



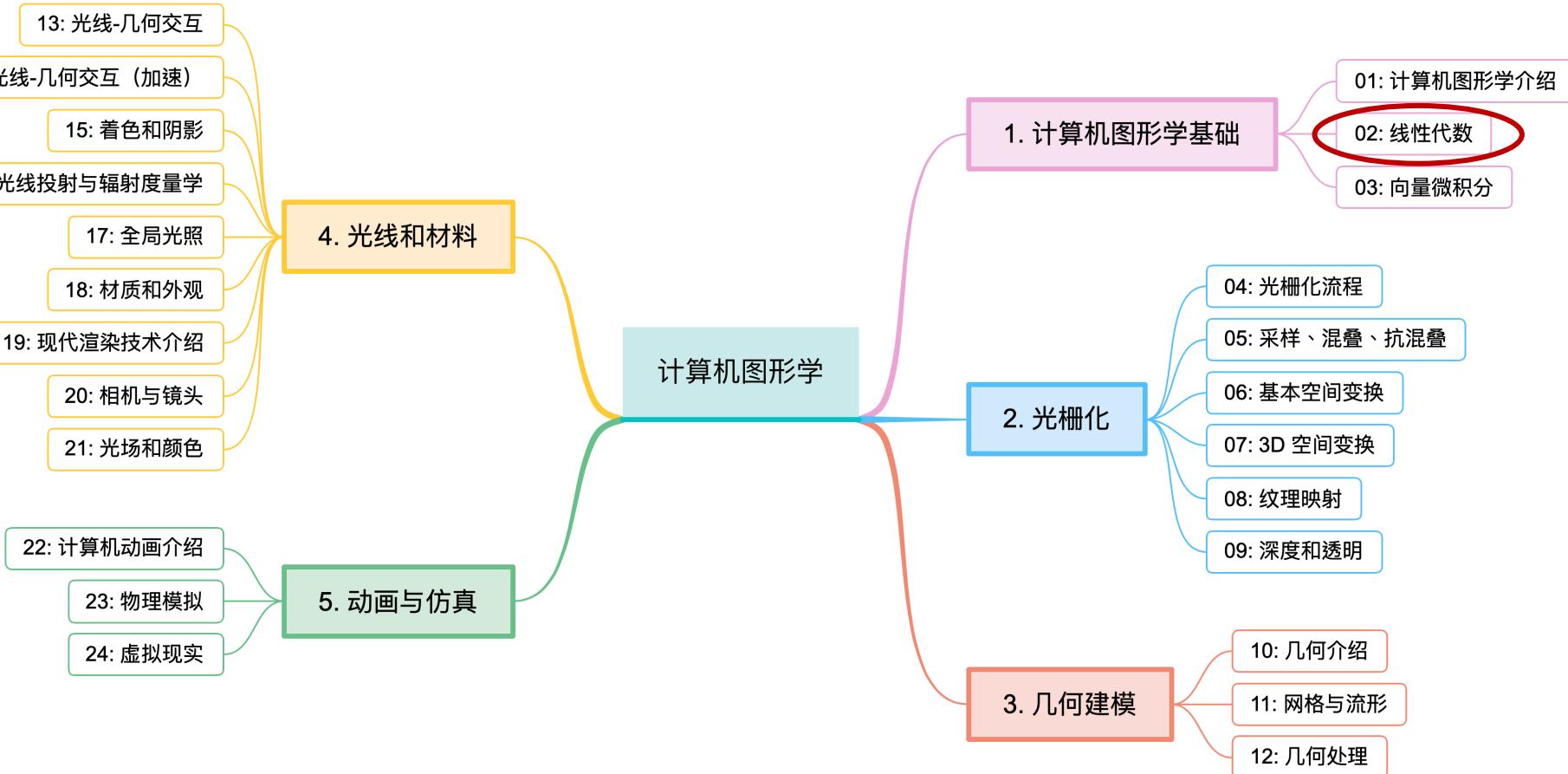
Lecture 02: 线性代数

SSE315: 计算机图形学
Computer Graphics

陈壮彬

软件工程学院

chenzhb36@mail.sysu.edu.cn



Today's topics

□ 从几何角度认识向量空间

□ 向量空间的测量

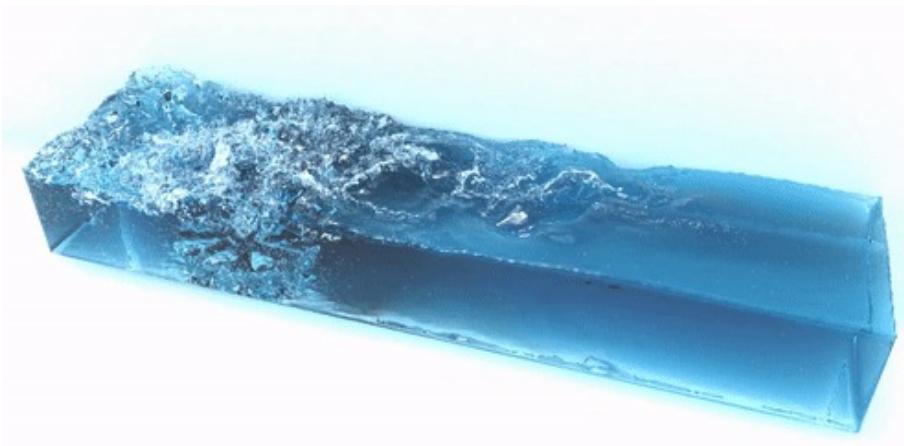
- 范数 Norm
- 内积 Inner product

□ 线性映射

计算机图形学中的线性代数

口为什么线性代数对计算机图形学很重要？

- **几何、物理等与计算之间的桥梁**
- 在图形的许多领域，一旦你能用线性代数表达问题的解决方法，你就基本上得到答案了：让计算机求解 $Ax = b$
- **高效的线性代数求解器/工具使现代计算机图形学成为可能**
(图像处理、基于物理的动画、几何处理...)



线性代数定义

□ 线性代数研究**向量空间 (vector spaces)** 及其之间的
线性映射 (linear maps), 正式定义:

For all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$ and scalars a, b :

- $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
- $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
- There exists a zero vector “ $\mathbf{0}$ ” such that $\mathbf{v} + \mathbf{0} = \mathbf{0} + \mathbf{v} = \mathbf{v}$
- For every \mathbf{v} there is a vector “ $-\mathbf{v}$ ” such that $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
- $1\mathbf{v} = \mathbf{v}$
- $a(b\mathbf{v}) = (ab)\mathbf{v}$
- $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
- $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$

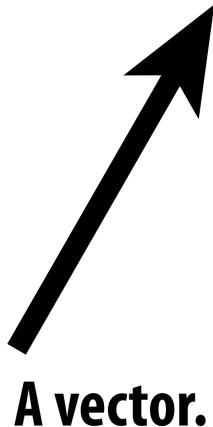
□ 上述规则是怎么来的?

□ 让我们试着理解它们!

向量 Vectors

□首先，**向量**是什么？

□直观地看，向量就是一个小箭头



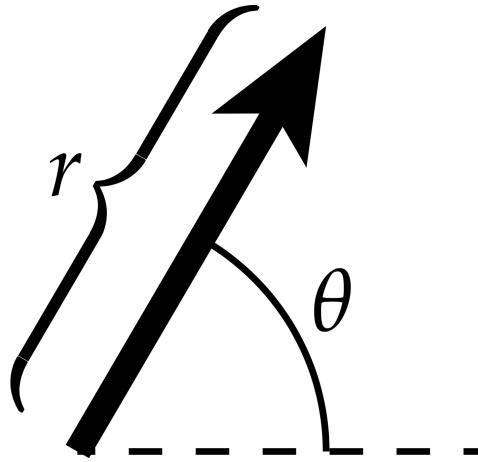
□在计算机图形学中，我们处理的许多数据可能看起来不像小箭头，如多项式、图像、辐射度...

□但它们仍在很多方面呈现出向量的特性；因此，这个小箭头仍然是一个有用的模型

向量 - 我们能测量什么?

□ 向量编码了什么信息?

□ 方向 (direction) 和大小 (magnitude)*



□ 例如，2D 中的向量可以通过相对于某个固定方向的长度和角度进行编码，即极坐标 (polar coordinates)

□ 还有其他方式编码向量吗？

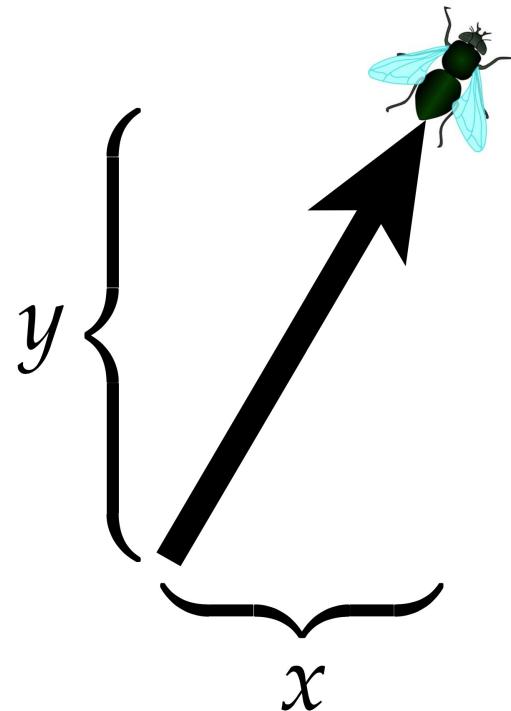
*传统上，向量不包括“基点”；带基点的向量有时被称为切向量 (tangent vector)

笛卡尔坐标系中的向量

□ 我们同样可以在笛卡尔坐标系中描述向量的信息



René Descartes, Est. 1596



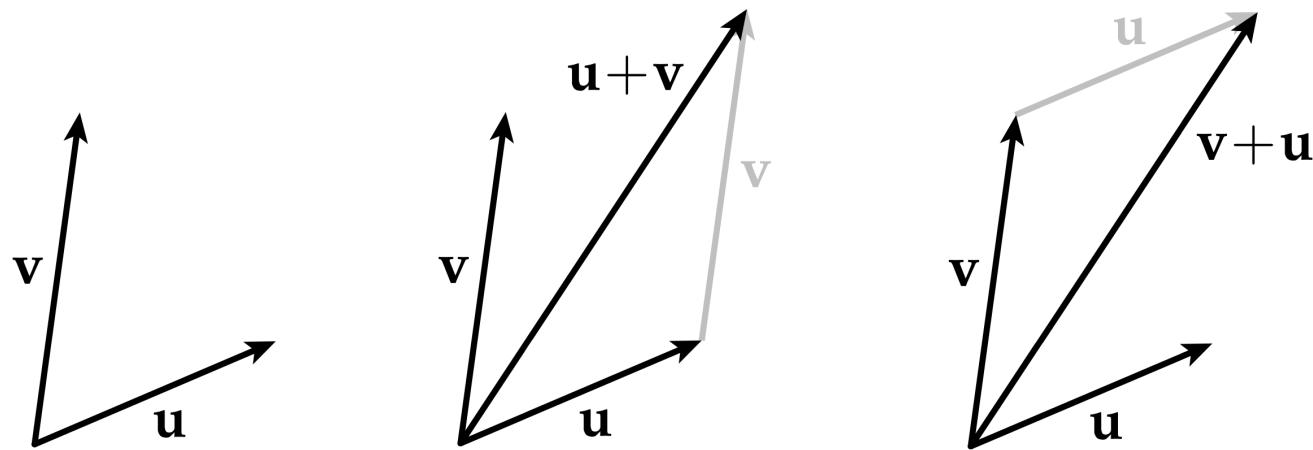
□ 注意⚠：不能直接比较不同系统中的坐标！

- 比如不能比较 (r, θ) 和 (x, y)

向量操作

□ 两个基本操作

□ 首先，我们可以“端到端”地把它们加起来 (addition)：



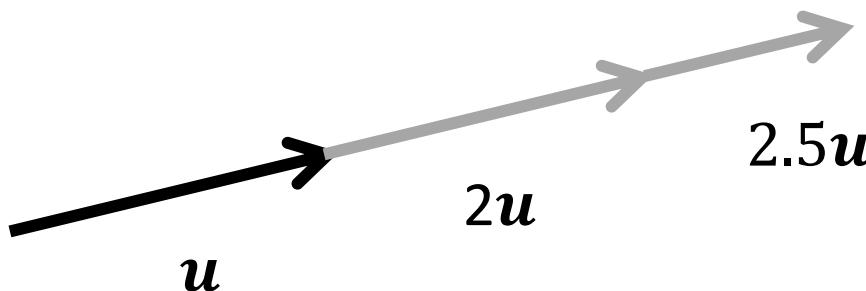
□ 如果我们先沿着 v 走，然后沿着 u 走呢？

□ 事实上，顺序没有关系： $u + v = v + u$

□ 专业术语：向量加法是可交换的 (commutative)

向量操作

□ 另一个基本操作：缩放 (Scaling)



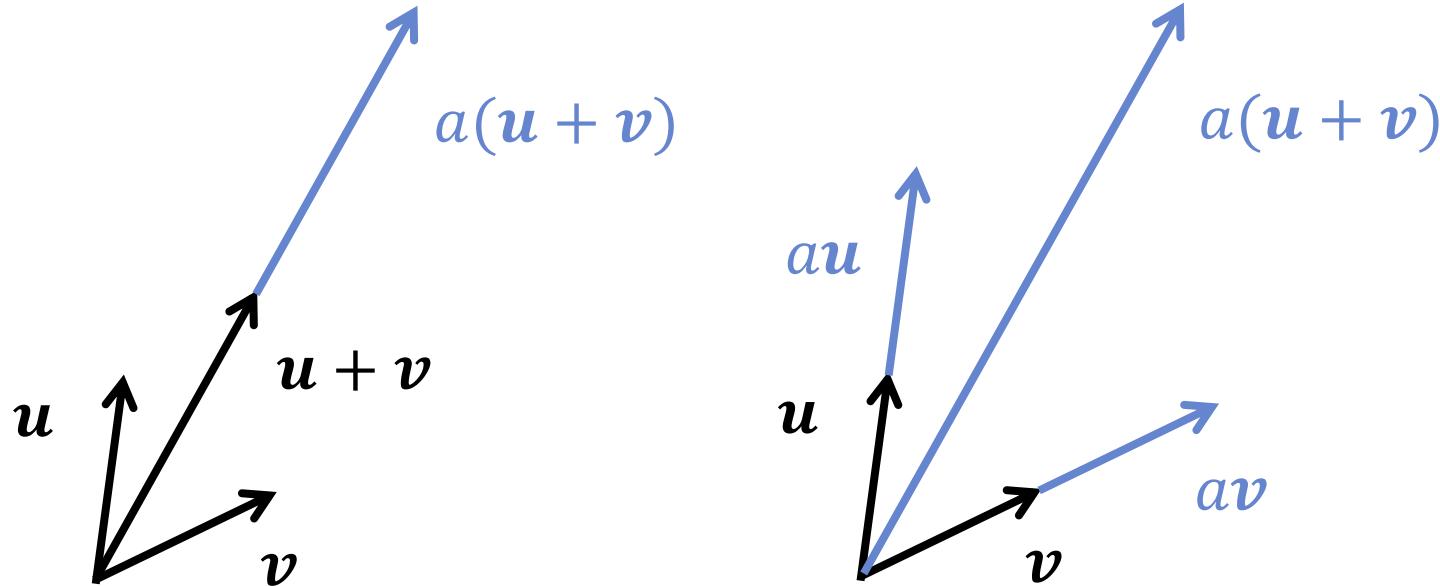
□ 通常，我们可以将向量 \mathbf{u} 乘以一个数字或“标量” a ，得到一个新的向量 $a\mathbf{u}$

□ 根据缩放“小箭头”的几何行为，容易推测出向量乘法具有如下性质：

$$a(b\mathbf{u}) = (ab)\mathbf{u}$$

加法和缩放的组合顺序

□ 缩放已相加的两个向量？或者将两个已缩放的向量相加？



□ 两种顺序得到的结果是一样的

$$a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$$

向量空间 - 正式定义

口重复组合不同操作，可以得到完整的向量应遵守的“规则”

For all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$ and scalars a, b :

- $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
- $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
- There exists a *zero vector* “ $\mathbf{0}$ ” such that $\mathbf{v} + \mathbf{0} = \mathbf{0} + \mathbf{v} = \mathbf{v}$
- For every \mathbf{v} there is a vector “ $-\mathbf{v}$ ” such that $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
- $1\mathbf{v} = \mathbf{v}$
- $a(b\mathbf{v}) = (ab)\mathbf{v}$
- $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
- $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$

口这些规则并不是“从天而降”的，每一条都来自“小箭头”的几何行为

任何满足所有这些属性的对象集合都是向量空间（即使它们看起来不像小箭头）



欧几里德向量空间

□ 最常见的例子：欧几里得 n 维空间 (Euclidean n -dimensional space)

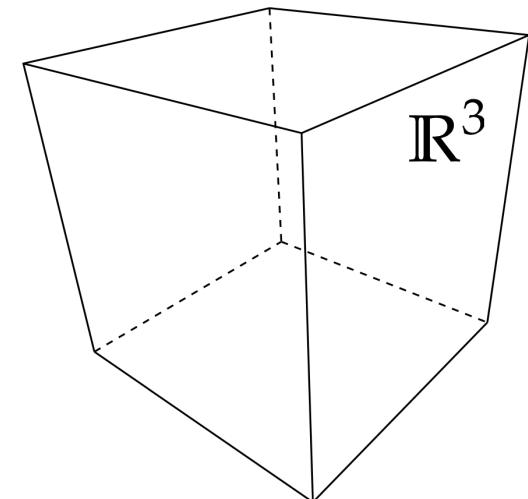
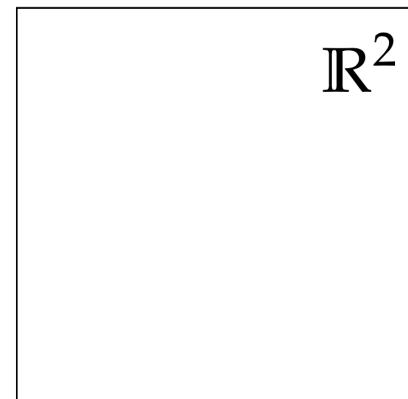
□ 通常记为 \mathbb{R}^n , 表示“ n 个实数”

□ $(1.23, 4.56, \pi/2)$ 即为 \mathbb{R}^3 中的一个点

□ 为什么欧几里德 n 维空间很常见？

- 基于我们生存的世界建模的
- 容易在计算机中编码 (即浮点数列表)

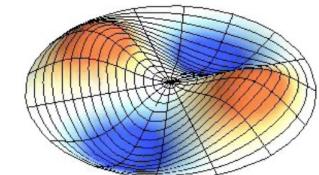
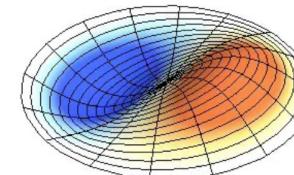
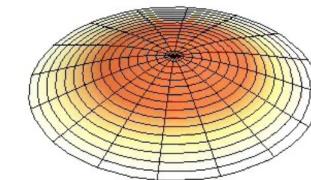
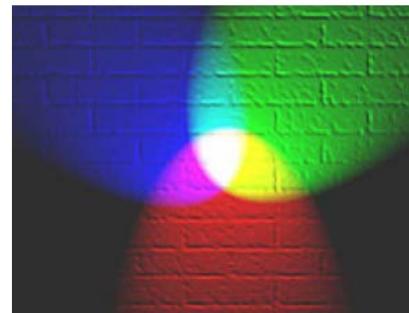
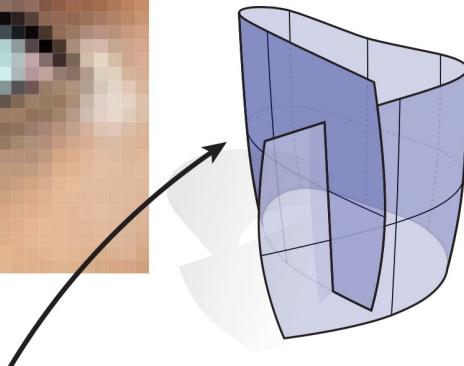
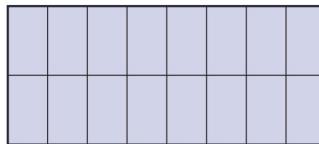
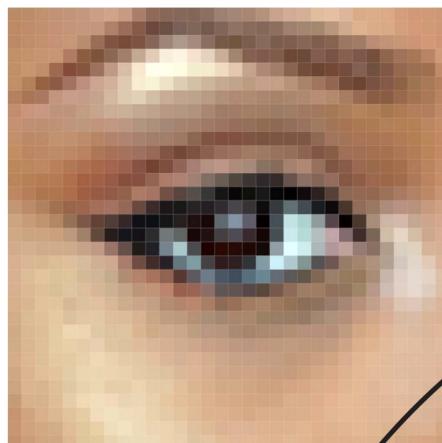
\mathbb{R}



函数向量 Functions as vectors

口计算机图形学中向量空间的另一个非常重要的例子是**函数空间 (function space)**

口为什么？因为图形中的许多对象都可以看作是函数，如图像、光源的辐射度、表面、模态振动等

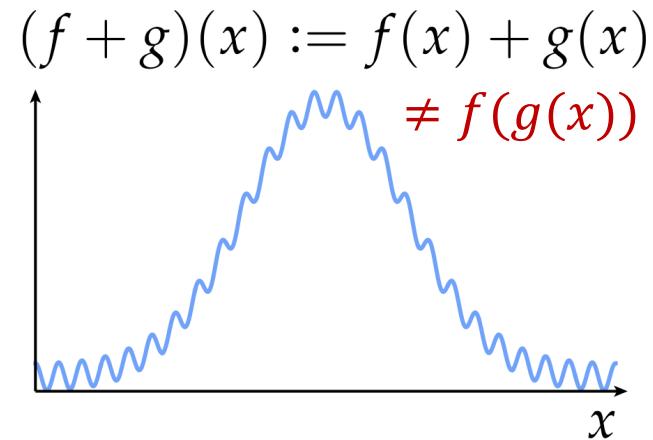
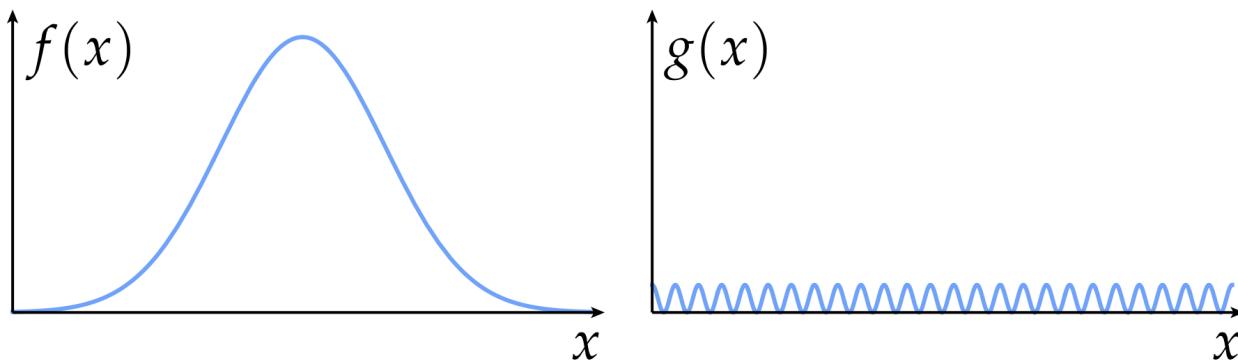


These are all vectors! :-)

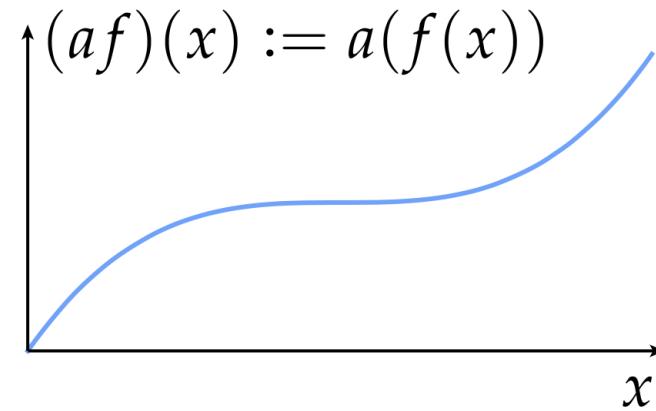
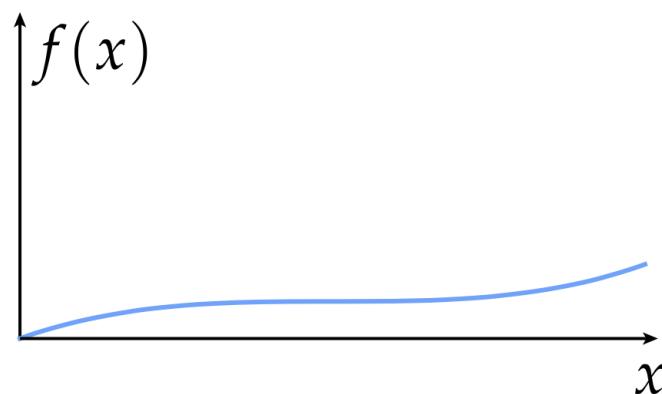
函数向量

□ 函数是否表现出与“小箭头”相同的行为？

□ 我们可以将两个函数相加（例如重叠两张图片）：



□ 我们同样可以缩放一个函数：



函数向量

口其他的属性呢？

For all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$ and scalars a, b :

- $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
- $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
- There exists a *zero vector* “ $\mathbf{0}$ ” such that $\mathbf{v} + \mathbf{0} = \mathbf{0} + \mathbf{v} = \mathbf{v}$
- For every \mathbf{v} there is a vector “ $-\mathbf{v}$ ” such that $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
- $1\mathbf{v} = \mathbf{v}$
- $a(b\mathbf{v}) = (ab)\mathbf{v}$
- $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
- $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$

口比如说，“零向量”是对所有 x 等于 0 的函数

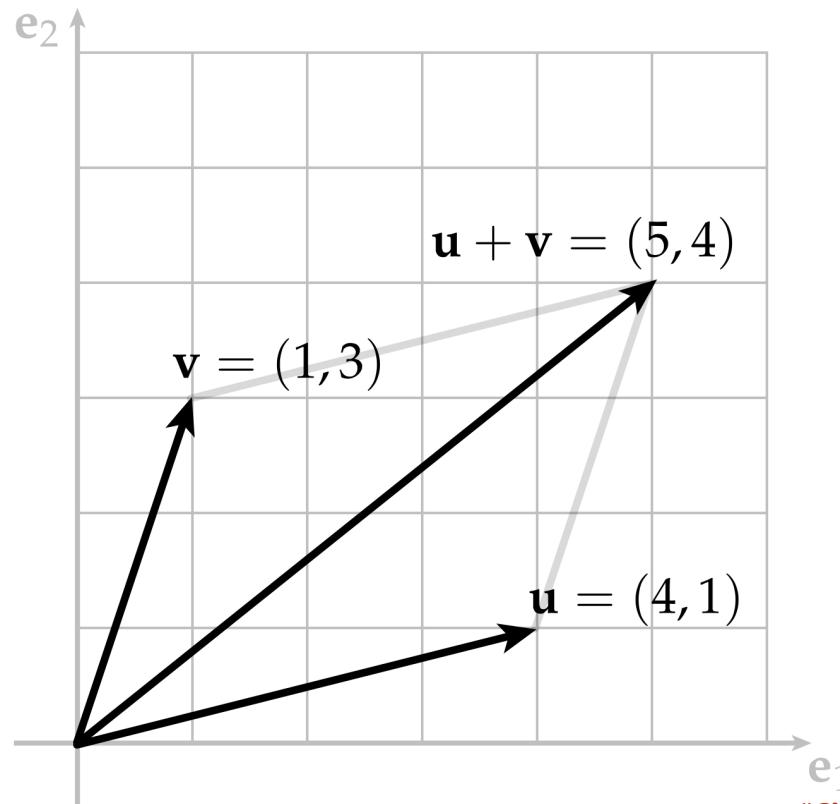
口答案是肯定的，函数也满足所有向量属性，即便它们看起来并不像一个“小箭头”

坐标系中的向量

□ 到目前为止，我们仅以图片的形式展示向量运算

□ 我们如何用数字对向量进行计算？

□ 向量的坐标表示：

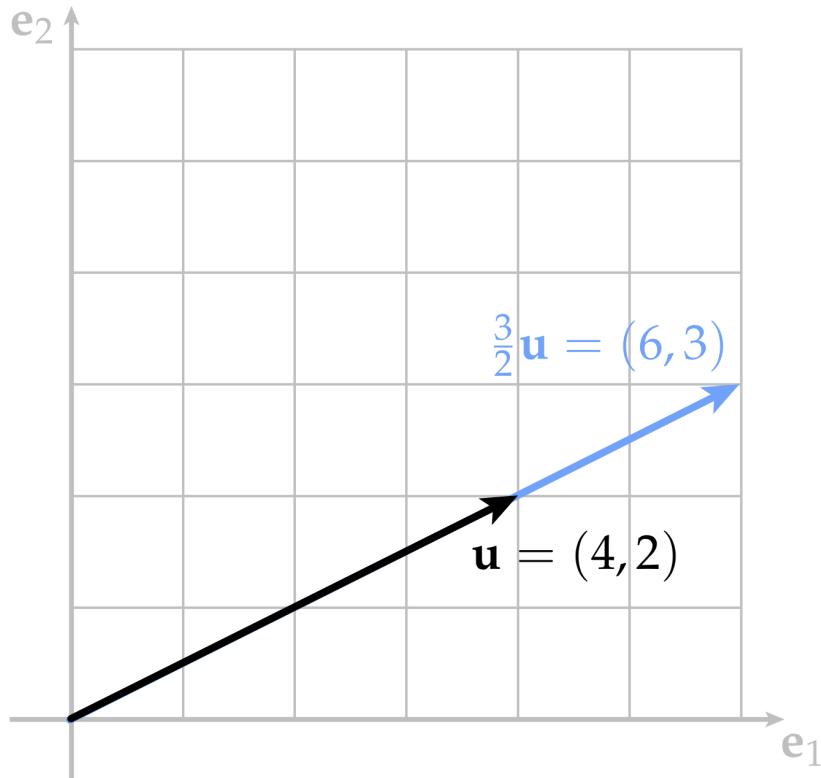


$$\begin{aligned}\mathbf{u} + \mathbf{v} \\ &= (1, 3) + (4, 1) \\ &= (1 + 4, 3 + 1) \\ &= (5, 4)\end{aligned}$$

*Side note: does it make sense to add vectors encoded as (r, θ) ?

在坐标系中缩放向量

□ 我们同样可以在坐标中缩放向量



$$\begin{aligned}\frac{3}{2}\mathbf{u} &= \frac{3}{2}(4, 2) \\ &= (4 \cdot 3/2, 2 \cdot 3/2) \\ &= (12/2, 6/2) \\ &= (6, 3)\end{aligned}$$

从几何到代数

□上述计算是否满足下列向量应遵守的“规则？”

For all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$ and scalars a, b :

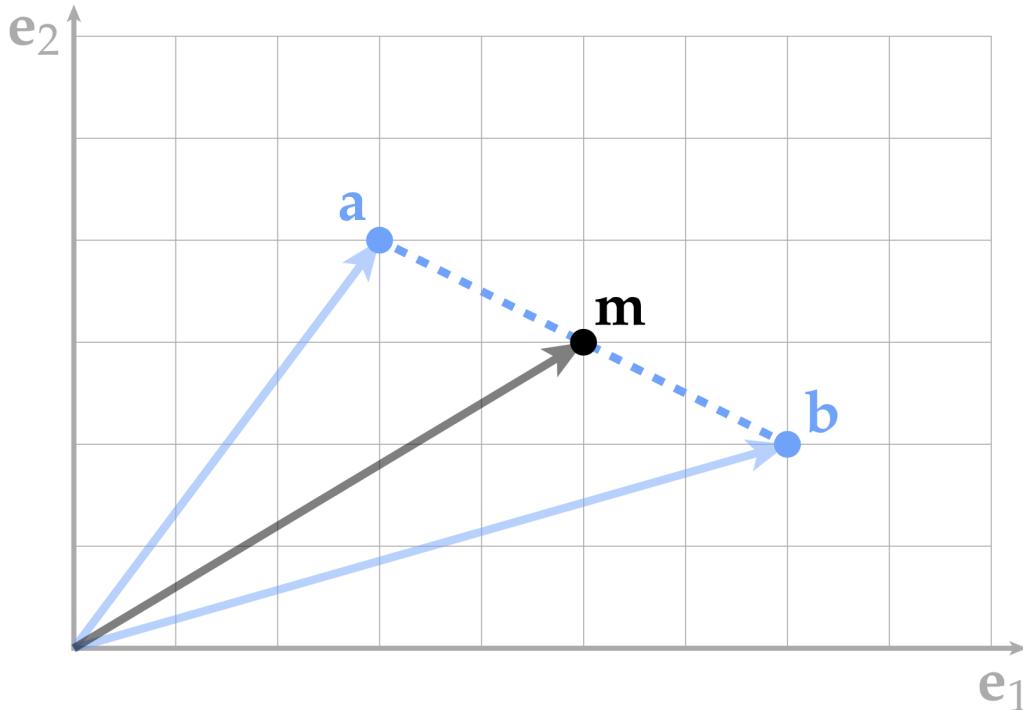
- $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
- $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
- There exists a *zero vector* “ $\mathbf{0}$ ” such that $\mathbf{v} + \mathbf{0} = \mathbf{0} + \mathbf{v} = \mathbf{v}$
- For every \mathbf{v} there is a vector “ $-\mathbf{v}$ ” such that $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
- $1\mathbf{v} = \mathbf{v}$
- $a(b\mathbf{v}) = (ab)\mathbf{v}$
- $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
- $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$

□例如，对于任意两个向量 $\mathbf{u} = (u_1, u_2)$ 和 $\mathbf{v} = (v_1, v_2)$:

$$\begin{aligned}\mathbf{u} + \mathbf{v} &= (u_1, u_2) + (v_1, v_2) = (u_1 + v_1, u_2 + v_2) = \\ &= (v_1 + u_1, v_2 + u_2) = (v_1, v_2) + (u_1, u_2) = \mathbf{v} + \mathbf{u}\end{aligned}$$

计算中点

- 将几何观测转化为代数规则便于**符号操作**和**数值计算**
- 例如，我如何计算 $a = (3, 4)$ 和 $b = (7, 2)$ 的中点 m ？

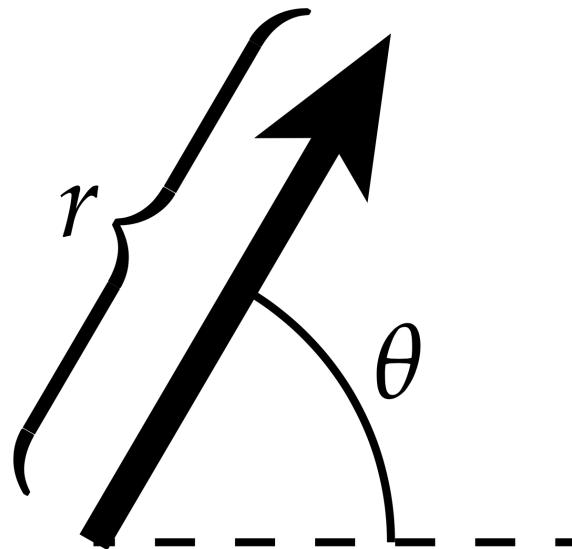


$$\begin{aligned}\mathbf{m} &= \frac{1}{2}(\mathbf{a} + \mathbf{b}) \\ &= \frac{1}{2}((3, 4) + (7, 2)) \\ &= \frac{1}{2}(10, 6) \\ &= (5, 3)\end{aligned}$$

Questions?

测量向量

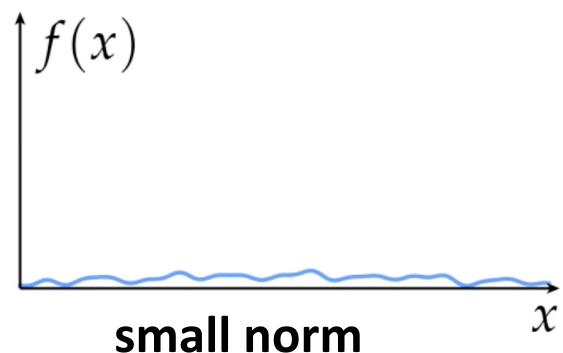
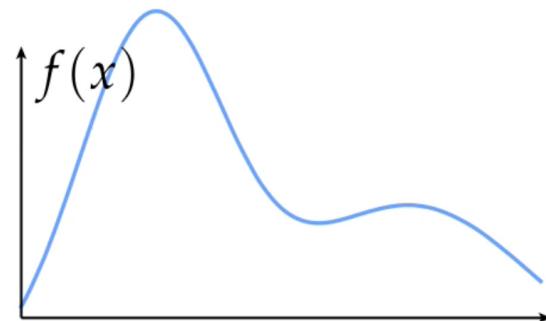
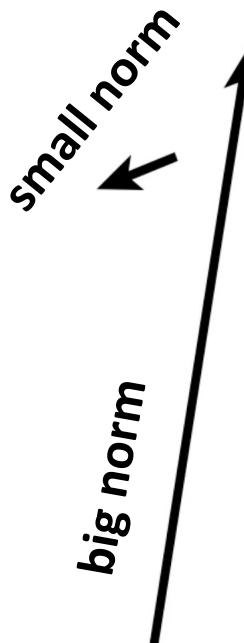
- 早些时候我们问，“向量编码了什么信息？”
- 方向 (orientation) 和大小 (magnitude)**
- 我们如何测量这些值？



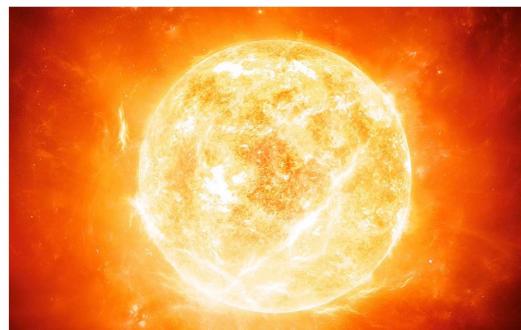
向量的范数 Norm

□大小 (magnitude)：对于给定的向量 v , 我们想给它关联一个数字 $|v|$, 称为它的**长度 (length)**或**范数 (norm)**

□直观地说, 范数应该捕捉向量“大”的程度



small norm



big norm

长度的自然属性 - 非负性

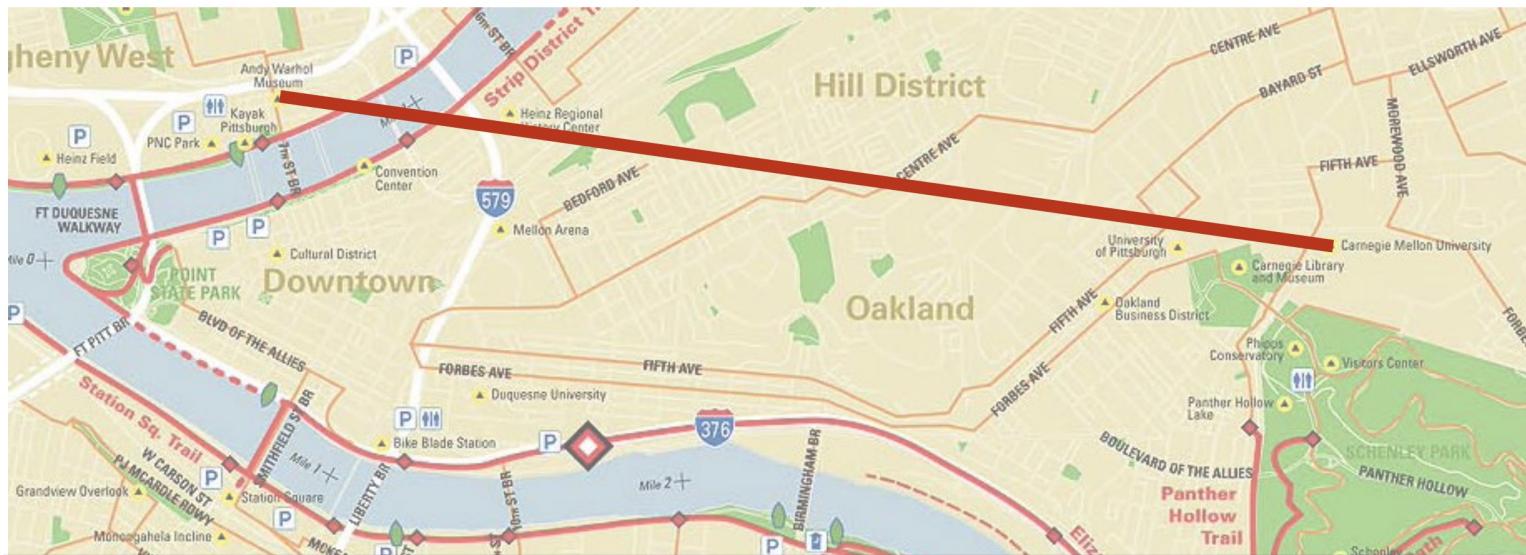
□ 向量的范数（或长度）满足什么性质？

□ 首先，它不应该是负数

$$|\mathbf{u}| \geq 0$$

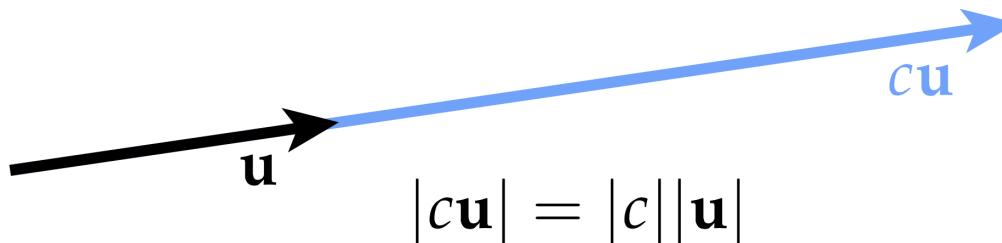
□ 并且只有零向量的长度是 0

$$|\mathbf{u}| = 0 \iff \mathbf{u} = 0$$

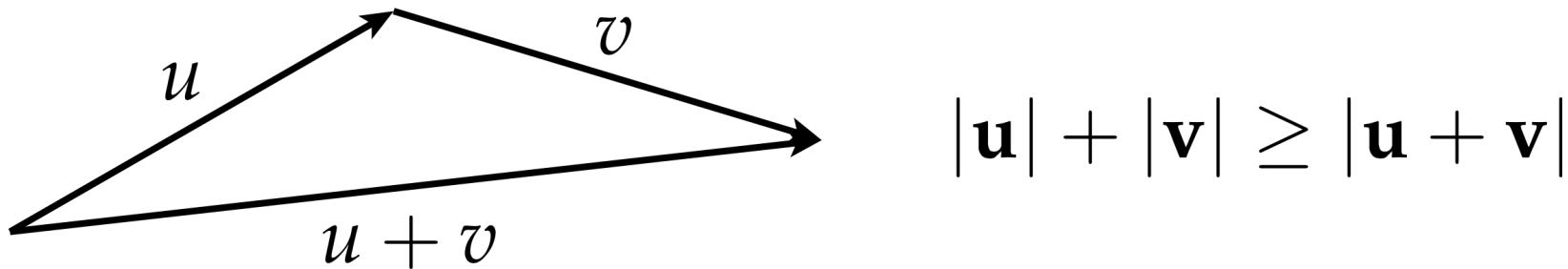


长度的自然属性

□ 然后，用因子 c 缩放向量，其范数应该按相同的量缩放



□ 最后，我们知道两点之间的最短路径总是沿着一条直线



□ 该性质也被称为“三角不等式 (Triangle inequality)”

范数 - 正式定义

□ 范数是一个可以为任意向量分配一个数值并满足以下性质的函数，适用于所有向量 \mathbf{u} , \mathbf{v} 和所有标量 a

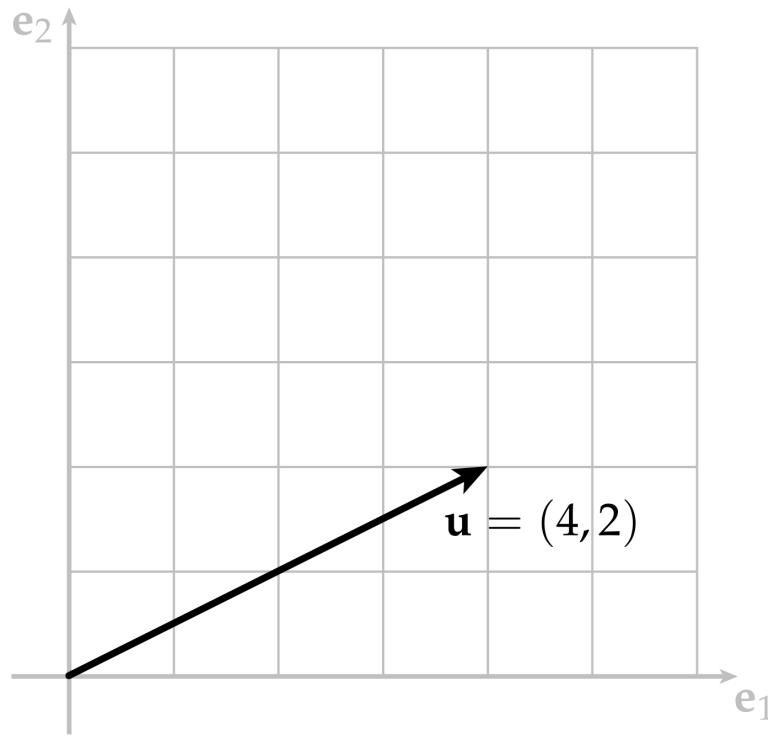
- $|\mathbf{v}| \geq 0$
- $|\mathbf{v}| = 0 \iff \mathbf{v} = \mathbf{0}$
- $|\alpha\mathbf{v}| = |\alpha||\mathbf{v}|$
- $|\mathbf{u}| + |\mathbf{v}| \geq |\mathbf{u} + \mathbf{v}|$

□ 每一条规则都可以通过画一幅具体的几何图来解释

笛卡尔坐标系中的欧氏范数

□ 标准范数 (standard norm) 是 n 维向量的欧几里得范数 (Euclidean norm)

$$|\mathbf{u}| = |(u_1, \dots, u_n)| := \sqrt{\sum_{i=1}^n u_i^2}$$



Example: $\mathbf{u} = (4, 2)$

$$\begin{aligned} |\mathbf{u}| &= \sqrt{4^2 + 2^2} \\ &= 2\sqrt{5} \end{aligned}$$

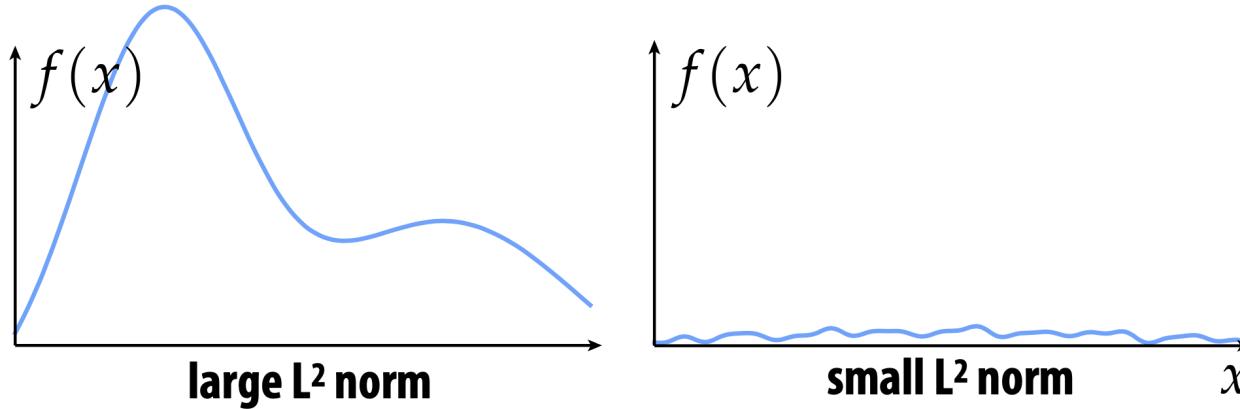
函数的 L^2 范数

□一个更少见的范数， L^2 范数，其测量了一个函数的“大小”

□考虑单位区间 $[0,1]$ 上的实值函数，其平方具有定义良好的积分。其 L^2 范数定义为：

$$\|f\| := \sqrt{\int_0^1 f(x)^2 dx}$$

□与欧式范数没有太大区别，只是用积分代替了和



函数的 L^2 范数 - 举例

□ 考虑一个定义在单位区间 $[0, 1]$

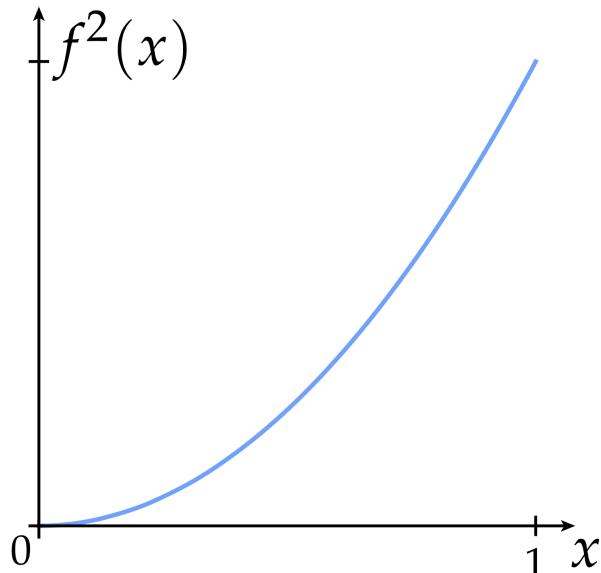
的函数 $f(x) := x\sqrt{3}$

□ Q: 该函数的 L^2 范数是什么?

$$\|f\| := \sqrt{\int_0^1 f(x)^2 dx}$$

□ A:

$$\|f\|^2 = \int_0^1 3x^2 dx = [x^3]_0^1 = 1^3 - 0^3 = 1$$



$$\Rightarrow \|f\| = \sqrt{1} = 1.$$

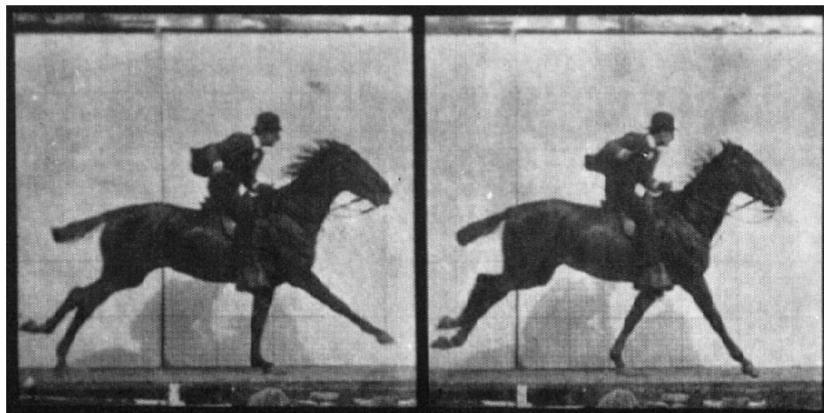
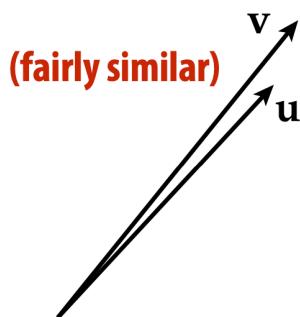
为清楚起见，我们使用 $\|\cdot\|$ 表示函数的范数，使用 $|\cdot|$ 表示 \mathbb{R}^n 空间中向量的范数

内积 Inner product - 动机

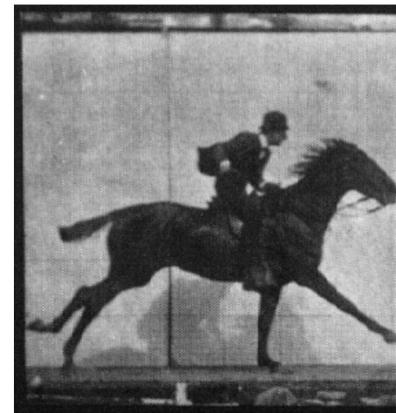
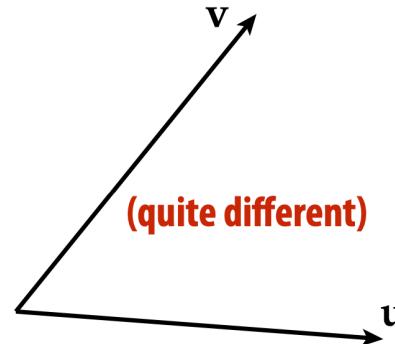
□ 我们还能测量什么？

□ 除了长度，我们说向量还有方向属性

□ 正如范数测量长度一样，内积测量向量“对齐”的程度



(fairly similar)



(quite different!)

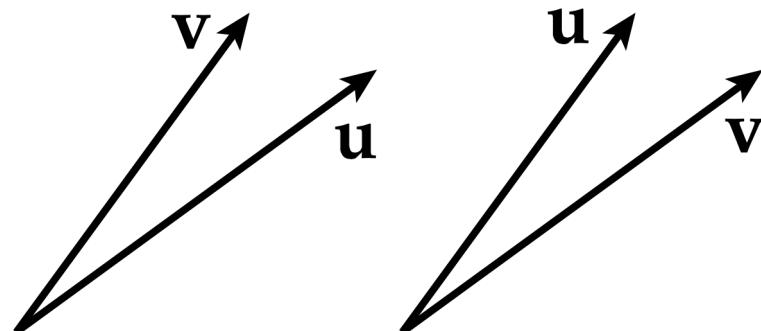
这两张图相似吗？



内积 - 对称性

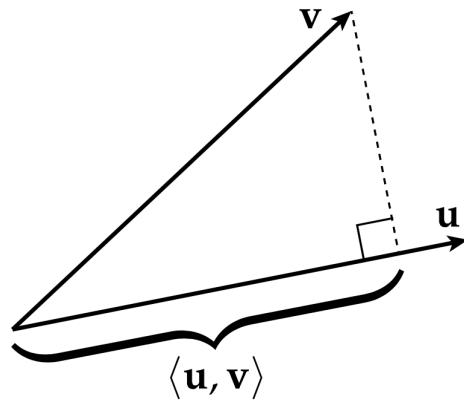
- 使用符号 $\langle u, v \rangle$ (有些也写成 $u \cdot v$) 来表示内积，有时也称为**标量积 scalar product** 或**点积 dot product**
- 当测量两个向量 u, v 的对齐 (alignment) 程度时，你会期望哪些性质？
- 一个显而易见的性质是：**顺序不重要**
 - u 与 v 的对齐和 v 与 u 的对齐是一样的：

$$\langle u, v \rangle = \langle v, u \rangle$$



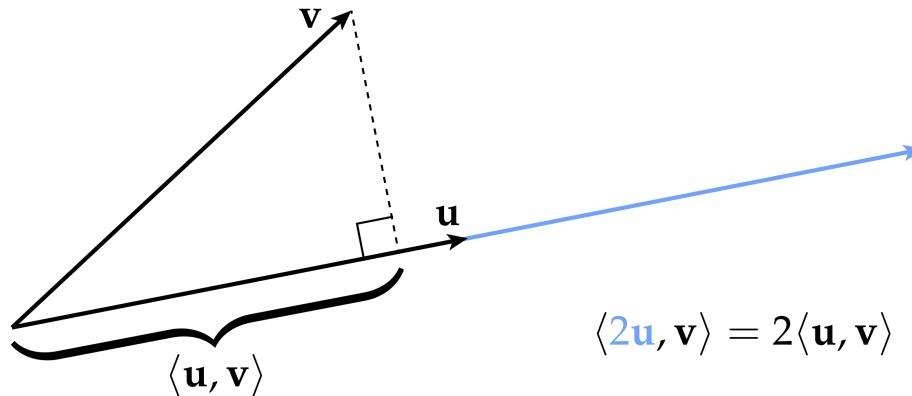
内积 - 投影和伸缩

对于单位向量 $|\mathbf{u}| = |\mathbf{v}| = 1$, 内积测量一个向量沿另一个向量方向的程度



这个属性是不是对称的?

如果我们缩放任一向量, 内积也会缩放



内积 - 非负性

□ 此外，向量应始终与自身对齐

- 向量与自身的内积是正的（或至少是非负的）



$$\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$$

□ 对于单位长度的向量，我们有

$$\langle \mathbf{u}, \mathbf{u} \rangle = 1$$

□ Q：一般来说， $\langle \mathbf{u}, \mathbf{u} \rangle$ 必须等于什么？

□ A：令 $\hat{\mathbf{u}} := \mathbf{u}/|\mathbf{u}|$ (单位向量)，我们有

$$\langle \mathbf{u}, \mathbf{u} \rangle = \langle |\mathbf{u}| \hat{\mathbf{u}}, |\mathbf{u}| \hat{\mathbf{u}} \rangle = |\mathbf{u}|^2 \langle \hat{\mathbf{u}}, \hat{\mathbf{u}} \rangle = |\mathbf{u}|^2 \cdot 1 = |\mathbf{u}|^2$$

内积 - 正式的定义

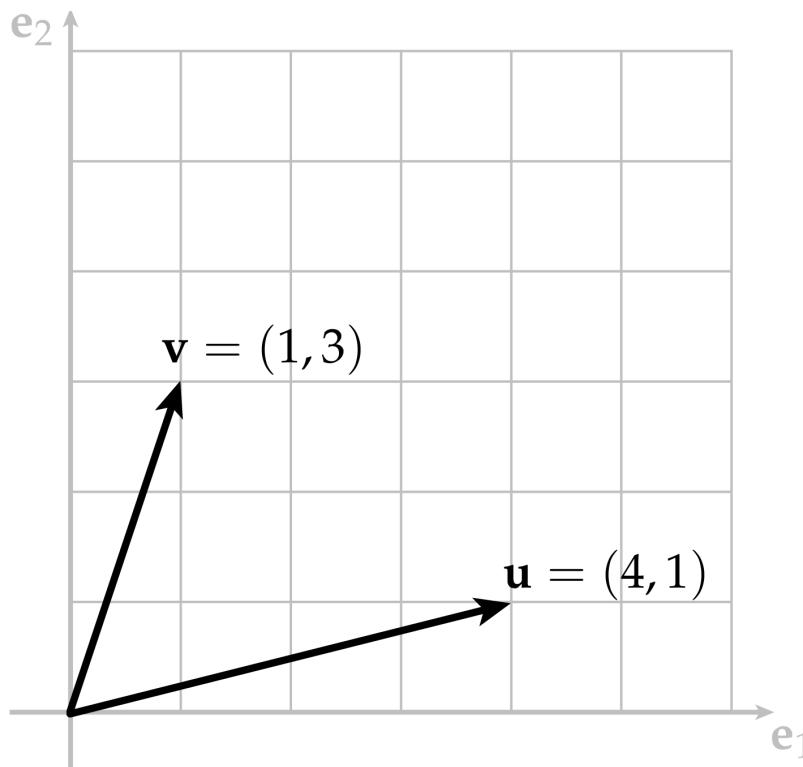
口内积是将任意两个向量 \mathbf{u}, \mathbf{v} 映射到一个数字 $\langle \mathbf{u}, \mathbf{v} \rangle$ 的函数，满足以下属性

- $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$
- $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$
- $\langle \mathbf{u}, \mathbf{u} \rangle = 0 \iff \mathbf{u} = \mathbf{0}$
- $\langle a\mathbf{u}, \mathbf{v} \rangle = a\langle \mathbf{u}, \mathbf{v} \rangle$
- $\langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle$

笛卡尔坐标系中的内积

口标准内积 (standard inner product) , 即欧几里得内积 (Euclidean inner product), 对两个 n 维向量进行运算

$$\langle \mathbf{u}, \mathbf{v} \rangle = \langle (u_1, \dots, u_n), (v_1, \dots, v_n) \rangle := \sum_{i=1}^n u_i v_i$$



Example:

$$\mathbf{u} = (4, 1)$$
$$\mathbf{v} = (1, 3)$$

$$\langle \mathbf{u}, \mathbf{v} \rangle = 4 \cdot 1 + 1 \cdot 3 = 7$$

函数的 L^2 内积 - 例子

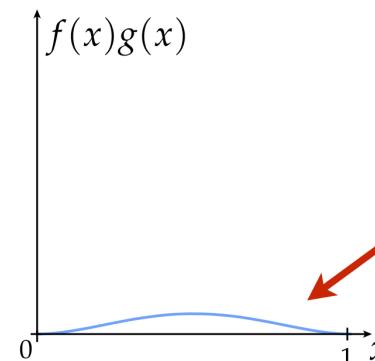
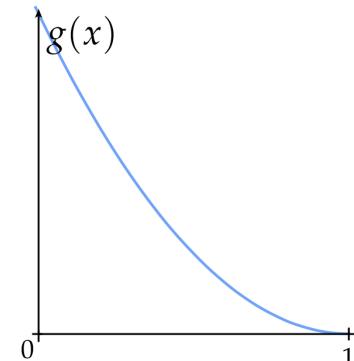
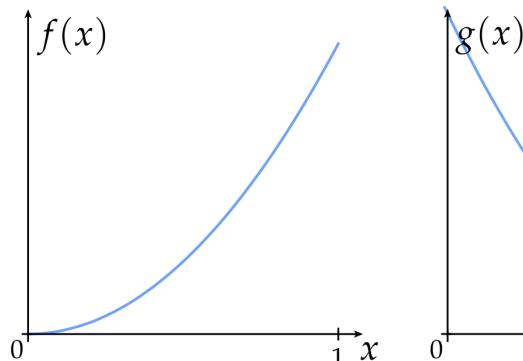
口就像函数有范数一样，我们也可以定义一个内积来衡量两个函数“对齐”的程度

口例如，对于单位区间上的平方可积函数：

$$\langle\langle f, g \rangle\rangle := \int_0^1 f(x)g(x) dx$$

Example: $f(x) := x^2, \quad g(x) := (1 - x)^2$

$$\langle\langle f, g \rangle\rangle = \int_0^1 x^2(1 - x)^2 dx = \dots = \frac{1}{30}$$



small number;
functions don't
“line up” much!

其他范数和内积

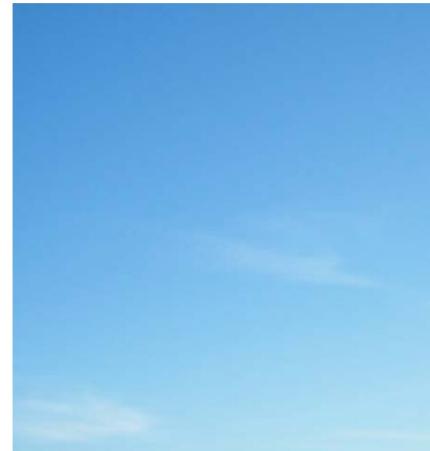
- 口有多种方法测量一个信号“有多大”（范数）或两个信号“对齐得有多好”（内积）
- 口范数或内积的选择取决于具体的应用
- 口例如，假设我们想要判断图像是否包含“有趣的东西”
- 口可以尝试测量**导数的范数（捕捉边缘）**：



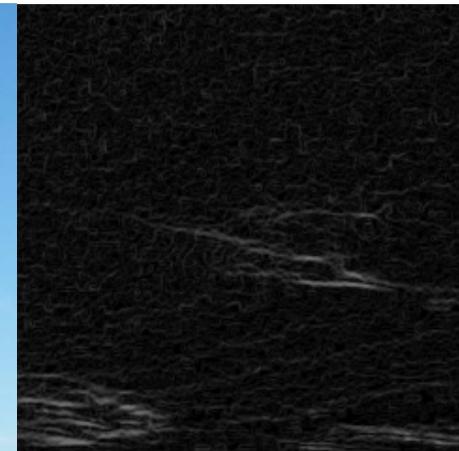
暗淡的



大



明亮的



小

Questions?

线性映射

□一开始我们说线性代数研究**向量空间 (vector spaces)**及其之间的**线性映射 (linear maps)**

□我们已经学习了如何处理向量空间：范数、内积...

□那什么是线性映射？它对图形学有什么用？

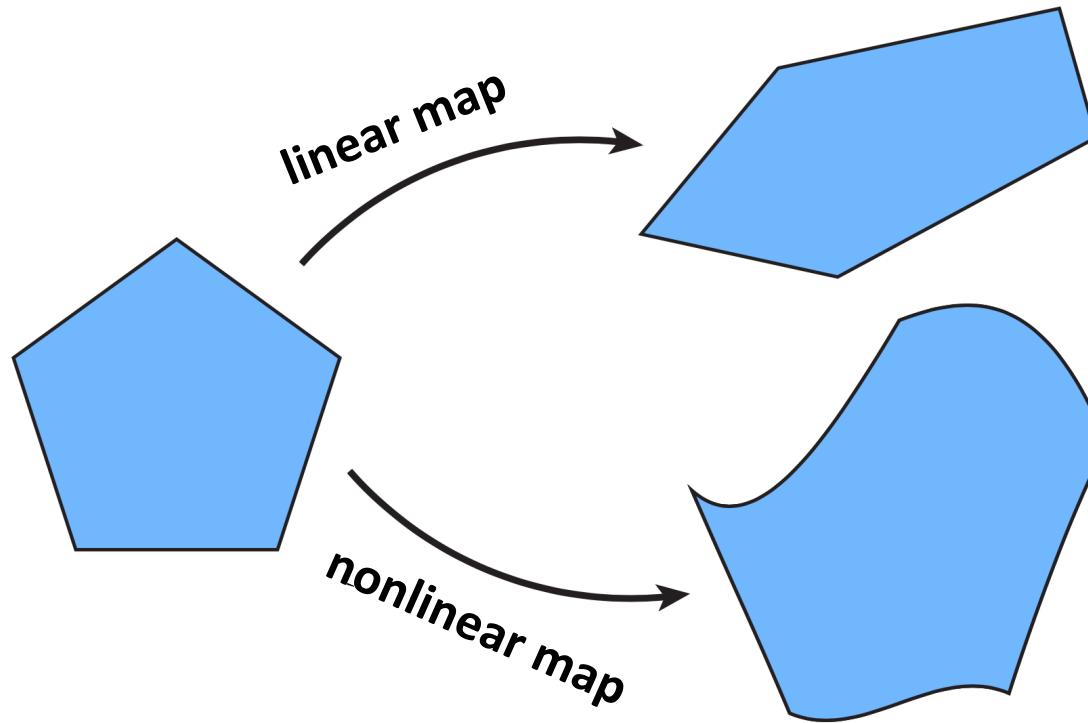
□先回答第二个问题：

- 计算机易于求解线性方程组
- 基本变换（旋转、平移、缩放）可以表示为线性映射
- 所有映射都可以近似为短距离/短时间内的线性映射（泰勒定理，Taylor' theorem)
 - 这种近似适用于所有几何体、动画、渲染和图像处理

线性映射 - 几何定义

□ 线性映射是什么？

□ 我们可以先从视觉上思考这个问题

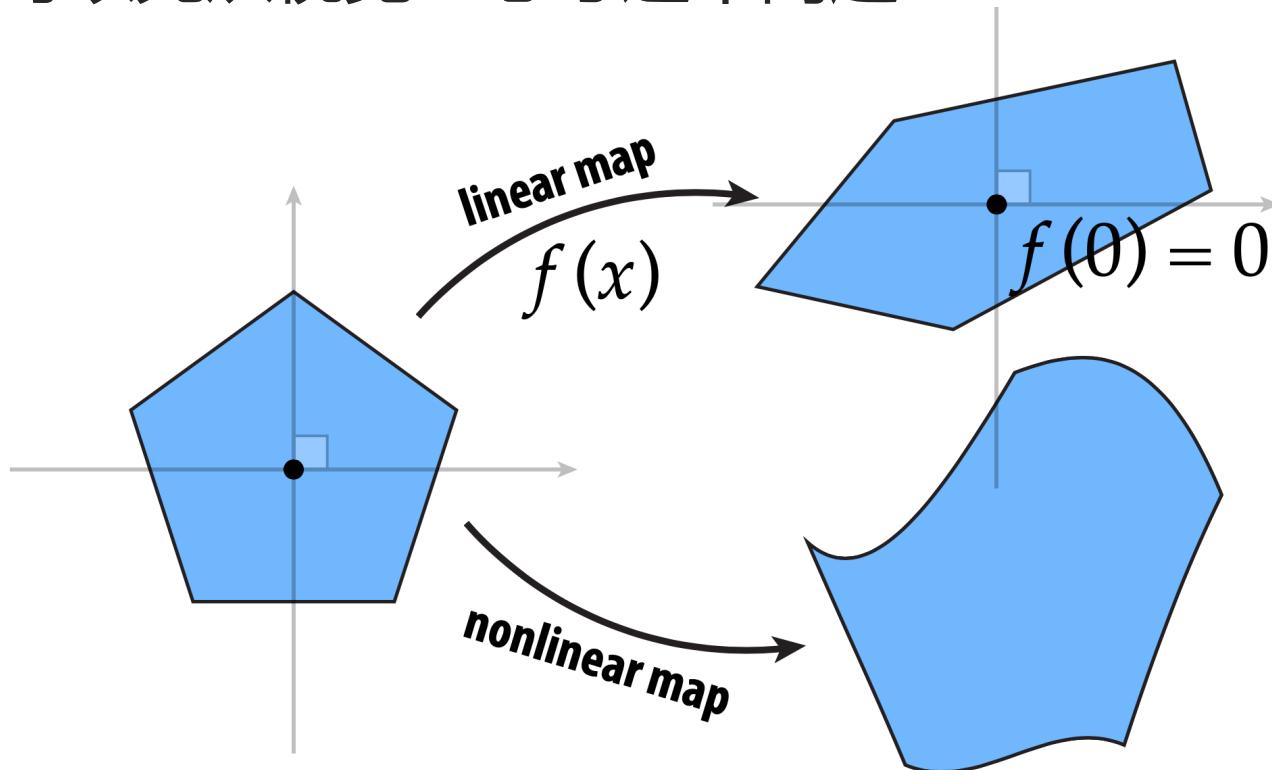


Key idea: *linear maps take lines to lines**

线性映射 - 几何定义

□ 线性映射是什么？

□ 我们可以先从视觉上思考这个问题



Key idea: *linear maps take lines to lines**

*...while keeping the origin fixed.

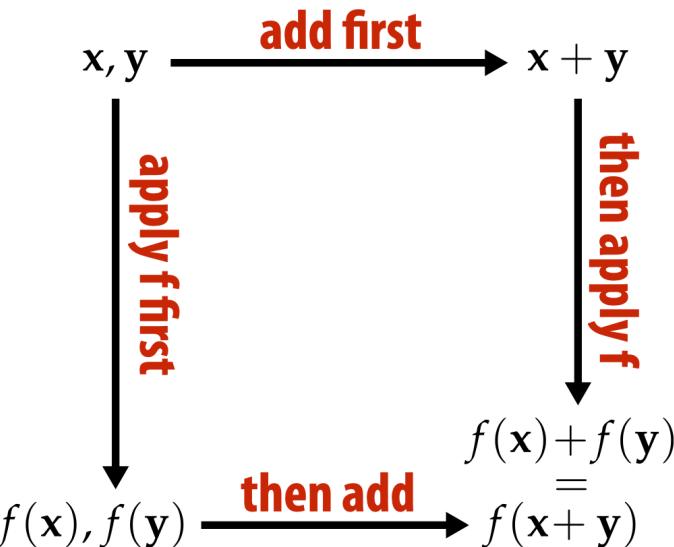
线性映射 - 代数定义

口映射 f 是线性的，如果它将向量映射到向量（不一定是相同维度），并且对于所有向量 \mathbf{u}, \mathbf{v} 和标量 a ，有

$$f(\mathbf{u} + \mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$$

$$f(a\mathbf{u}) = af(\mathbf{u})$$

口即，不管是先将向量相加再应用映射，还是先映射再相加（缩放也一样），结果都相同：



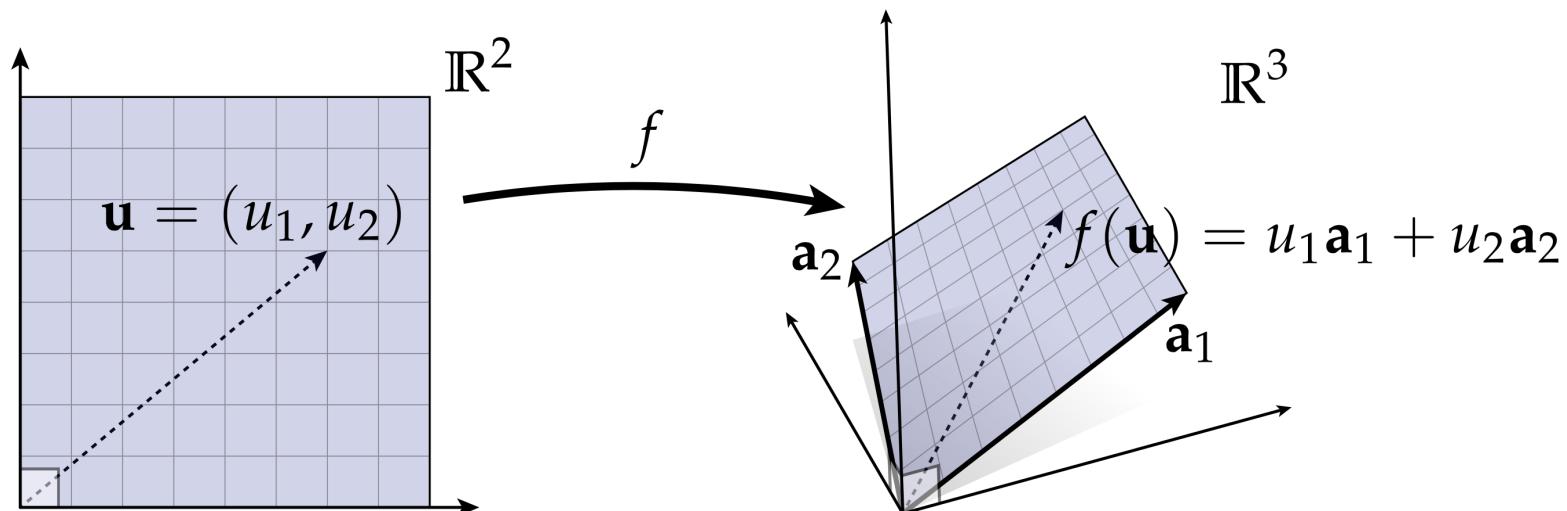
坐标系中的线性映射

□对于 \mathbb{R}^m 和 \mathbb{R}^n 之间的映射（例如 2D 到 3D 的映射），我们可以给出更明确的定义

□一个映射是线性的，当且仅当它可以表示为

$$f(u_1, \dots, u_m) = \sum_{i=1}^m u_i \mathbf{a}_i$$

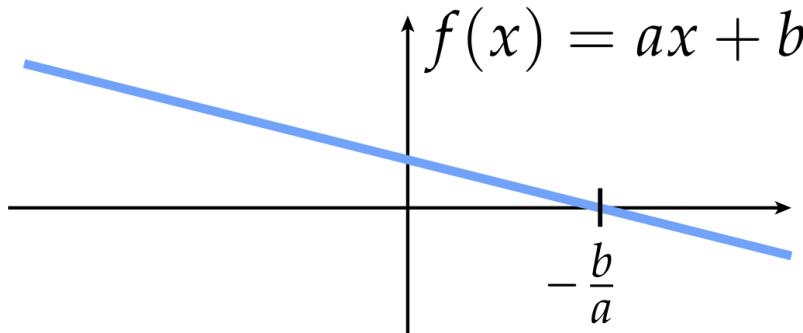
□换句话说，它是一组固定向量 \mathbf{a}_i 的线性组合



Q: Is $f(x) := ax + b$ a linear function?

线性 (linear) 映射和仿射 (affine) 映射

口 答案是否定的（但很容易出错），因为该函数是一条直线



口 然而，它不是一条穿过原点的线，即 $f(0) \neq 0$

口 另一种判断不是线性的方式： $f(x_1 + x_2) \neq f(x_1) + f(x_2)$

$$f(x_1 + x_2) = a(x_1 + x_2) + b = ax_1 + ax_2 + b$$

$$f(x_1) + f(x_2) = (ax_1 + b) + (ax_2 + b) = ax_1 + ax_2 + 2b$$

口 这个函数被称为 affine 函数，而不是 linear 函数

口 之后我们将学习通过齐次坐标将仿射函数转换为线性函数

线性方程组

口线性方程组 (System of linear equations) : 一组方程，
左边是线性函数，右边是常数，比如

$$\begin{aligned}x + 2y &= 3 \\4x + 5y &= 6\end{aligned}$$

口未知变量有时被称为**自由度 (degrees of freedom)**；方
程有时被称为**约束 (constraints)**

口目标：求解同时满足所有约束的自由度：

$$x = 3 - 2y$$

$$4(3 - 2y) + 5y = 6$$

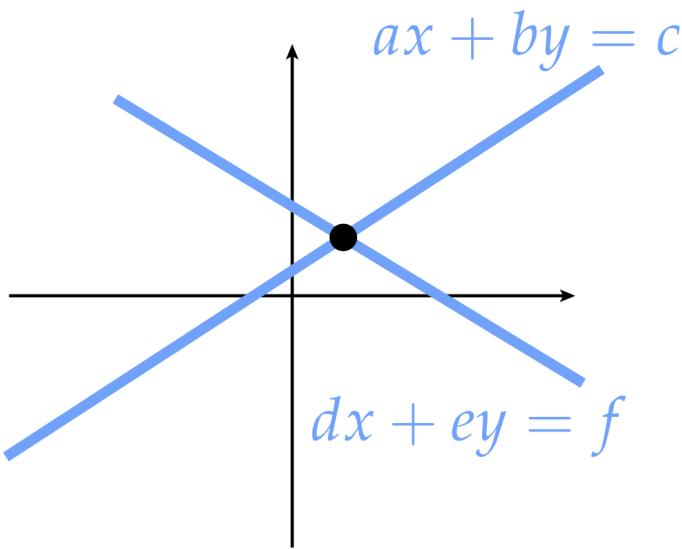
$$\boxed{\begin{aligned}y &= 2 \\x &= -1\end{aligned}}$$

可视化的线性系统

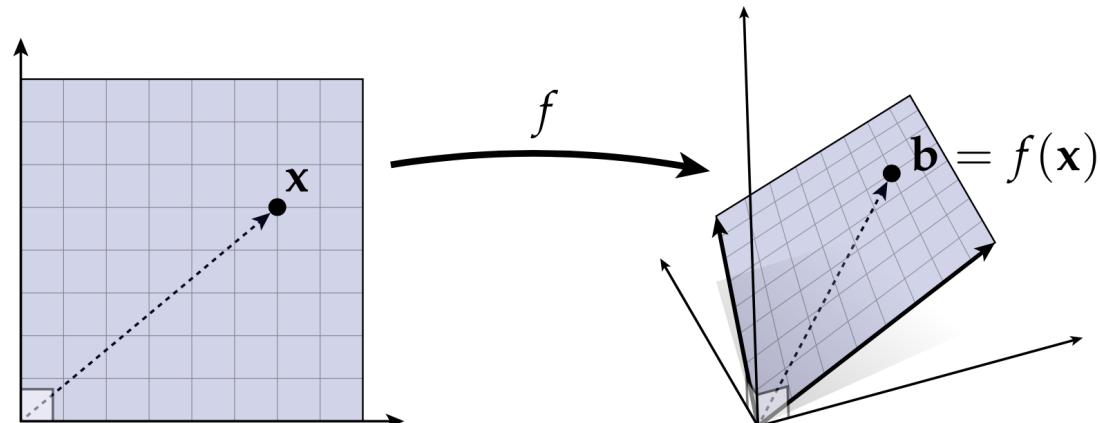
口线性系统可以表示许多不同的任务（模拟、图像处理等）

口对于任何线性系统，都存在一些很好的可视化模型：

Find the point where two lines meet:



GIVEN a point b , FIND the point x that maps to it:

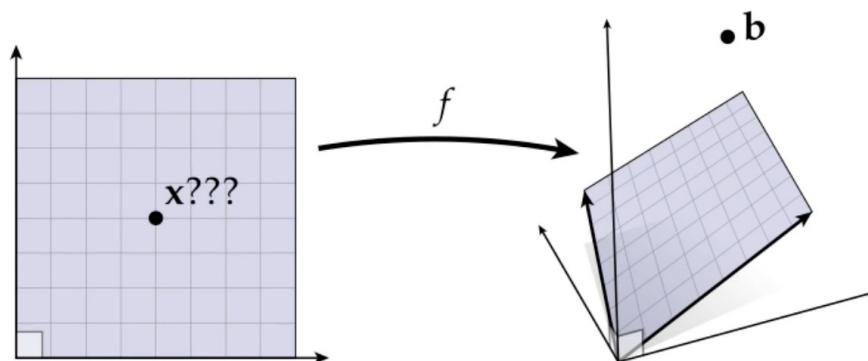
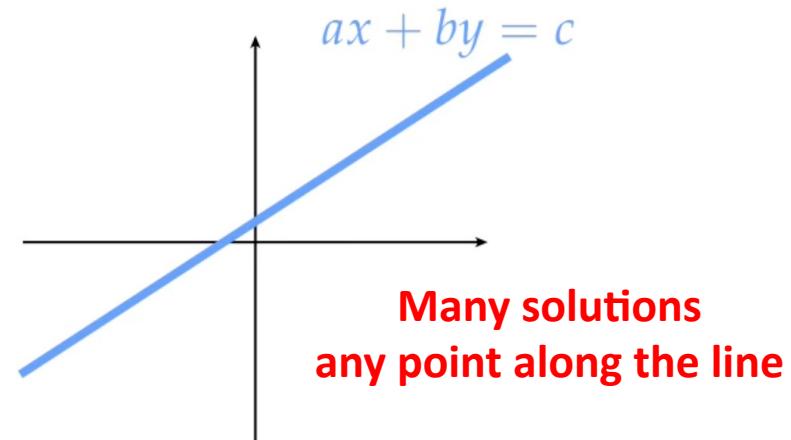
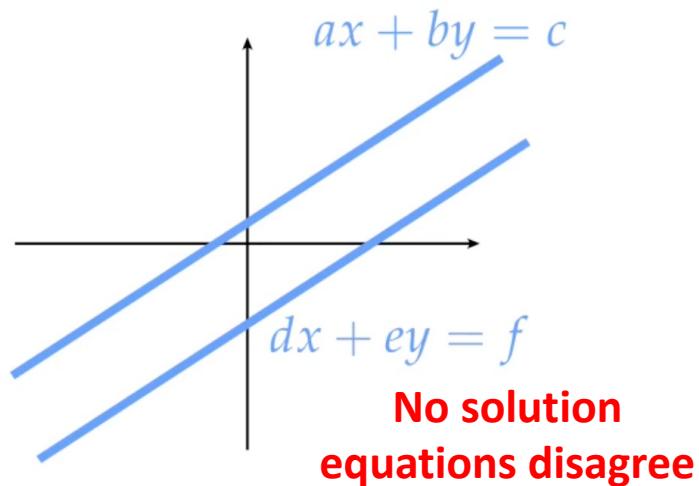


求解线性系统

解的唯一性 uniqueness 和存在性 existence

□ 并不是所有的线性系统都能求解

□ 线性系统也可能有多个解



用矩阵表示线性映射

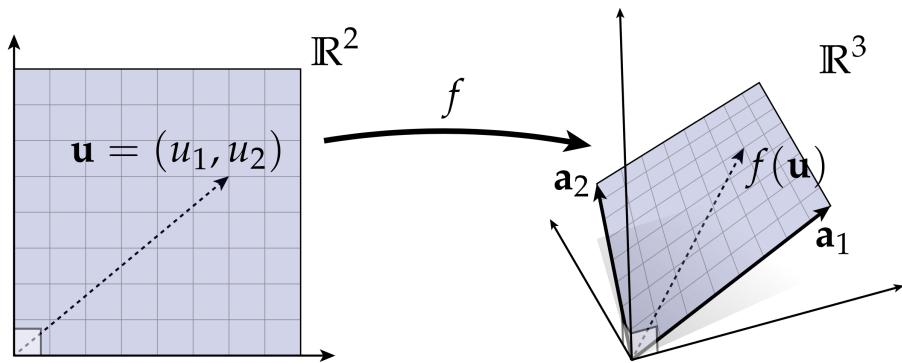
□ 假设我有如下线性变换

$$f(\mathbf{u}) = u_1 \mathbf{a}_1 + u_2 \mathbf{a}_2$$

□ Q：如何把它编码成矩阵？

□ A：直接向量 \mathbf{a} 变成矩阵的列：

$$A := \begin{bmatrix} a_{1,x} & a_{2,x} \\ a_{1,y} & a_{2,y} \\ a_{1,z} & a_{2,z} \end{bmatrix}$$



转换成矩阵乘法的优点：
• 符号操作
• 数值计算加速

□ 原本的线性映射可以用矩阵乘法表示

$$\begin{bmatrix} a_{1,x} & a_{2,x} \\ a_{1,y} & a_{2,y} \\ a_{1,z} & a_{2,z} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} a_{1,x}u_1 + a_{2,x}u_2 \\ a_{1,y}u_1 + a_{2,y}u_2 \\ a_{1,z}u_1 + a_{2,z}u_2 \end{bmatrix} = u_1 \mathbf{a}_1 + u_2 \mathbf{a}_2$$



中山大學 软件工程学院
SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

谢谢

陈壮彬
软件工程学院
chenzhb36@mail.sysu.edu.cn