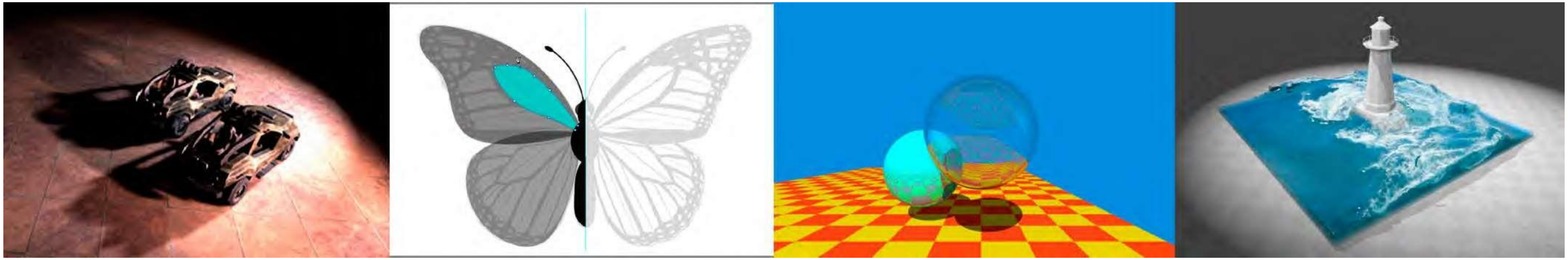
Computer Graphics

Review of Linear Algebra



Last Lecture

- Course Topics/ Logistics
- What is Computer Graphics?
- Short history of Computer Graphics
- Applications of Computer Graphics
- Difference between CG and CV ?

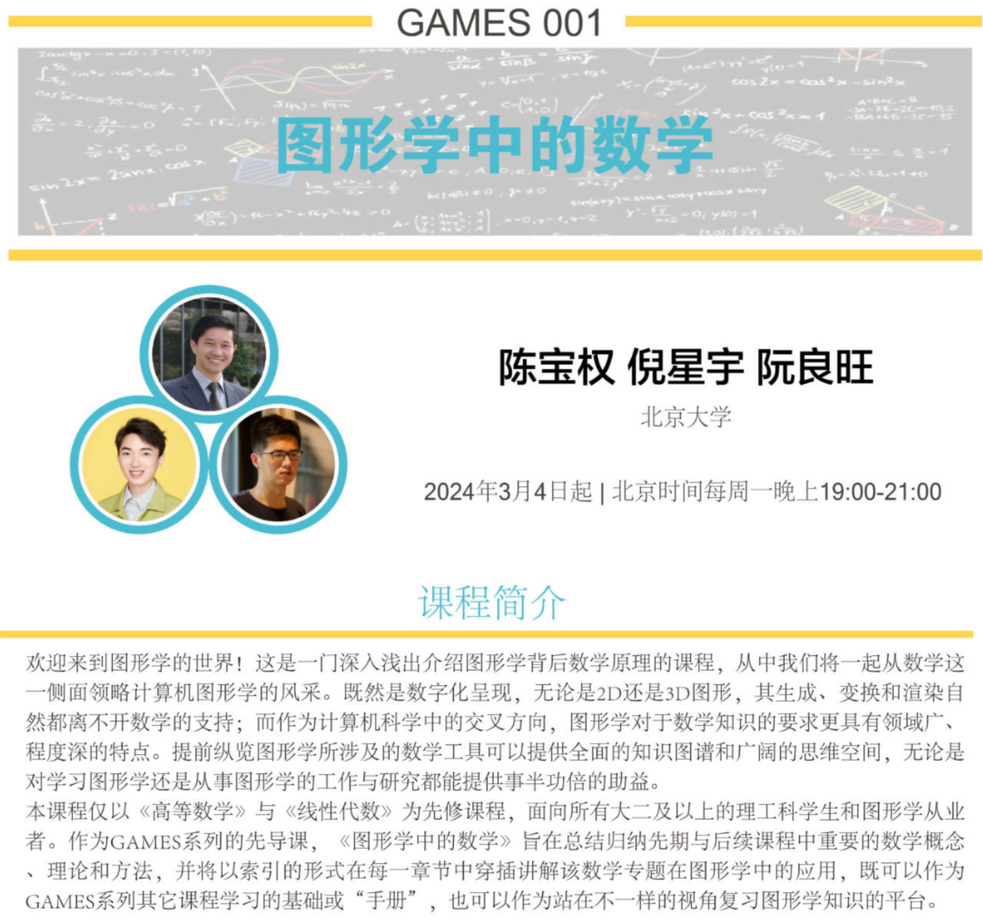
Introduction to Linear Algebra!

(Linear Algebra in Computer Graphics…)

Graphics' Dependencies

- Basic mathematics
  - Linear algebra, calculus, statistics
- Basic physics
  - Optics, Mechanics
- Misc
  - Signal processing
  - Numerical analysis
- And a bit of aesthetics

Graphics’ Dependencies



GAMES 001

图形学中的数学

陈宝权 倪星宇 阮良旺

北京大学

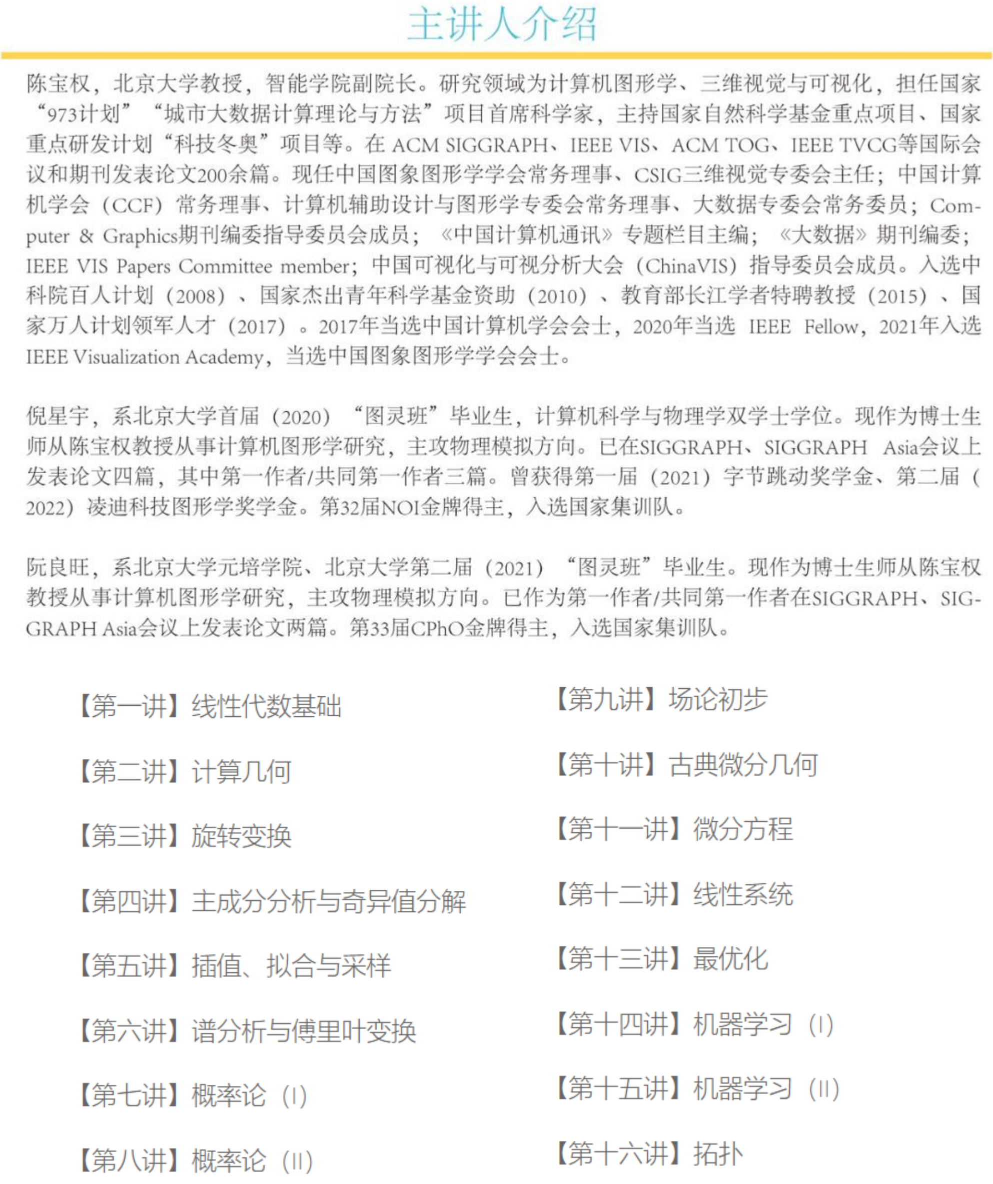
2024年3月4日起 | 北京时间每周一晚上19:00-21:00

课程简介

欢迎来到图形学的世界！这是一门深入浅出介绍图形学背后数学原理的课程，从中我们将一起从数学这一侧面领略计算机图形学的风采。既然是数字化呈现，无论是2D还是3D图形，其生成、变换和渲染自然都离不开数学的支持；而作为计算机科学中的交叉方向，图形学对于数学知识的要求更具有领域广、程度深的特点。提前纵览图形学所涉及的数学工具可以提供全面的知识图谱和广阔的思维空间，无论是对学习图形学还是从事图形学的工作与研究都能提供事半功倍的助益。

本课程仅以《高等数学》与《线性代数》为先修课程，面向所有大二及以上的理工科学生和图形学从业者。作为GAMES系列的先导课，《图形学中的数学》旨在总结归纳先期与后续课程中重要的数学概念、理论和方法，并将以索引的形式在每一章节中穿插讲解该数学专题在图形学中的应用，既可以作为GAMES系列其它课程学习的基础或“手册”，也可以作为站在不一样的视角复习图形学知识的平台。

https://games-cn.org/games001/



主讲人介绍

陈宝权，北京大学教授，智能学院副院长。研究领域为计算机图形学、三维视觉与可视化，担任国家“973计划”“城市大数据计算理论与方法”项目首席科学家，主持国家自然科学基金重点项目、国家重点研发计划“科技冬奥”项目等。在 ACM SIGGRAPH、IEEE VIS、ACM TOG、IEEE TVCG等国际会议和期刊发表论文200余篇。现任中国图象图形学学会常务理事、CSIG三维视觉专委会主任；中国计算机学会（CCF）常务理事、计算机辅助设计与图形学专委会常务理事、大数据专委会常务委员；Computer & Graphics期刊编委指导委员会成员；《中国计算机通讯》专题栏目主编；《大数据》期刊编委；IEEE VIS Papers Committee member；中国可视化与可视分析大会（ChinaVIS）指导委员会成员。入选中科院百人计划（2008）、国家杰出青年科学基金资助（2010）、教育部长江学者特聘教授（2015）、国家万人计划领军人才（2017）。2017年当选中国计算机学会会士，2020年当选 IEEE Fellow，2021年入选IEEE Visualization Academy，当选中国图象图形学学会会士。

倪星宇，系北京大学首届（2020）“图灵班”毕业生，计算机科学与物理学双学士学位。现作为博士生师从陈宝权教授从事计算机图形学研究，主攻物理模拟方向。已在SIGGRAPH、SIGGRAPH Asia会议上发表论文四篇，其中第一作者/共同第一作者三篇。曾获得第一届（2021）字节跳动奖学金、第二届（2022）凌迪科技图形学奖学金。第32届NOI金牌得主，入选国家集训队。

阮良旺，系北京大学元培学院、北京大学第二届（2021）“图灵班”毕业生。现作为博士生师从陈宝权教授从事计算机图形学研究，主攻物理模拟方向。已作为第一作者/共同第一作者在SIGGRAPH、SIGGRAPH Asia会议上发表论文两篇。第33届CPhO金牌得主，入选国家集训队。

【第一讲】线性代数基础

【第二讲】计算几何

【第三讲】旋转变换

【第四讲】主成分分析与奇异值分解

【第五讲】插值、拟合与采样

【第六讲】谱分析与傅里叶变换

【第七讲】概率论（I）

【第八讲】概率论（II）

【第九讲】场论初步

【第十讲】古典微分几何

【第十一讲】微分方程

【第十二讲】线性系统

【第十三讲】最优化

【第十四讲】机器学习（I）

【第十五讲】机器学习（II）

【第十六讲】拓扑

This Course

- More dependent on Linear Algebra
  - Vectors (dot products, cross products, …)
  - Matrices (matrix-matrix, matrix-vector mult., …)

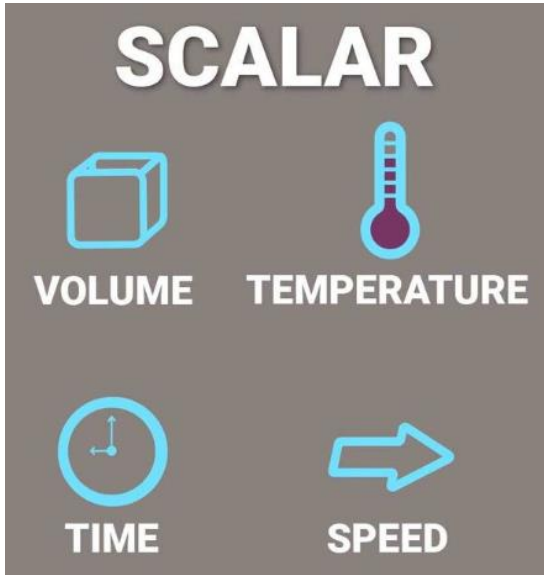
- For example,
  - A point is a vector (?)
  - An operation like translating or rotating objects can be matrix-vector multiplication

An Example of Rotation

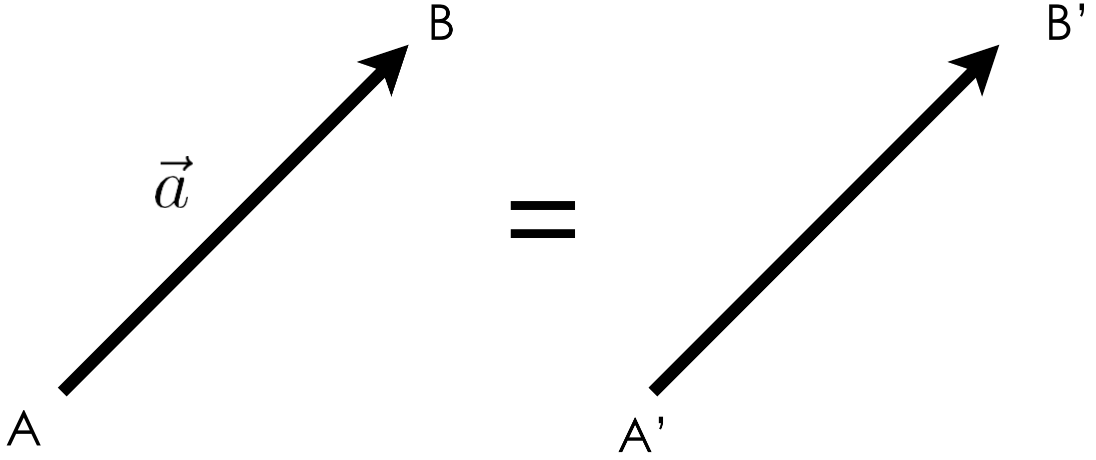
Rendering Glints on High-Resolution Normal-Mapped Specular Surfaces, Lingqi Yan, 2014

Scalar

- Examples: 1, 2, 3, 0.5, 0.9, 4.7 …
- Usually written as a
- How many / much
- Scalar in CG: length, number, ….
- Operation: + - X /



Vectors



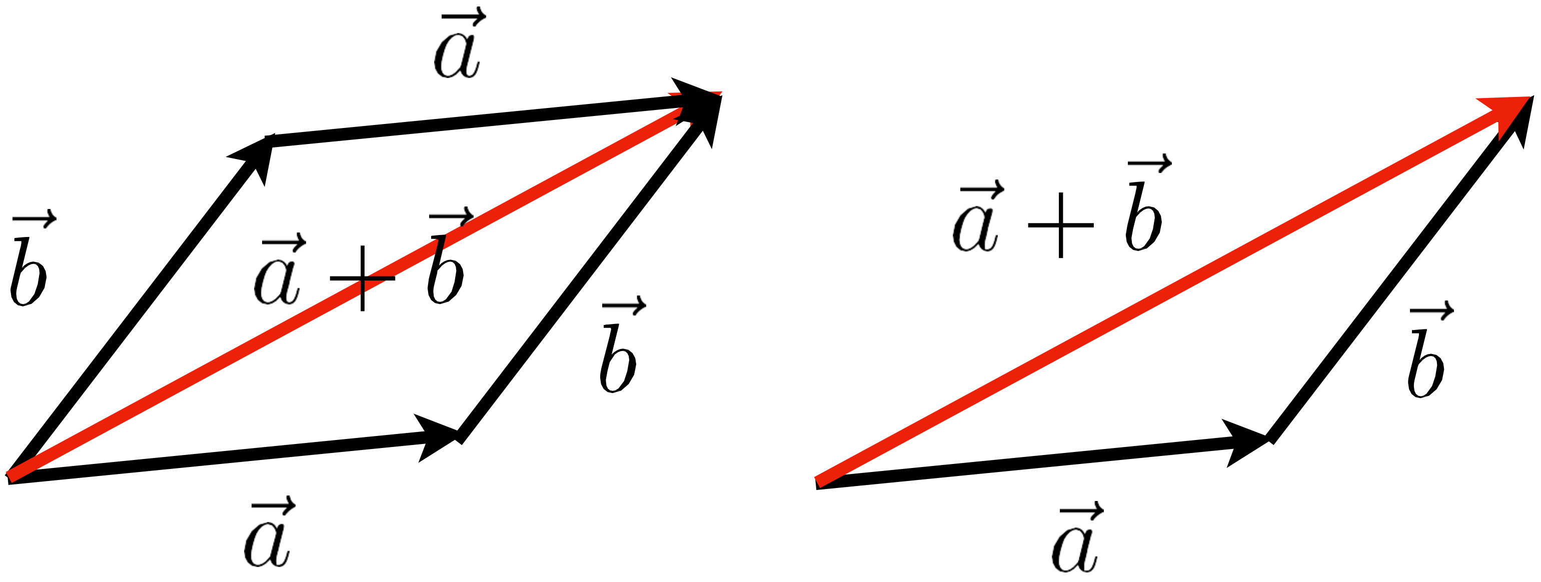
- Usually written as $\vec{a}$ or in bold $\boldsymbol{a}$
- Or using start and end points $\overrightarrow{AB} = B - A$
- Direction and length
- No absolute starting position

Vector Normalization

- Magnidude (length) of a vector written as $\|\vec{a}\|$
- Unit vector
  - A vector with magnitude of 1
  - Finding the unit vector of a vector (normalization): $\hat{a} = \vec{a}/\|\vec{a}\|$
  - Used to represent directions

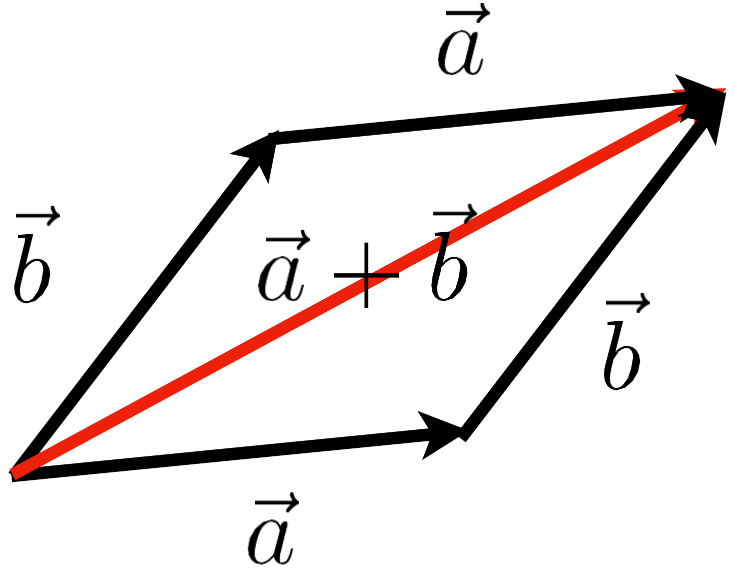
注：计算几何中通常使用归一化向量

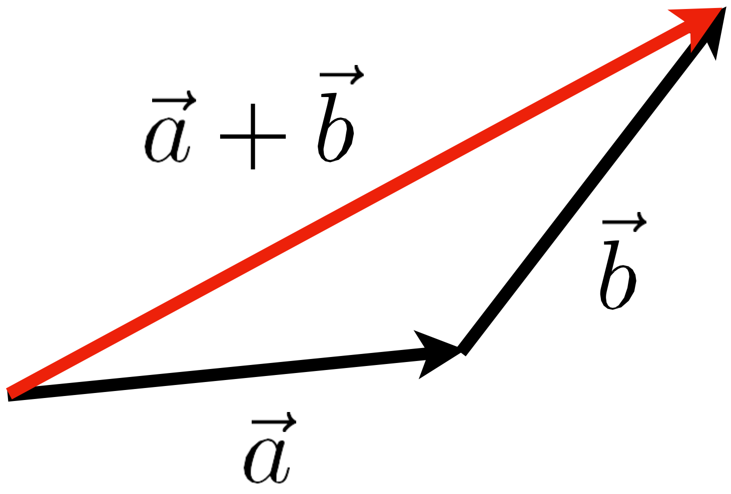
Vector Addition



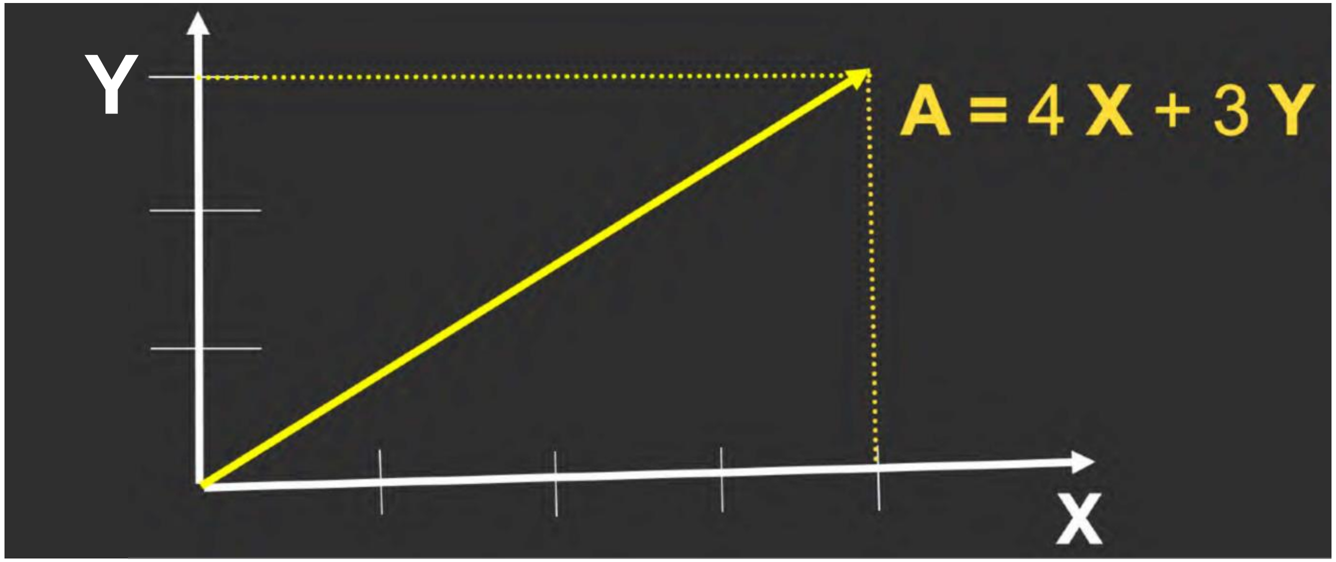
- Geometrically: Parallelogram law & Triangle law
- Algebraically: Simply add coordinates

Vector Subtraction





Cartesian Coordinates



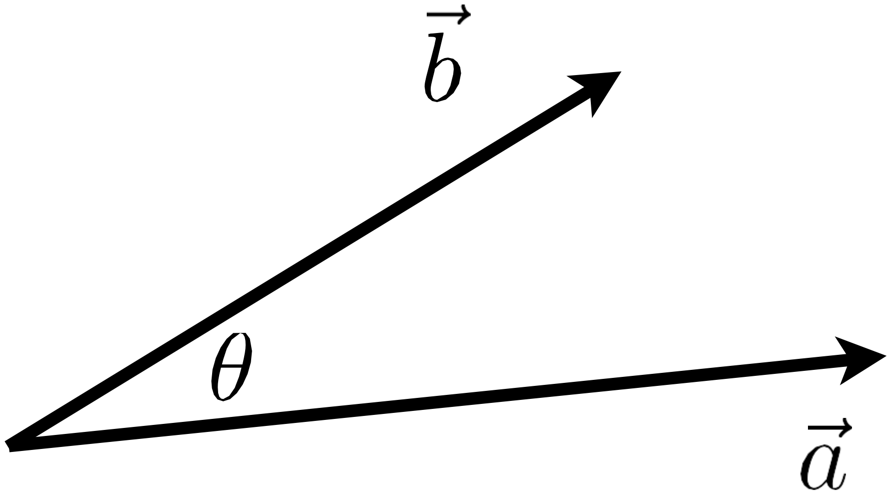
- X and Y can be any (usually orthogonal unit) vectors

$$\mathbf{A} = \begin{pmatrix} x \\ y \end{pmatrix} \qquad \mathbf{A}^T = (x, y) \qquad \|\mathbf{A}\| = \sqrt{x^2 + y^2}$$

Vector Multiplication

- Dot product (点乘)
- Cross product
- Orthonormal bases and coordinate frames

Dot (scalar) Product



$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos\theta$$

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$$

- For unit vectors

$$\cos\theta = \hat{a} \cdot \hat{b}$$

Dot (scalar) Product

- Properties

$$\vec{a} \cdot \vec{b} = \vec{b} \cdot \vec{a}$$

$$\vec{a} \cdot (\vec{b} + \vec{c}) = \vec{a} \cdot \vec{b} + \vec{a} \cdot \vec{c}$$

$$(k\vec{a}) \cdot \vec{b} = \vec{a} \cdot (k\vec{b}) = k(\vec{a} \cdot \vec{b})$$

Dot Product in Cartesian Coordinates

- Component-wise multiplication, then adding up
  - In 2D

$$\vec{a} \cdot \vec{b} = \begin{pmatrix} x_a \\ y_a \end{pmatrix} \cdot \begin{pmatrix} x_b \\ y_b \end{pmatrix} = x_a x_b + y_a y_b$$

  - In 3D

$$\vec{a} \cdot \vec{b} = \begin{pmatrix} x_a \\ y_a \\ z_a \end{pmatrix} \cdot \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} = x_a x_b + y_a y_b + z_a z_b$$

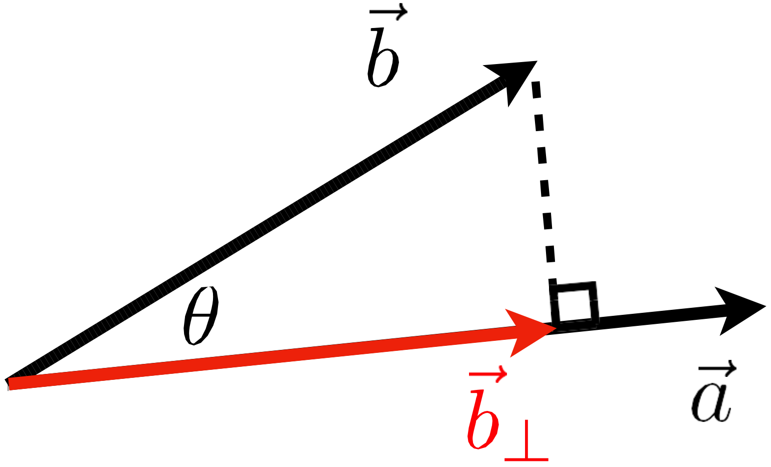
Dot Product in Graphics

- Find angle between two vectors
  (e.g. angle between light source and surface)
- Finding projection of one vector on another

请问Dot Product如何实现上述两个功能?

Dot Product for Projection

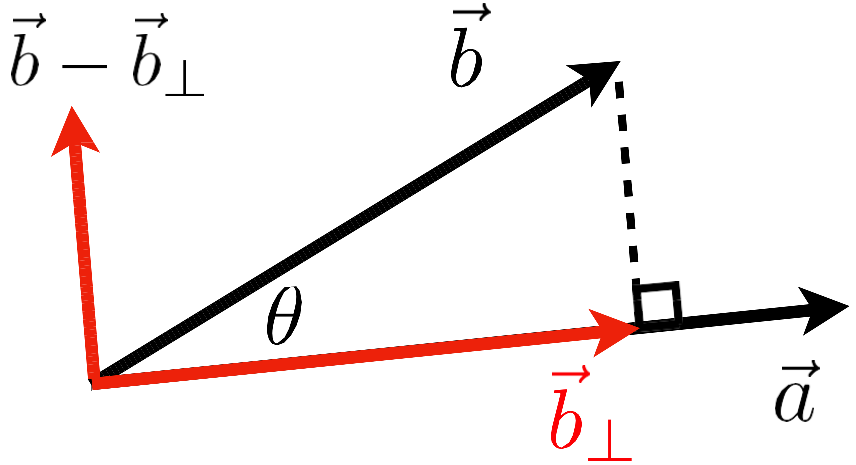
- $\vec{b}_\perp$: projection of $\vec{b}$ onto $\vec{a}$
  - $\vec{b}_\perp$ must be along $\vec{a}$ (or along $\hat{a}$)
    - $\vec{b}_\perp = k\hat{a}$
  - What's its magnitude k?
    - $k = \|\vec{b}_\perp\| = \|\vec{b}\| \cos\theta$



Dot Product在CG中还有什么功能呢?

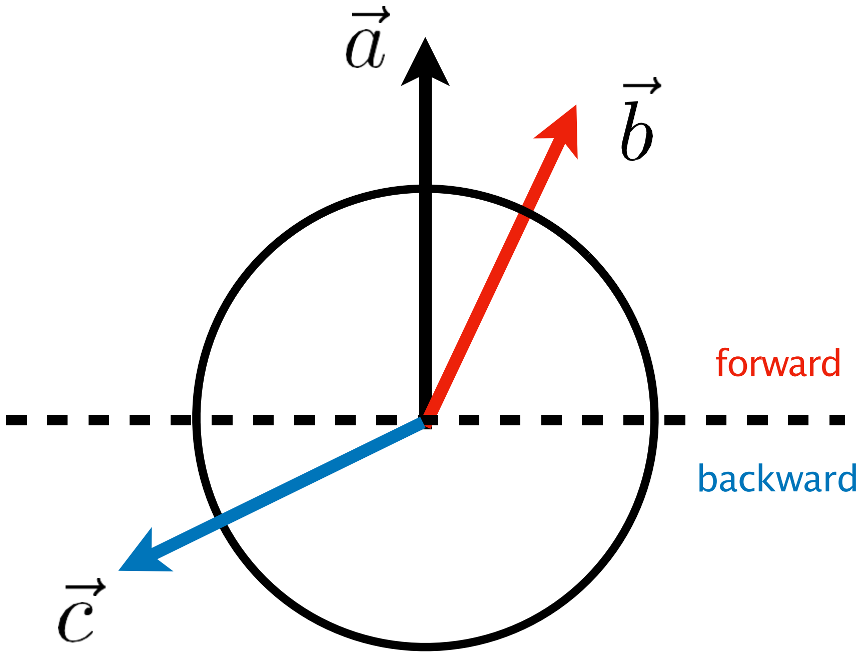
Dot Product in Graphics

- Measure how close two directions are
- Decompose a vector



Dot Product in Graphics

- Measure how close two directions are
- Decompose a vector
- Determine forward / backward

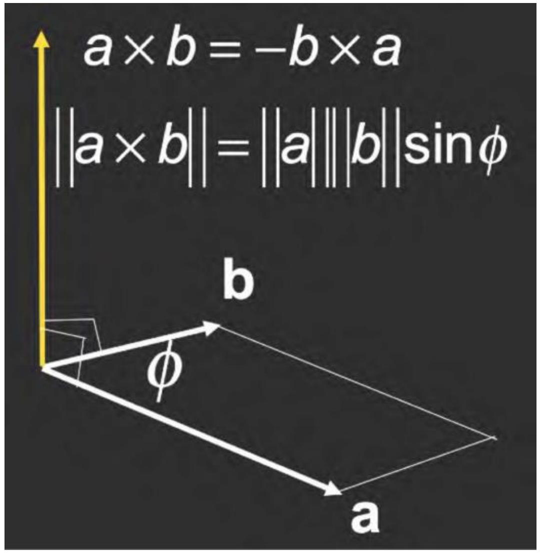


dot product > or < 0

Vector Multiplication

- Dot product
- Cross product （叉乘）
- Orthonormal bases and coordinate frames

Cross (vector) Product


$a \times b = -b \times a$
$\|a \times b\| = \|a\|\|b\|\sin\phi$
b
ϕ
a

- Cross product is orthogonal to two initial vectors
- Direction determined by right-hand rule
- Useful in constructing coordinate systems (later)

Cross product: Properties

$$\vec{x} \times \vec{y} = +\vec{z}$$

$$\vec{y} \times \vec{x} = -\vec{z}$$

$$\vec{y} \times \vec{z} = +\vec{x}$$

$$\vec{z} \times \vec{y} = -\vec{x}$$

$$\vec{z} \times \vec{x} = +\vec{y}$$

$$\vec{x} \times \vec{z} = -\vec{y}$$

$$\vec{a} \times \vec{b} = -\vec{b} \times \vec{a}$$

$$\vec{a} \times \vec{a} = \vec{0}$$

$$\vec{a} \times (\vec{b} + \vec{c}) = \vec{a} \times \vec{b} + \vec{a} \times \vec{c}$$

$$\vec{a} \times (k\vec{b}) = k(\vec{a} \times \vec{b})$$

Cross Product: Cartesian Formula?

$$\vec{a} \times \vec{b} = \begin{pmatrix} y_a z_b - y_b z_a \\ z_a x_b - x_a z_b \\ x_a y_b - y_a x_b \end{pmatrix}$$

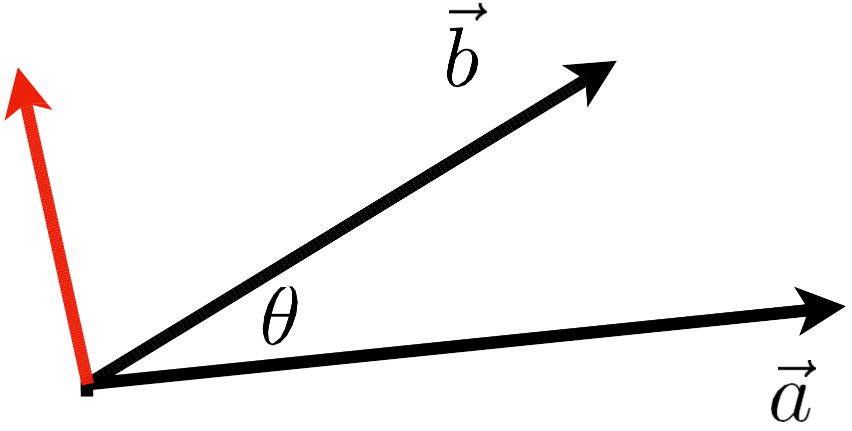
- Later in this lecture

$$\vec{a} \times \vec{b} = A^* b = \begin{pmatrix} 0 & -z_a & y_a \\ z_a & 0 & -x_a \\ -y_a & x_a & 0 \end{pmatrix} \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix}$$

dual matrix of vector a

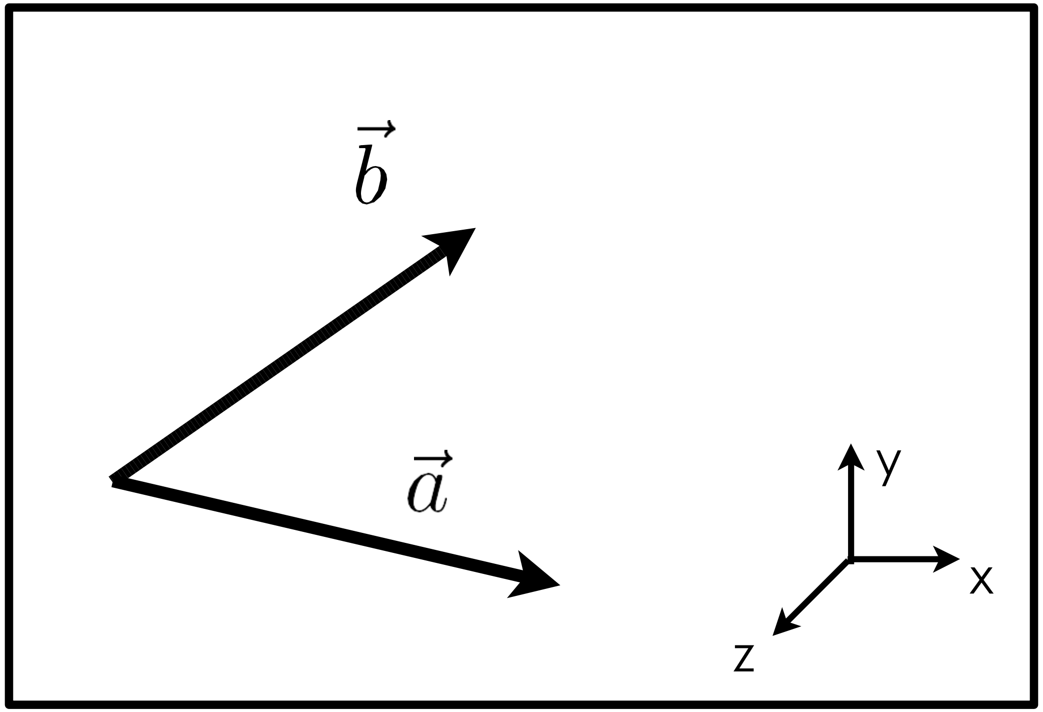
Cross Product in Graphics

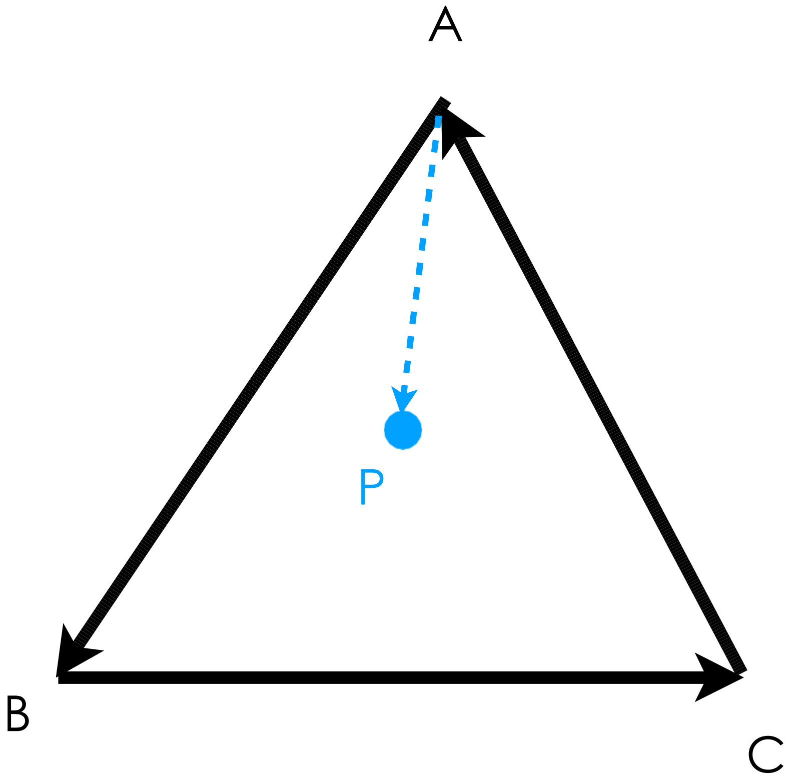
- Determine left / right
- Determine inside / outside



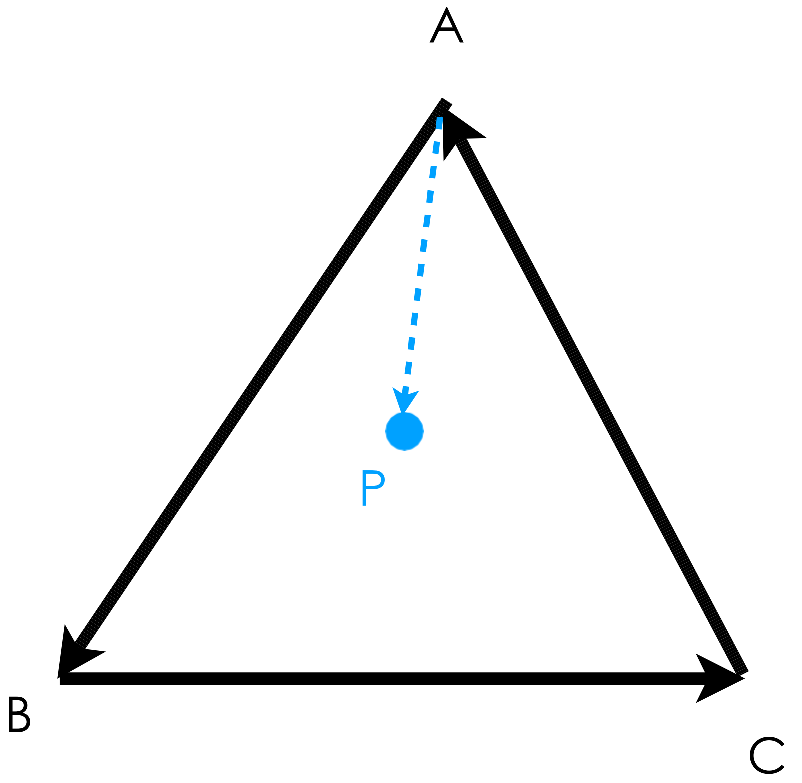
请问Cross Product如何实现上述功能?

Cross Product in Graphics





Any other methods?



Vector Multiplication

- Dot product
- Cross product
- Orthonormal bases and coordinate frames （正交基和坐标系）

Orthonormal Bases / Coordinate Frames

- Important for representing points, positions, locations
- Often, many sets of coordinate systems
  - Global, local, world, model, parts of model (head, hands, …)
- Critical issue is transforming between these systems/ bases
  - A topic for next few weeks

Orthonormal Coordinate Frames

- Any set of 3 vectors (in 3D) that

$$\|\vec{u}\| = \|\vec{v}\| = \|\vec{w}\| = 1$$

$$\vec{u} \cdot \vec{v} = \vec{v} \cdot \vec{w} = \vec{u} \cdot \vec{w} = 0$$

$$\vec{w} = \vec{u} \times \vec{v}$$ (right-handed)

$$\vec{p} = (\vec{p} \cdot \vec{u})\vec{u} + (\vec{p} \cdot \vec{v})\vec{v} + (\vec{p} \cdot \vec{w})\vec{w}$$

(projection)

Questions?

Matrices

- Magical 2D arrays that haunt in every CS course
- In Graphics, pervasively used to represent transformations
  - Translation, rotation, shear, scale
    (more details in the next few lectures)

What is a matrix

- Array of numbers (m × n = m rows, n columns)

$$\begin{pmatrix} 1 & 3 \\ 5 & 2 \\ 0 & 4 \end{pmatrix}$$

- Addition and multiplication by a scalar are trivial: element by element

Matrix-Matrix Multiplication

- # (number of) columns in A must = # rows in B
  (M x N) (N x P) = (M x P)

$$\begin{pmatrix} 1 & 3 \\ 5 & 2 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} 3 & 6 & 9 & 4 \\ 2 & 7 & 8 & 3 \end{pmatrix}$$

Matrix-Matrix Multiplication

- # (number of) columns in A must = # rows in B
  (M x N) (N x P) = (M x P)

$$\begin{pmatrix} 1 & 3 \\ 5 & 2 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} 3 & 6 & 9 & 4 \\ 2 & 7 & 8 & 3 \end{pmatrix} = \begin{pmatrix} 9 & ? & 33 & 13 \\ 19 & 44 & 61 & 26 \\ 8 & 28 & 32 & ? \end{pmatrix}$$

- Element (i, j) in the product is
  the dot product of row i from A and column j from B

Matrix-Matrix Multiplication

- Properties
  - Non-commutative (不满足交换律)
    (AB and BA are different in general)
  - Associative and distributive (结合律和分配律)
    - $(AB)C=A(BC)$
    - $A(B+C) = AB + AC$
    - $(A+B)C = AC + BC$

Matrix-Vector Multiplication

- Treat vector as a column matrix (m×1) (why?)
- Key for transforming points
- Official spoiler: 2D reflection about y-axis

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -x \\ y \end{pmatrix}$$

Transpose of a Matrix

- Switch rows and columns (ij -> ji)

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

- Property

$$(AB)^T = B^T A^T$$

Identity Matrix and Inverses

$$I_{3\times 3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$AA^{-1} = A^{-1}A = I$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

Vector multiplication in Matrix form

- Dot product?

$$\vec{a} \cdot \vec{b} = \vec{a}^T \vec{b}$$

$$= \begin{pmatrix} x_a & y_a & z_a \end{pmatrix} \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} = \begin{pmatrix} x_a x_b + y_a y_b + z_a z_b \end{pmatrix}$$

- Cross product?

$$\vec{a} \times \vec{b} = A^* b = \begin{pmatrix} 0 & -z_a & y_a \\ z_a & 0 & -x_a \\ -y_a & x_a & 0 \end{pmatrix} \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix}$$

dual matrix of vector a

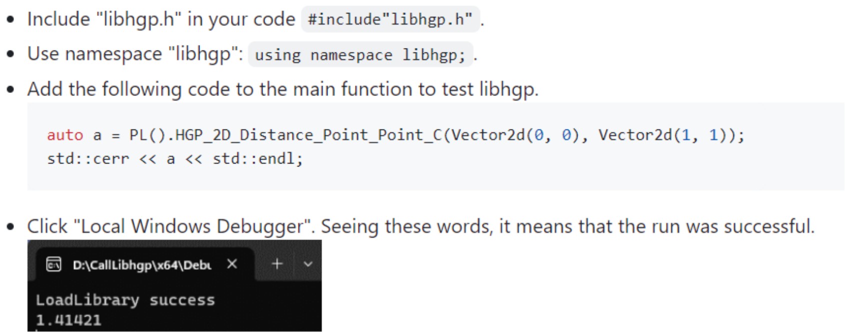
An Example of General Transformation

Dagon engine - Sponza scene (glTF with PBR materials)

Liblgp (Low Level Geo Pack)

▶ StatisticsCombinationSet	统计和组合函数，共11种
▶ StringDataStructure	字符串处理函数，共22种
▶ BasicGeomFunctions	基本几何函数，共43种
▶ BasicMathFunctions	基本数学函数，共16种
▶ Transformation	坐标变换函数，共14种
▶ IOFunctions	输入输出函数，共16种
▶ Graph	与图相关的函数，共3种
▶ DevelopmentRelated	扩展功能，共13种

https://github.com/haisenzhao/liblgp



- Include "libhgp.h" in your code `#include"libhgp.h"`.
- Use namespace "libhgp": `using namespace libhgp;`.
- Add the following code to the main function to test libhgp.

```
auto a = PL().HGP_2D_Distance_Point_Point_C(Vector2d(0, 0), Vector2d(1, 1));
std::cerr << a << std::endl;
```

- Click "Local Windows Debugger". Seeing these words, it means that the run was successful.

```
LoadLibrary success
1.41421
```

Libhgp (High Level Geo Pack)

核心功能

- 计算距离（点、直线、线段、射线、多边形、多边形组等）
- 求交点（直线之间、线段之间、多边形之间等）
- 网格操作（计算面积、采样、细分、平滑等）
- 获取信息（获取三角形网格中邻居、面、邻边的向量信息）
- 计算（梯度、曲率、发现等）

https://github.com/haisenzhao/libhgp

Questions?

Next

- Transformation

Thank you!