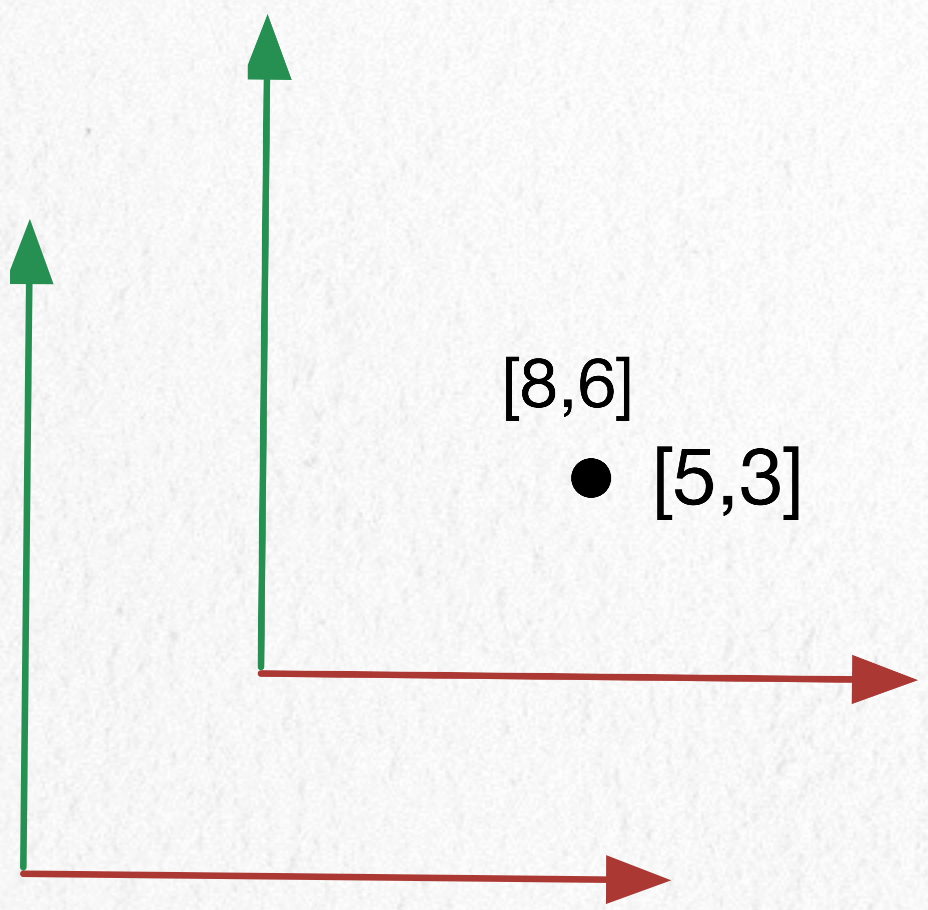
TRANSFORMACIONES LINEALES

Computación Gráfica

Tipos de Datos Geométricos

- Un punto se puede representar con tres números reales $[x,y,z]^T$ que llamaremos **vector coordenado**.
- Los números especifican la posición del punto respecto a un **sistema coordenado** acordado con un punto origen y tres direcciones.



- Para definir la localización actual de un punto hay que definir:
  - **sistema coordenado.**
  - **vector coordenado.**

Tipos de Datos Geométricos

- **Punto** : objeto geométrico, no objeto numérico. $\tilde{p}$
- **Vector**: también un objeto geométrico. Los puntos representan lugares, los vectores representan el movimiento para ir de un punto a otro. $\vec{v}$
- **Vector coordenado:** objeto numérico, compuesto de números reales. Representado con negritas. $\mathbf{c}$
- **Sistema coordenado:** no-numérico, colección de vectores. Hay dos tipos de sistemas coordenados: $\vec{\mathbf{f}}^t$
  - Una base: describe vectores.
  - Un marco de referencia: describe puntos.

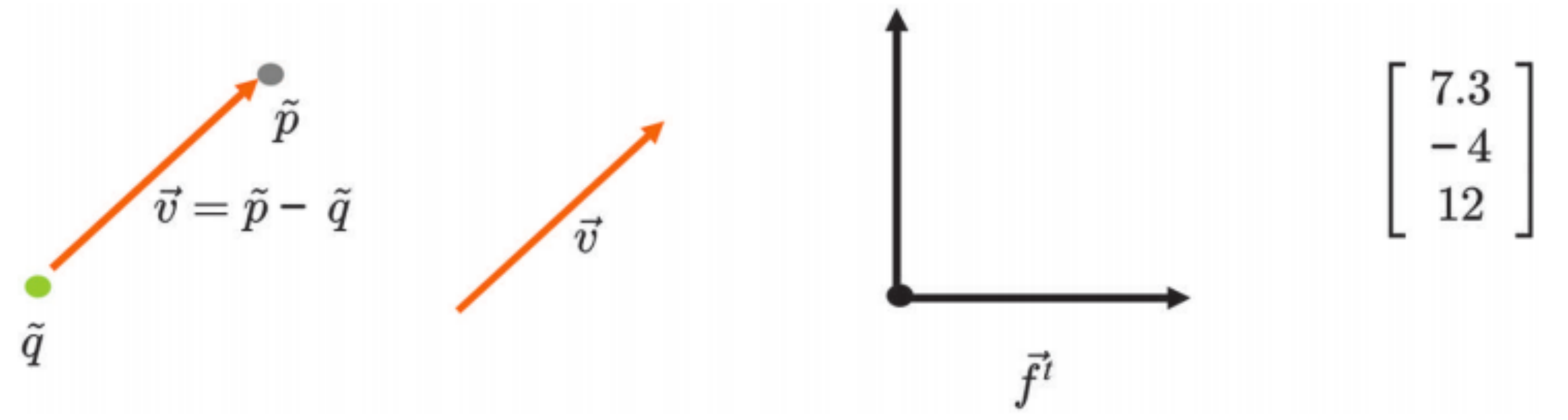


Figure 2.1
Geometric data types: points are shown as dots and vectors as arrows. Vectors connect two dots and do not change when translated. A frame represents a coordinate system and consists of one origin point and a basis of d vectors. A coordinate vector is a triple of real numbers.

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Vectores y vectores coordenados

- **Vector**: entidad geométrica abstracta que representa el movimiento entre dos puntos en el mundo. (ej. una milla al norte)
- **Vector coordenado**: conjunto de números reales utilizados para especificar un vector, una vez que se ha definido un sistema coordenado.
- **Espacio vectorial V**: conjunto de elementos $\vec{v}$ que satisfacen ciertas reglas, p.e.
  - operación **adición** (toma dos vectores y los transforma a un tercer vector) - asociativa y conmutativa.
  - operación **multiplicación** de un escalar real con un vector para obtener un vector - distributiva
- Muchas familias de objetos con estructura de espacio vectorial . Aquí nos interesamos en el espacio vectorial que consiste en "movimientos" entre puntos geométricos.
- En particular, **no** vamos a pensar en los vectores como un conjunto de tres números.

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Sistema coordenado (Base)

- Conjunto de vectores que se pueden usar para producir todo el conjunto de vectores del espacio utilizando las operaciones definidas.
- Un conjunto de vectores $\vec{b_1} \ldots \vec{b_n}$ es linealmente dependiente si existen escalares diferentes a cero $\alpha_1 \ldots \alpha_n$ tal que $\sum_i \alpha_i \vec{b_i} = \vec{0}$
- Si $\vec{b_1} \ldots \vec{b_n}$ son linealmente independientes, y podemos generar todo V usando adiciones y multiplicaciones, entonces el conjunto $\vec{b_i}$ se llama una base de V.
- n es la dimensión de la base/espacio.
- Para movimientos libres en el espacio la dimensión es 3.
- Nos referimos a cada uno de los vectores de una base como eje (axis).
- Podemos usar una base para producir cualquier vector del espacio. Esto se puede expresar usando un conjunto único de coord. c_i

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Sistema coordenado (Base)

$$\overrightarrow{v} = \sum_i c_i \overrightarrow{b_i}$$

- O escrito de otra forma:

$$\vec{v} = \sum_i c_i \vec{b_i} = \begin{bmatrix} \vec{b_1} & \vec{b_2} & \vec{b_3} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}.$$

- Podemos abreviarlo:

$$\vec{v} = \vec{\mathbf{b}}^t \mathbf{c}$$

- donde $\vec{v}$ es un vector, $\vec{\mathbf{b}}^t$ es un renglón de vectores base y $\mathbf{c}$ es un vector coordenado (columna).

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Transformaciones Lineales

- Una transformación lineal $\mathcal{L}$ es una transformación de V a V que satisface las siguientes propiedades:

$$\mathcal{L}(\vec{v}+\vec{u}) = \mathcal{L}(\vec{v}) + \mathcal{L}(\vec{u})$$

$$\mathcal{L}(\alpha\vec{v}) = \alpha\mathcal{L}(\vec{v}).$$

- Usamos la notación $\vec{v} \Rightarrow \mathcal{L}(\vec{v})$ para decir que el vector $\vec{v}$ se transforma al vector $\mathcal{L}(\vec{v})$ por medio de $\mathcal{L}$.

- La clase de transformaciones lineales es exactamente la clase que se puede expresar usando matrices (la transformación refleja su efecto en los vectores base).

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Transformaciones Lineales

- Estas propiedades implican la relación:

$$\vec{v} \Rightarrow \mathcal{L}(\vec{v}) = \mathcal{L}\left(\sum_i c_i \vec{b_i}\right) = \sum_i c_i \mathcal{L}(\vec{b_i})$$

- Lo que podemos escribir también cómo:

$$\begin{bmatrix} \vec{b_1} & \vec{b_2} & \vec{b_3} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \Rightarrow \begin{bmatrix} \mathcal{L}(\vec{b_1}) & \mathcal{L}(\vec{b_2}) & \mathcal{L}(\vec{b_3}) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

- Cada uno de los tres vectores $\mathcal{L}(\vec{b_i})$ es un elemento del campo V.

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Transformaciones Lineales

- Cada uno de éstos vectores se puede escribir como una combinación lineal:

$$\mathcal{L}(\vec{b_1}) = \begin{bmatrix} \vec{b_1} & \vec{b_2} & \vec{b_3} \end{bmatrix} \begin{bmatrix} M_{1,1} \\ M_{2,1} \\ M_{3,1} \end{bmatrix}$$

- para valores apropiados de $M_{j,l}$.
- Haciendo lo mismo para todos los vectores base, tenemos:

$$\begin{bmatrix} \mathcal{L}(\vec{b_1}) & \mathcal{L}(\vec{b_2}) & \mathcal{L}(\vec{b_3}) \end{bmatrix} = \begin{bmatrix} \vec{b_1} & \vec{b_2} & \vec{b_3} \end{bmatrix} \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix}$$

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Transformaciones Lineales

- En resumen, podemos usar una matriz para transformar un vector en otro:

$$\vec{\mathbf{b}}^t \mathbf{c} \Rightarrow \vec{\mathbf{b}}^t M \mathbf{c}$$

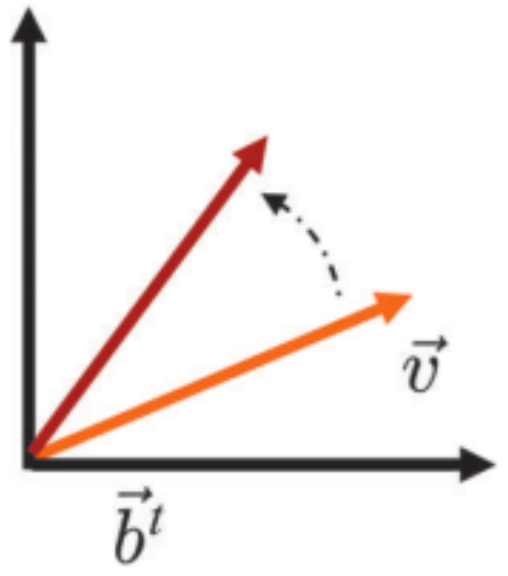


Figure 2.2
A vector undergoes a linear transformation $\vec{v} = \vec{\mathbf{b}}^t \mathbf{c} \Rightarrow \vec{\mathbf{b}}^t M \mathbf{c}$. The matrix M depends on the chosen linear transformation.

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Transformaciones Lineales

- Si aplicamos la transformación a cada vector de la base, obtenemos una nueva base:

$$\begin{bmatrix} \vec{b_1} & \vec{b_2} & \vec{b_3} \end{bmatrix} \Rightarrow \begin{bmatrix} \vec{b_1} & \vec{b_2} & \vec{b_3} \end{bmatrix} \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix}$$

$$\vec{\mathbf{b}}^t \Rightarrow \vec{\mathbf{b}}^t M$$

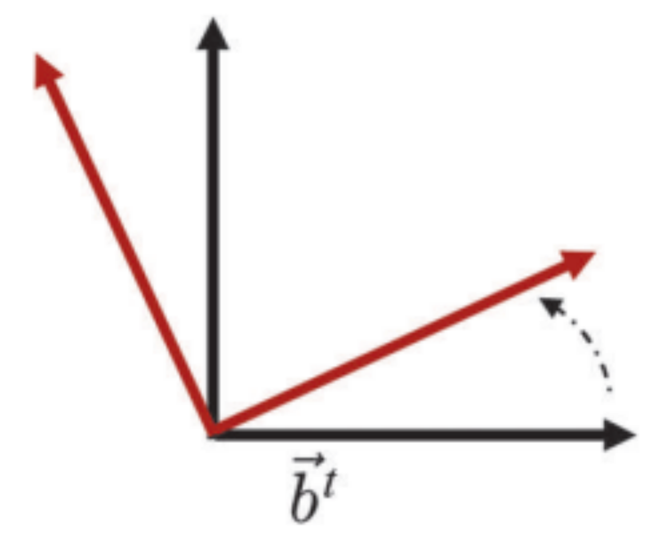


Figure 2.3
A basis undergoes a linear transformation $\vec{\mathbf{b}}^t \Rightarrow \vec{\mathbf{b}}^t M$.

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

- Y por supuesto es válido multiplicar una matriz por un vector coordenado:

$$\mathbf{c} \Rightarrow M\mathbf{c}$$

Identidad e Inversa

- El transformación identidad mantiene los vectores sin cambios. La matriz es:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- La inversa de una matriz M es la matriz única M^{-1} con la propiedad $MM^{-1} = M^{-1}M = I$.
- Representa la transformación inversa sobre vectores.
- Si una transformación lineal transforma más de un vector de entrada al mismo vector de salida, la transformada no será invertible y la matriz asociada no tendrá inversa.
- En CG cuando elegimos transformaciones de 3D a 3D para mover objetos en el espacio en el mayor de los casos no tiene sentido usar transformaciones no-invertibles.

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Matrices para cambios de base

- Además de usarse para describir transformaciones (⇒), una matriz se puede usar para describir igualdades (=) entre un par de bases o un par de vectores.
- Podemos dar una relación de equivalencia entre bases:

$$\vec{\mathbf{a}}^t = \vec{\mathbf{b}}^t M \qquad\qquad \vec{\mathbf{a}}^t M^{-1} = \vec{\mathbf{b}}^t$$

- Y así cambiar la base de vectores coordenados:

$$\vec{v} = \vec{\mathbf{b}}^t \mathbf{c} = \vec{\mathbf{a}}^t M^{-1} \mathbf{c}$$

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Producto punto

- Los vectores en 3D también tienen la operación de producto punto $\vec{v} \cdot \vec{w}$ que toma dos vectores y regresa un número real.
- El producto punto nos permite definir la longitud cuadrada (norma cuadrada) de un vector:

$$\|\vec{v}\|^2 := \vec{v} \cdot \vec{v}$$

- El producto punto se relaciona con el ángulo entre dos vectores:

$$\cos(\theta) = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\|\|\vec{w}\|}$$

- Decimos que dos vectores son **ortogonales** si $\vec{v} \cdot \vec{w} = 0.$
- Decimos que una **base** es **ortonormal** si todos los vectores base son de longitud unitaria y ortogonales entre si.

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Producto punto

- Producto punto con una base ortonormal:

$$\vec{\mathbf{b}}^t\mathbf{c} \cdot \vec{\mathbf{b}}^t\mathbf{d} = \left(\sum_i c_i \vec{b}_i\right) \cdot \left(\sum_j d_j \vec{b}_j\right)$$

$$= \sum_i \sum_j c_i d_j \left(\vec{b}_i \cdot \vec{b}_j\right)$$

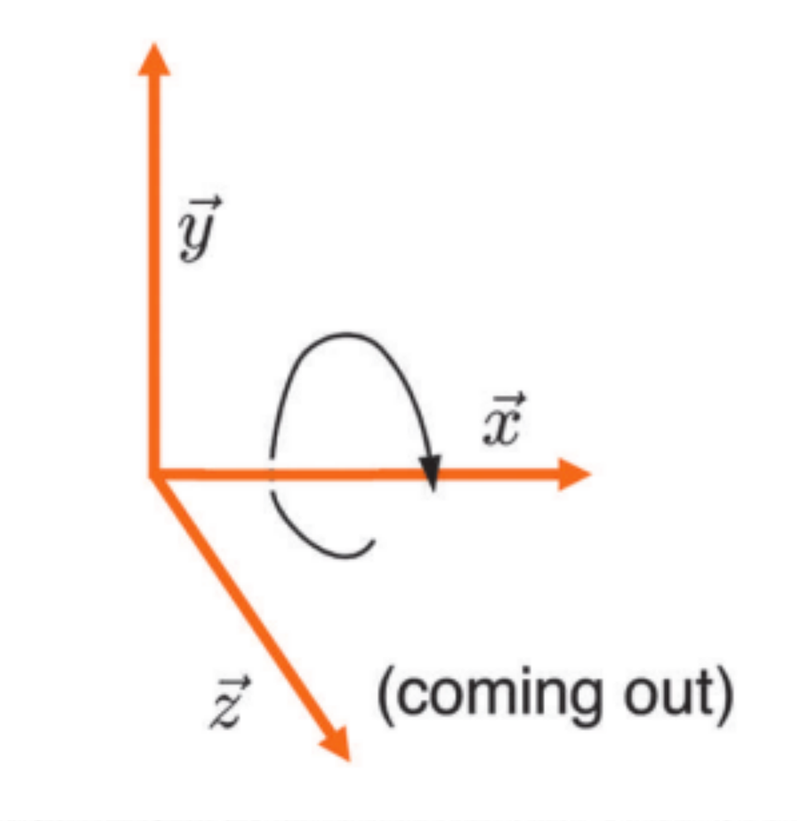
$$= \sum_i c_i d_j$$

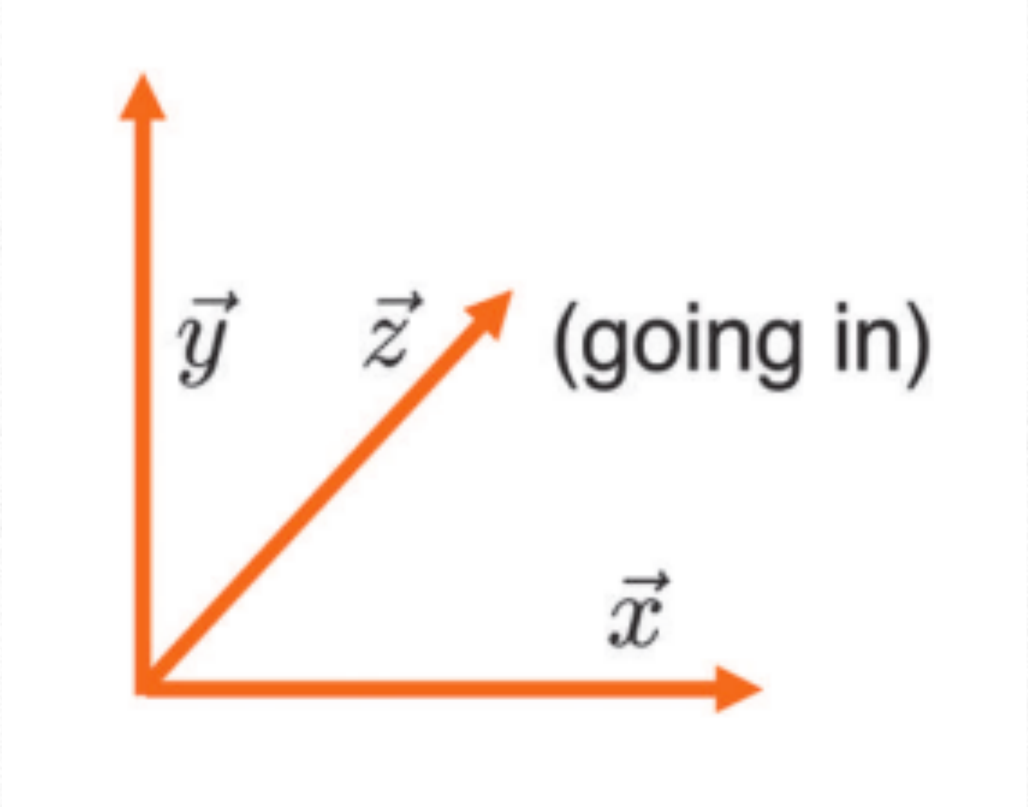
S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Más estructura

- Decimos que una **base ortonormal** en **2D** es de **mano derecha** si el segundo vector base se puede calcular a partir del primero con una rotación de 90 grados en sentido contrario a las manecillas del reloj. (el orden de los vectores en la base es claramente importante)
- Decimos que una **base ortonormal** en **3D** es de **mano derecha** si los tres vectores base (ordenados) se arreglan:

Mano derecha





Mano izquierda

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Producto vectorial

- En 3D también se puede definir la operación de producto vectorial o producto cruz que toma dos vectores y sale un vector definido como:

$$\vec{v} \times \vec{w} := \|v\|\|w\| \sin(\theta)\vec{n}$$

- donde $\vec{n}$ es un vector unitario que es ortogonal al plano generado por $\vec{v}$ y $\vec{w}$ tal que $[\vec{v}, \vec{w}, \vec{n}]$ forman una base de mano derecha.

$$(\vec{\mathbf{b}}^t\mathbf{c}) \times (\vec{\mathbf{b}}^t\mathbf{d}) = \begin{bmatrix} c_2d_3 - c_3d_2 \\ c_3d_1 - c_1d_3 \\ c_1d_2 - c_2d_1 \end{bmatrix}$$

S.J. Gortler. Foundations of 3D Computer Graphics. MIT Press. 2012.

Translación de objetos en 2D

- Podemos transladar puntos en el plano (x,y) a nuevas posiciones sumando cantidades de translación a las coordenadas de los puntos.
- Para cada punto P(x,y) a ser movido por d_x unidades paralelo al eje-x y d_y coordenadas paralelo al eje-y, el nuevo punto se escribe:

$$x' = x + d_x, \quad y' = y + d_y.$$

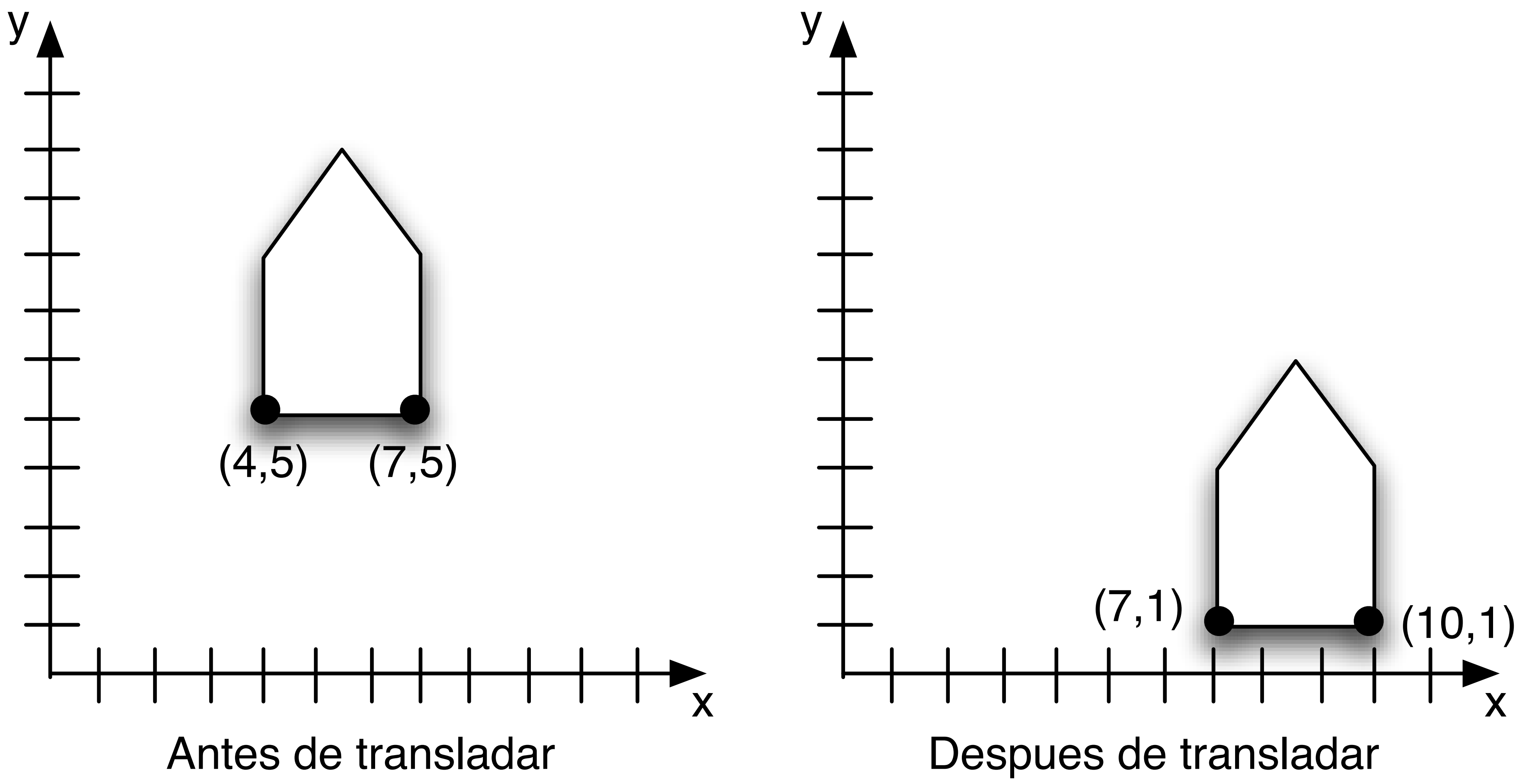
- Si definimos los vectores columna:

$$P = \begin{bmatrix} x \\ y \end{bmatrix}, P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, T = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

- también se puede expresar:

$$P' = P + T$$

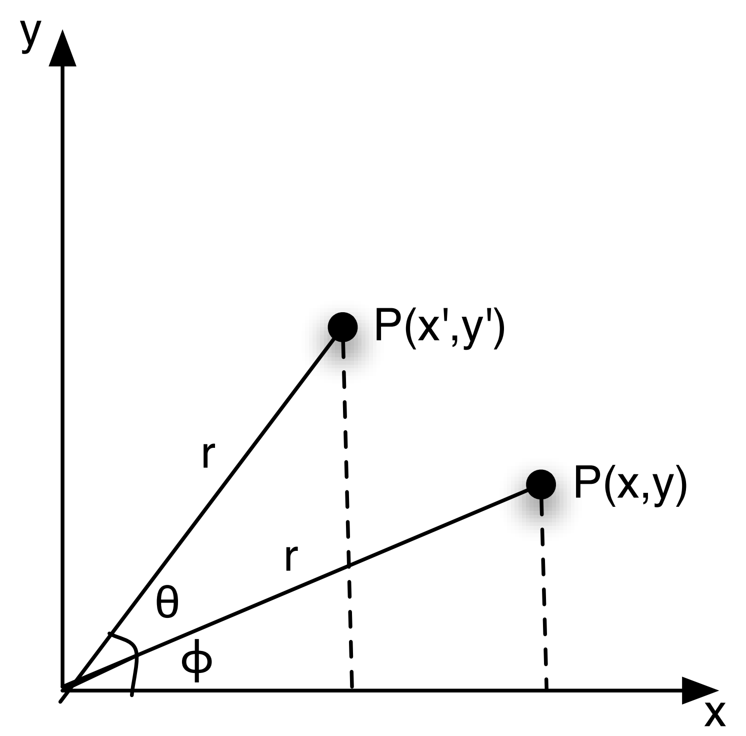
Translación de objetos en 2D



- Podríamos transladar un objeto aplicando la primera ecuación a cada punto pero como cada línea en un objeto está formada por un número infinito de puntos, este proceso sería infinito.
- También podemos transladar un objeto moviendo solamente los puntos extremos de cada línea y dibujando una nueva línea entre ellos.

Rotación de objetos en 2D

- Los ángulos positivos se miden en sentido contrario a las manecillas del reloj (CCW) de x hacia y.
- Para ángulos negativos (CW) se pueden usar las identidades *cos(-θ)=cos(θ)* y *sin(-θ) = -sin(θ)*.
- La ecuación de rotación se puede obtener de la figura:



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

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

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

Rotación de objetos en 2D

- Los puntos se pueden rotar por un ángulo θ alrededor del origen.
- Una rotación se define:

$$x' = x \cdot \cos\theta - y \cdot \sin\theta, \quad y' = x \cdot \sin\theta + y \cdot \cos\theta.$$

- En forma matricial:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \text{ o } P' = R \cdot P$$

donde R es la matriz de rotación: