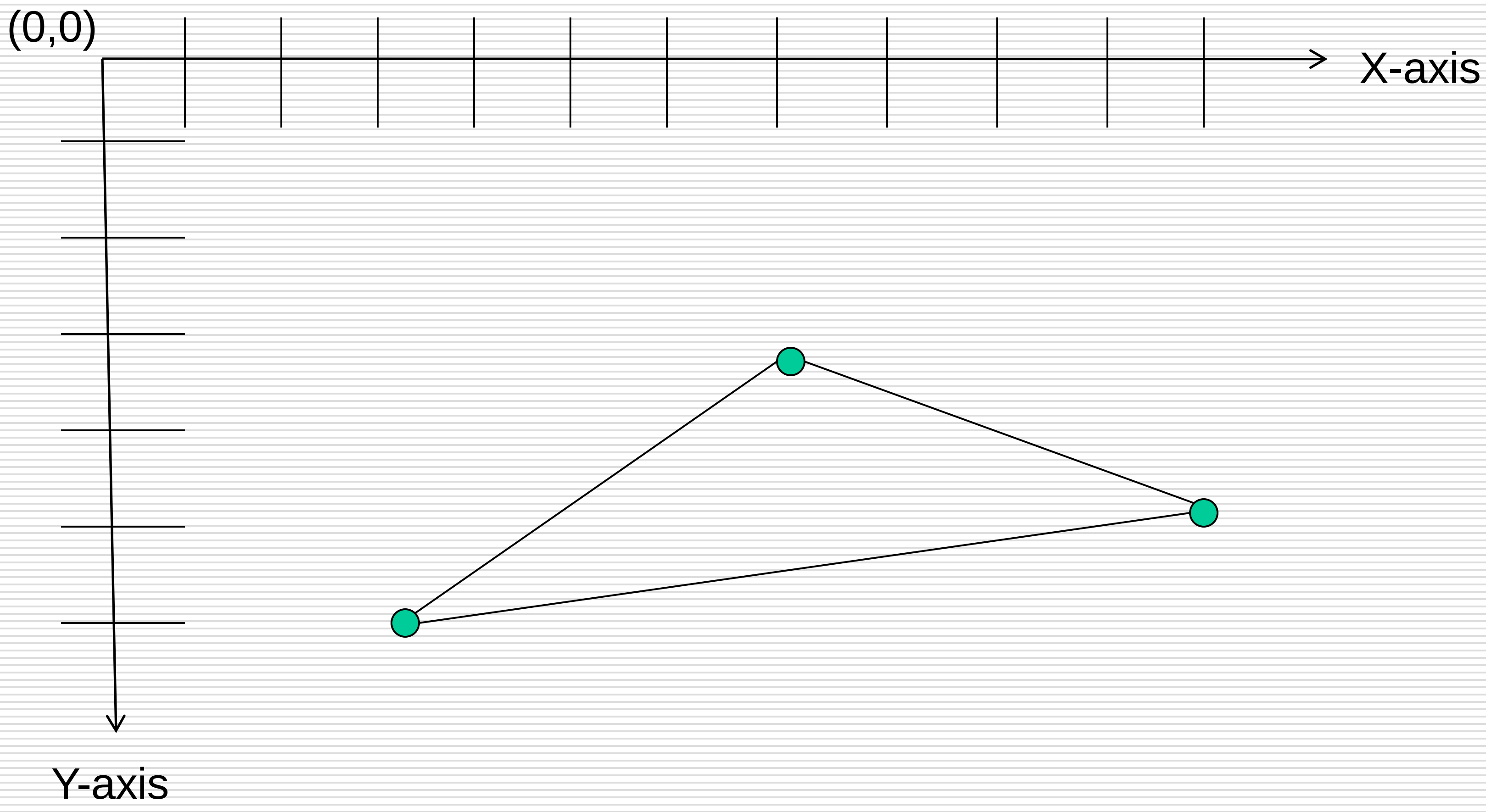
Computer Graphics Math

Sandro Spina
Computer Graphics and Simulation Group

Computer Science Department
University of Malta

Coordinate Spaces (2D)



The Euclidean Space (vector)

- The n-dimensional real Euclidean Space is denoted $\mathbb{R}^n$
- A vector **v** in $\mathbb{R}^n$ is an n-tuple, i.e. an ordered list of real numbers.

$$\mathbf{v} \in \mathbb{R}^n \Leftrightarrow \mathbf{v} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix} \text{ with } v_i \in \mathbb{R},\ \text{i=0}, \ldots, \text{n-1}$$

- Note that the vector above is represented in column-major form.
- Vectors can be added together or multiplied by a scalar.

Transpose of a vector

- We can write row vectors (as opposed to column vectors as seen in the previous slide) as the transpose of their column vectors.
- $\mathbf{v}^T = [\ v_1, v_2, \ldots., v_n]$
- The subscripts are usually labelled in a more meaningful way … not just numbers.
- For example a vector **v** in 3D space would have the subscripts x, y and z representing the x-coordinate, y-coordinate and z-coordinate of the vector point.

The Euclidean Space (+ and *)

$$\mathbf{u} + \mathbf{v} = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{pmatrix} + \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix} = \begin{pmatrix} u_0 + v_0 \\ u_1 + v_1 \\ \vdots \\ u_{n-1} + v_{n-1} \end{pmatrix} \in \mathbb{R}n$$

$$\mathrm{a}\mathbf{u} = \begin{pmatrix} au_0 \\ au_1 \\ \vdots \\ au_{n-1} \end{pmatrix} \in \mathbb{R}n$$

*The Euclidean Space (+ and *)*

- A vector may be multiplied by a scalar to produce a new vector whose components retain the same relative proportions.
- a**v** = **v**a, where a is a scalar quantity
- When a = -1, we get –**v** which represent the negation of the vector
- Addition and subtraction is component wise.
- <u>IMPORTANT:</u> **p** – **q** = **p** + (-**q**)

Basic theorems on vectors … (i)

- $\mathbf{p} + \mathbf{q} = \mathbf{q} + \mathbf{p}$ (commutativity)
- $(\mathbf{p} + \mathbf{q}) + \mathbf{r} = \mathbf{p} + (\mathbf{q} + \mathbf{r})$ (associativity)
- $a(\mathbf{p} + \mathbf{q}) = a\mathbf{p} + a\mathbf{q}$ (distributive law)
- $(a + b)\mathbf{p} = a\mathbf{p} + b\mathbf{p}$ (distributive law)

Basic theorems on vectors … (ii)

- $(ab)\,\mathbf{p} = a\,(b\,\mathbf{p})$
- $0 + \mathbf{v} = \mathbf{v}$ (zero identity)
- $\mathbf{v} + (-\mathbf{v}) = 0$ (additive inverse)
- $1\mathbf{u} = \mathbf{u}$ (identity mult)

Dot (Inner) Product of Vectors

- For a Euclidean space we may compute the **dot product** of two vectors, denoted by **u.v** and defined as follows:

$$\boldsymbol{u}.\boldsymbol{v} = \sum_{i=o}^{n-1} u_i v_i$$

- Which is essentially the summation of the products of the respective components of **u** and **v** .

Some rules for the dot product

- **u** . **u** ≥ 0, with **u** . **u** = 0 iff **u** = (0, 0, ..., 0) = 0
- (**u** + **v**) . **w** = **u** . **w** + **v** . **W** (additivity)
- (a**u**) . **v** = a (**u** . **v**) (homogeneity)
- **u** . **v** = **v** . **u** (symmetry)
- **u . v** = 0 iff **u** is perpendicular to **v**

The norm of a vector

- The norm of a vector **v**, denoted by ||**v**||, is a nonnegative number that can be expressed using the dot product as follows ...

$$\|\boldsymbol{u}\| = \sqrt{\boldsymbol{u} \,.\, \boldsymbol{u}} = \sqrt{\sum_{i=0}^{n-1} u_i^2}$$

- The importance of the norm will be evident when used to normalise a vector.

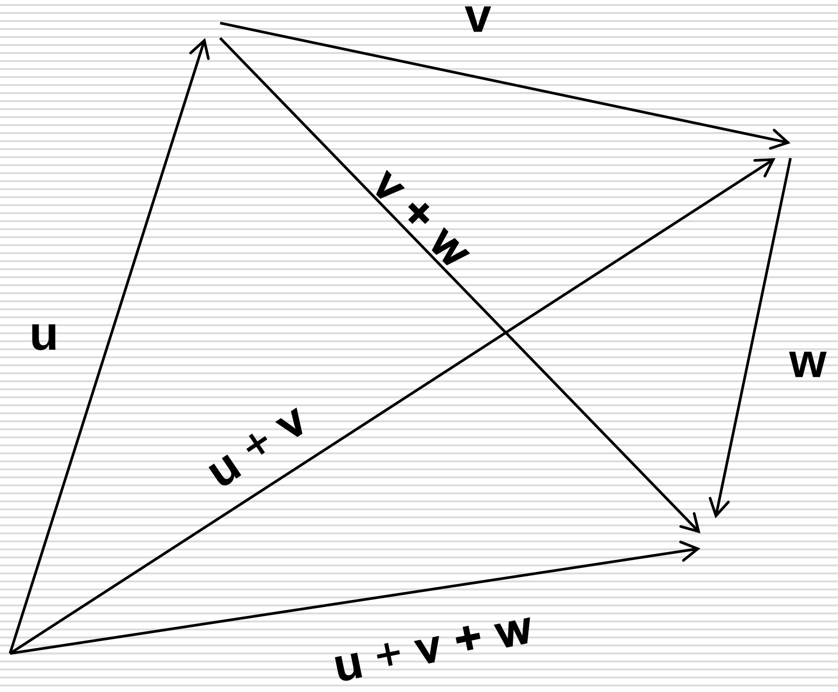
Some rules for the norm $||\mathbf{v}||$

- $||\,\mathbf{u}\,|| = 0$, iff $\mathbf{u} = (0, 0, \ldots, 0) = \mathbf{0}$
- $||\,a\mathbf{u}\,|| = |a|\;||\,\mathbf{u}\,||$
- $||\,\mathbf{u} + \mathbf{v}\,|| \leq ||\,\mathbf{u}\,|| + ||\,\mathbf{v}\,||$
- The norm of a vector gives us an indication of the its magnitude.

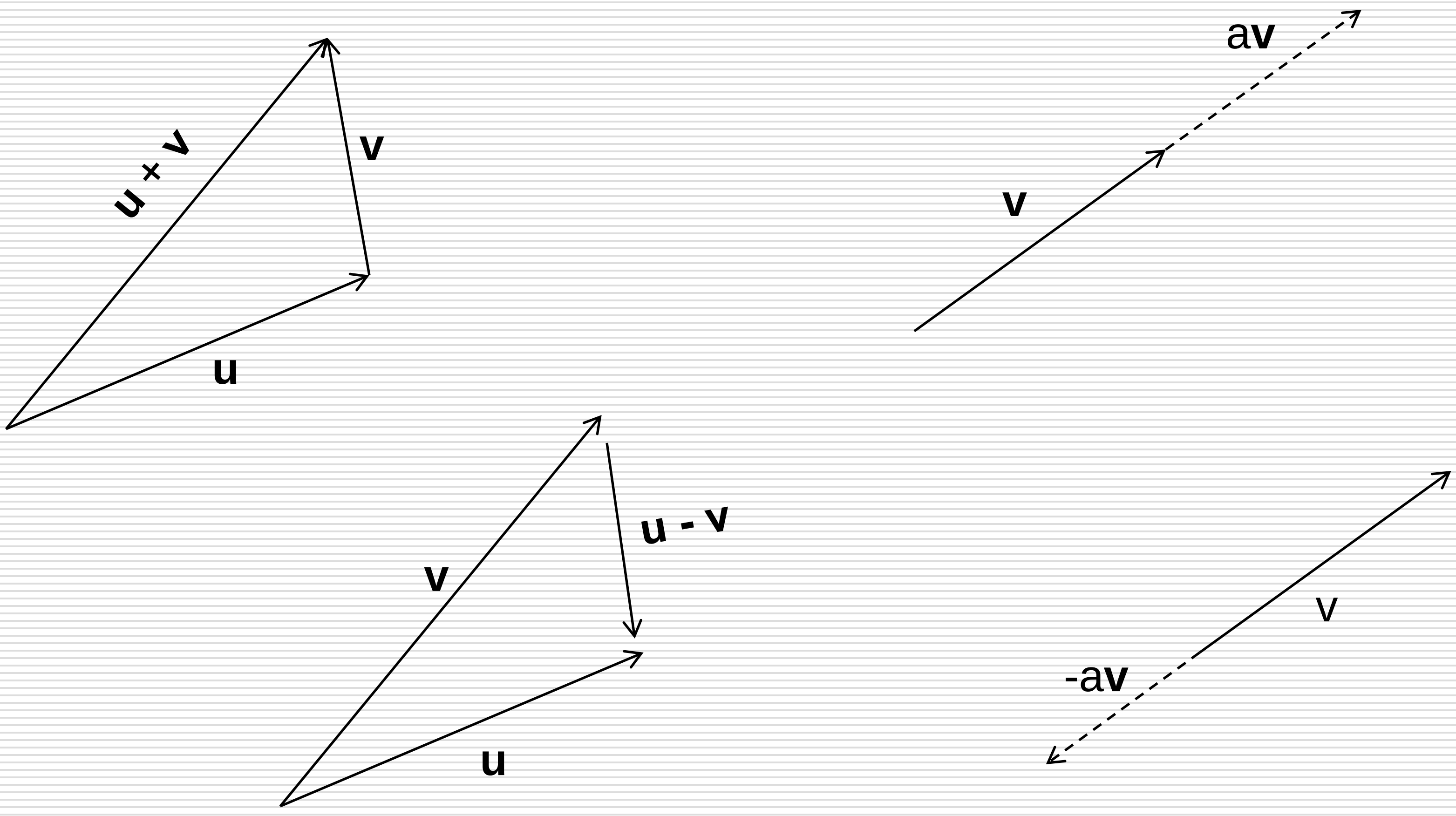
Vectors ...

- In our Euclidean space with basis vectors (1,0,0), (0,1,0) and (0,0,1), since the basis vectors are common for all vectors we can omit them when representing the vector.
- We simply write the scalar components of the vector. For eg v=(4,5,6) ....
- A vector **v** can be interpreted in two ways:
  - **Point in space**
  - **Directed line segment (i.e. A direction vector)**

Vector Diagrams (ii)...



Vector Diagrams …



Normalisation of Vectors

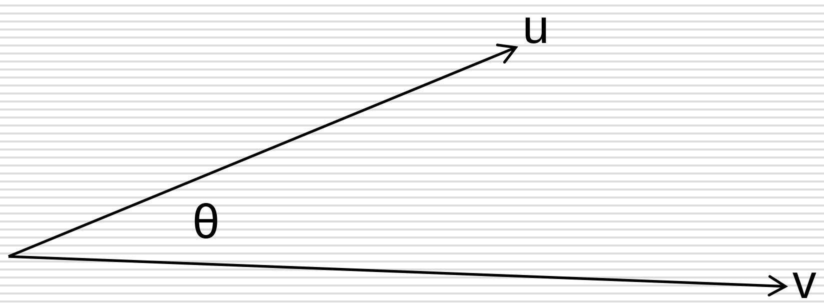
- The norm of a vector gives us a measure of the length (magnitude) of the vector ...
- Sometimes we'll need to normalise vectors (loose magnitude information but retain direction) with the help of the norm.
- This can be done by dividing by the length of the vector (the norm)

$$\frac{v}{\|v\|}$$

- This is also called the unit vector

Dot Product (ii)

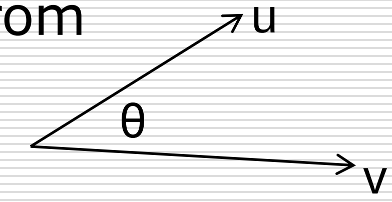
- We have already seen how to calculate the dot product between two vectors **u** and **v**.



- The dot product is also related to the angle θ between the vectors as follows:
  - $\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}||\ ||\mathbf{v}|| \cos \theta$,where θ is the smallest angle between **u** and **v**
  - We'll see how this equation is heavily used in CG for lighting calculations

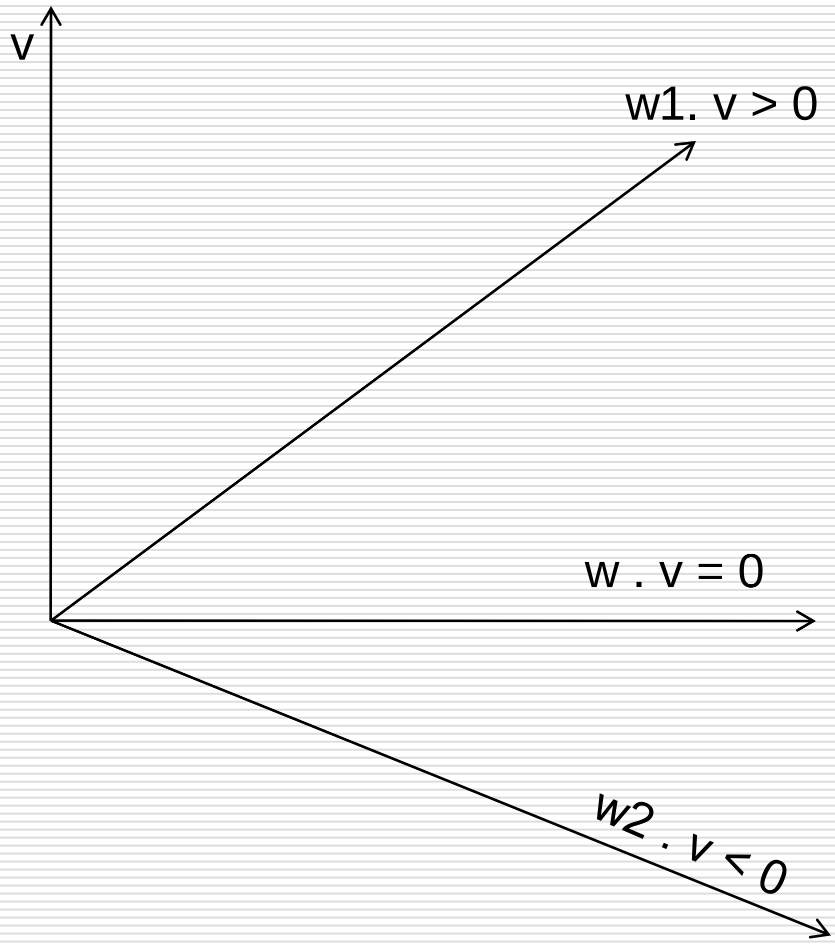
Dot Product (iii)

- A number of conclusions can be drawn from the sign of the dot product.



- Important (as we've already seen) is when the dot product is 0, indicates that the vectors are orthogonal.
- This is clear here as well given that cos(90deg) = 0
- If **u**.**v** > 0 then angle θ lies between 0 and 90 degrees
- If **u**.**v** < 0 then angle θ lies between 90 and 180 degrees

Dot Product (iv)



Linear Independence (i)

- Vectors that are parallel are linearly dependent.

- More formally, given the following equation:

  - $v_0 \mathbf{u}_0 + \ldots + v_{n-1} \mathbf{u}_{n-1} = 0$

- If only assigning the scalars $v_0 = \ldots = v_{n-1}$ to 0 solves the above equation then the vectors $u_0, \ldots, u_{n-1}$ are linearly independent.

- For example vectors (3,5) and (6,10) are not independent since $v_0=2$ and $v_1=-1$ solves the equation.

Linear Independence + Basis

- Linear independent vectors give us a way how to define all the space in which the vectors reside.

- If a set of n vectors, $\mathbf{u}_0, \ldots, \mathbf{u}_{n-1} \in \mathbb{R}^n$, is linearly independent and any vector $\mathbf{v}$ can be written as

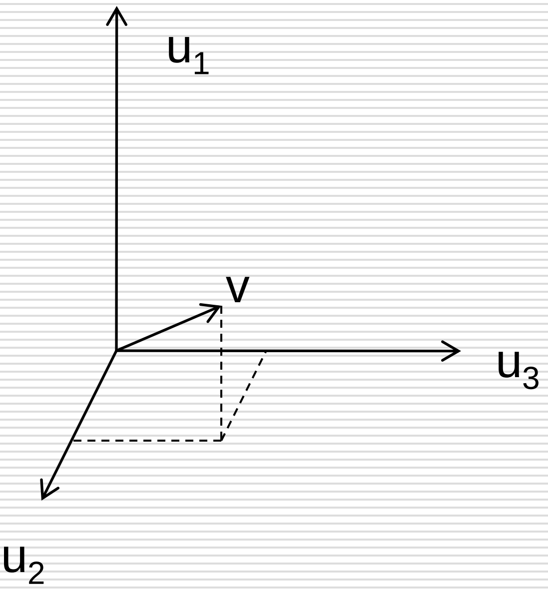
$$\boldsymbol{v} = \sum_{i=o}^{n-1} v_i \boldsymbol{u}_i$$

- ... then the vectors $u_0, \ldots, u_{n-1}$ are said to span Euclidean space $\mathbb{R}^n$

- Moreover if the scalars v_0 to v_{n-1} are uniquely determined by the vector $\mathbf{v}$, for all $\mathbf{v} \in \mathbb{R}^n$, then $\mathbf{u}_0,\ldots,\mathbf{u}_{n-1}$ form a basis in $\in \mathbb{R}^n$

Basis Vectors

- A three dimensional vector $\mathbf{v} = (v_0, v_1, v_2)$ expressed in the basis formed by u_1, u_2 and u_3 in $\mathbb{R}^3$



- Take for example the basis vectors in 2D: (4,3) and (2,6).
- If I want to describe the vector (-5,-6) I simply need to multiply (4,3) by -1 and (2,6) by 0.5 ... This will give me the new vector.
- I can describe all vectors in this way

Orthonormal Basis

- In CG we shall be making use of orthonormal basis ...
- For such a basis, consisting of base vectors $u_0, ..., u_{n-1}$ the following must hold:

$$\boldsymbol{u}_i . \boldsymbol{u}_j = \begin{cases} 0, & i \neq j, \\ 1, & i = j \end{cases}$$

- What's important here is that each pair of basis vectors must be orthogonal and have unit length.
- The vectors (1,0,0) (0,1,0) and (0,0,1) form an ortho normal basis which we refer to as the *standard basis*.
- The standard basis is **orthogonal**

Cross Product

- Suppose we have two vectors **v** and **w** …. And we need to generate a new vector which is orthogonal (perpendicular) to both vectors.

- The operation that computes this is the cross product.

- This property has many uses in computer graphics (as we shall see) one of which is a method for calculating a surface normal at a particular point given two distinct tangent vectors.

Cross Product (Definition)

- The cross product of two vectors **u** and **v**, is another vector whose components are defined as follows:

- $\mathbf{u} \times \mathbf{v} = (u_y v_z - u_z v_y, u_z v_x - u_x v_z, u_x v_y - u_y v_x)$

- There a *two* vectors that are perpendicular to **u** x **v** … which are **w** and **–w**. One the negation of the other.

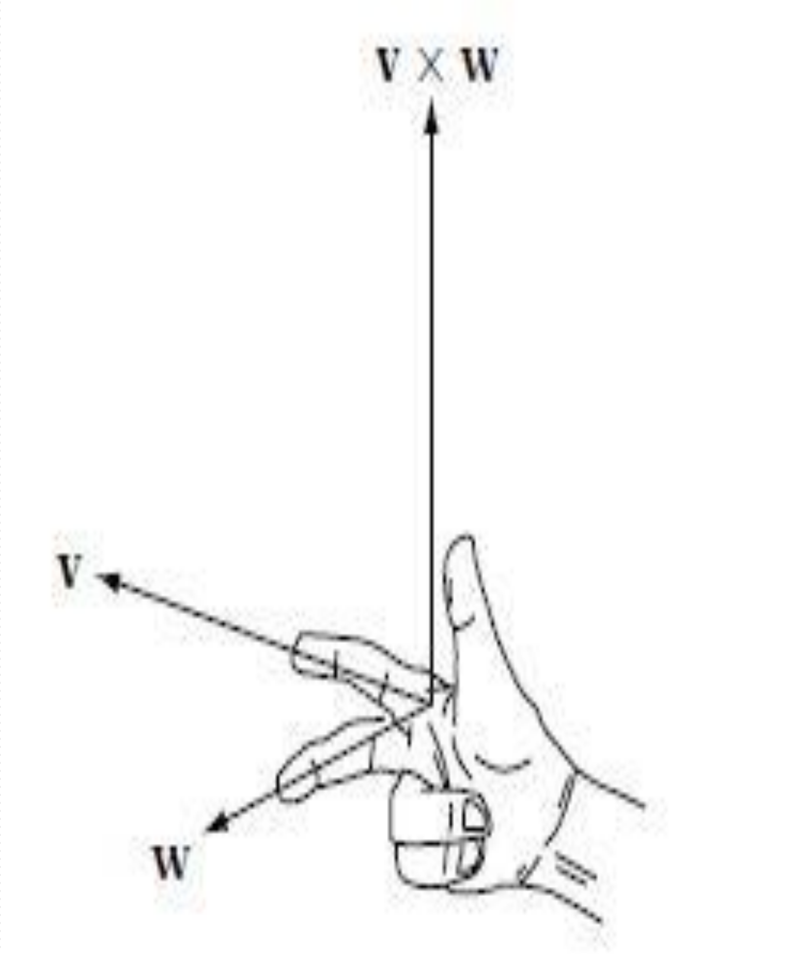
- The one we choose is determined by what we refer to as the right hand rule …. (in which you use your right hand obviously)

Right-Hand Rule

- With your right hand … align

- Forefinger with **v**,
- Middle finger with **w**,
- The cross product will point in the direction of the thumb.

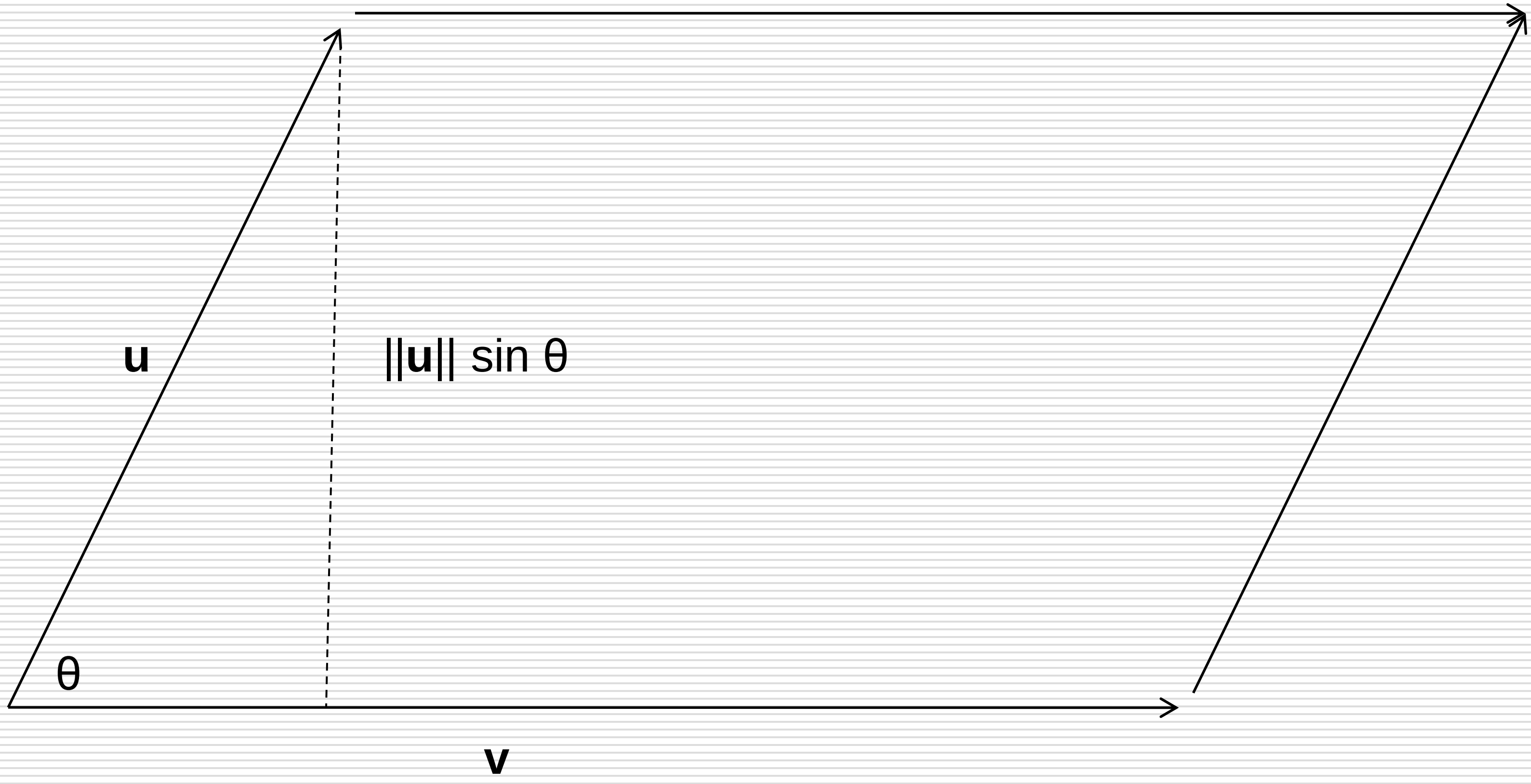
- If you negate **w**, then the direction of the cross product changes as well.



Cross Product (Magnitude of ...)

- The length of the cross product of two vectors **u** and **v** is equal to the area of the parallelogram extended by the two vectors.
- This can be computed using the formula
- $\| \mathbf{u} \times \mathbf{v} \| = \| \mathbf{u} \| \, \| \mathbf{v} \| \sin \theta$
- Where θ is the angle between the vectors **u** and **v**

Cross Product (Magnitude of (ii))



Cross Product (Properties of ...)

- The cross product is not commutative (i.e. order is important)
  - $\mathbf{u} \times \mathbf{v} = -(\mathbf{v} \times \mathbf{u})$
- $\mathbf{u} \times (\mathbf{v} + \mathbf{w}) = (\mathbf{u} \times \mathbf{v}) + (\mathbf{u} \times \mathbf{w})$
- $(\mathbf{u} + \mathbf{v}) \times \mathbf{w} = (\mathbf{u} \times \mathbf{w}) + (\mathbf{v} \times \mathbf{w})$
- $a (\mathbf{v} \times \mathbf{w}) = (a\mathbf{v}) \times \mathbf{w} = \mathbf{v} \times (a\mathbf{w})$
- $\mathbf{v} \times \mathbf{v} = 0$ and $\mathbf{v} \times 0 = 0 \times \mathbf{v} = 0$

Cross Product *(Small Proof of correctness)*

- We can use the result from the dot product to show that the cross product of two vectors **u** and **v** is correct (i.e. it is perpendicular to both vectors)

- Let u and v be any two 3D vectors. Then $(\mathbf{u} \times \mathbf{v}) \cdot \mathbf{u} = 0$ and $(\mathbf{u} \times \mathbf{v}) \cdot \mathbf{v} = 0$

- $(\mathbf{u} \times \mathbf{v}) \cdot \mathbf{u} = (u_y v_z - u_z v_y, u_z v_x - u_x v_z, u_x v_y - u_y v_x) \cdot \mathbf{u}$

- $= u_x u_y v_z - u_x u_z v_y + u_y u_z v_x - u_y u_x v_z + u_x v_y u_z - u_y v_x u_z$
- $= 0$

Matrices

- **Matrices in 3D computer graphics are ubiquitous !!**
- "Matrices are the mathematical currency for 3D graphics" – OpenGL Bible
- We shall be using matrices to move (transform) points and direction vectors …
- Matrices provide us with a tool to manipulate vectors and points.
- We shall be looking at a semi-formal mathematical description of matrices in the next few slides.

Matrices (Definition)

- A matrix **M** is described by p x q scalars, m_{ij}, where $0 \leq i \leq p-1$, $0 \leq j \leq q-1$, ordered in a rectangular fashion (with p rows and q columns) as shown below …

$$\mathbf{M} = \begin{pmatrix} m_{00} & \cdots & m_{0,q-1} \\ \vdots & \ddots & \vdots \\ m_{p-1,0} & \cdots & m_{p-1,q-1} \end{pmatrix} = [m_{ij}]$$

Identity Matrix

- The identity matrix **I**, is a special matrix which is <u>square</u> and contains ones in the diagonal and zeros everywhere else. Also called the *unit matrix.*

- It is the matrix-form counterpart of the scalar number one.

- The following represents the 3x3 identity matrix …

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matrix Addition

- Matrices add entry-wise. Because of this, the addition of two matrices M and N is only possible for equal sized matrices ….
- $\mathbf{M} + \mathbf{N} = [m_{ij}] + [n_{ij}] = [m_{ij} + n_{ij}]$
- Pictorially we have :

$$\mathbf{M}+\mathbf{N}=\begin{pmatrix} m_{00}+n_{00} & \cdots & m_{0,p-1}+n_{0,q-1} \\ \vdots & \ddots & \vdots \\ m_{p-1,0}+n_{q-1,0} & \cdots & m_{p-1,q-1}+n_{p-1,q-1} \end{pmatrix}$$

Matrix Addition Properties

- The resulting matrix is of the same size of the operands
- $(\mathbf{L} + \mathbf{M}) + \mathbf{N} = \mathbf{L} + (\mathbf{M} + \mathbf{N})$
- $\mathbf{M} + \mathbf{N} = \mathbf{N} + \mathbf{M}$
- $\mathbf{M} + \mathbf{0} = \mathbf{M}$
- $\mathbf{M} - \mathbf{M} = \mathbf{0}$

Matrix Scalar Multiplication

- We can (similar to vectors) multiply our matrix by a scalar quantity.
- A scalar *a* and a matrix **M**, can be multiplied as follows:
- $a\mathbf{M} = [am_{ij}]$ …. Pictorially we have:

$$\boldsymbol{a}\mathbf{M} = \begin{pmatrix} am_{00} & \cdots & am_{0,p-1} \\ \vdots & \ddots & \vdots \\ am_{p-1,0} & \cdots & am_{p-1,q-1} \end{pmatrix}$$

Matrix Scalar Multiplication Properties

- $0\mathbf{M} = 0$
- $1\mathbf{M} = \mathbf{M}$
- $a(b\mathbf{M}) = (ab)\mathbf{M}$
- $a\mathbf{0} = \mathbf{0}$
- $(a+b)\,\mathbf{M} = a\mathbf{M} + b\mathbf{M}$
- $a(\mathbf{M} + \mathbf{N}) = a\mathbf{M} + a\mathbf{N}$

Transpose of a Matrix

- The transpose of a matrix **M** is referred to as $\mathbf{M}^T$.
- If $\mathbf{M} = [m_{ij}]$ then $\mathbf{M}^T$ is defined as $\mathbf{M} = [m_{ji}]$
- In practice we are switching the rows with the columns.
- Hence the transpose of a matrix of size n x m, is a matrix with size m x n.
- In a square matrix the diagonal scalars remain the same with all the other values are transposed.

Transpose of a Matrix (Properties)

- $(a\mathbf{M})^T = a\mathbf{M}^T$
- $(\mathbf{M} + \mathbf{N})^T = \mathbf{M}^T + \mathbf{N}^T$
- $(\mathbf{M}^T)^T = M$
- $(\mathbf{MN})^T = \mathbf{N}^T \mathbf{M}^T$

Matrix Multiplication

- Matrix multiplication, denoted **MN** between two matrices **M** and **N**, is defined only if the size of **M** is *p* x *q* and the size of **N** is *q* x *r*

- If this is the case then the resultant matrix **T** = **MN**, would be of size *p* x *r*

- Each cell in the new matrix **T** is computed as follows:

$$\boldsymbol{T}_{pr} = \sum_{k=1}^{q} \boldsymbol{M}_{pk} \boldsymbol{N}_{kr}$$

Matrix Multiplication (Pictorially)

$$\boldsymbol{T} = \boldsymbol{MN} = \begin{pmatrix} m_{00} & \cdots & m_{0,q-1} \\ \vdots & \ddots & \vdots \\ m_{p-1,0} & \cdots & m_{p-1,q-1} \end{pmatrix} \begin{pmatrix} n_{00} & \cdots & n_{0,r-1} \\ \vdots & \ddots & \vdots \\ n_{q-1,0} & \cdots & n_{q-1,r-1} \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{i=0}^{q-1} m_{0,i} n_{i,0} & \cdots & \sum_{i=0}^{q-1} m_{0,i} n_{i,r-1} \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{q-1} m_{p-1,i} n_{i,0} & \cdots & \sum_{i=0}^{q-1} m_{p-1,i} n_{i,r-1} \end{pmatrix}$$

Matrix Multiplication with Vector

- If we consider a vector **v** as an *n* x 1 sized matrix then we can multiply a vector by a matrix using the method in the previous slide.

- Note that this will give us a new vector **w** with dimensions *m* x 1. Pictorially we have :

$$\boldsymbol{w} = \boldsymbol{M}\boldsymbol{v} = \begin{pmatrix} \boldsymbol{m}_0 \,.\, \boldsymbol{v} \\ \vdots \\ \boldsymbol{m}_{p-1} \,.\, \boldsymbol{v} \end{pmatrix} = \begin{pmatrix} \boldsymbol{w}_0 \\ \vdots \\ \boldsymbol{w}_{q-1} \end{pmatrix}$$

Matrix Multiplication Properties (Imp)

- $(\mathbf{LM})\mathbf{N} = \mathbf{L}(\mathbf{MN})$
- $(\mathbf{L}+\mathbf{M})\mathbf{N} = \mathbf{LN} + \mathbf{LM}$
- $\mathbf{MI} = \mathbf{IM} = \mathbf{M}$
- Important: Matrix multiplication is not commutative … which means that $\mathbf{MN} \neq \mathbf{NM}$ in general (there could be cases where it is)

Determinant of a Matrix (i)

- An important value associated with every square matrix is the value of its determinant.
- The determinant, |**M**| or det(**M**), is a scalar quantity derived from the entries of the matrix.
- For 2 x 2 square matrix the determinant is equal to :

$$|\mathbf{M}| = \begin{vmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{vmatrix} = m_{00}m_{11} - m_{01}m_{10}$$

Determinant of a Matrix (ii)

- In the case of a 3 x 3 matrix, he determinant is equal to:

$$|\mathbf{M}| = \begin{vmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{vmatrix}$$

$$= m_{00}m_{11}m_{22} + m_{01}m_{12}m_{20} + m_{02}m_{10}m_{21} - m_{00}m_{12}m_{21} - m_{01}m_{10}m_{22} - m_{02}m_{11}m_{20}$$

- We are adding diagonals (from top) going to the right then subtracting diagonals (from top) going to the left.

Determinant of a Matrix (iii)

- If we assume that the rows in the matrix represent three different vectors, i.e.

$$|\mathbf{M}| = \begin{vmatrix} e_x & e_y & e_z \\ u_x & u_y & u_z \\ u_x & u_y & u_z \end{vmatrix}$$

- $|\mathbf{M}| = (m_x \times m_y) \cdot m_z$

Properties of the matrix determinant (i)

- For an *n* x *n* matrix the following apply to determinant calculations:
- $|\mathbf{M}^{-1}| = 1 / |\mathbf{M}|$
- $|\mathbf{MN}| = |\mathbf{M}|\ |\mathbf{N}|$
- $|a\mathbf{M}| = a|\mathbf{M}|$
- $|\mathbf{M}^{T}| = |\mathbf{M}|$

Properties of the matrix determinant (ii)

- If all elements of a row (or column) of a matrix **M** are multiplied by a scalar *a*, then the new determinant is *a*|**M**|
- **IMP:** If two rows (or columns) coincide (i.e. the cross product between them is 0) then the determinant of matrix **M**, |**M**| = 0
- This last property is important whenever we need to calculate the inverse of a matrix (as we shall see when working on geometric transformations in 3D pipeline)

Subdeterminants (Cofactors) and Adjoints (i)

- An adjoint is a form of a matrix.
- The subdeterminant (cofactor) of an *n* x *n* matrix **M**, denoted by $d^{\mathbf{M}}_{ij}$, is equal to the determinant (of the resulting *n-1* x *n-1* matrix) obtained when deleting row *i* and column *j* from **M**.

$$d_{02}^{\boldsymbol{M}} = \begin{vmatrix} m_{10} & m_{11} \\ m_{20} & m_{21} \end{vmatrix}$$

Subdeterminants (Cofactors) and Adjoints (ii)

- The adjoint of a matrix **M** is obtained by taking the subdeterminants for every component in the matrix, resulting in the following:

$$adj\,(M) = \begin{pmatrix} d_{00} & -d_{10} & d_{20} \\ -d_{01} & d_{11} & -d_{21} \\ d_{02} & -d_{12} & d_{22} \end{pmatrix}$$

Inverse of a Matrix (i)

- The multiplicative inverse of a matrix, **M**, denoted by $\mathbf{M}^{-1}$, (which is dealt with here), exists only for square matrices with $|\mathbf{M}| \neq 0$.

- This is one of the reasons why we need to be able to calculate the determinant of a matrix.

- If $\mathbf{N} = \mathbf{M}^{-1}$ then to prove the inverse is correct it suffices to show that $\mathbf{NM} = \mathbf{I}$ and $\mathbf{MN} = \mathbf{I}$

- I.e. a matrix multiplied by its inverse results in the identity matrix … which produces no effect.

Inverse of a Matrix (ii)

- The equation outlined in the previous slide can be formulated in a slightly different way … using vectors.
- If **u** = **Mv** and the matrix **N** exists such that **v** = **Nu**, then **N** = $\mathbf{M}^{-1}$
- This formulation makes it immediately more relevant to computer graphics.
- The adjoint method can be used to calculate the inverse.
- The inverse of a matrix is useful geometrically because it allows us to 'undo' another transformation.

Inverse of a Matrix (iii)

- In the case of a 2 x 2 matrix we have:

$$\mathbf{M}^{-1} = \frac{1}{|\mathbf{M}|}\begin{vmatrix} m_{11} & -m_{01} \\ -m_{10} & m_{00} \end{vmatrix}$$

- In the general case we have the following:

$$\mathbf{M}^{-1} = \frac{1}{|\mathbf{M}|}\, adj\,(\boldsymbol{M})$$

Orthogonal Matrices

- A square matrix **M**, with only real elements, is orthogonal if and only if $\mathbf{M}\mathbf{M}^T = \mathbf{M}^T\mathbf{M} = \mathbf{I}$
- This means that the transpose of **M** is equal to the inverse of **M** , i.e. $\mathbf{M}^{-1} = \mathbf{M}^T$
- The standard basis is orthonormal, since the basis vectors are orthogonal to each other and of length one (unit vectors). Representing this basis as a matrix $\mathbf{E} = (\mathbf{e}_x\ \mathbf{e}_y\ \mathbf{e}_z) = \mathbf{I}$, gives us an orthogonal matrix.

Transforms …

- For our next module we'll see how matrices (and their properties) as discussed here are used in CG to describe point and vector transformations.

- Matrices are used to describe
  - Rotations
  - Scaling
  - Translation

- Once that's done we'll be able to start writing some simple programs.