

CAD

THIẾT KẾ NHỜ MÁY TÍNH

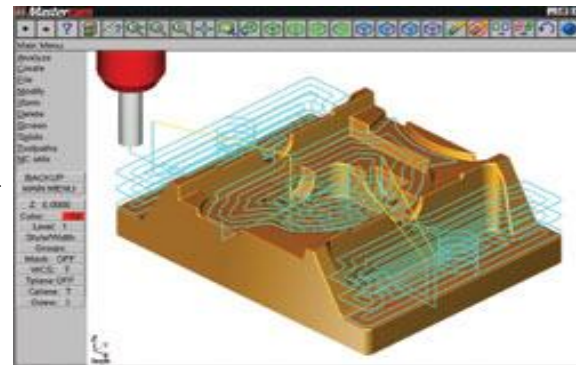
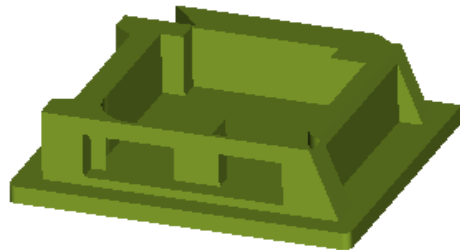
# NỘI DUNG

- Tổng quan
- Đồ hoạ máy tính
- Mô hình hoá hình học
- Cấu trúc dữ liệu và Tiêu chuẩn đồ hoạ
- Giới thiệu phần mềm CAD

# Tổng quan

Chúng ta biết mối liên hệ giữa CAD/CAM và CNC

CAD



CAM

CNC



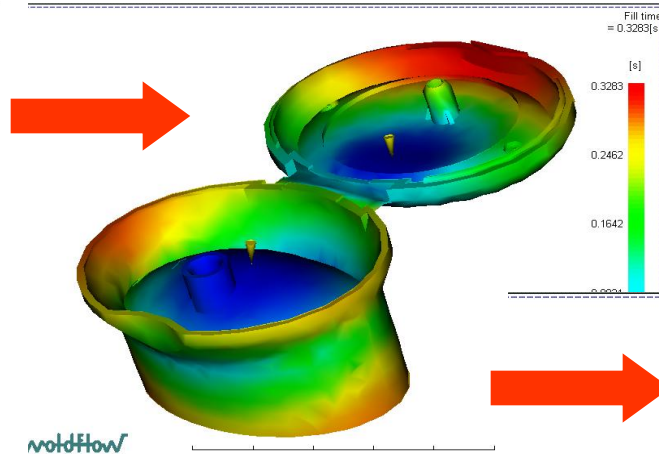
```
N010 G71 G90 G94  
N020 M03 F200 S1000  
N030 G00 X10.00Y10.00  
N040 G00 Z2.00  
N050 G01 Z-10.00  
N060 G00 Z2.00  
N070 G00 X50.00  
N080 G01 Z-10.00  
N090 G00 Z2.00  
N100 G00 Y30.00  
N120 G00 Z20.00  
N130 G00 X00 Y00  
N140 M02
```

# Tổng quan

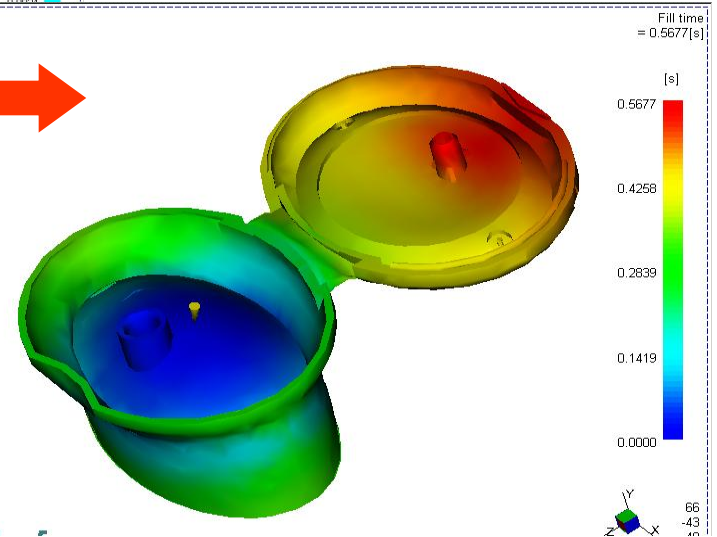
Hệ thống CAD dùng máy tính để



Phát triển



Phân tích



Sửa chữa

# Tổng quan

Vấn đề cốt lõi trong hệ thống CAD là *đồ họa máy tính*. Khi thiết kế người thiết kế dùng máy tính để *tạo ra, biến đổi và đưa dữ liệu* lên màn hình ở dạng hình ảnh và ký hiệu.



# Tổng quan

Một hệ tương tác đồ họa máy tính ICG (Interactive Computer Graphics) điển hình bao gồm *phần cứng* và *phần mềm*.



# Tổng quan

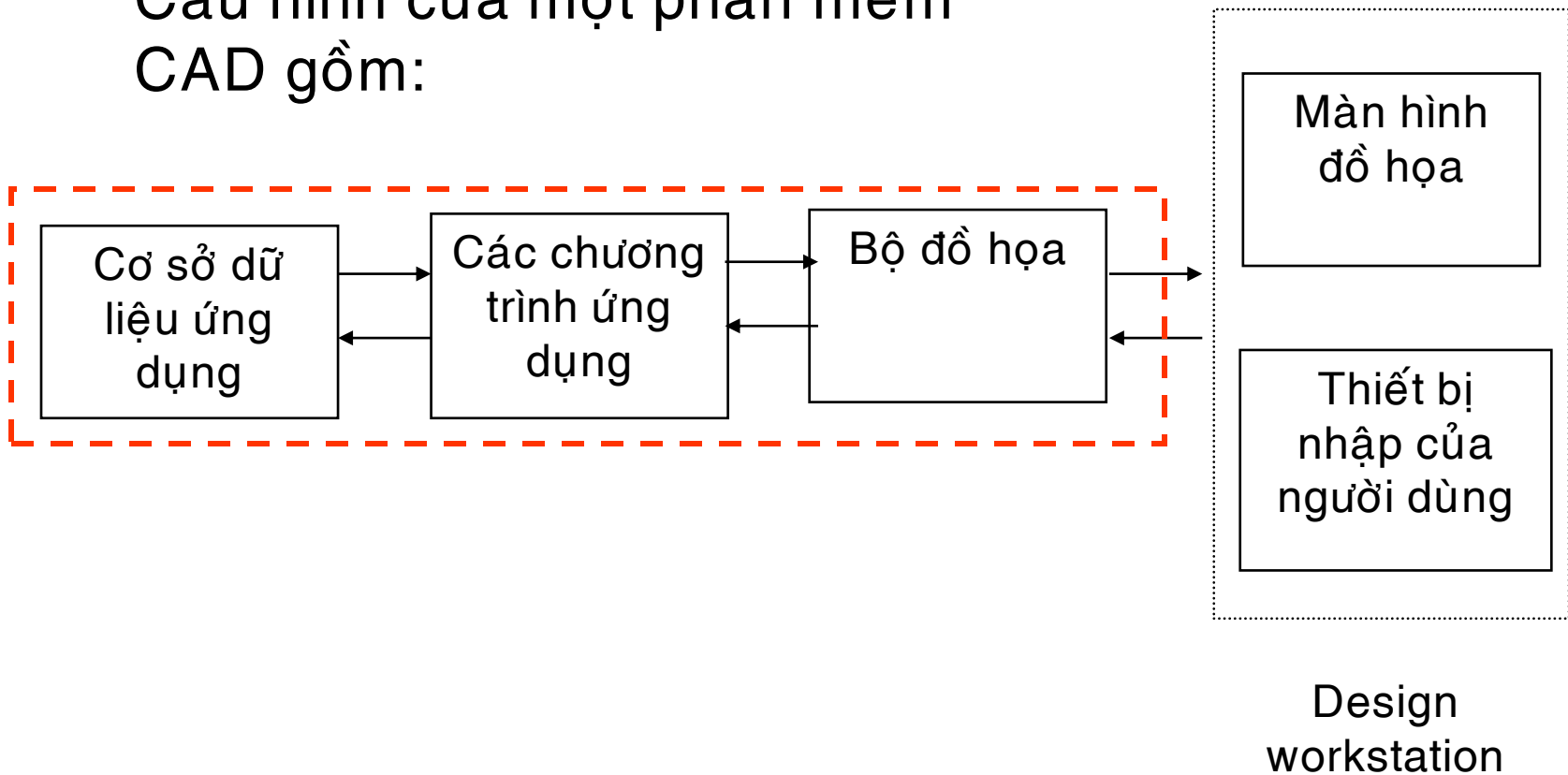
## *Phần cứng gồm:*

- Bộ xử lý trung tâm
- Một hoặc vài trạm làm việc kể cả màn hình
- Các thiết bị như máy in, máy vẽ, .....

*Phần mềm gồm:* Các chương trình cần thiết để đưa quá trình đồ họa lên hệ thống, kèm theo các *chương trình ứng dụng* cho các nhiệm vụ thiết kế riêng biệt theo yêu cầu của người dùng.

# Tổng quan

Cấu hình của một phần mềm CAD gồm:





# Tổng quan

*Hệ thống hay bộ đồ họa* (Graphics Package) là phần mềm hỗ trợ giữa người dùng và màn hình đồ họa. Nó *quản lý sự tương tác* giữa người dùng và hệ thống. Nó cũng dùng như là *giao diện* (kết nối trung gian) giữa người dùng và phần mềm ứng dụng. Hệ thống đồ họa gồm các *chương trình con nhập* (input subroutines) và các *chương trình con xuất* (output subroutines).

# Tổng quan

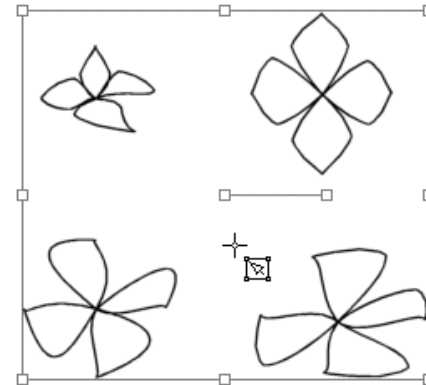
**Các chương trình con nhập** nhận các lệnh và dữ liệu từ người dùng và đưa chúng đến chương trình ứng dụng.

**Chương trình con xuất** điều khiển màn hình (hoặc các thiết bị khác) và biến đổi model ứng dụng sang hình ảnh đồ họa 2 hoặc 3 chiều.

# ĐỒ HOẠ MÁY TÍNH

Sự tiến hoá của đồ hoạ máy tính:

- Từ 1960 Màn hình ký tự
- Từ 1963- 1980s Màn hình vector
- từ 1972 - Màn hình raster 2D
- từ 1984 - 3D graphics workstation
- 1990 - 2005 Multimedia và Hypermedia, thực tế ảo,...

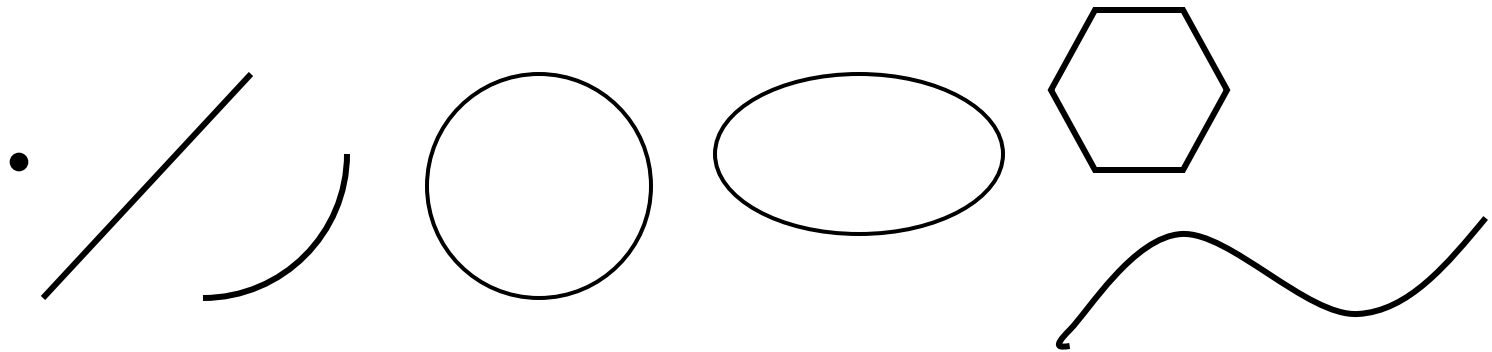


# Các chức năng của bộ đồ họa

1. *Tạo ra các phân tử đồ họa (Create)*
2. *Biến đổi (Trasform): Di chuyển, phóng đại, xoay*
3. *Chỉnh sửa (Modify): cắt xén,...*
4. *Tạo đặc tính (Attribute): nét vẽ, màu, độ dày.*
5. *Chia nhỏ để chọn xoá đối tượng (Delete)*
6. *Tạo hàm nhập của người dùng (User input Function)*

# Tạo ra các phần tử đồ họa

*Phần tử trong đồ họa máy tính* là một đối tượng hình ảnh cơ sở như *điểm, đường thẳng, đường tròn,...* Tập hợp những phần tử này trong hệ thống cũng có thể bao gồm cả chữ viết và ký hiệu đặc biệt



# Tạo ra các phần tử đồ họa

Thí dụ AutoCAD có các công cụ tạo các phần tử hình học sau



# Biến đổi (Transformation)

Biến đổi hình học trong không gian 2 chiều (2D):

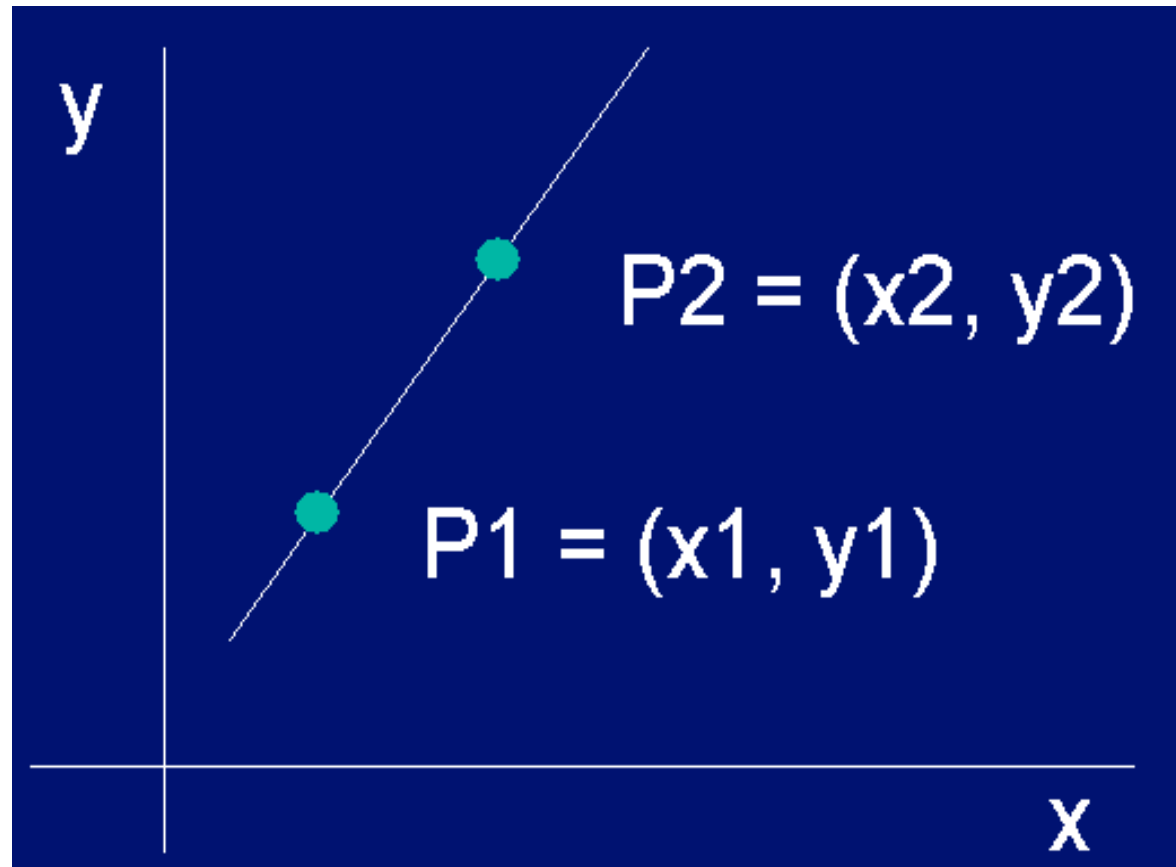
Để định vị một điểm trong hệ tọa độ 2 trục, phải chỉ ra tọa độ X, Y . Những tọa độ này có thể coi như một ma trận  $1 \times 2$  : (x,y).

Thí dụ: Ma trận (2,5) là một điểm có tọa độ  $x = 2$  và  $y = 5$  tính từ gốc tọa độ.

Phương pháp này có thể phát triển cho việc xác định đường như là ma trận  $2 \times 2$  bằng cách cho tọa độ 2 điểm của đường thẳng. Công thức có dạng:

$$\mathcal{L} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$$

# Biến đổi (Transformation)





# Biến đổi (Transformation)

Bằng cách dùng các quy tắc ma trận, một điểm hay một đường (hoặc phần tử hình học khác viết dưới dạng ma trận) có thể được biến đổi để tạo ra một phần tử mới.

Có nhiều phép biến đổi được dùng trong đồ họa máy tính. Chúng ta sẽ bàn về 3 phép biến đổi: *di chuyển*, *khuyến đại* và *xoay*.

# Phép di chuyển

**Di chuyển:** Là đưa phần tử hình học từ chỗ này tới chỗ kia. Trong trường hợp 1 điểm, phương trình được viết như sau:

$$x' = x + m, y' = y + n$$

Trong đó  $x', y'$  là tọa độ của điểm di chuyển tới  
 $x, y$  là tọa độ ban đầu của điểm di chuyển đi,  
 $m, n$  - lượng di chuyển tương ứng theo phương  $x, y$ .

Viết dưới dạng ma trận:

$$(x', y') = (x, y) + T$$

Trong đó  $T = (m, n)$  ma trận di chuyển

Ký hiệu  $T$  là do chữ Translate = di chuyển.

# Phép di chuyển

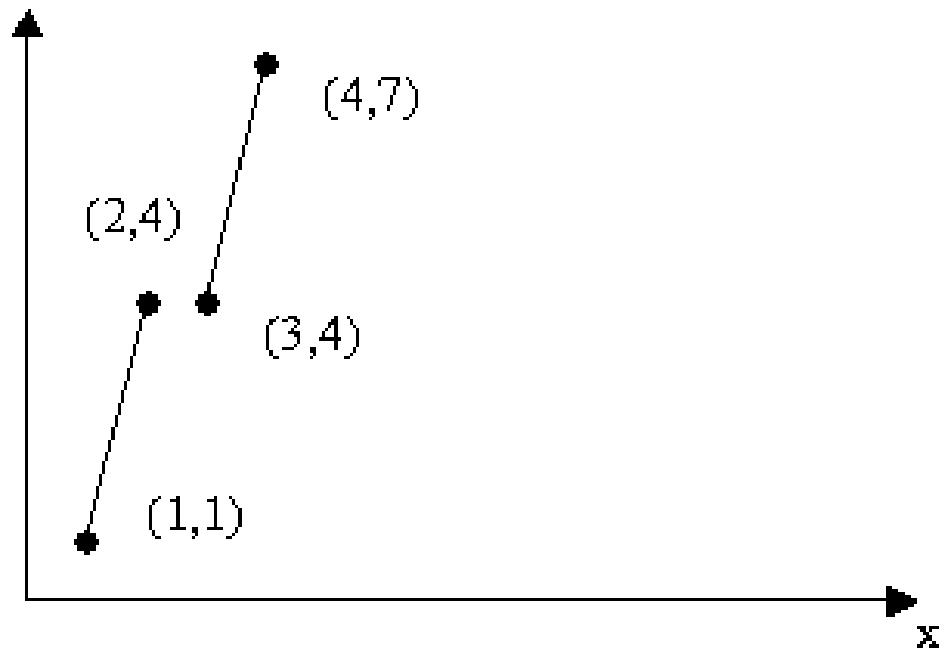
Để minh họa các biến đổi trong mặt phẳng 2D hãy xét một đường thẳng được xác định bởi ma trận.

$$L = \begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix}$$

Giả sử ta cần di chuyển L đi 2 đơn vị theo x và 3 đơn vị theo y

Việc này có nghĩa là ta phải cộng thêm 2 vào các giá trị x hiện tại và 3 vào giá trị y hiện tại của các điểm đầu và cuối của đường thẳng. Nghĩa là đường thẳng mới có các điểm cuối là (3,4) và (4,7). Hiệu quả của phép di chuyển minh họa trên hình sau

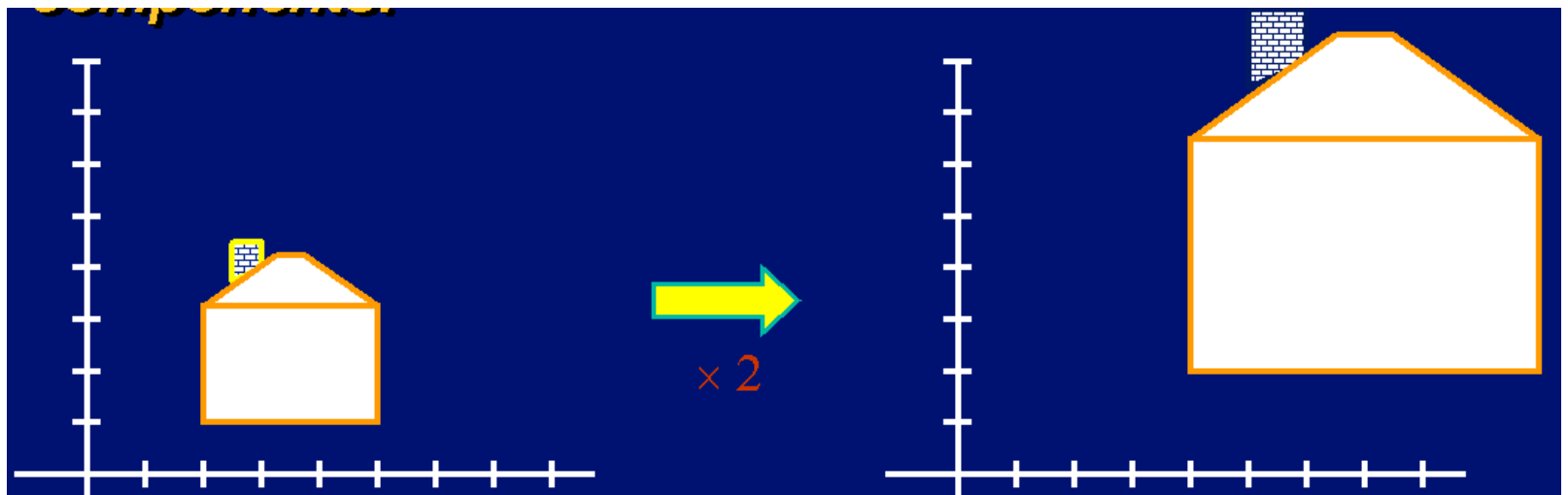
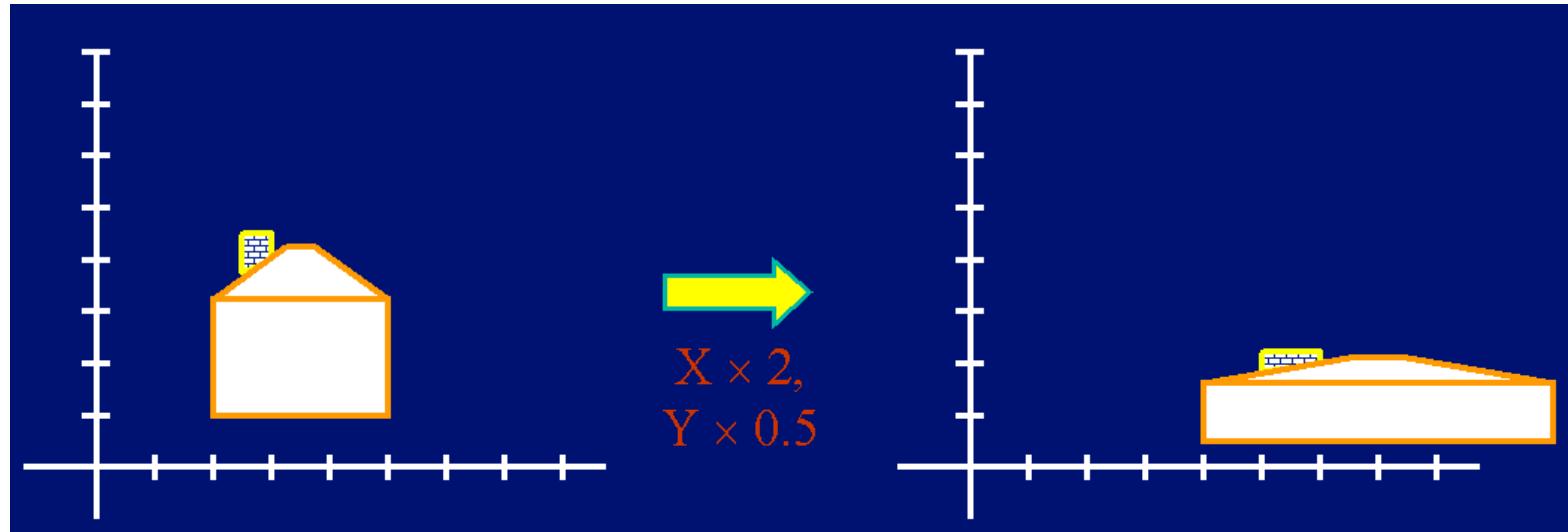
# Phép di chuyển



# *Scaling*: lấy tỉ lệ

Dùng để tăng hoặc giảm kích thước của đối tượng hình học. Không nhất thiết phải có tỉ lệ khuếch đại như nhau theo cả 2 phương  $x, y$ . Thí dụ một đường tròn có thể biến đổi thành ellipse bằng cách lấy tỉ lệ khuếch đại theo phương  $x$  và  $y$  khác nhau.

# *Scaling: lấy tỉ lệ*



# *Scaling*: lấy tỉ lệ

Các điểm của một phân tử có thể được lấy tỉ lệ khuếch đại bằng ma trận tỉ lệ như sau:

$$(x', y') = (x, y) S$$

Trong đó

$$S = \begin{bmatrix} m & 0 \\ 0 & n \end{bmatrix}$$

Là ma trận tỉ lệ. Điều này có nghĩa là kích thước của phân tử được thay đổi bởi hệ số tỉ lệ  $m$  theo phương  $x$  và hệ số tỉ lệ  $n$  theo phương  $y$ .

# Scaling: lấy tỉ lệ

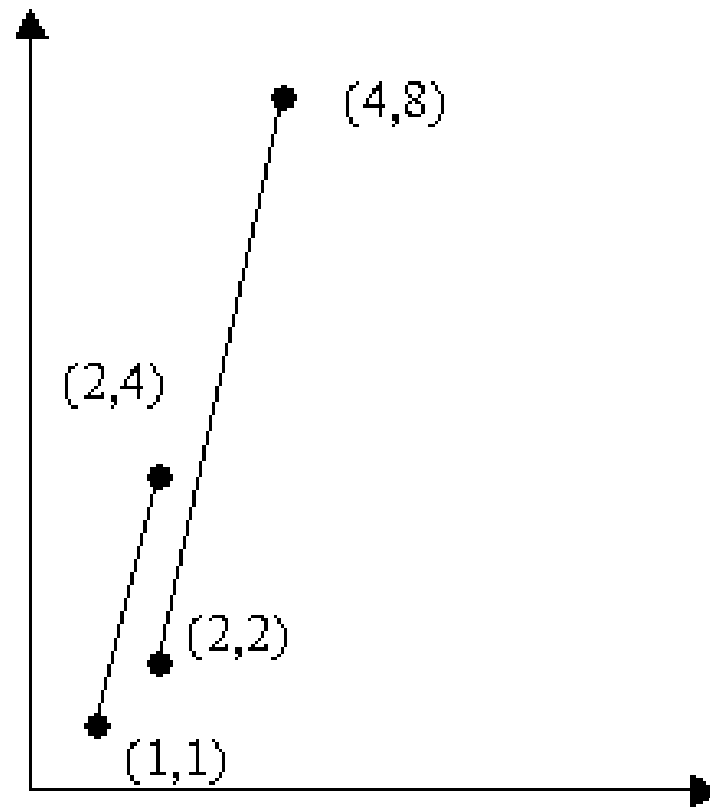
## Thí dụ

Cũng với đường thẳng như trong thí dụ trên, ta hãy nhân tỉ lệ lên 2. Khi đó ma trận tỉ lệ ( 2x2) được xác định như sau:

$$S = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Phương trình khuếch đại được viết

$$\begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 8 \end{bmatrix}$$



Kết quả khuếch đại



# Phép xoay

Trong phép biến đổi này các điểm của một phân tử được xoay quanh gốc tọa độ một góc  $\theta$ . Đối với góc  $\theta$  dương, chi tiết xoay ngược chiều kim đồng hồ. Viết dưới dạng ma trận sẽ như sau:

$$(x', y') = (x, y) R$$

Trong đó

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

Là ma trận xoay. R là viết tắt của chữ rotation = xoay

# Phép xoay

Thí dụ: Chúng ta sẽ sử dụng cùng đường thẳng trên và cho xoay một góc  $30^0$  quanh gốc tọa độ. Ma trận xoay được viết:

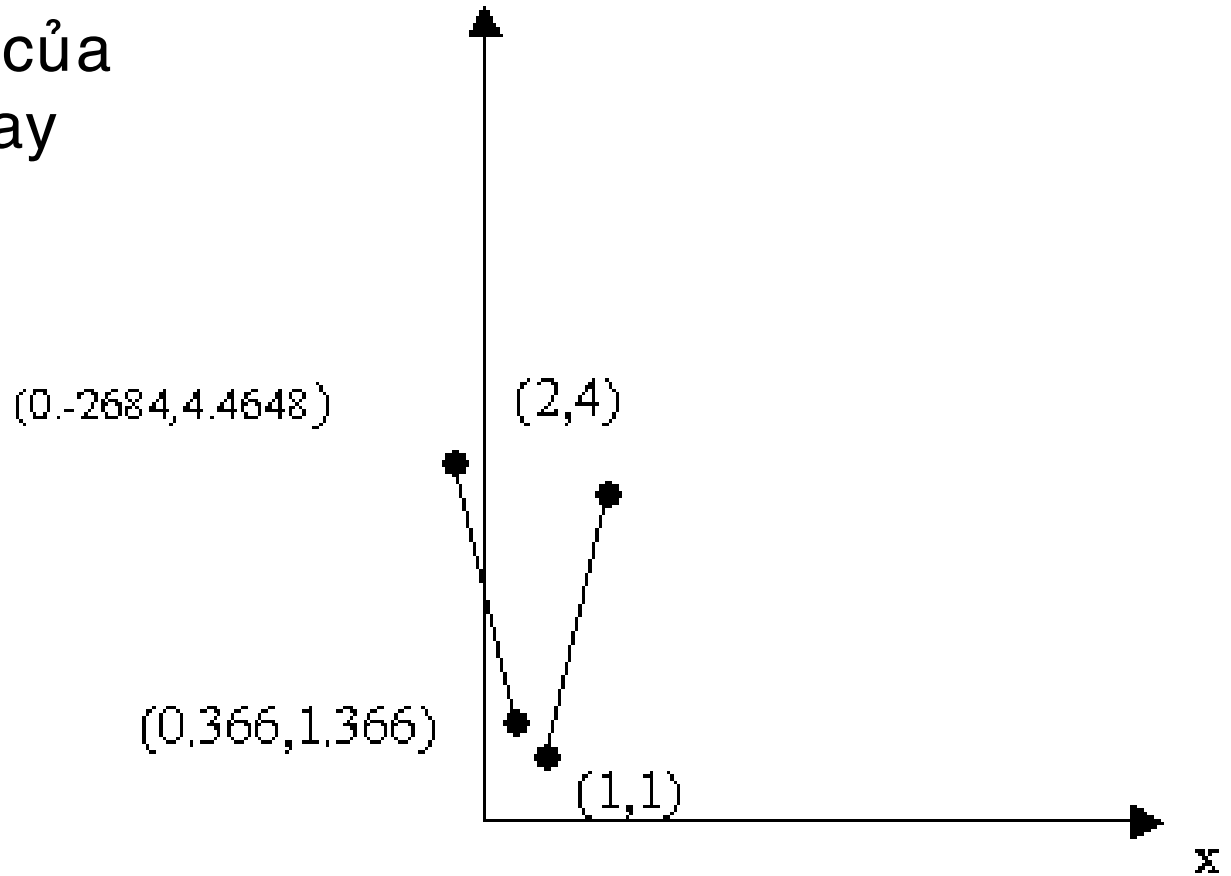
$$R = \begin{bmatrix} \cos 30 & \sin 30 \\ -\sin 30 & \cos 30 \end{bmatrix} = \begin{bmatrix} 0,866 & 0,5 \\ -0,5 & 0,866 \end{bmatrix}$$

Đường thẳng mới được xác định như sau:

$$\begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 0,866 & 0,5 \\ -0,5 & 0,866 \end{bmatrix} = \begin{bmatrix} 0,366 & 1,366 \\ -0,268 & 4,464 \end{bmatrix}$$

# Phép xoay

Kết quả của  
phép xoay



## *Biến đổi trong không gian 3 chiều (3D)*

Biến đổi nhờ phương pháp ma trận có thể phát triển cho không gian 3D. Chúng ta hãy xem cũng với 3 phép biến đổi như trên, xét trong không gian 3D thì sẽ như thế nào

# *Biến đổi trong không gian 3 chiều (3D)*

*Di chuyển*: Ma trận dịch chuyển cho một điểm trong không gian 3D là:

$$T = (m, n, p)$$

Các tọa độ điểm của đối tượng sẽ phải cộng thêm

m đơn vị theo phương X

n đơn vị theo phương Y

p đơn vị theo phương Z

để tạo nên phần tử mới trong không gian 3D

# *Biến đổi trong không gian 3 chiều (3D)*

## *Scaling (nhân tỉ lệ)*

Nhân tỉ lệ được cho bởi ma trận tỉ lệ

$$S = \begin{bmatrix} m & 0 & 0 \\ 0 & n & 0 \\ 0 & 0 & p \end{bmatrix}$$

Khi những giá trị  $m, n, p$  bằng nhau, tỉ lệ là tuyến tính.

# *Biến đổi trong không gian 3 chiều (3D)*

*Rotation – xoay:*

Xoay trong không gian 3D có thể được thực hiện đối với mỗi trục tọa độ.

*Xoay một góc  $\theta$  được thực hiện bởi ma trận*

*quanh trục Z*

$$R_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

*quanh trục Y*

$$R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

*quanh trục X*

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

# *Biểu diễn trong hệ tọa độ đồng nhất*

Hệ tọa độ đồng nhất biểu diễn tọa độ 2D thành vector 3D

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{\text{homogeneous coords}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Hệ tọa độ đồng nhất tuy không trực giác nhưng cho phép thực hiện các phép biến đổi trong đồ họa rất dễ dàng



# *Biểu diễn trong hệ tọa độ đồng nhất*

Di chuyển

$$\mathbf{T}_{translation} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix}$$

Khuyến đại

$$\mathbf{S}_{scale} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Xoay

$$\mathbf{R}_{rotation} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$