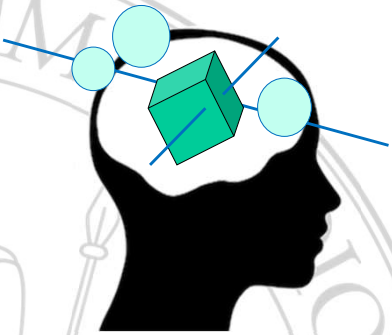
Marco Tarini - Computer Graphics 2020/2021
Università degli Studi di Milano

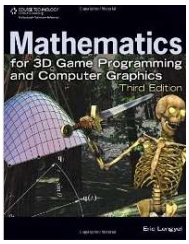
Vector and Point algebra



1

Algebra di punti e vettori – come studiare

- ✓ Anche se non si tratta di CG, queste basi sono necessarie ad una piena comprensione degli argomenti
  - ⇒ (e vengono chieste all'esame)
- ✓ Come appianare le eventuali lacune:
  - ⇒ Libri di testo consigliati
  - ⇒ In rete



Possibile libro di testo:
Mathematics for 3D Games ans Computer Graphics
Eric Lengyel
Chapters 2, 3

2

Points and vectors: what they are

- ✓ Points
  - ⇒represent positions in space
- ✓ Vectors
  - ⇒represent displacements in space
  - ⇒they have no position!
  - ⇒they have a *length*
  - ⇒they have a *direction*
  - ⇒used to move in space

Points and vectors: we draw them as…

- ✓ a dot
- ✓ an arrow

3

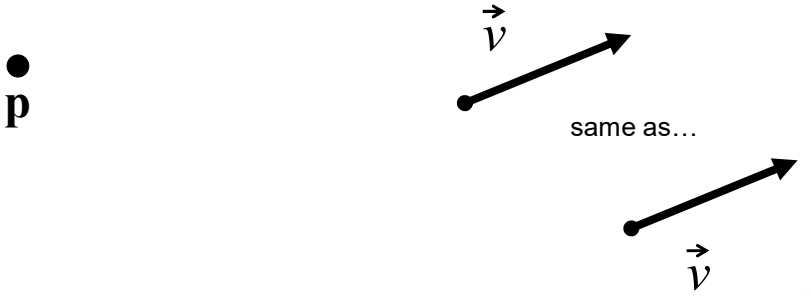
Points & Vectors

	represents:	examples:	imagine it as…
a Point	A position A location	Where something is The center of a sphere	a small floating dot :-D
a Vector	A displacement The difference between 2 points. The vector that connects them.	The velocity of an object The gravity acceleration How to reach point A from point B	a small arrow :-D (with a given length and direction)

4

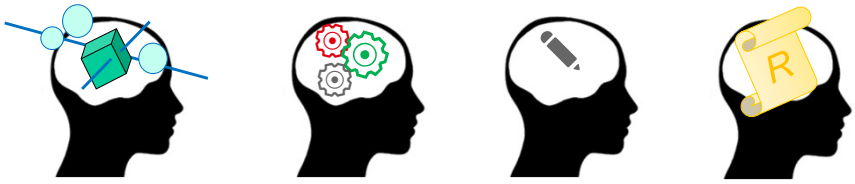
Points and vectors: we draw / write them as…

- ✓ Point: a dot
- ✓ Vector: a small arrow
  - ⇒ Note: the arrow is drawn in some place, but they have no position



5

Point, vector *algebra*



spatial intuition

operational: how to compute

syntax: how to write

rules: to manipulate equations

6

Points & vectors algebra

✓ In the following, make sure to know / understand each operation we will see in 3 ways:

intuitive / spatial: what does it do conceptually

operational: how to compute the result
(1) starting from the coordinates of the operand(s)
(2) (for products only) also, starting from the angles between the two operands, and their lengths

syntactic: how to write them down
(1) on paper (math-notation)
(2) in a programming language (in a C++ library, GLSL…)

7

Point, vector, versor *algebra*

✓ *A*lso, familiarize with the way the operations behave, i.e. with their…

rules such as
(1) commutativity? associativity? (of each operation)
(2) distributivity? (between pairs of operations)
(3) inverse operation? identity element? absorbing element?

8

Point, Vectors, Versors: Internal representation

✓triplets of Cartesian Coordinates

⇒(scalar values)

- they are the of the point/vector

⇒e.g.:

```
class Vector3 {
  // fields:
  public float coords[3];

  // methods:
  …
}
```

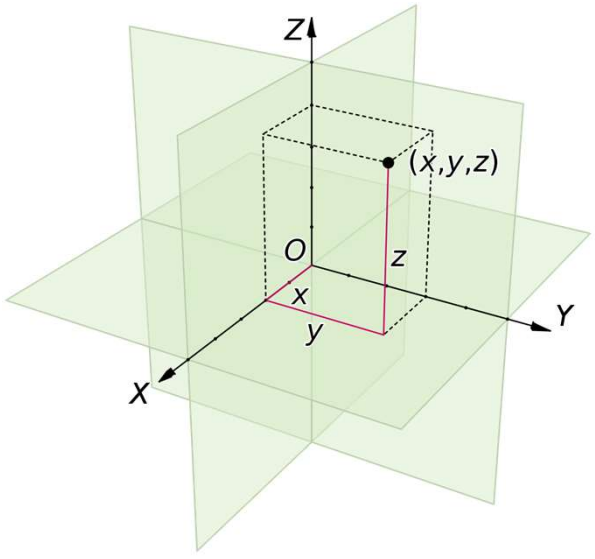
or:

```
class Vector3 {
  // fields:
  public float x, y, z;

  // methods:
  …
}
```

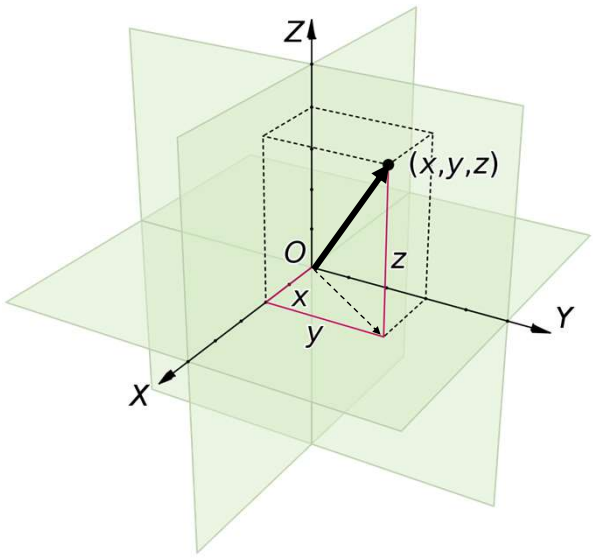
9

Expressed in a (“Cartesian”) Coordinate System!



10

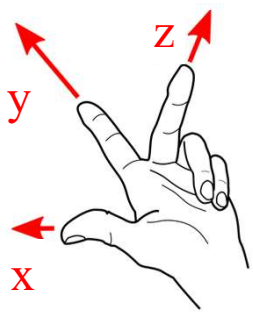
Expressed in a (“Cartesian”) Coordinate System!

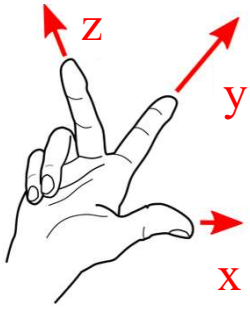


11

Right-handed or left-handed coord. system?

✓Ambiguity: how are we *imagining* things?





12

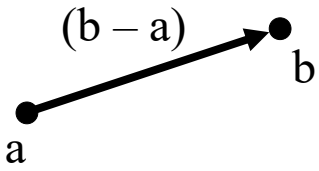
Caveat: one data type, multiple semantics

- ✓ Even if, often, libraries/languages choose to use the same data type ("vec3d", "Point3D", "vector" etc) for 3D points & 3D vectors (&… 3D versors, colors, etc) they are not the same thing,
  - ⇒ that's nothing new!
    we use the same scalar data types ("float", "doubles") for widely different things (e.g. weight, or volume, or temperature).
- ✓ They important thing is to operate on them accordingly.
  - ⇒ e.g.: not ok to sum a temperature with a surface
  - ⇒ e.g.: ok to divide a weight for a volume (and get a specific weight)
- ✓ which operations make sense on points, vectors, versors?
  - ⇒ that is, what about their *algebra* ?

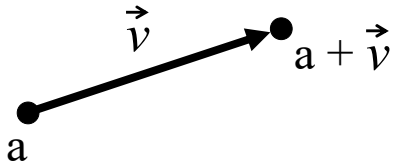
13

Point and vector algebra (summary)

- ✓ Difference:
  point – point = vector



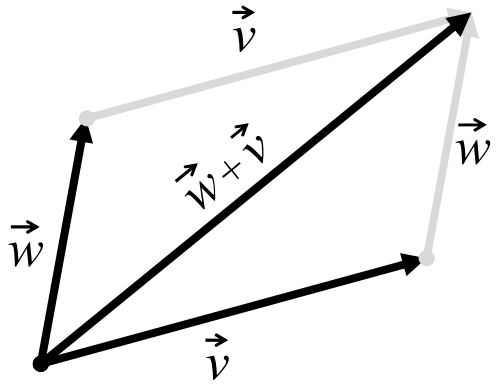
- ✓ Addition:
  point + vector = point



14

Vector algebra: operation between vectors

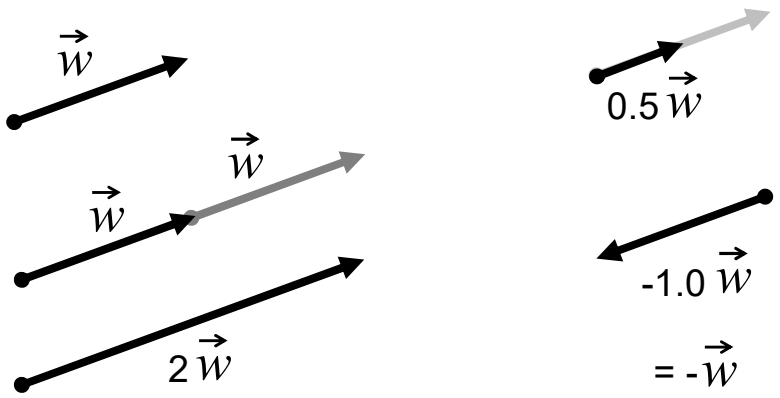
✓addition (between vectors):
vector + vector = vector



15

Vector algebra: operation between vectors

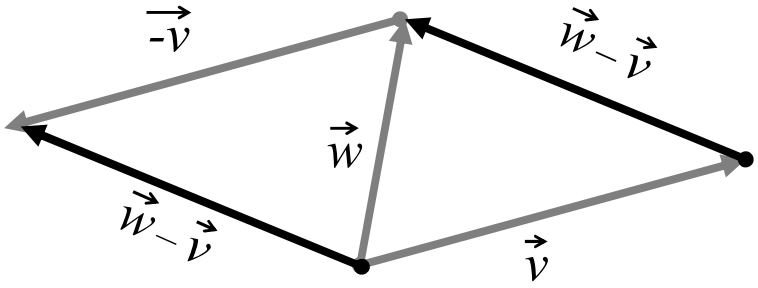
✓product of a vector with a scalar:
scalar · vector = vector



16

Vector algebra: operation between vectors

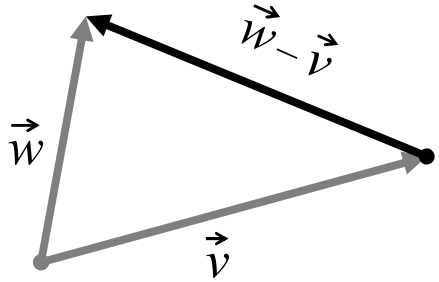
✓ difference (between vectors):
vector - vector = vector



17

Vector algebra: operation between vectors

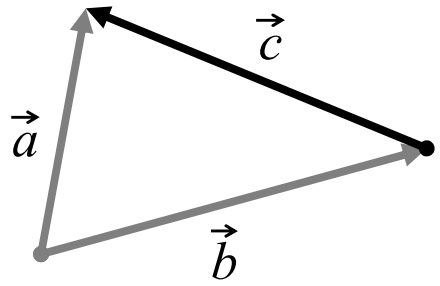
✓ difference (between vectors):
vector - vector = vector



18

Vector algebra: operation between vectors

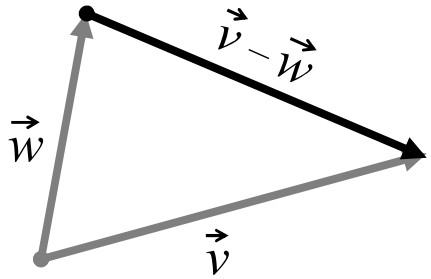
✓ difference (between vectors):
vector - vector = vector



19

Vector algebra: operation between vectors

✓ difference (between vectors):
vector - vector = vector



20

Vector algebra: operation between vectors

Linear operations :

- ✓ addition (between vectors):
  vector + vector = vector
- ✓ product with a scalar:
  scalar · vector = vector

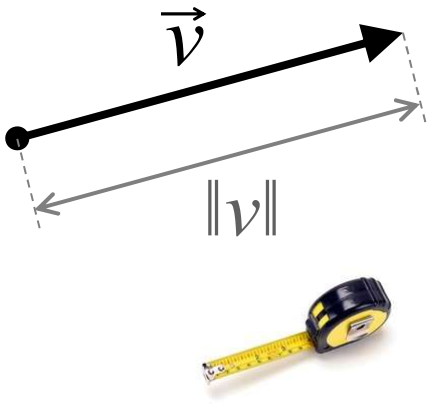
  ⇒therefore: interpolation (between vectors)

- ✓ opposite (flip verse):
  ⇒therefore: difference (between vectors)

21

Point and vector algebra

- ✓ Norm:
  norm(vector) = scalar
  ⇒or length / magnitude / Euclidean Norm / 2-Norm



More about this in the next lecture!

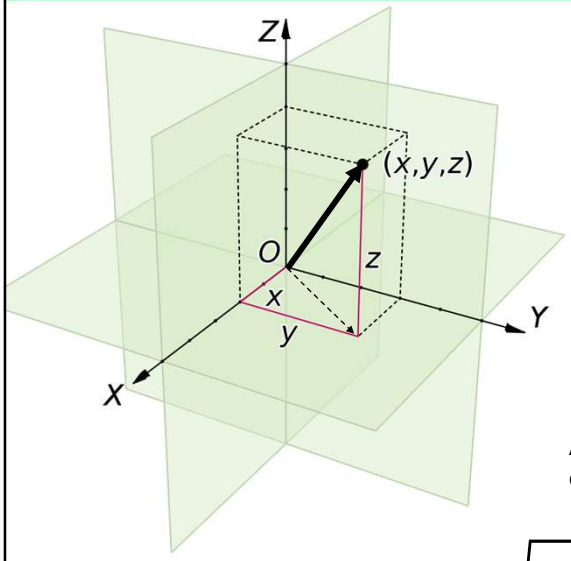
22

Norma di un vettore

- ✓Terminologia: detta anche:
  - ⇒lunghezza, magnitudine, estensione, norma, norma 2, norma Euclidea
- ✓Scritta come $\|\vec{v}\|$ o $\|\vec{v}\|_2$ o $|\vec{v}|$
- ✓Scritta nel codice (in librerie o linguaggi) come funzione o metodo length(v) o v.length() o v.norm()
- ✓Rappresenta l'estensione di un vettore

23

Calcolo della norma



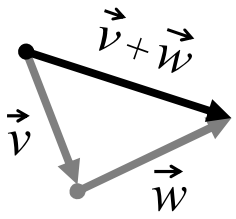
Applicando il teorema di pitagora (2 volte):

$$\|\vec{v}\| = \sqrt{{v_x}^2 + {v_y}^2 + {v_z}^2}$$

24

Usi e caratteristiche della norma

- ✓La norma è sempre ≥0, e 0 solo per vettori nulli
- ✓Distanza fra due punti a e b :
  norma della loro differenza $\|\mathrm{a}-\mathrm{b}\|$
- ✓Norma e prodotto con scalare:
  $\|k\,\vec{\boldsymbol{v}}\| = |k|\,\|\vec{\boldsymbol{v}}\|$
- ✓Norma e somma: (“disugualianza trangolare”)
  $\|\,\vec{\boldsymbol{v}} + \vec{\boldsymbol{w}}\,\| \leq \|\vec{\boldsymbol{v}}\| + \|\vec{\boldsymbol{w}}\|$



25

Un esempio di problema geometrico

- ✓Siano date due sfere, con i centri in posizione $\mathbf{c}_0$ e $\mathbf{c}_1$ e raggio r_0 e r_1
  C’è interesezione (si toccano / compenetrano?)
- ✓Soluzione: sì, se
  $$\|\mathbf{c}_0 - \mathbf{c}_1\| < r_0 + r_1$$
- ✓Cioè se…
  $$\|\mathbf{c}_0 - \mathbf{c}_1\|^2 < (r_0 + r_1)^2$$

Esercizi:

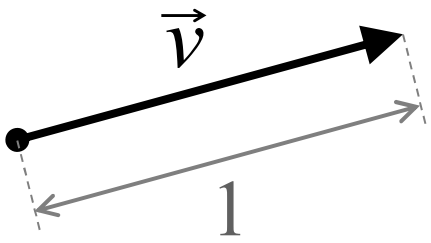
- ⇒Verifica il tipo della formula qui sopra, incluso il tipo (punto, vettore, scalare) di ogni sottoespressione
- ⇒Perchè la seconda espressione è più efficiente da computare della prima?

26

Unit vectors

Vettori a lunghezza 1

- ✓ anche detti:
  - ⇒vettori unitari
  - ⇒vettori normali
    o anche solo "normali"
    (sorattutto quando sono ortogonali ad una superficie qualcosa)
  - ⇒Versori
- ✓ A volte denotati con il cappuccio invece che la freccetta, così: $\hat{v}$
- ✓ Rappresentano una direzione
  - ⇒i versori sono un terzo dipo di "oggetto"



27

Normalizzazione

- ✓ Dato un vettore non unitario, ottenere il vettore unitario con la stessa direzione
  - ⇒In pratica, estrarre la direzione di un vettore
- ✓ Come: basta dividere per la sua norma (cioè scalare per 1/norma)

$$\hat{v} = \frac{\vec{v}}{\|\vec{v}\|}$$

- ✓ Nota: l'unico vettore che non può essere normalizzato è il vettore nullo (o degenere), quello di tutti 0
  - ⇒l'operazione causerebbe una divisione per zero

28