Theoretical Backgrounds of Audio & Graphics

Vectors Spaces

Angela Brennecke | Prof. Dr.-Ing.
Audio & Interactive Media Technologies

Filmuniversität Babelsberg
KONRAD WOLF

winter term 21/22

Motivation

"Computer graphics is concerned with the representation and manipulation of sets of geometric elements, such as points and line segments. The necessary mathematics is found in the study of various types of abstract spaces. […], we review the rules governing three such spaces: the (linear) vector space, the affine space, and the Euclidean space.

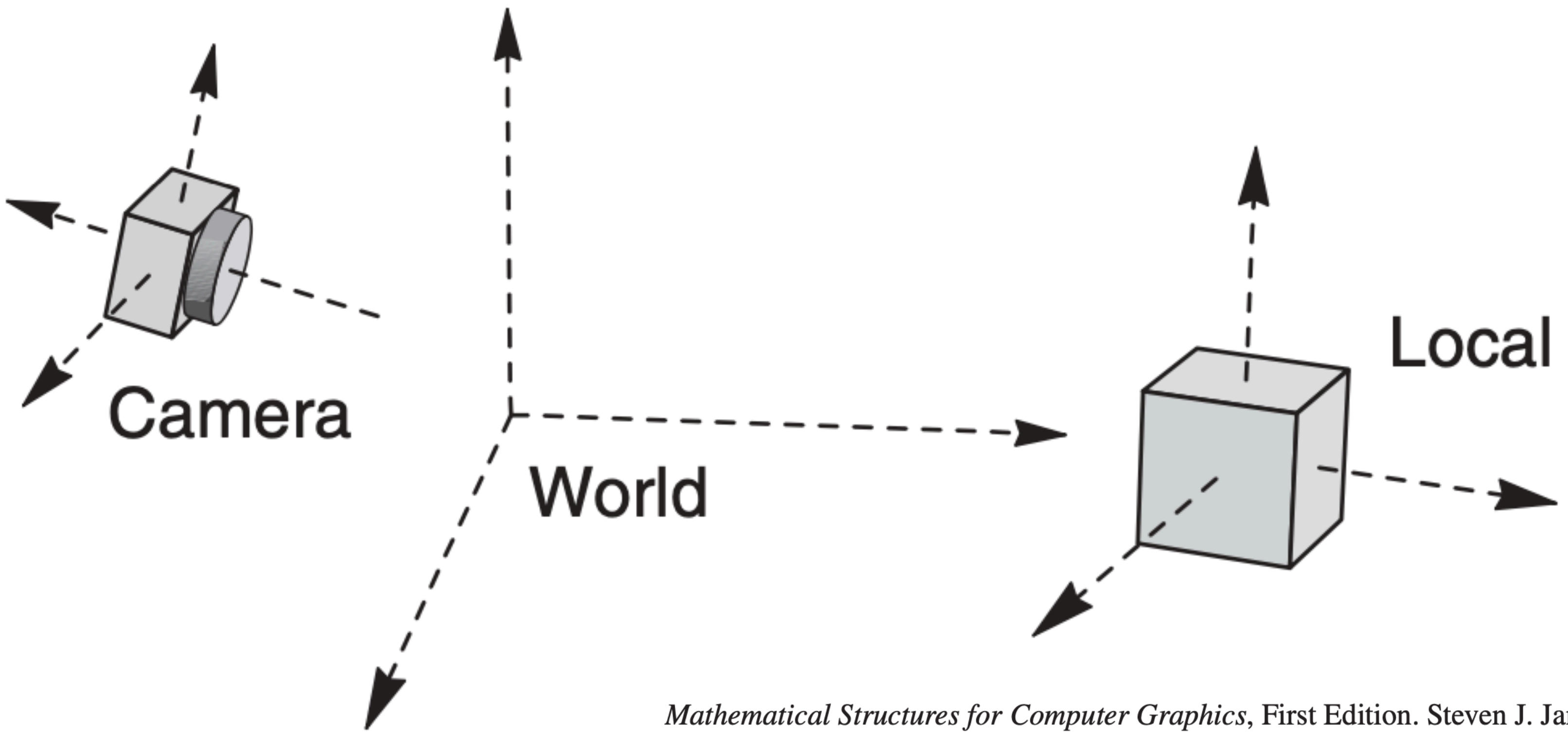
- The **(linear) vector space** contains only two types of objects: scalars, such as real numbers, and vectors.
- The **affine space** adds a third element: the point.
- **Euclidean spaces** add the concept of distance."

Angel, E. and Shreiner, D. (2012): **Interactive Computer Graphics—A Top-down Approach with Shader-based OpenGL**. Addison-Wesley, USA.

Motivation

- In computer graphics, we work with 2d or 3d **geometric objects**
- These are usually described as polygonal **vertex meshes** that are **located** in a specific 2d or 3d **scene**

- In order to describe their visual representation as well as their positioning in a 2D or 3D scene, the mathematics of the following notions and concepts are essential:
  - **scalars**
  - **vectors**
  - **points**
  - **matrices**
  - **transformations**



Mathematical Structures for Computer Graphics, First Edition. Steven J. Janke.

Linear Vector Spaces

- A linear vector space is defined by
  - a set of vectors **v**
    including a **zero vector** (0)
  - a set of scalar values **s,** for example,
    represented by real numbers

- It introduces two basic operations
  that work element-wise
  - **Vector-vector addition:**
    f(v1, v2) -> v
  - **Scalar-vector multiplication**:
    f(v1 * s) -> v

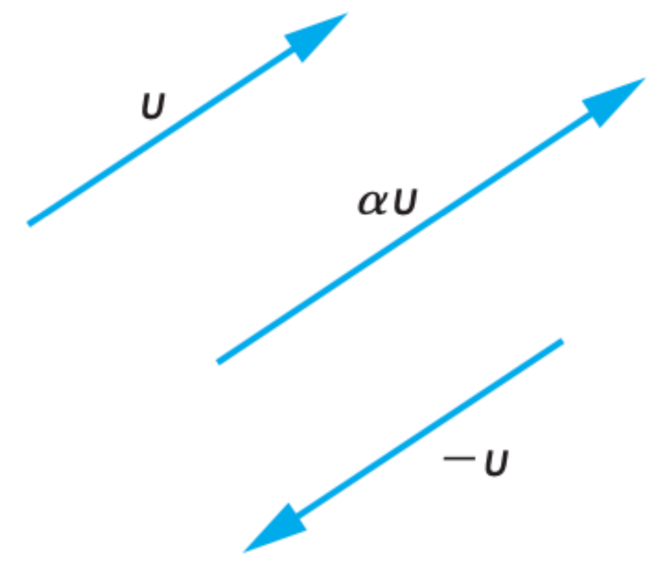


FIGURE B.2 Scalar–vector multiplication.

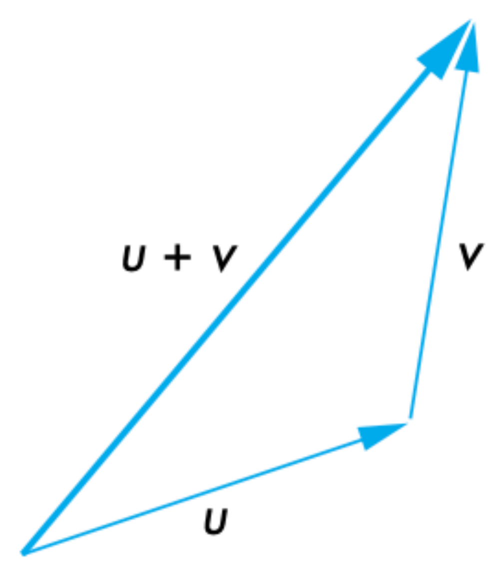


FIGURE B.3 Head-to-tail axiom for vectors.

Angel, E. and Shreiner, D. (2012): **Interactive Computer Graphics— A Top-down Approach with Shader-based OpenGL**. Addison-Wesley, USA.

Linear Vector Spaces

- Geometrically, vectors are interpreted as **directed line segments** or **translations**

- Mathematically, vectors are considered as **n-tuples** of scalars **s**
  v = (s1, s2, s3, …, sn)

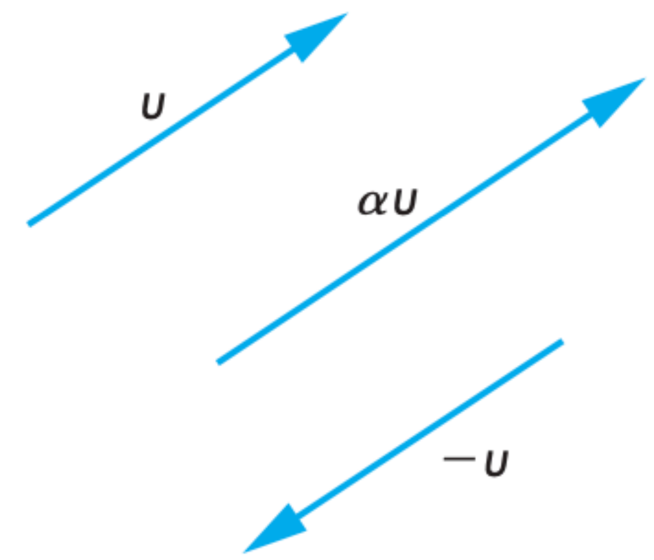


FIGURE B.2 Scalar–vector multiplication.

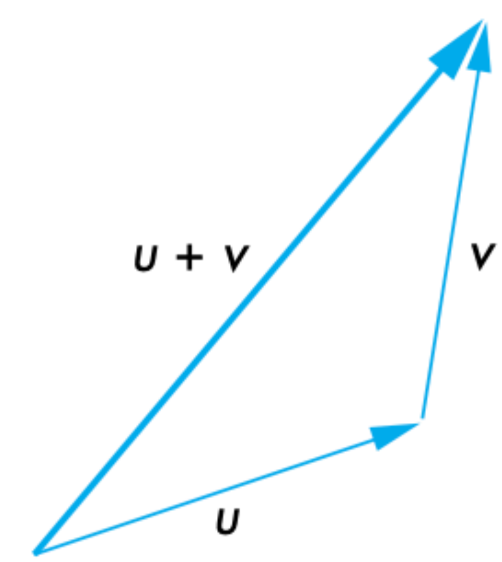


FIGURE B.3 Head-to-tail axiom for vectors.

Angel, E. and Shreiner, D. (2012): **Interactive Computer Graphics— A Top-down Approach with Shader-based OpenGL**. Addison-Wesley, USA.

Linear Vector Spaces

- **Basis vectors** of a vector space are **linearly independent** vectors
  - 2d: x = (1,0) and y = (0,1)
  - 3d: x = (1,0,0), y = (0,1,0) and z = (0,0,1)

- Every other vector of the vector space can be represented as a **linear combination** of the basis vectors of the vector space
  - (v1,v2,v3) = v1(1,0,0) + v2(0,1,0) + v3(0,0,1)

- The number of basis vectors that define the vector space also determine the **dimension** of the vector space

Affine Spaces

- An **affine space** is a vector space that introduces the set of points **P** to the vector space in addition to vectors **v** and scalars **s**

- Affines spaces add the operation
  - **Point-point subtraction**:
    f(p1, p2) -> v
    which yields a vector

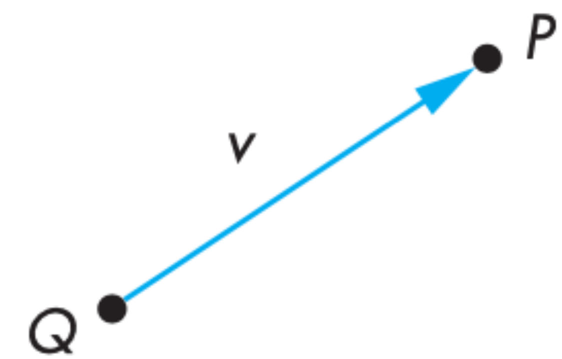


FIGURE 3.5 Point-vector addition.

Angel, E. and Shreiner, D. (2012): **Interactive Computer Graphics— A Top-down Approach with Shader-based OpenGL**. Addison-Wesley, USA.

Affine Spaces

- Geometrically, points are interpreted as **locations** in space

- Mathematically, points are specified like vectors as n-tuples of scalar values **s**
  - **P** = (s1, s2, s3, …, sn)

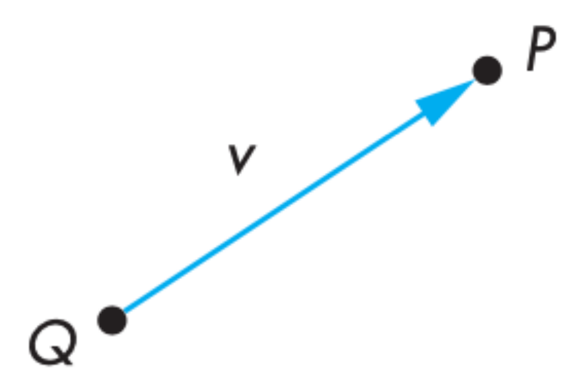


FIGURE 3.5 Point-vector addition.

Angel, E. and Shreiner, D. (2012): **Interactive Computer Graphics— A Top-down Approach with Shader-based OpenGL**. Addison-Wesley, USA.

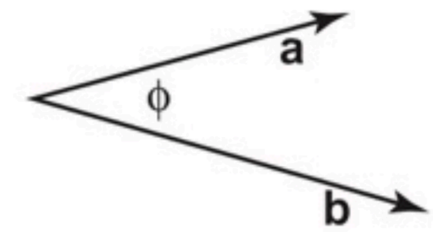
Euclidian Spaces

- **Euclidian spaces** are affine spaces that add an important operation
  - the **Dot product** also referred to as
    - **Scalar product** or
    - **Inner (dot) product**

- Euclidian spaces thus introduce the notion of
  - **distance / length** and
  - **angle** measurements

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos \phi$$

$$\cos \phi = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|}$$



$$\mathbf{a} \cdot \mathbf{b} = x_a x_b + y_a y_b + z_a z_b$$

Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

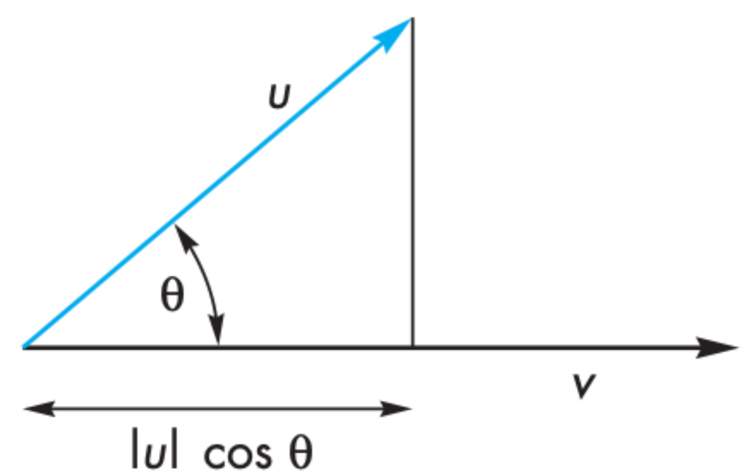
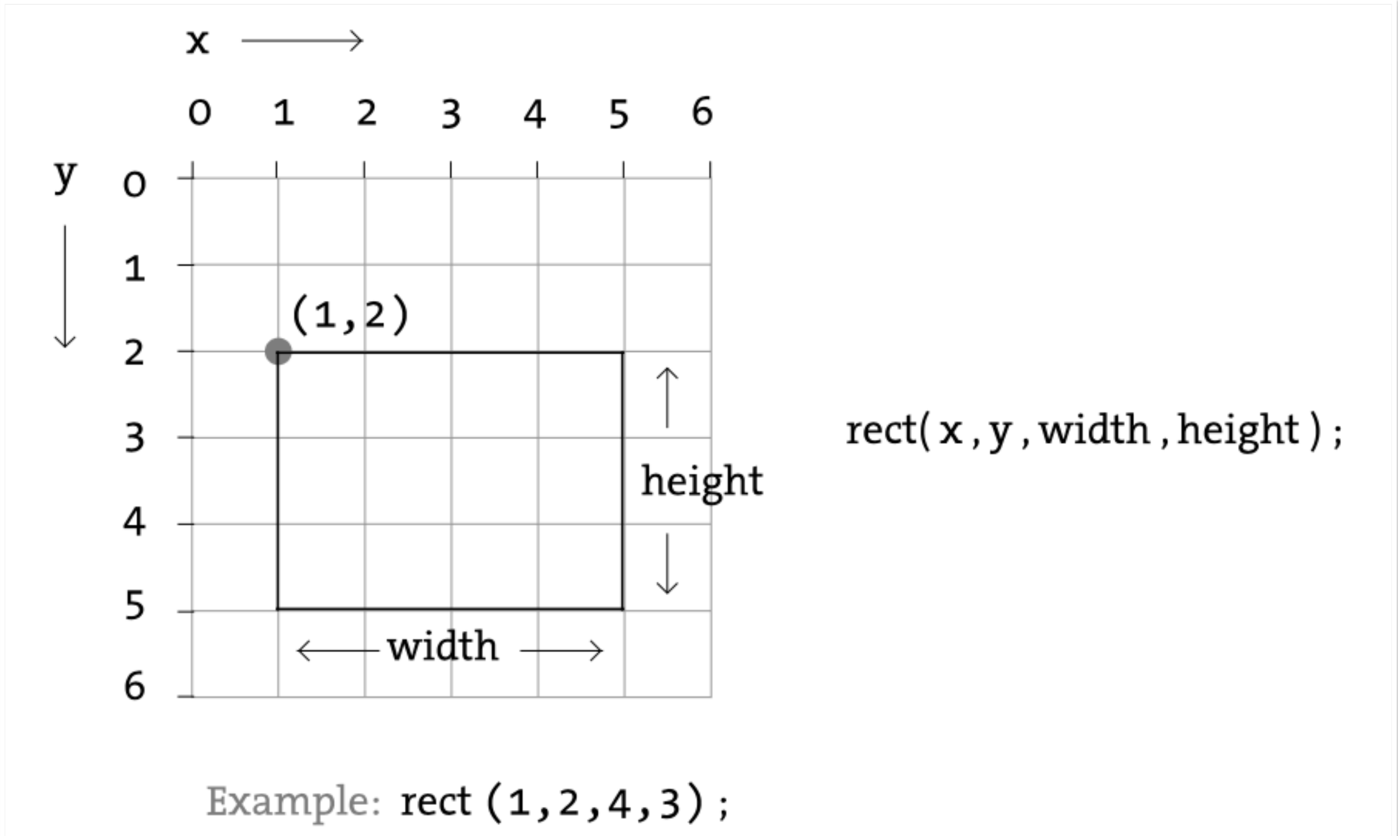


FIGURE 3.14 Dot product and projection.

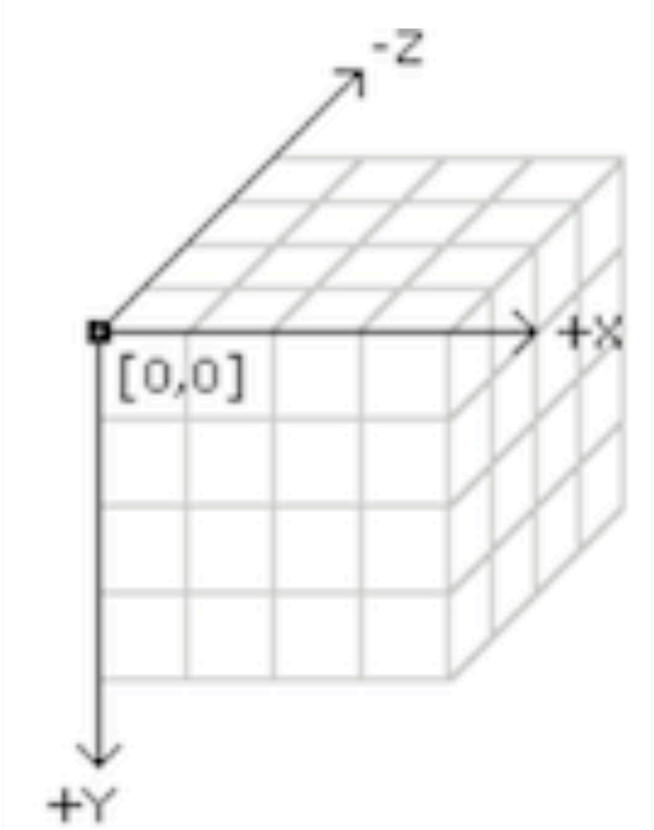
Angel, E. and Shreiner, D. (2012): **Interactive Computer Graphics— A Top-down Approach with Shader-based OpenGL**. Addison-Wesley, USA.

Cartesian Coordinate System

- A cartesian coordinate system: A set of **basis vectors** at **unit length** (e1, e2, e3, …, en) and a reference point called the **origin o**
- Vectors and points can be clearly defined in such a reference system by specifying their coordinates, linear combinations of the unit vectors



https://processing.org/tutorials/drawing/

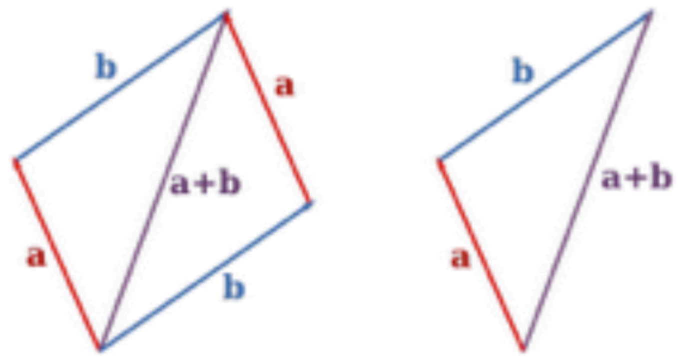


https://processing.org/tutorials/p3d/

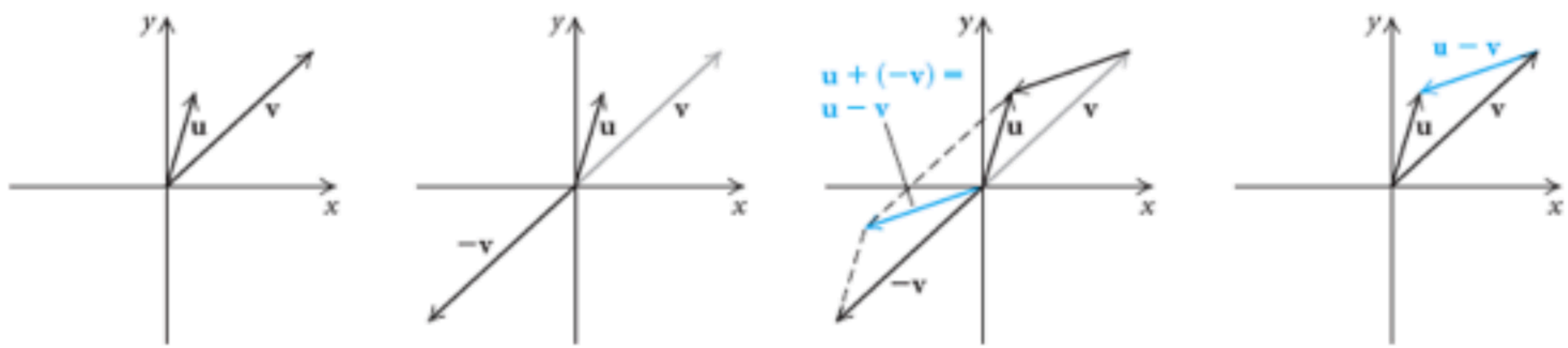
Basic Vector Operations

Addition & Subtraction

- Addition



- Subtraction



https://www.math10.com/en/geometry/vectors-operations/vectors-operations.html

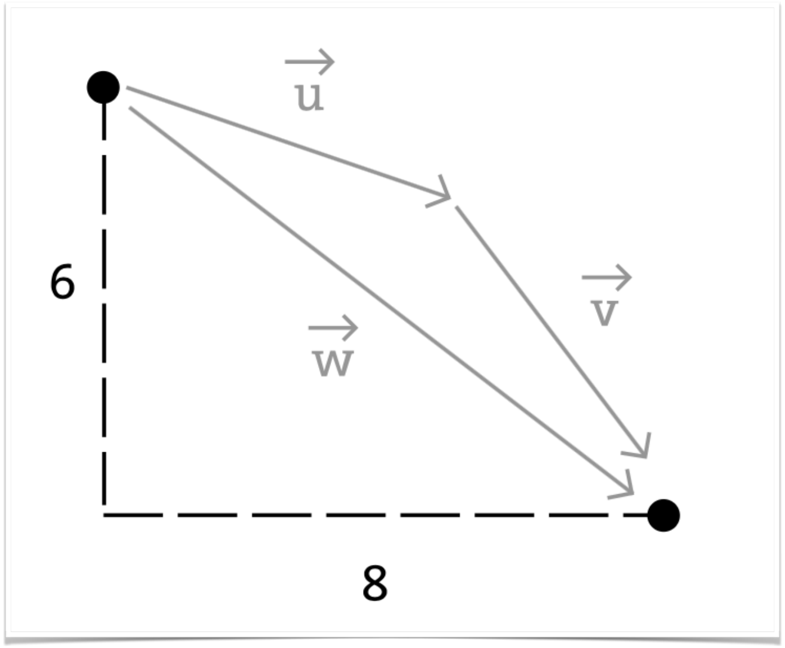
https://en.wikipedia.org/wiki/Euclidean_vector#Addition_and_subtraction

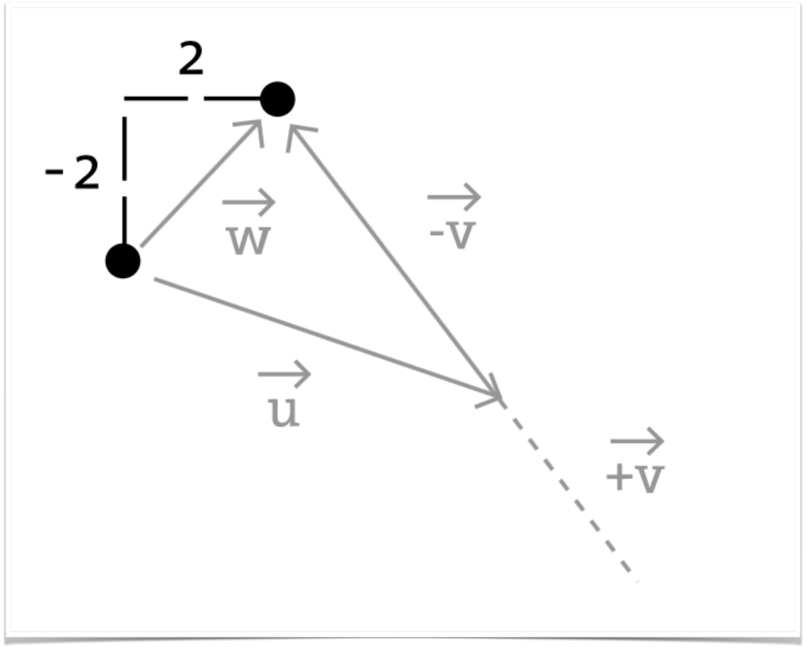
Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

Addition & Subtraction

$$w_x = u_x + v_x$$
$$w_y = u_y + v_y$$

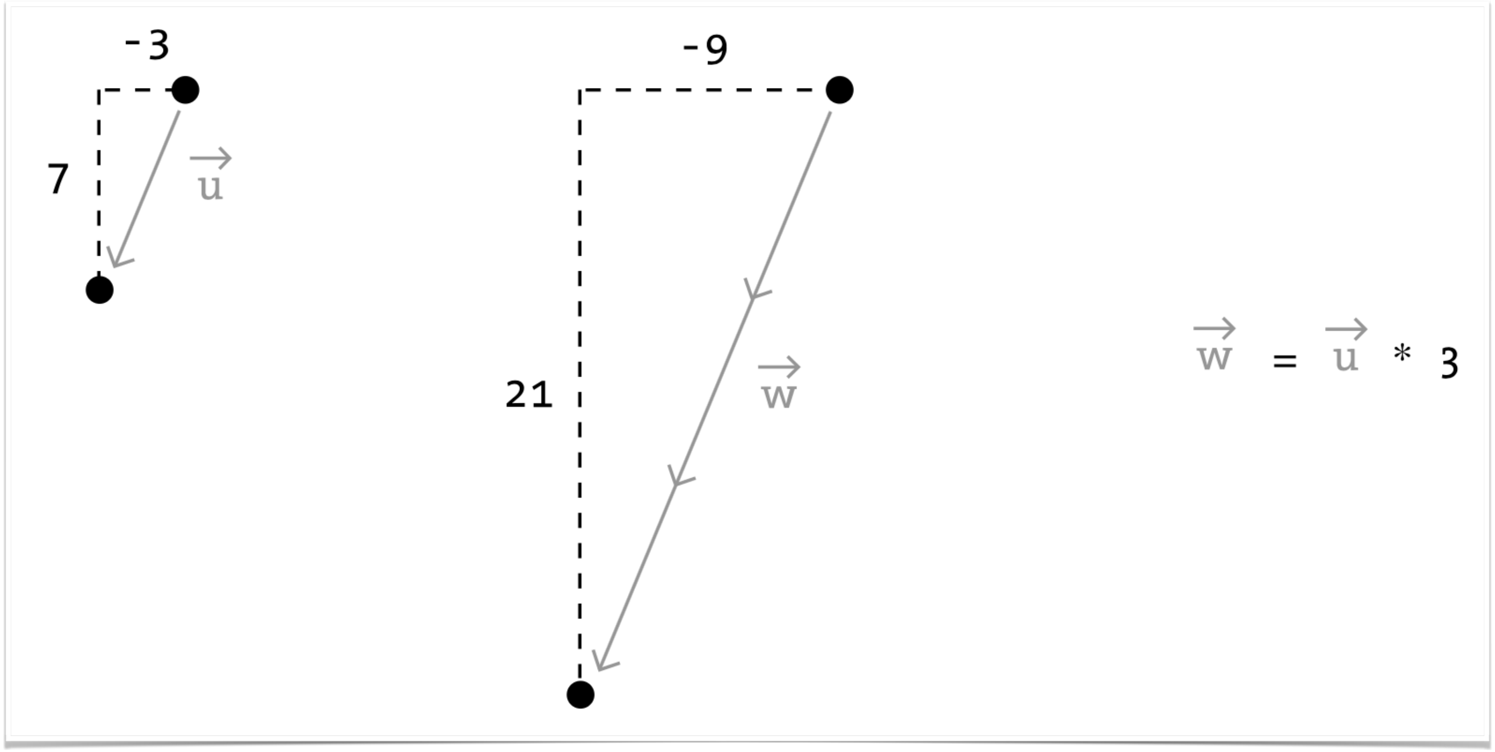
$$w_x = u_x - v_x$$
$$w_y = u_y - v_y$$

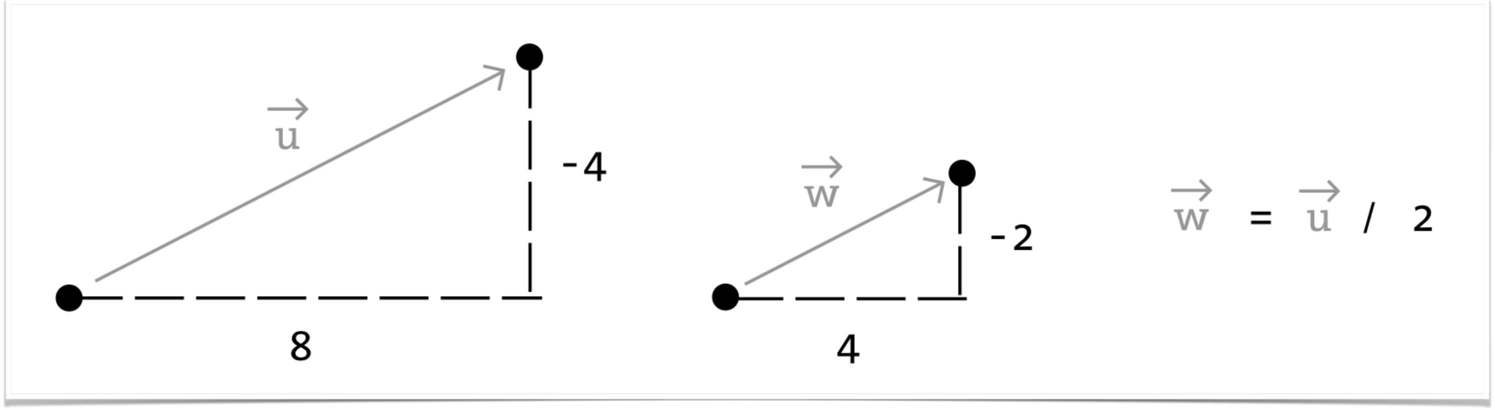




https://processing.org/tutorials/pvector/

Scalar Multiplication & Division



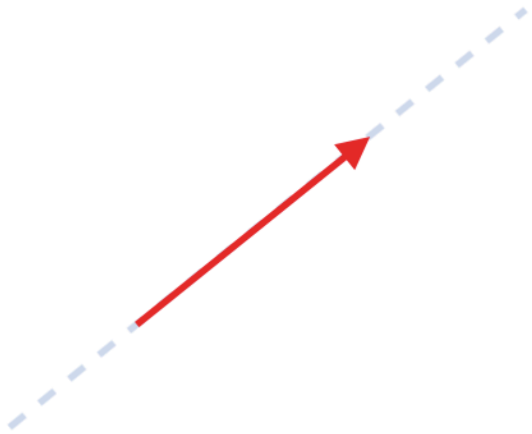


https://processing.org/tutorials/pvector/

Magnitude Calculation

- The **length** of a vector is also called the *magnitude*

$$\|\mathbf{a}\| = \sqrt{\sum_{i=1}^{n} a_i^2}$$



E.g. in 2D cartesian coordinates:

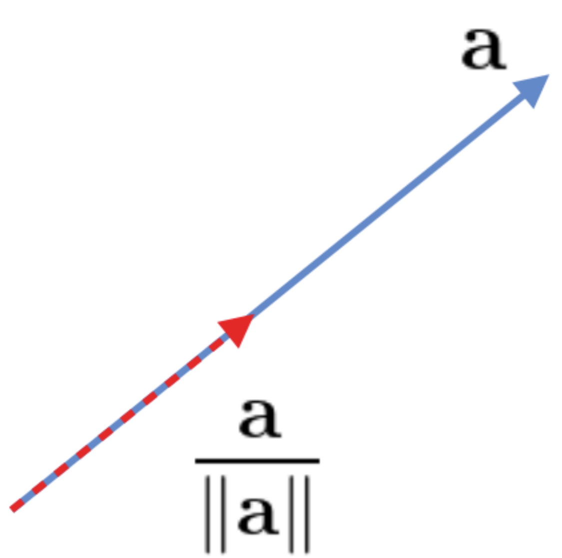
$$\|\mathbf{a}\| = \sqrt{x_a^2 + y_a^2}$$

Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

Normalization

- *Normalization* is the resizing of a vector to length 1 (unit size)
- Done through multiplication by

$$\frac{1}{\|\mathbf{a}\|}$$



- This is essential for many graphics operations

Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

The Dot Product

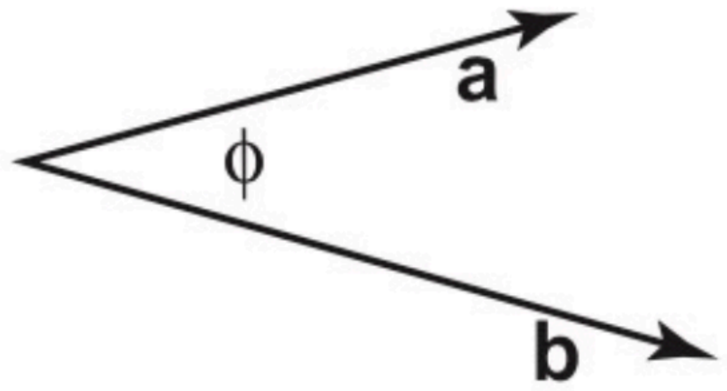
- The dot product is also known as the *scalar* product because it returns a **scalar**

- The dot product is of fundamental importance for computer graphic operations
    - The most common use is to compute the cosine of the angle between two vectors.
    - It supplies a measure of the difference between the directions in which the two vectors point.

Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

The Dot Product

- Returns a value related to its arguments' lengths and the angle between them



$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos \phi$$

Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

The Dot Product

- The dot product of two *n*-dimensional vectors **a** and **b**, written as **a** · **b**, is the scalar quantity given by the formula

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i$$

- This definition states that the dot product of two vectors is given by the sum of the products of each component
- In Cartesian coordinates in three dimensions, e.g.:

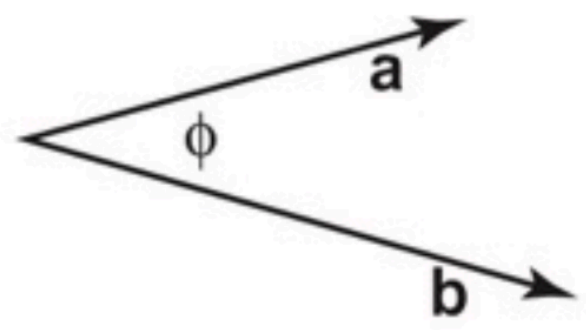
$$\mathbf{a} \cdot \mathbf{b} = x_a x_b + y_a y_b + z_a z_b$$

Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

The Dot Product

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos \phi$$

$$\cos \phi = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|}$$

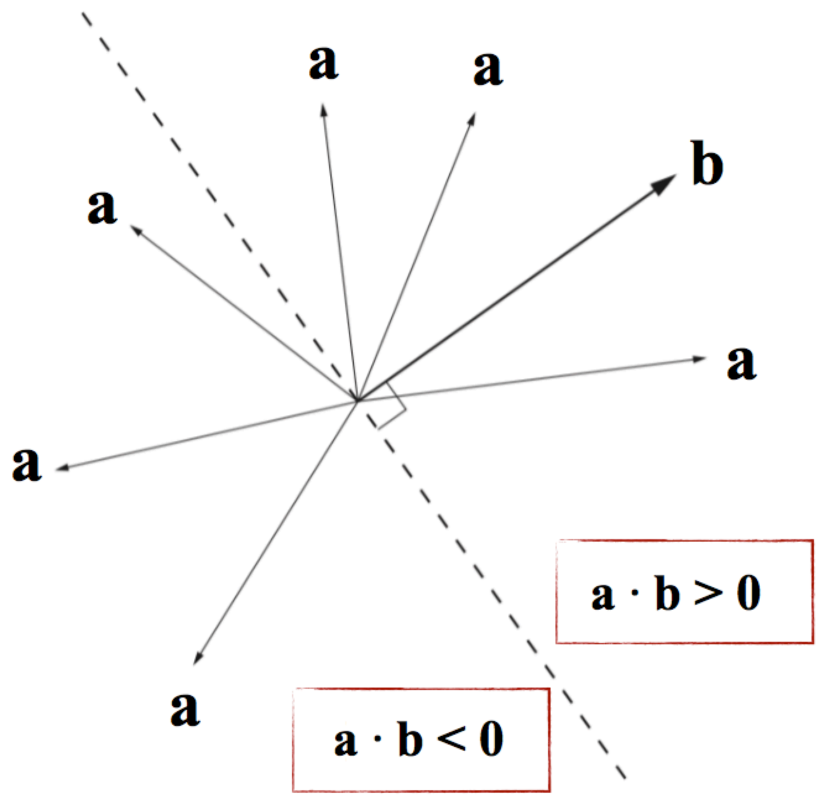


$$\mathbf{a} \cdot \mathbf{b} = x_a x_b + y_a y_b + z_a z_b$$

Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

The Dot Product

- The sign of the dot product tells us how close two vectors are to pointing in the same direction.
- Consider the plane passing through the origin and perpendicular to a vector **b**

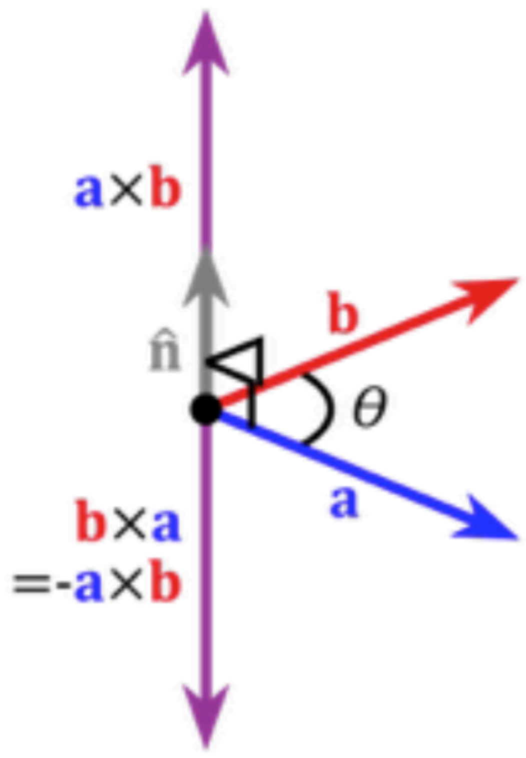


- Any vector lying on the same side of the plane as **b** yields a positive dot product with **b**,
- Any vector lying on the opposite side of the plane from **b** yields a negative dot product with **b**.

Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

The Cross Product

- Returns a new *vector* that is **perpendicular to both of the vectors** being multiplied together
  - One of its major uses in Computer Graphics is the calculation of a surface normal at a particular point given two distinct tangent vectors.



https://en.wikipedia.org/wiki/Euclidean_vector#Addition_and_subtraction

Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

The Cross Product

$$\mathbf{a} \times \mathbf{b} = (y_a z_b - z_a y_b, z_a x_b - x_a z_b, x_a y_b - y_a x_b).$$

- Given two 3D vectors **a** and **b**, the cross product **a** × **b** satisfies the equation

$$||\mathbf{a} \times \mathbf{b}|| = ||\mathbf{a}|| \; ||\mathbf{b}|| \sin \alpha$$

where α is the planar angle between **a** and **b.**

Gieseke, L. Mathematics for Audio & Graphics, WS17/18, Film University

References

- Buss, S. (2003): **3D Computer Graphics—A Mathematical introduction with OpenGL**. Cambridge University Press, New York, NY, USA.
- Angel, E. and Shreiner, D. (2012): **Interactive Computer Graphics—A Top-down Approach with Shader-based OpenGL**. Addison-Wesley, USA.
- Shiffman, D. (2012): **Nature of Code.** https://natureofcode.com/book/

- Lecture slides
  - Gieseke, L. Mathematics for Audio & Graphics, 2017/18, Film University
  - Jarosz, W., Computer Graphics, Fall 2016, Dartmouth College
  - Marschner S., Computer Graphics, Spring 2018, Cornell University
  - Schlechtweg, S. (2010/11), „Computergraphik Grundlagen“, Hochschule Anhalt (FH) Köthen.