção perspectiva com plano de projecção  $z=0$  e centro de projecção em  $z=-d$

$P(x, y, z)$  - ponto original

$P_p(x_p, y_p, z_p)$  - ponto projectado

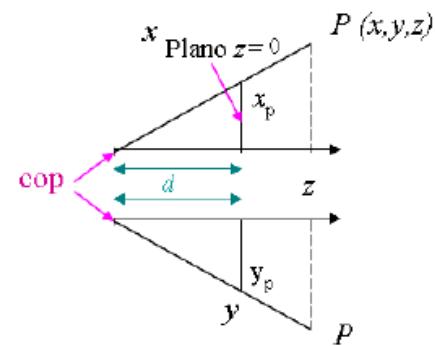
Por semelhança de triângulos:

$$x_p/d = x/(z+d) \quad y_p/d = y/(z+d)$$

Multiplicando por  $d$ :

$$x_p = \frac{d \cdot x}{z+d} = \frac{x}{z/d+1}$$

$$y_p = \frac{d \cdot y}{z+d} = \frac{y}{z/d+1}$$



$$M'_{pers} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}$$

Multiplicando  $P$  por  $M_{pers}$  obtém-se o ponto projectado – FAZER na AULA !

Esta formulação permite fazer  $d$  tender para  $\infty$ , obtendo-se a matriz de projecção paralela ortogonal num plano de projecção em  $z=0$ :

$$M_{ortho} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Como são as coordenadas do ponto projectado?  
Estávamos à espera deste resultado?

# **PHONG'S REFLECTION MODEL**

# Ambient illumination

- Ambient illumination is included as a **constant value** for the whole scene:

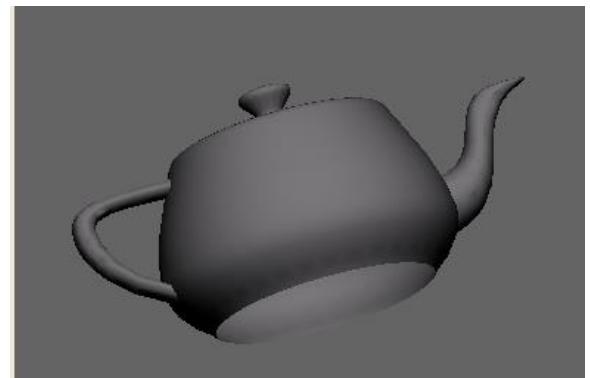
$$I_a$$

which entails a **uniform illumination** for all objects



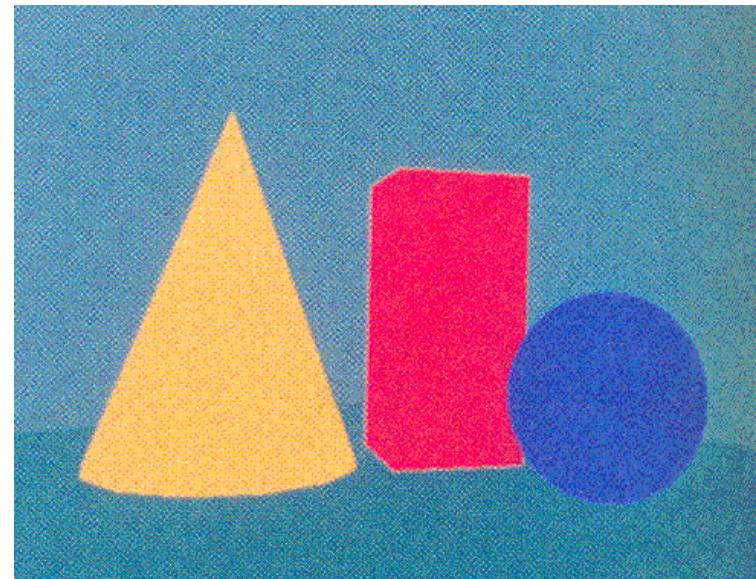
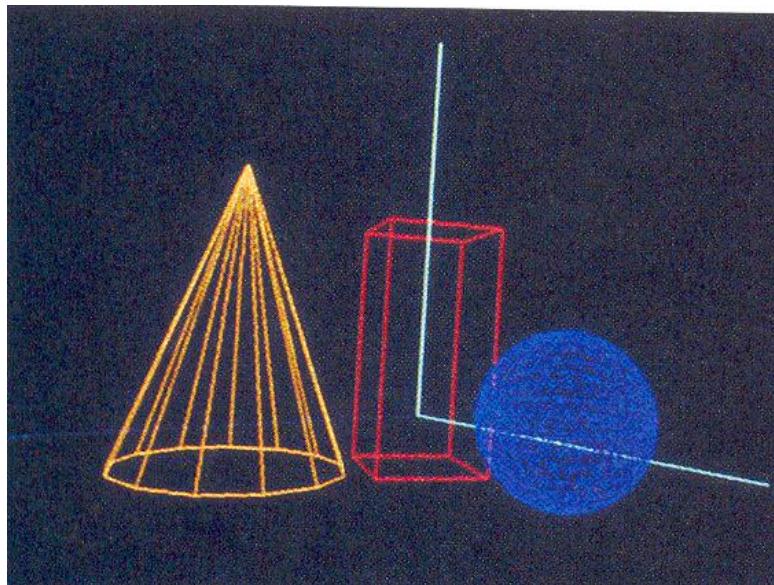
- Therefore, the **reflections** produced by the surfaces are:

- independent of **viewing direction**
- independent of **surface orientation**
- just depend on the **optical properties** of the surface



- Just by itself, **ambient illumination** produces uninteresting results.
- Why ?

$$I_{ambdiff} = K_d I_a$$

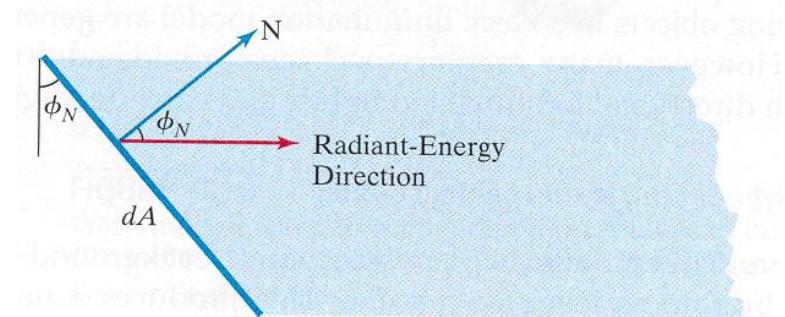


Just ambient illumination

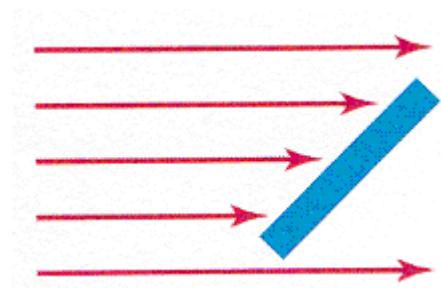
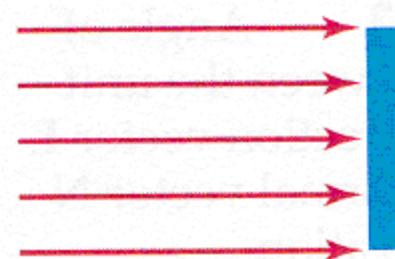
# Diffuse reflection

- It is considered that **incident light is spread with equal intensity in all directions**, regardless of the viewing direction
- Surfaces with that feature are called **Lambertian reflectors** or **ideal diffuse reflectors**
- The intensity of reflected light is computed by **Lambert's Law**:

$$\text{Intensity} = \frac{\text{Radiant-Energy per time unit}}{\text{Projected Area}}$$

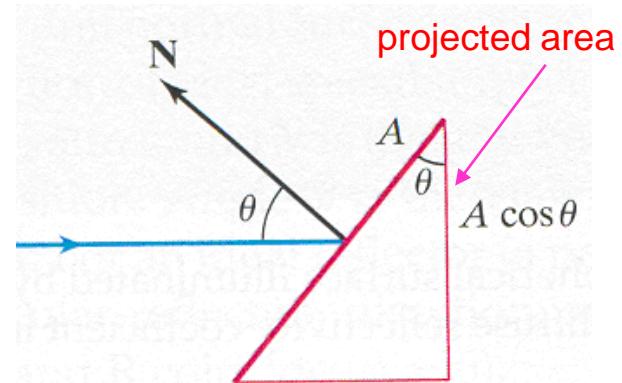


- There is, at least, **one point light source** (usually located at the viewpoint)
- The amount of incident light depends on the **surface orientation** regarding the **direction of the light source**
- A surface that is **orthogonal** to the light direction is “more illuminated” than an **oblique** surface, with the **same area**
- This effect can be seen by varying the orientation of a white paper-sheet relative to light direction



- $\theta$  is the **incidence angle** (between the surface normal and the light direction)
- Given a light source  $I_l$ , the **amount of diffusely reflected light** by a surface is:

$$I_{l,\text{diff}} = k_d I_l \cos \theta$$

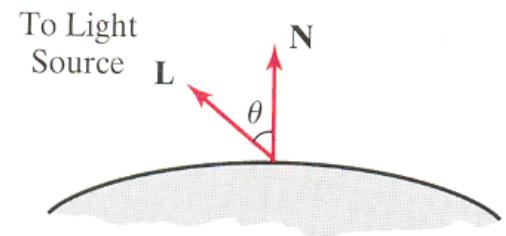


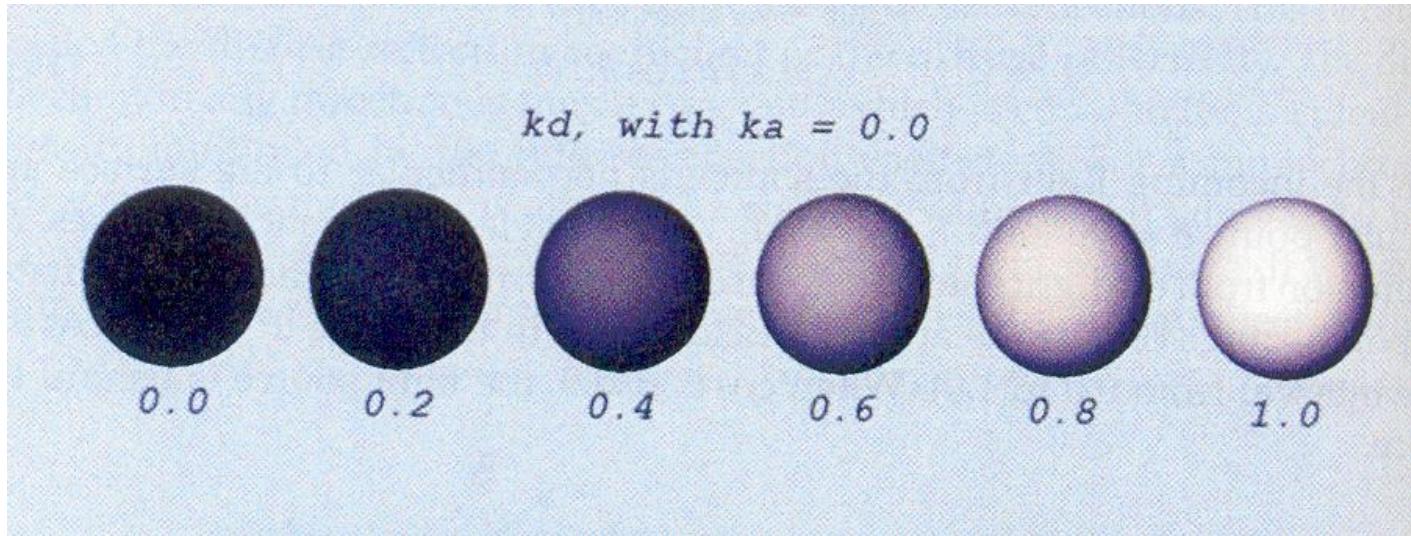
- Using **unit vectors**:

$\mathbf{N} \rightarrow$  surface normal

$\mathbf{L} \rightarrow$  light source direction

$$I_{l,\text{diff}} = \begin{cases} k_d I_l (\mathbf{N} \cdot \mathbf{L}), & \text{se } \mathbf{N} \cdot \mathbf{L} > 0 \\ 0.0, & \text{se } \mathbf{N} \cdot \mathbf{L} \leq 0 \end{cases}$$





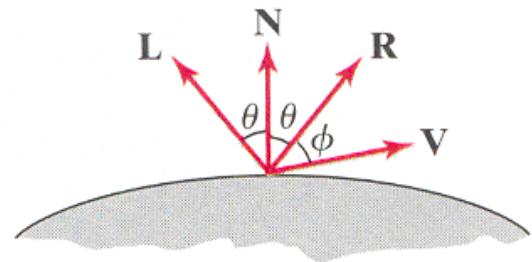
Diffuse reflection for a sphere illuminated by a white point light source, with  $0 < K_d < 1$ , and without ambient illumination ( $K_a = 0$ )

**Adding** the ambient and diffuse components:

$$I_{\text{diff}} = \begin{cases} k_a I_a + k_d I_l (\mathbf{N} \cdot \mathbf{L}), & \text{se } \mathbf{N} \cdot \mathbf{L} > 0 \\ k_a I_a, & \text{se } \mathbf{N} \cdot \mathbf{L} \leq 0 \end{cases}$$

# Specular Reflection and Phong's Model

- The shinier areas, **specular reflections** or **highlights**, that can be seen on shiny surfaces, result from the reflection of most light around concentrated areas
- The **specular reflection angle** is equal to the incidence angle (relative to the normal)
- **R** is the **unit vector** defining the ideal specular reflection direction
- **V** is the **unit vector** defining the viewing direction
- An **ideal reflector** reflects light only in the specular reflection direction (the viewer perceives the specular reflection only if **V** and **R** are coincident  $\Phi = 0$ )

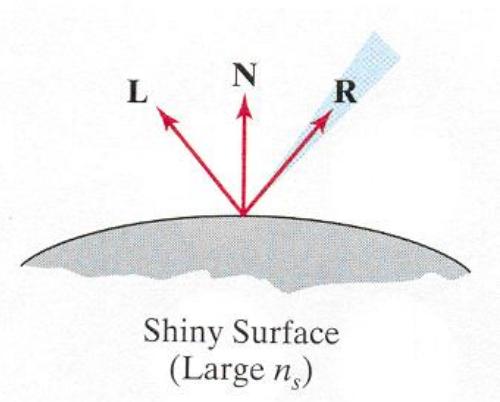
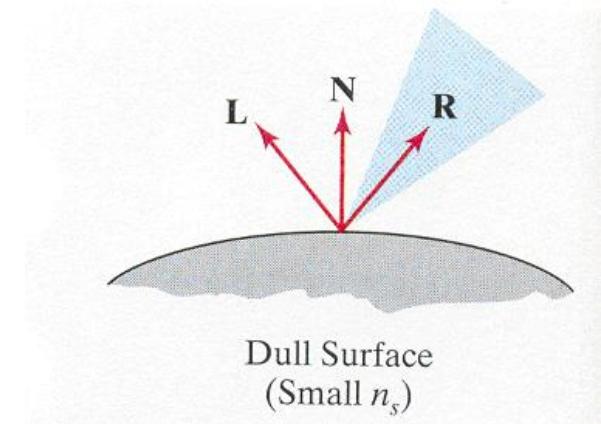


- Objects that are not ideal reflectors produce specular reflections in a **finite set of directions** around vector  $\mathbf{R}$
- **Shiny** surfaces have a **narrow** set of reflection directions
- The empirical **Phong specular reflection model** sets the intensity of the specular reflections as proportional to

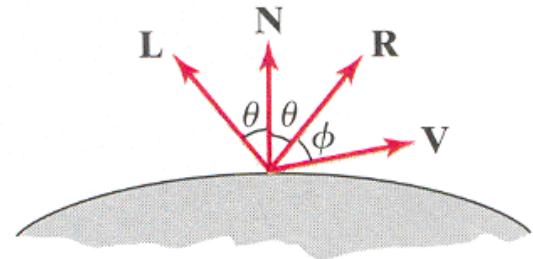
$$\cos^{n_s} \phi$$

$$I_{l,\text{spec}} = W(\theta) I_l \cos^{n_s} \phi$$

$W(\theta)$  is the **specular reflection coefficient**



- The value of  $\cos \Phi$  can be computed by the scalar product of unit vectors  $\mathbf{V} \cdot \mathbf{R}$
- There are no specular reflections, whenever:
  - The light source is behind the surface
  - $\mathbf{V}$  and  $\mathbf{L}$  “are on the same side” of vector  $\mathbf{N}$
- Assuming that the specular reflection coefficient is a constant for each material, the intensity of the specular reflection, for a single point light source, can be computed as:



$$I_{l, \text{spec}} = \begin{cases} k_s I_l (\mathbf{V} \cdot \mathbf{R})^{n_s}, & \text{if } \mathbf{V} \cdot \mathbf{R} > 0 \quad \text{and} \quad \mathbf{N} \cdot \mathbf{L} > 0 \\ 0.0, & \text{if } \mathbf{V} \cdot \mathbf{R} < 0 \quad \text{or} \quad \mathbf{N} \cdot \mathbf{L} \leq 0 \end{cases}$$

$\cos \Phi$

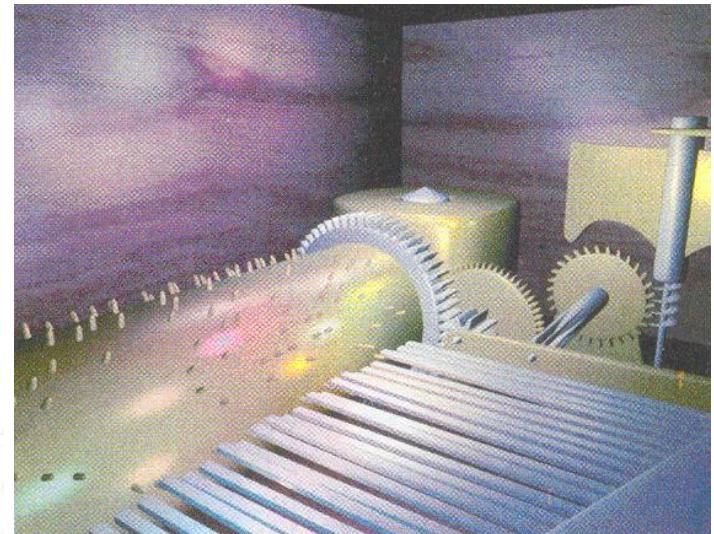
R and V on opposite sides      Light source behind the surface

# More than one point light-source

- For a single point light-source, we add the three components:
  - ambient illumination
  - diffuse reflection
  - specular reflection

$$I = k_a I_a + k_d I_l (\mathbf{N} \cdot \mathbf{L}) + k_s I_l (\mathbf{N} \cdot \mathbf{H})^{n_s}$$

Various light-sources, with different colors



- If there is more than one light-source:

$$I = k_a I_a + \sum_{l=1}^n I_l [k_d (\mathbf{N} \cdot \mathbf{L}) + k_s (\mathbf{N} \cdot \mathbf{H})^{n_s}]$$

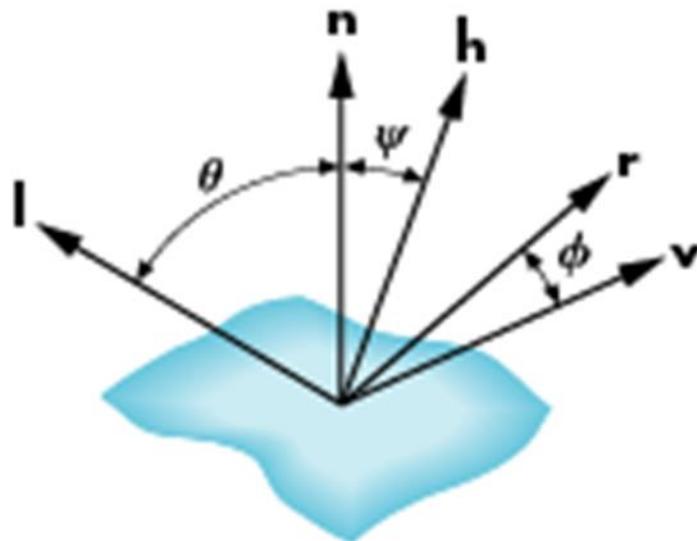


The *halfway vector*  $\mathbf{H}$  (between  $\mathbf{L}$  and  $\mathbf{V}$ ) is sometimes used as an approximation to  $\mathbf{R}$ , in a version of Phong's model simpler to Compute for non-planar surface (with varying  $\mathbf{N}$ )

# The Halfway Vector

- $\mathbf{h}$  is the unit vector “halfway” between  $\mathbf{l}$  and  $\mathbf{v}$

$$\mathbf{h} = (\mathbf{l} + \mathbf{v}) / \|\mathbf{l} + \mathbf{v}\|$$



[Angel]

## RGB color model

- When using the RGB model, light intensity is specified in the illumination model as a RGB-vector:

$$I_l = (I_{lR}, I_{lG}, I_{lB})$$

- The same for the reflection coefficients:

$$k_a = (k_{aR}, k_{aG}, k_{aB}) \quad k_d = (k_{dR}, k_{dG}, k_{dB}) \quad \bar{k_s} = (k_{sR}, k_{sG}, k_{sB})$$

- For instance, the blue component for the diffuse reflection is:

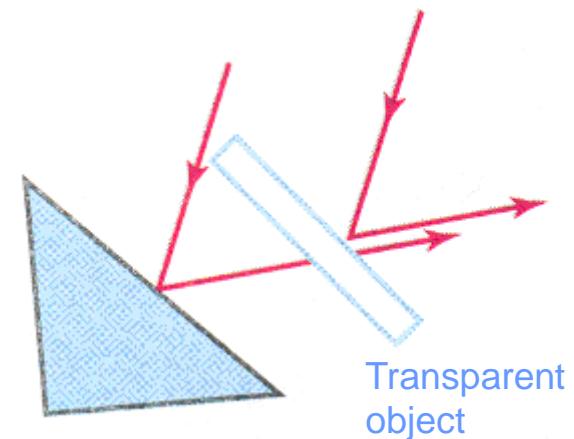
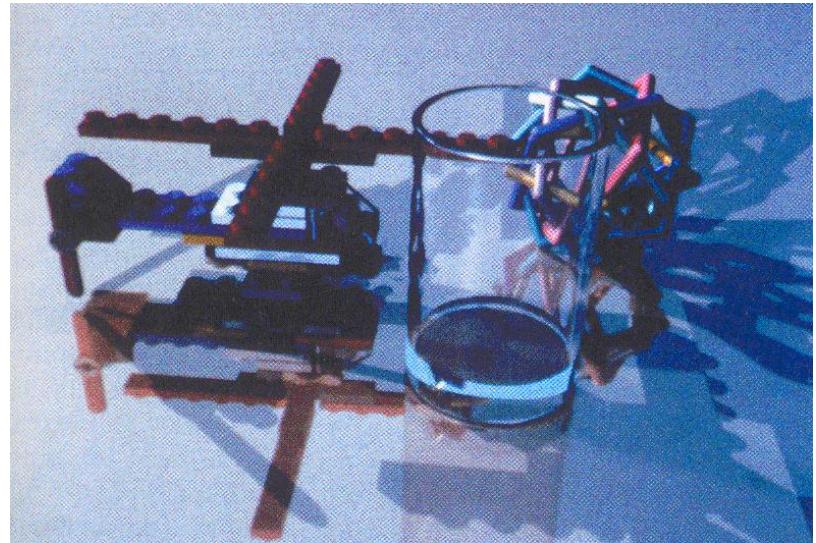
$$I_{lB, \text{diff}} = k_{dB} I_{lB} (\mathbf{N} \cdot \mathbf{L}_l)$$

- It is possible to use other color models (e.g., CMY)

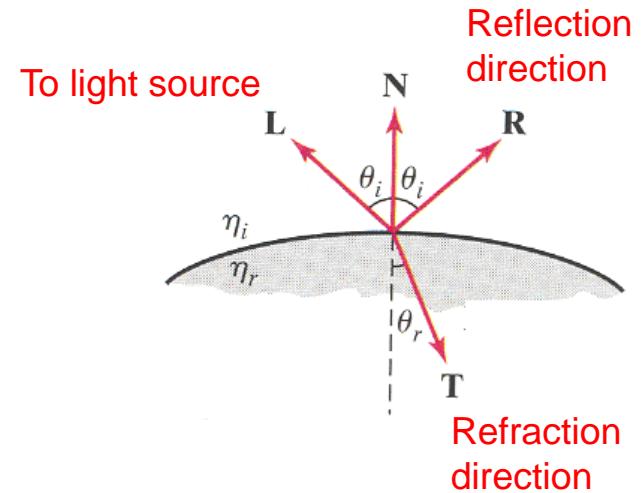
# **TRANSPARENCIES AND SHADOWS**

# Transparency

- A **transparent** object allows seeing other objects behind it
- There are also **translucent** objects, which transmit light, but scatter it in all directions
- Transparency can be modeled with different degrees of realism:
  - with **refraction** (light rays change direction)
  - without refraction

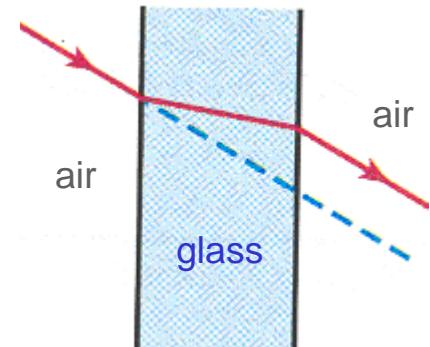


- For greater realism, light refraction has to be considered
- The direction of refracted light is different from the direction of incident light



- The **refraction angle** varies with:
  - the media **refraction indices**  $\eta_i \eta_r$
  - the **incidence angle**  $\theta_i$
- According to **Snell's Law** :

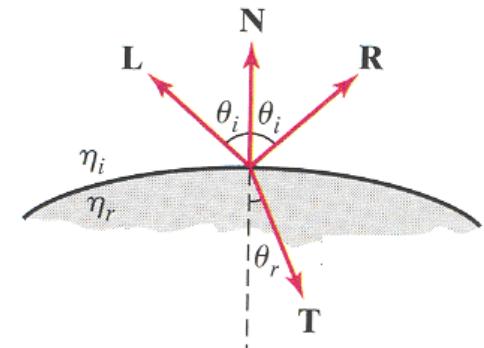
$$\sin \theta_r = \frac{\eta_i}{\eta_r} \sin \theta_i$$



The emerging ray is parallel to the incident light ray

- To compute  $\mathbf{T}$  (unit vector for the **refraction direction**):

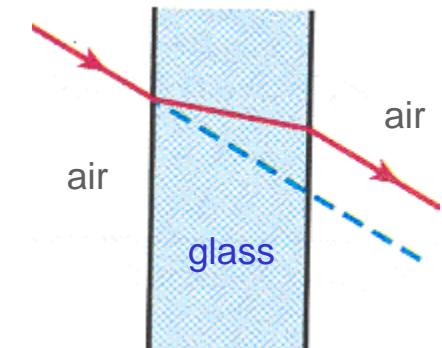
$$\mathbf{T} = \left( \frac{\eta_i}{\eta_r} \cos \theta_i - \cos \theta_r \right) \mathbf{N} - \frac{\eta_i}{\eta_r} \mathbf{L}$$



$\mathbf{N}$  – surface unit normal vector

$\mathbf{L}$  – unit vector towards the light source

- The general effect of light passing through a glass slice is an emerging light ray parallel to incident light ray



The emerging ray is parallel to the incident light ray

- It is possible to **combine** light transmitted through a transparent surface with light reflected by that surface:

$$I = (1 - k_t)I_{\text{refl}} + k_t I_{\text{trans}}$$

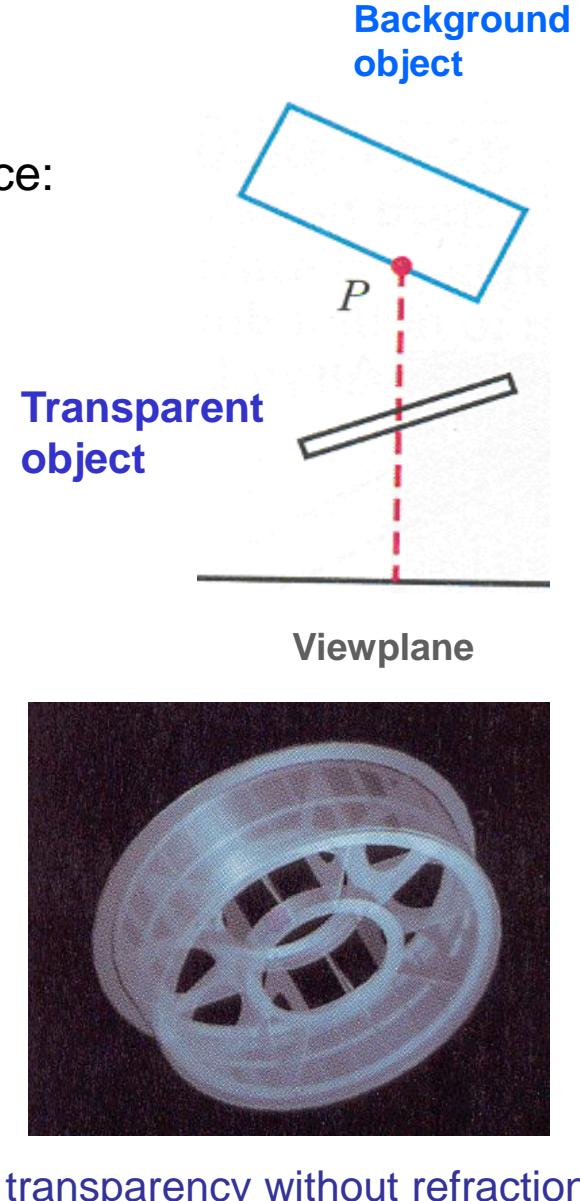
$k_t$  - transparency coefficient [0, 1]

1 – fully transparent

0 – totally opaque

- Can also define the **opacity coefficient**:

$$(1 - k_t)$$



transparency without refraction

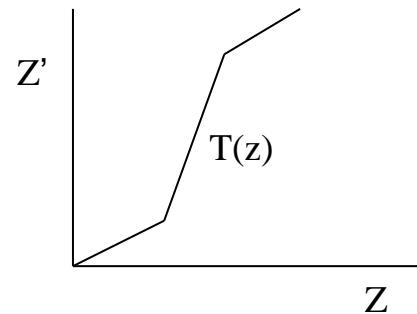
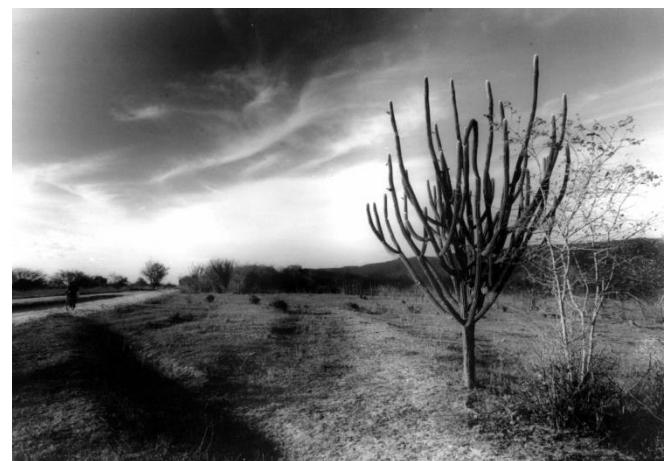
# **INTENSITY TRANSFORMATIONS**

## 1. Intensity transformations

- The same transformation is applied to all image pixels
- **Contrast stretching** is one of the mostly used transformations
- For **grayscale** images that do not use the entire **dynamic range** (possible gray values), the histogram can be modified by a transformation function  $T$

$$z' = T(z)$$

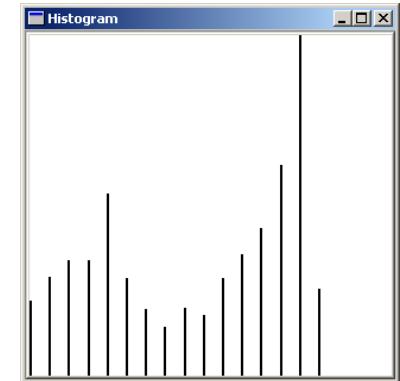
New gray value                      Original gray value



# **HISTOGRAM-BASED PROCESSING**

# Contrast-stretching – Another example

$$final[x, y] = \frac{original[x, y] - \min}{\max - \min} \times 255$$



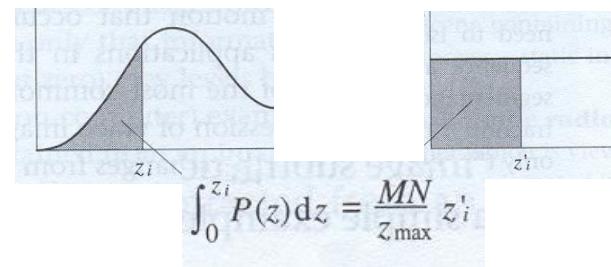
# 5. Histogram Equalization

- It can be regarded as an extension of **contrast-stretching**
- The image is analyzed and a function  $T(z)$  is computed in order to obtain an image with a **equiprobable histogram**
- It is assumed that:
  - $z$  is continuous
  - the image has a continuous probability density function  $P(z)$
  - the resulting image has an **equiprobable distribution**  $Q(z)$

$$\int_0^{z_1} P(z) dz = \int_0^{z'_1} Q(z) dz = \frac{MN}{z_{\max}}$$

$$z'_i = T(z_i) = \frac{z_{\max}}{MN} \int_0^{z_1} P(z) dz$$

$$z'_i = \frac{z_{\max}}{MN} \sum_{z=0}^{z_1} H(z) dz$$



$M \times N - n^o$  de pixels da imagem

é a aproximação discreta e  
H é o histograma discreto

# **FILTERING**

## 7. Image smoothing

- Also called *neighbourhood averaging*, can be used to reduce image noise
- Or as pre-processing step prior to edge detection
- It is a simple filtering operation, where the **filter kernel** slides over the original image and the **intensity (gray level)** of the resulting pixel in the final image is computed as:

$$\sum_{i=0}^{N-1} W_i \cdot I_i$$

Pixel intensity values in the original image  
Filter weights

- Other filters allow reducing noise *without blurring* (e.g., median filter)

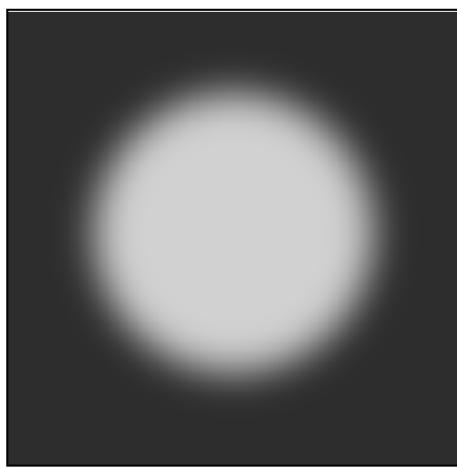
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3 x 3 Averaging Filter

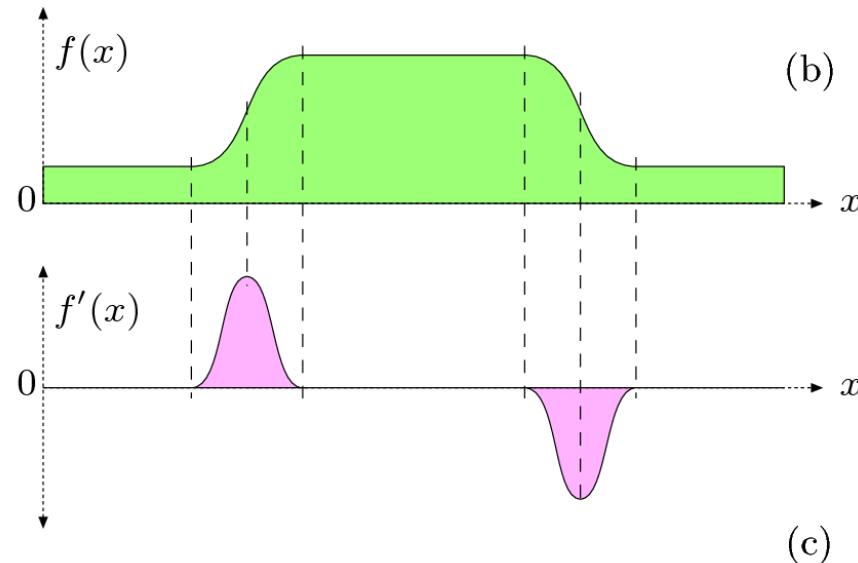
All weights are 1/9

# **EDGE DETECTION**

# 1st Derivative



(a)

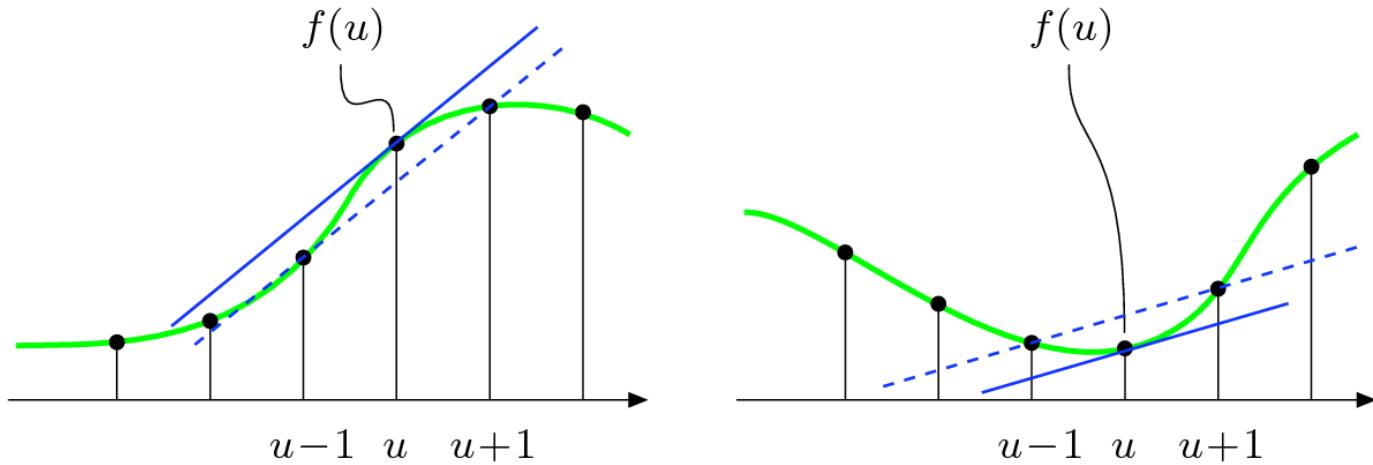


$$f'(x) = \frac{df}{dx}(x)$$

$$\frac{df}{du}(u) \approx \frac{f(u+1) - f(u-1)}{2} = 0.5 \cdot (f(u+1) - f(u-1))$$

[Burger & Burge]

# 1st Derivative – Simple estimation ?



$$\frac{df}{du}(u) \approx \frac{f(u+1) - f(u-1)}{2} = 0.5 \cdot (f(u+1) - f(u-1))$$

[Burger & Burge]

# Partial Derivatives and Gradient

$$H_x^D = \begin{bmatrix} -0.5 & \mathbf{0} & 0.5 \end{bmatrix} = 0.5 \cdot \begin{bmatrix} -1 & \mathbf{0} & 1 \end{bmatrix}$$

$$H_y^D = \begin{bmatrix} -0.5 \\ \mathbf{0} \\ 0.5 \end{bmatrix} = 0.5 \cdot \begin{bmatrix} -1 \\ \mathbf{0} \\ 1 \end{bmatrix}$$

$$\nabla I(u, v) = \begin{bmatrix} \frac{\partial I}{\partial u}(u, v) \\ \frac{\partial I}{\partial v}(u, v) \end{bmatrix}$$

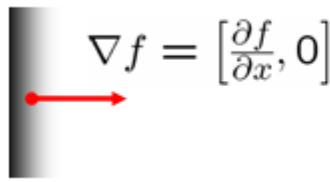
$$|\nabla I|(u, v) = \sqrt{\left(\frac{\partial I}{\partial u}(u, v)\right)^2 + \left(\frac{\partial I}{\partial v}(u, v)\right)^2}$$

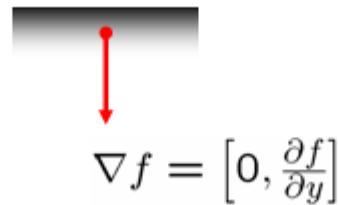
[Burger & Burge]

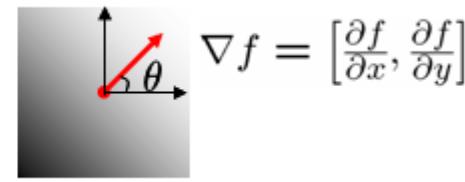
# Edge detection with derivative operators

- The **gradient vector** of image  $f(x,y)$  points towards the direction of fastest intensity change

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$


$$\nabla f = \left[ \frac{\partial f}{\partial x}, 0 \right]$$


$$\nabla f = \left[ 0, \frac{\partial f}{\partial y} \right]$$


$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

Gradient **direction**

$$\theta = \tan^{-1} \left( \frac{\partial f / \partial y}{\partial f / \partial x} \right)$$

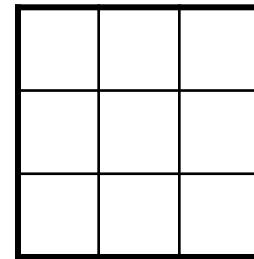
Gradient **magnitude**

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- How to compute derivatives of a discrete image  $f(x,y)$ ?
- One option is to compute **finite differences**:

$$\frac{\partial f}{\partial x}[x, y] \approx f[x + 1, y] - f[x, y]$$

and apply the corresponding **kernel**



- Several **operators** that **approximate the gradient** can be used to detect edges:
  - Roberts
  - Prewitt
  - Sobel

# **OPERATORS FOR EDGE DETECTION**

# Prewitt and Sobel Operators

$$H_x^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & \mathbf{0} & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad H_y^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & \mathbf{0} & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\nabla I(u, v) \approx \frac{1}{6} \cdot \begin{bmatrix} (I * H_x^P)(u, v) \\ (I * H_y^P)(u, v) \end{bmatrix}$$

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & \mathbf{0} & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & \mathbf{0} & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\nabla I(u, v) \approx \frac{1}{8} \cdot \begin{bmatrix} (I * H_x^S)(u, v) \\ (I * H_y^S)(u, v) \end{bmatrix}$$

[Burger & Burge]