# Шугаман алгебр

Dr. Juan Carlos Niebles
Stanford AI Lab

Prof. Fei–Fei Li
Stanford Vision Lab

# Агуулга

- Вектор болон матриц
  - Мартицын үндсэн үйлдлүүд
  - Тусгай матриц
- Хувиргалтын матриц
  - Ижил(Homogeneous) координат
  - хувиргах
- Урвуу матриц
- Матрицын эгнээ
- Ганц тооны задаргаа (Singular Value Decomposition- SVD)
  - Зургын файлын хэмжээг шахахад хэрэглэх
  - Гол компонентийн дүн шинжилгээнд ашиглах (Principal Component Analysis-PCA)
  - Компьютер алгоритм

# Агуулга

- Вектор болон матриц
  - Мартицын үндсэн үйлдлүүд
  - Тусгай матриц
- Хувиргалтын матриц
  - Ижил(Homogeneous) координат
  - Шилжүүлэх
- Урвуу матриц
- Матрицын эгнээ
- Ганц тооны задаргаа (Singular Value Decomposition- SVD)
  - Зургын файлын хэмжээг шахахад хэрэглэх
  - Гол компонентийн дүн шинжилгээнд ашиглах (Principal Component Analysis-PCA)
  - Компьютер алгоритм

Вектор болон матрицууд нь ямар нэг зүйлийг илэрхийлэх дараалсан тоонуудын цуглуулга: орон зайн шилжилт, маштаблах хүчин зүйл, цэгийн тодрол гэх мэт. Бид эдгээр зүйлсийн нийтлэг хэрэглээ, стандарт үйлдлүүд ийг тодорхойлно.

# Вектор

- Баганан вектор $\mathbf{v} \in \mathbb{R}^{n \times 1}$

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

- Мөрөн вектор $\mathbf{v}^T \in \mathbb{R}^{1 \times n}$

$$\mathbf{v}^T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

$T$ Хувиргалтын үйлдлийг илэрхийлнэ.

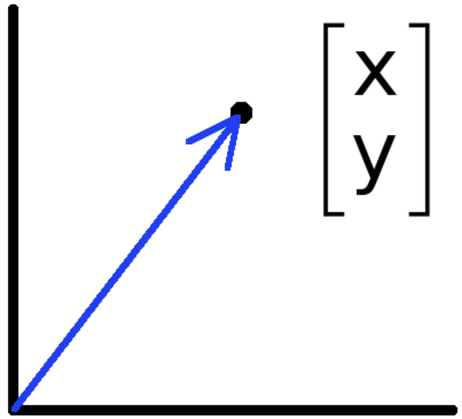
# Вектор

- Бид энэ хичээл дээр баганан векторын хувьд авч үзнэ.

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

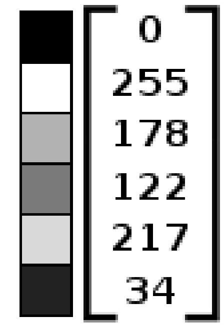
- Магадгүй та MATLAB дээр програмчлахдаа векторын байршилыг мөрөн хэлбэрээр авч үзэж болох юм.
- MATLAB дээр **V'** гэж бичин *V* векторыг шилжүүлнэ. (Гэвч хичээлийн материалд бид ихэнхдээ шилжилтийг заахдаа "V *эхлэл*" утга бүхий $V^T$ ашиглана. )

# Векторын чухал 2 хэрэглээ




- Вектор нь 2D, 3D орон зай дахь эхлэлийг (offset) дүрслэнэ.
- Цэгүүд нь эхлэлээс салаасан вектор

- Өгөгдөл (пиксэл, дүрсийн гол цэг дээр градиент, гэх мэт)-ийг мөн вектор гэж үзэж болно.




- Ийм векторууд нь геометр тайлбаргүй боловч “алслалт” – тай адил тооцоолол нь тодорхой утгатай байна.

# Матриц

- A матриц $\mathbf{A} \in \mathbb{R}^{m \times n}$ нь m мөр n багана тай тоонуудын жагсаалт юм.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & & & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

- Хэрэв $m = n$, $\mathbf{A}$ Мартицыг квадрат матриц гэж нэрлэнэ.

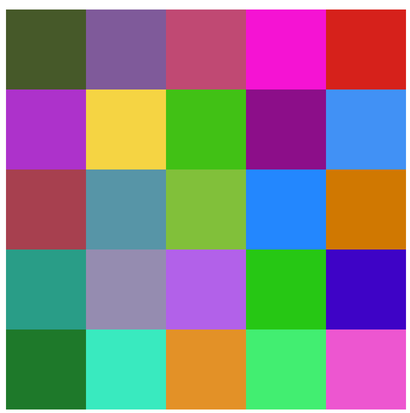
# Зураг

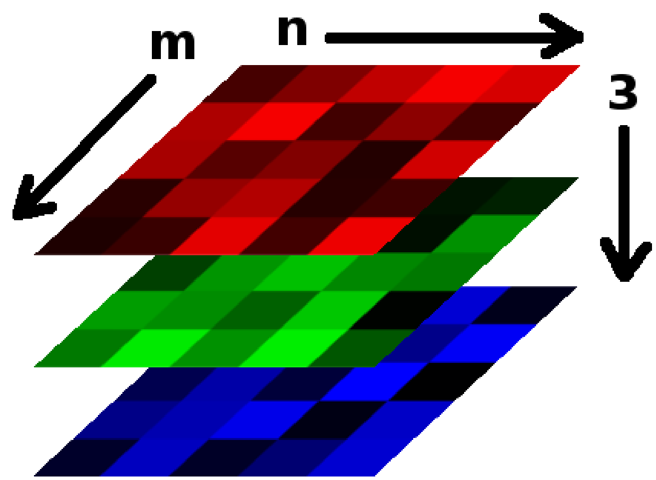
$$= \begin{bmatrix} 193 & 180 & 210 & 112 & 125 \\ 189 & 8 & 177 & 97 & 114 \\ 100 & 71 & 81 & 195 & 165 \\ 167 & 12 & 242 & 203 & 181 \\ 44 & 25 & 9 & 48 & 192 \end{bmatrix}$$

- MATLAB-д зургыг пикселийн тодролын матрицаар д үрсэлнэ.
- Матрицын координат нь Декартын координат биш гэдгийг тэмдэглэж хэлэх нь зүйтэй. Зүүн дээд өнцөг нь [y,x] = (1,1)

# Өнгөт зураг

- Хар цагаан зураг пиксел бүр нэг тоо байх б өгөөд m × n матрицад хадаглагдана.
- Өнгөт зурагын хувьд пиксел бүрт 3 too бай на – улаан, ногоон, хөх өнгөний тодрол (R GB)
- m × n × 3 матрицад хадаглагдана.





# Матрицын үндсэн үйлдлүүд

- Бид дараах  үйлдүүдийг авч үзнэ :
  - Нэмэх
  - Масштаблах
  - Скаляр үржвэр
  - Үржвэр
  - Хөрвүүлэх (transpose)
  - Урвуу / pseudoinverse
  - Тодорхойлогч / trace

# Матриц үйлдлүүд

- Нэмэх

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} a+1 & b+2 \\ c+3 & d+4 \end{bmatrix}$$

  – Зөвхөн скаляр болон тохирох хэмжиглэхүүнтэй матрицыг нэмнэ.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + 7 = \begin{bmatrix} a+7 & b+7 \\ c+7 & d+7 \end{bmatrix}$$

- Масштаблах (Scaling)

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times 3 = \begin{bmatrix} 3a & 3b \\ 3c & 3d \end{bmatrix}$$

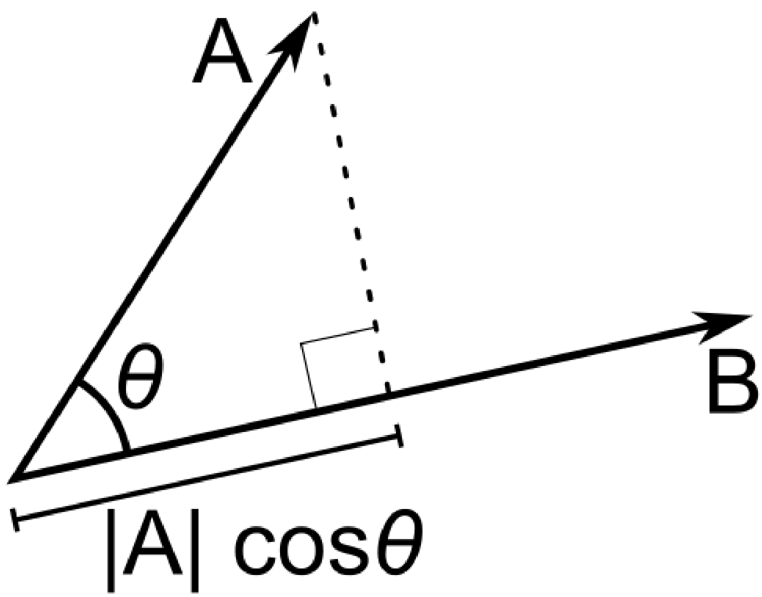
# Матриц үйлдлүүд

- Inner product (dot product) of vectors
  - Multiply corresponding entries of two vectors and add up the result
  - x··y is also |x||y|Cos( *the angle between x and y* )

$$\mathbf{x}^T\mathbf{y} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i \qquad \text{(scalar)}$$

# Матриц үйлдлүүд

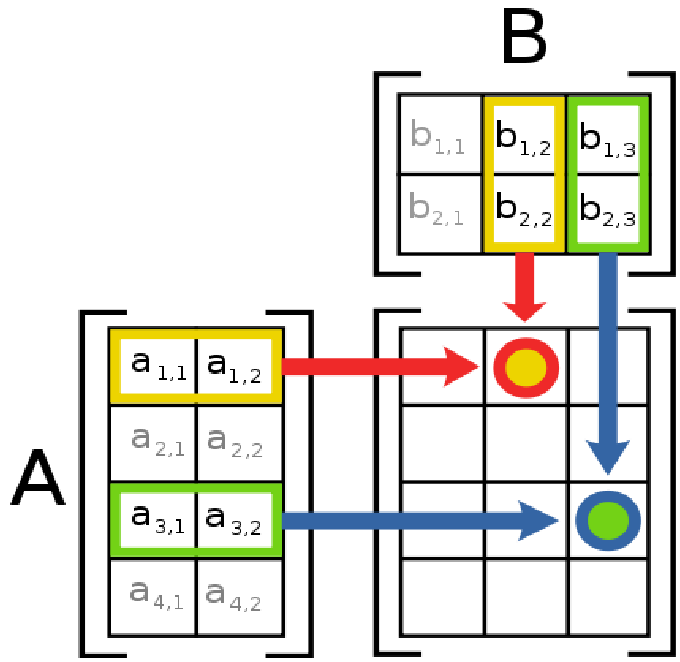
- Векторын скаляр үржвэр (Inner product)
  - Хэрэв B нэгж вектор бол A·B нь B чиглэлд орших A-ын уртыг авна.



# Матриц үйлдлүүд

- Үржүүлэх

- AB –ын үр дүн:




- Утга бүр нь A-ын мөр ба B-ын баганы элемен түүдийн скаляр үржвэрийн үр дүн юм.

# Матриц үйлдлүүд

- Үржвэрийн жишээ:

A x B

$$\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 2 \\ 4 & 6 \end{bmatrix} \quad \begin{bmatrix} 10 & 14 \\ 34 & 54 \end{bmatrix}$$

$$0 \cdot 3 + 2 \cdot 7 = 14$$

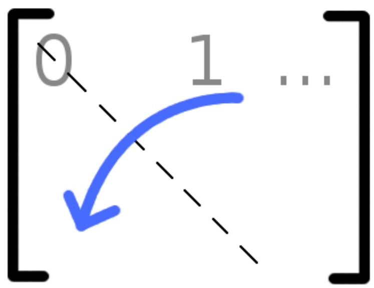
  – Матрицын гарах утга бүр нь зүүн матрицын мөрийн элементүүдийг баруун матрицын баганын элементүүдтэй харгалзуулсан скаляр үржвэрийн үр дүн байна.

# Матриц үйлдлүүд

- Зэрэг дэвшүүлэх
  - Тогтсон журмын дагуу, AA марицыг $A^2$ , AAA матрицыг $A^3$ гэх мэтээр авч үзнэ.
  - Мэдээж зөвхөн квадрат матрицыг ингэж үржүүлж болно.

# Матриц үйлдлүүд

- Хөрвүүлэлт – матриц эргүүлэх, мөр нь багана болно.



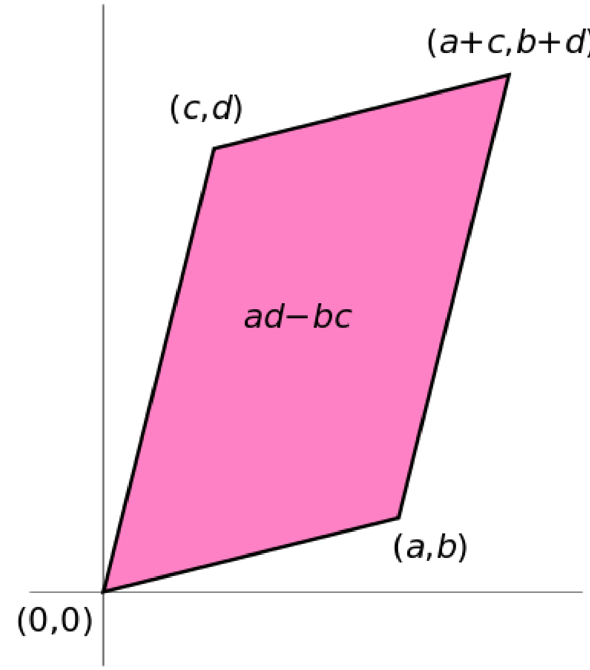

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}^T = \begin{bmatrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$

- Тэнцэтгэл нь:

$$(ABC)^T = C^T B^T A^T$$

# Матриц үйлдлүүд

- Тодорхойлогч
  - det(A) нь скаляр утгыг буцаана.
  - Параллелограммын тайлбай н ь матрицын мөрийн вектороор тодорхойлогдоно.




  - $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$-ын хувд, $\det(\mathbf{A}) = ad - bc$
  - Шинж чанар:

$$\det(\mathbf{AB}) = \det(\mathbf{BA})$$

$$\det(\mathbf{A}^{-1}) = \frac{1}{\det(\mathbf{A})}$$

$$\det(\mathbf{A}^T) = \det(\mathbf{A})$$

$$\det(\mathbf{A}) = 0 \Leftrightarrow \mathbf{A} \text{ is singular}$$

# Матриц үйлдэлүүд

- Trace

$$\mathrm{tr}(\mathbf{A}) = \text{sum of diagonal elements}$$

$$\mathbf{tr}\left(\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}\right) = 1 + 7 = 8$$

  – Өөрчлөлтийн талаар маш олон вариантууд бий тул зарим тохиолд туршилтын маягаар хэрэглэ дэг. (Хичээлийн туршид цөөн яригдана.)

  – Шинж чанар:

$$\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$$

$$\mathrm{tr}(\mathbf{A} + \mathbf{B}) = \mathrm{tr}(\mathbf{A}) + \mathrm{tr}(\mathbf{B})$$

# Тусгай матриц

- Нэгж матриц **I**
  - Квадрат матриц, диагональ утга 1 бусад нь 0 байх
  - **I** · [бусад матриц] = [тухайн матриц]

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Диагональ матриц
  - Диагоналийн утгуудтай квадрат матриц, бусад нь 0 байх
  - Диагональ·[бусад матриц] тухайн матрицын мөрийг дүрсэлнэ.

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

# Тусгай матриц

- Тэгш хэмт матриц

$$\mathbf{A}^T = \mathbf{A}$$

$$\begin{bmatrix} 1 & 2 & 5 \\ 2 & 1 & 7 \\ 5 & 7 & 1 \end{bmatrix}$$

- Налуу - тэгш хэмт матриц

$$\mathbf{A}^T = -\mathbf{A}$$

$$\begin{bmatrix} 0 & -2 & -5 \\ 2 & 0 & -7 \\ 5 & 7 & 0 \end{bmatrix}$$

# Агуулга

- Vectors and matrices
  - Basic Matrix Operations
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- Matrix inverse
- Matrix rank
- Singular Value Decomposition (SVD)
  - Use for image compression
  - Use for Principal Component Analysis (PCA)
  - Computer algorithm

Матриц үржүүлэг нь векторуудыг хувиргахад хэрэглэгдэнэ. Энэ тохиолдолд А матрицыг хувиргалт-ын матриц гэж нэрлэнэ.

# Хувиргалт (Transformation)

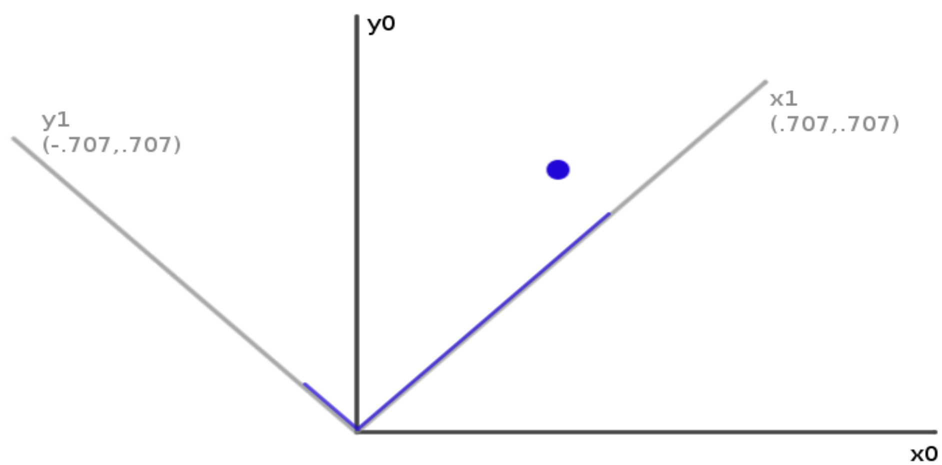
- Матрицууд нь үржих үйлдлээр векторууды г хувиргахад хэрэглэгдэнэ: x’= Ax
- Хамгийн энгийн нь масштабаар тооцох:

$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix}$$

# Эргүүлэлт

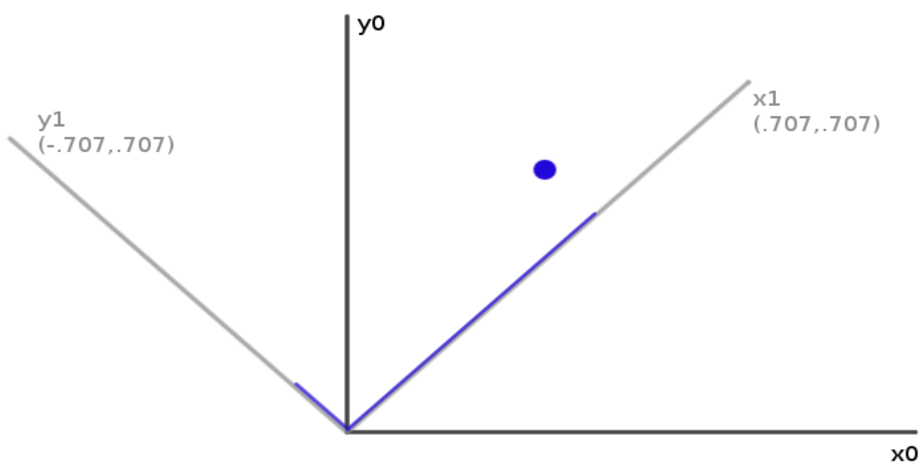
- frame “0” –д дүрслэгдсэн векторыг эргүүлэлт хийгдсэн координат frame “1”руу хэрхэн хөрвүүл эх вэ?
- Вектор гэж юу вэ гэдгийг эргэн санацгаая:

[frame-ийн x тэнхлэгийн чиглэл дэх компонент, frame-ийн y тэнхлэгийн чиглэл дэх компонент]



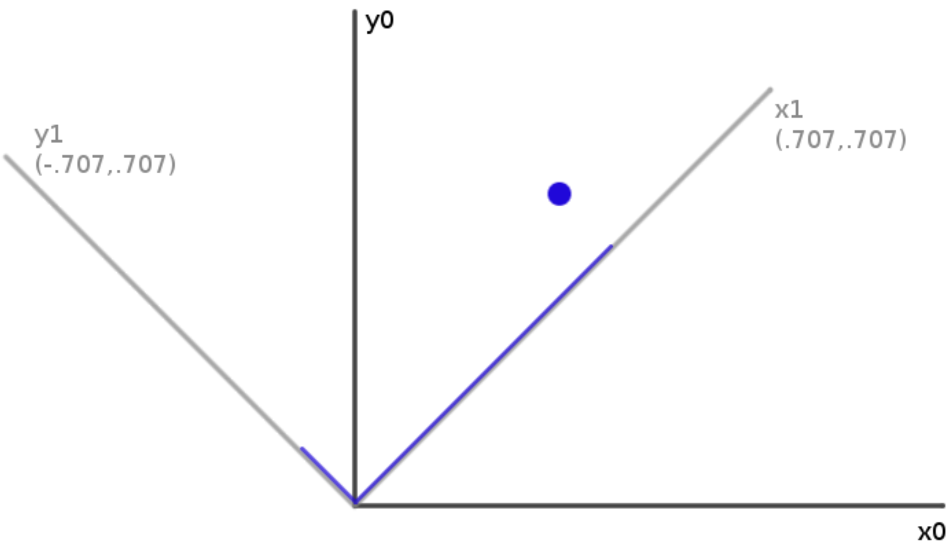
# Эргүүлэлт

- Үүнийг эргүүлэхийн тулд бид векторыг байгуулах шаардлагатай:
  [**шинэ** x тэнхлэгийн чиглэл дэх компонент, **шинэ** y тэнхлэгийн чиглэл дэх компонент]
- Скаляр үржвэрээр бид үүнийг хялбархан шийдэж болно!
- New x coordinate is [original vector] **dot** [the new x axis]
- New y coordinate is [original vector] **dot** [the new y axis]



# Эргүүлэлт

- Ойлголт:  матриц*вектор 2-ын үржвэр
  - X координатын үр дүн нь: [original vector] **dot** [matrix row 1]
  - Матрицын үржвэр нь вектор p эргүүлнэ:

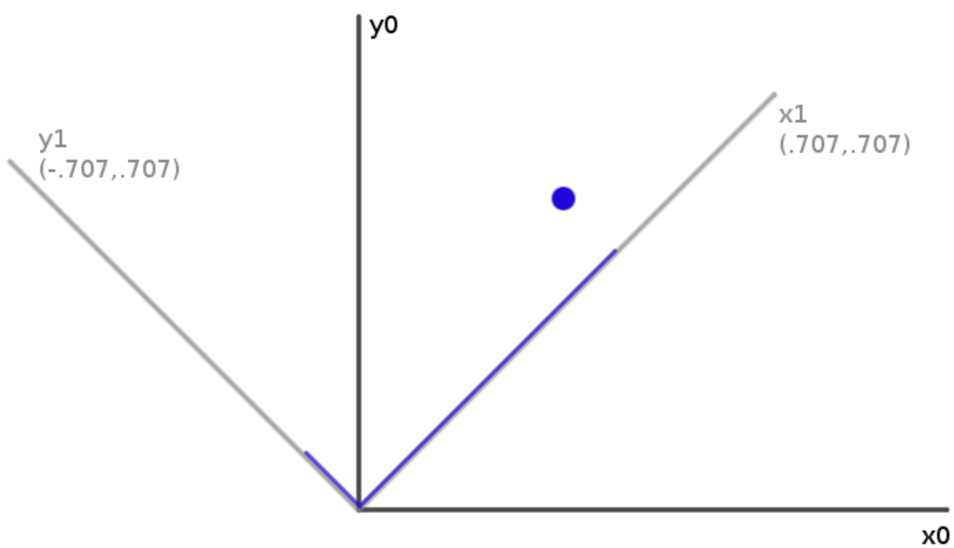



R x p = rotated p'

$$\underbrace{\begin{bmatrix} .707 & .707 \\ -.707 & .707 \end{bmatrix}}_{R} \overbrace{\begin{bmatrix} px \\ py \end{bmatrix}}^{p} \qquad \begin{bmatrix} px' \\ py' \end{bmatrix}$$

# Эргүүлэлт

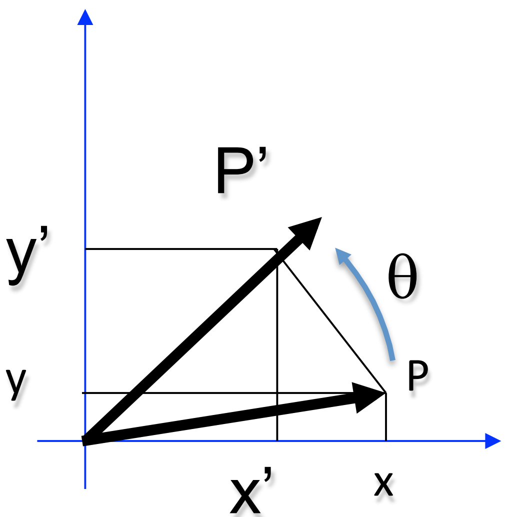
- Шинэ координатын систем дэх цэгийг зү үн тийш эргүүлсэн цэг гэж үзье
- Бид үр дүнг анхдагч коодинатын систем дээр байрлуулвал цэгийг баруун тийш эргүүлнэ.




– Ийм маягаар, векторыг э ргүүлэхэд эргүүлэлтийн м атрицыг ашигладаг. Вект ор эргүүлэх үйлдлийг мэд рэмж гэж ойлгож болно.

# 2D эргүүлэлтийн матрицын томъёо

Counter-clockwise rotation by an angle θ
Counter-clockwise rotation by an angle θ




$$x' = \cos\theta\ x - \sin\theta\ y$$

$$y' = \cos\theta\ y + \sin\theta\ x$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{R}\,\mathbf{P}$$

# Хувиргалтын матриц

- Цэгийг өөрчлөхөд хэд хэдэн хувиргалтын м атрицуудыг хэрэглэдэг:
  $p' = R_2 R_1 S p$
- Өөрчлөлт нь зүүнээс баруун, нэг өөрчлөлт нь нөгөөгийнхөө дараа хэрэгжинэ.
- Дээрх жишээн дэх үр дүн нь:
  $(R_2 (R_1 (S p)))$
- Эхлээд матрицуудаа үржүүлээд дан ганц хув иргалтын матриц хэлбэрт болгох нь үр дүнги йн хувьд ижил
  $p' = (R_2 R_1 S) p$

# Homogeneous систем

- Ерөнхийдөө, матриц үржвэр нь векторын  ком понентуудыг  шугаман байдлаар  нэгтгэнэ.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

  – масштаб, эргүүлэлт, хазайлгах хувиргалтууд
  – Гэвч тогтмолууд нэмэх боломжгүй! ☹

# Homogeneous систем

– (somewhat hacky) шийдэл? Вектор бүрийн төгсгөлд “1” байрлуулах:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

– Дээрх маягаар эргүүлэлт, масштаблах, хазайлгах хувиргалтуудыг хийнэ.

– Үүнийг “homogeneous coordinates” гэж нэрлэдэг.

# Homogeneous систем

– homogeneous координатад уржвэр нь матриц ын баруун талын баганыг вектор гэж тооцоол дог.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

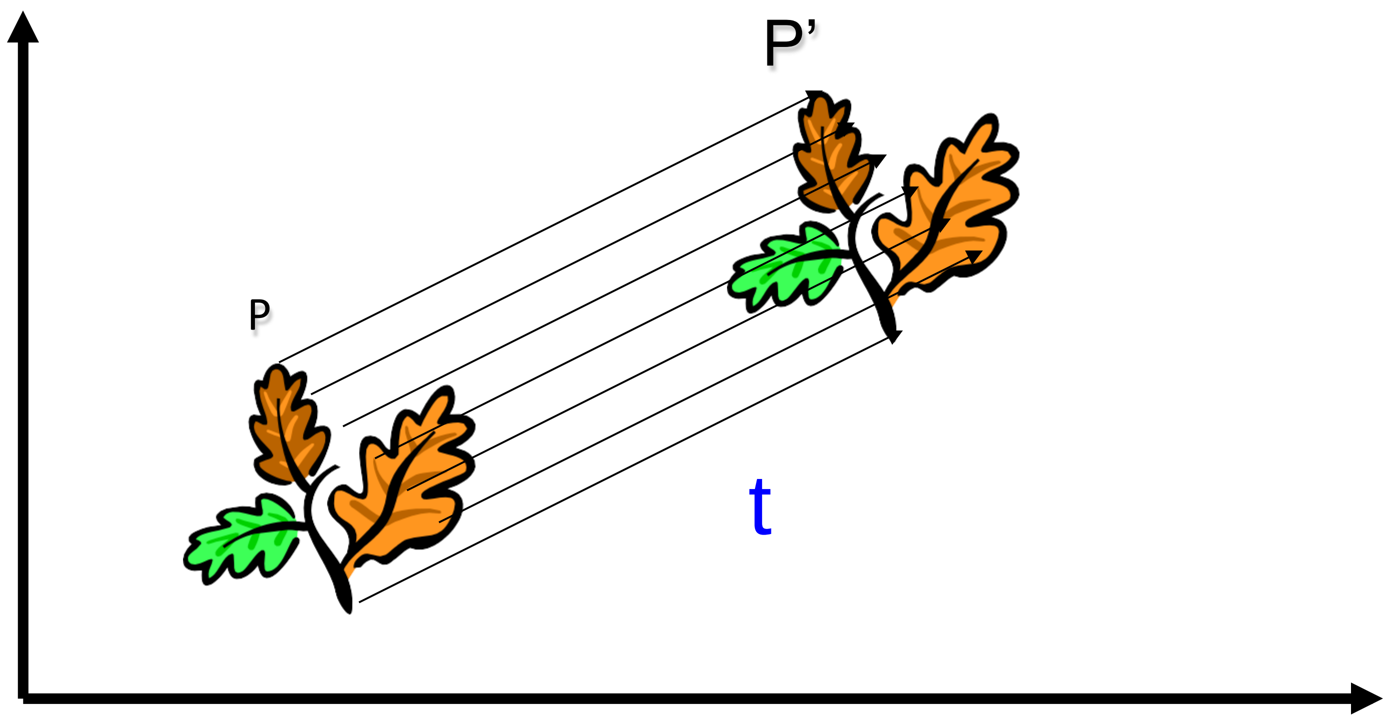
– homogeneous хувиргалтын матриц нь доод тал ын мөрөнд [0 0 1] байх ба  үр дүн нь мөн адил “1” байна.

# Homogeneous систем

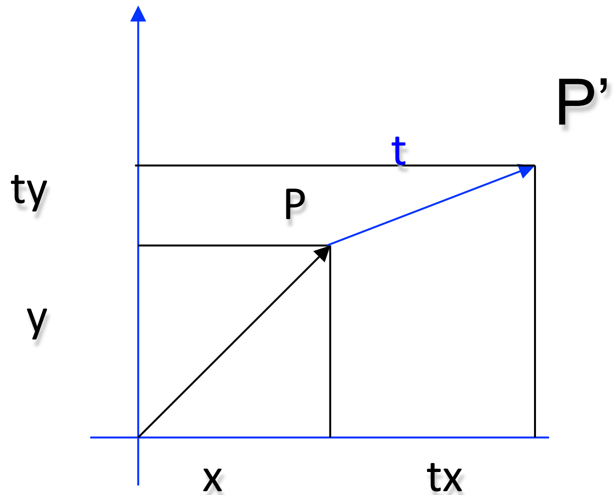
- Өөр нэг тохиолдол: үр дүнг ямар нэг зүйлд хуваах
  - Жишээлбэл, камерийг зургаас холтгоход координатаар нь хувааж болно.
  - Матриц үржвэр нь хуваалт хийж чадахгүй
  - Томъёоллын хувьд матрицыг үржүүлсэний дараагаар үр дүнг сүүлийн координатад хуваана.

$$\begin{bmatrix} x \\ y \\ 7 \end{bmatrix} \Rightarrow \begin{bmatrix} x/7 \\ y/7 \\ 1 \end{bmatrix}$$

# 2D Хувиргалт



# Homogeneous координатыг ашигласан 2D хувиргалт



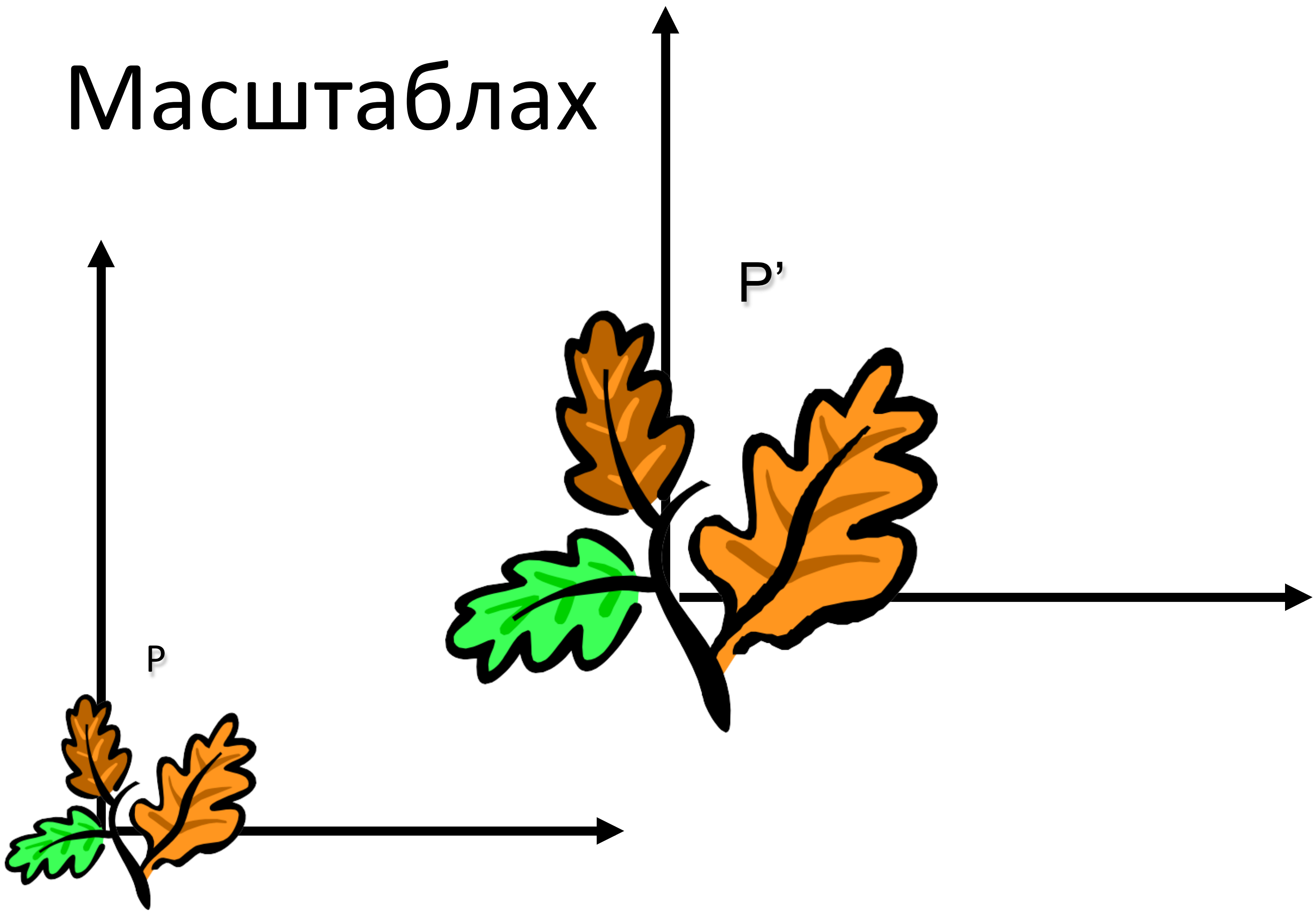

$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{t} = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

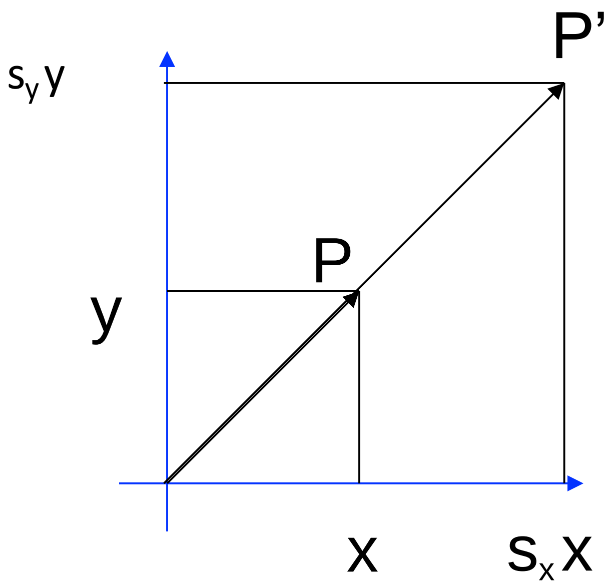
$$\mathbf{P'} \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \cdot \mathbf{P} = \mathbf{T} \cdot \mathbf{P}$$

# Масштаблах



# Масштаблах томъёо



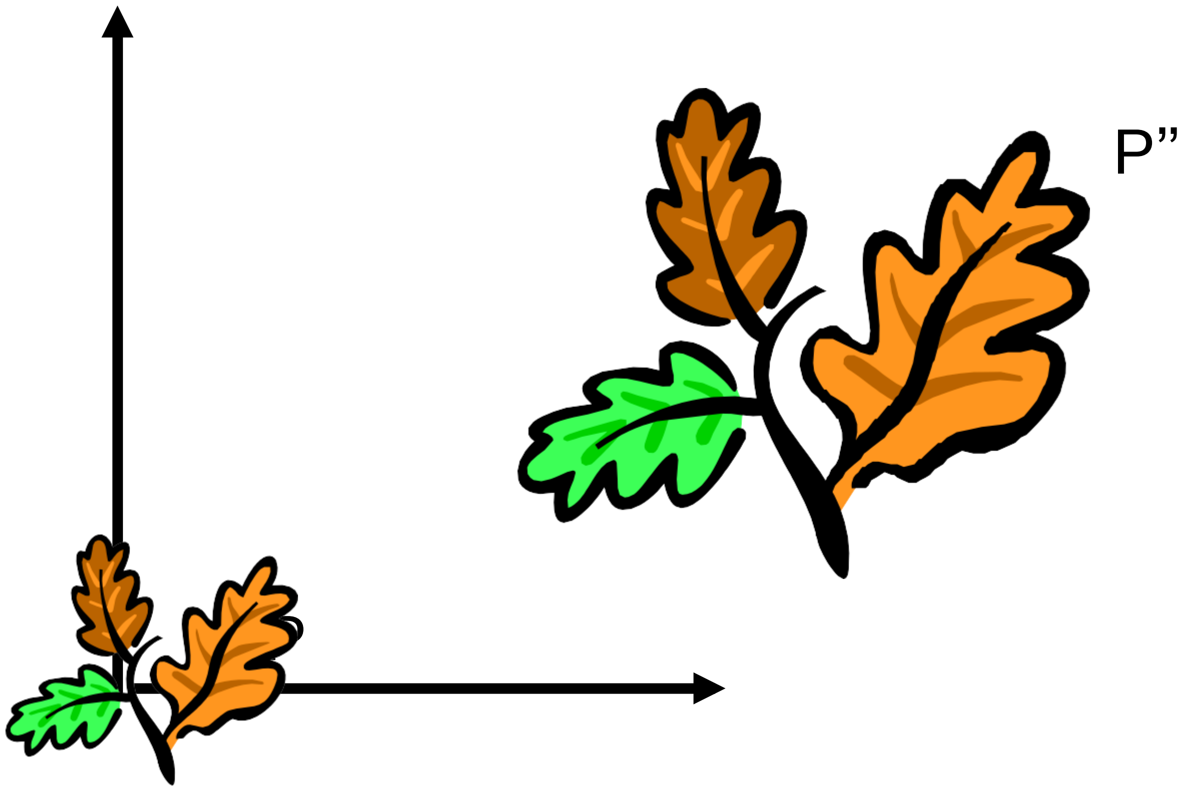

$$\mathbf{P} = (x, y) \rightarrow \mathbf{P'} = (s_x x, s_y y)$$

$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{P'} = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

$$\mathbf{P'} \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{S'} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \mathbf{P} = \mathbf{S} \cdot \mathbf{P}$$

# Масштаблах болон хувиргах




$$P' = S \cdot P \quad P'' = T \cdot P'$$

$$P'' = T \cdot P' = T \cdot (S \cdot P) = T \cdot S \cdot P = A \cdot P$$

# Масштаблах болон хувиргах

$$\mathbf{P''} = \mathbf{T}\cdot\mathbf{S}\cdot\mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

$$= \underbrace{\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathsf{A}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} S & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

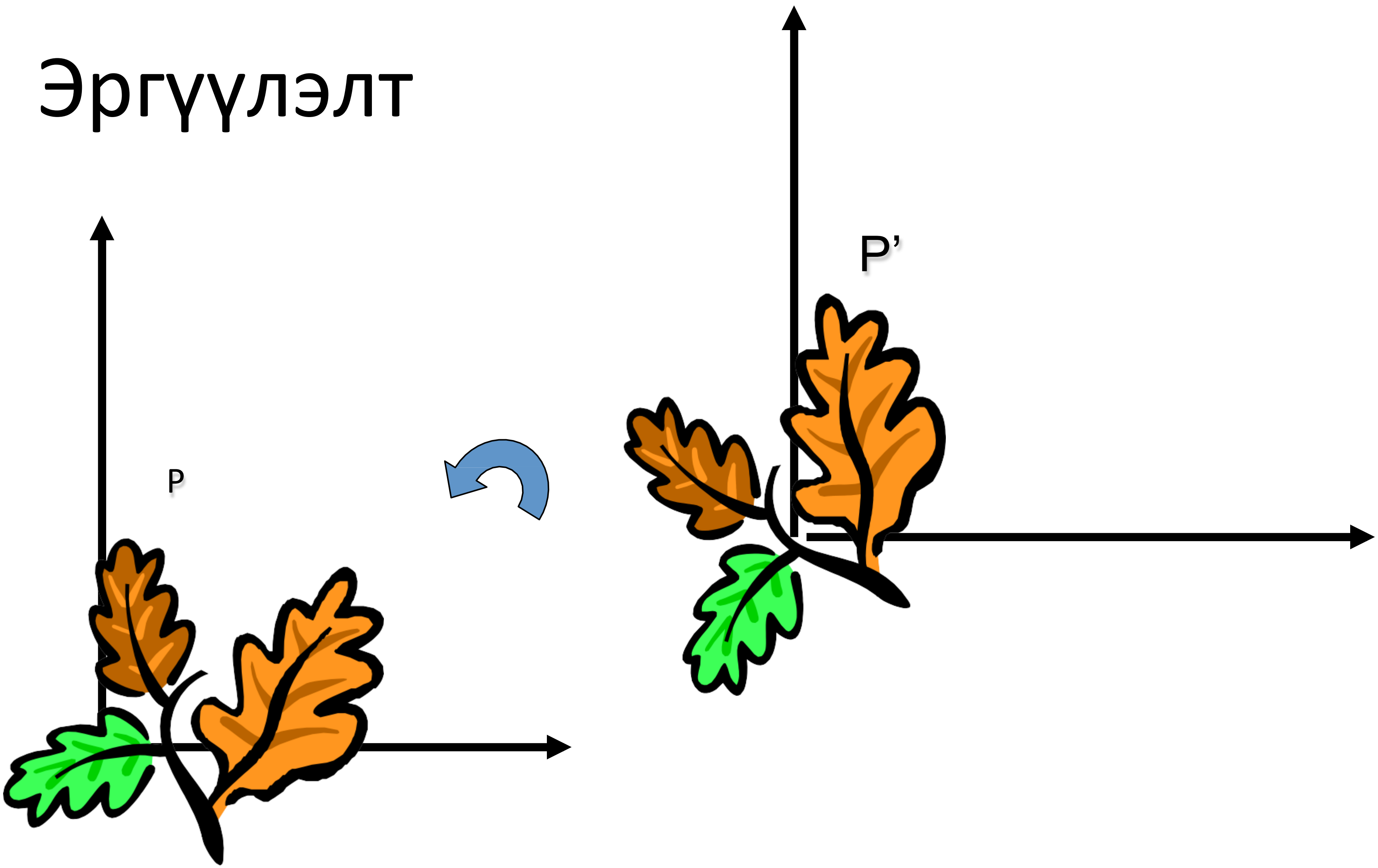
# Хөрвүүлэх болон Масштаблах != Масштаблах болон хөрвүүлэх

$$\mathbf{P'''} = \mathbf{T}\cdot\mathbf{S}\cdot\mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

$$\mathbf{P'''} = \mathbf{S}\cdot\mathbf{T}\cdot\mathbf{P} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

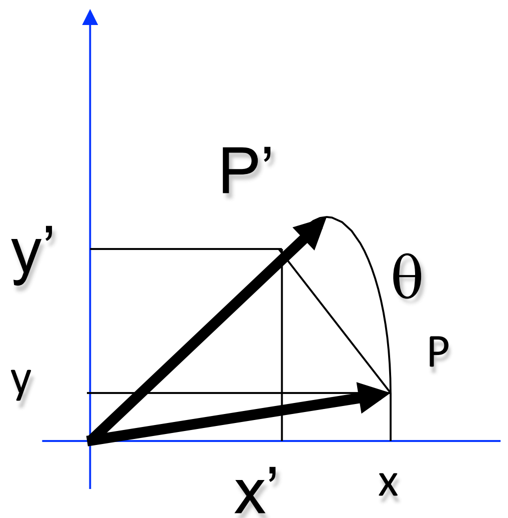
$$= \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & s_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$$

# Эргүүлэлт



# Эргүүлэлтийн томъёо

Counter-clockwise rotation by an angle $\theta$




$$x' = \cos\theta\, x - \sin\theta\, y$$

$$y' = \cos\theta\, y + \sin\theta\, x$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P'} = \mathbf{R}\,\mathbf{P}$$

# Эргүүлэлтийн матрицын шинж чанар

- Transpose of a rotation matrix produces a rotation in the opposite direction

$$\mathbf{R} \cdot \mathbf{R}^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}} \cdot \mathbf{R} = \mathbf{I}$$

$$\det(\mathbf{R}) = 1$$

- The rows of a rotation matrix are always mutually perpendicular (a.k.a. orthogonal) unit vectors
  – (and so are its columns)

# Шинж чанар

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2x2 нь 2D эргүүлэлтийн матриц

A 2D rotation matrix is 2x2

Жич: R нь нормаль матрицын категорит хамаар ах ба олон сонирхолтой шинж чанаруудыг агуул даг:

$$\mathbf{R} \cdot \mathbf{R}^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}} \cdot \mathbf{R} = \mathbf{I}$$

$$\det(\mathbf{R}) = 1$$

# Масштаблах + Эргүүлэх + Хөрвүүлэх

P'= (T R S) P

P'= (T R S) P

$$\mathbf{P'} = \mathbf{T}\cdot\mathrm{R}\cdot\mathrm{S}\cdot\mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

This is the form of the general-purpose transformation matrix

$$= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \boxed{\begin{bmatrix} R\,S & t \\ 0 & 1 \end{bmatrix}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# Агуулга

- Vectors and matrices
  - Basic Matrix Operations
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- Matrix inverse
- Matrix rank
- Singular Value Decomposition (SVD)
  - Use for image compression
  - Use for Principal Component Analysis (PCA)
  - Computer algorithm

Хувиргалт ын матрицын урвуу үйлдэл

# Урвуу үйлдэл

- **A** матриц өгөгдсөн ба $\mathbf{A}^{-1}$ нь урвуу матриц ба дараах хэлбэртэй. $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$
- Жишээ нь: $$\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$$
- Урвуу үйлдэл байнга тохиолдоггүй. Хэрэв $\mathbf{A}^{-1}$ үүссэн бол **A** нь урвуу эсвэл ганц тоо би ш байна. Өөрөөр хэлбэл ганц тоо.
- Урвуу матрицын тэнцэтгэл нь:

$$(\mathbf{A}^{-1})^{-1} = \mathbf{A}$$
$$(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$
$$\mathbf{A}^{-T} \triangleq (\mathbf{A}^{T})^{-1} = (\mathbf{A}^{-1})^{T}$$

# Матриц үйлдэл

- Псевдо урвуу
  - Хэрэв A ба B нь мэдэгдэж байгаа AX = B матриц тэ гшитгэлээс X-г олох бол
  - MATLAB ашиглан үржвэр болон урвууг тооцоолж олвол: $A^{-1}AX=A^{-1}B \rightarrow X=A^{-1}B$
  - MATLAB комманд нь **inv(A)*B** байна.
  - Гэвч том матрицын хувьд урвууг тооцож олоход компьютерийн хөвөгч таслалыг шийдэх асуудал тулгардаг. (Учир нь их тоо ба бага тоо аль алин ыг нь агуулдаг).
  - Эсвэл уг матрицын урвуу огт гарахгүй байж болно.

# Матриц үйлдэл

- Псевдо урвуу
  - Эдгээр нөхцөлд AX=B шийдэх арга замууд бий. MAT LAB үүнийг шийдэж чадна!
  - MATLAB –д урвууг тооцохын оронд шууд **A\B** гэж бичээд AX=B тэгшитгэлийн X-ийг олж болно.
  - MATLAB-д тохиромжит олон аргууд байдаг.
  - MATLAB томъёогоор тооцож олсон X-ийн утгыг буцаана.
    - Хэрэв зөв шийдэл байхгүй бол ойцролцоо утгыг буцаана.
    - Хэрэв олон шийдэлтэй бол хамгийн бага утгыг буцаана.

# Матриц үйлдэл

- MATLAB жишээ:

$$AX = B$$

$$A = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

```
>> x = A\B
x =
      1.0000
     -0.5000
```

# Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- Matrix inverse
- **Matrix rank**
- Singular Value Decomposition (SVD)
  - Use for image compression
  - Use for Principal Component Analysis (PCA)
  - Computer algorithm

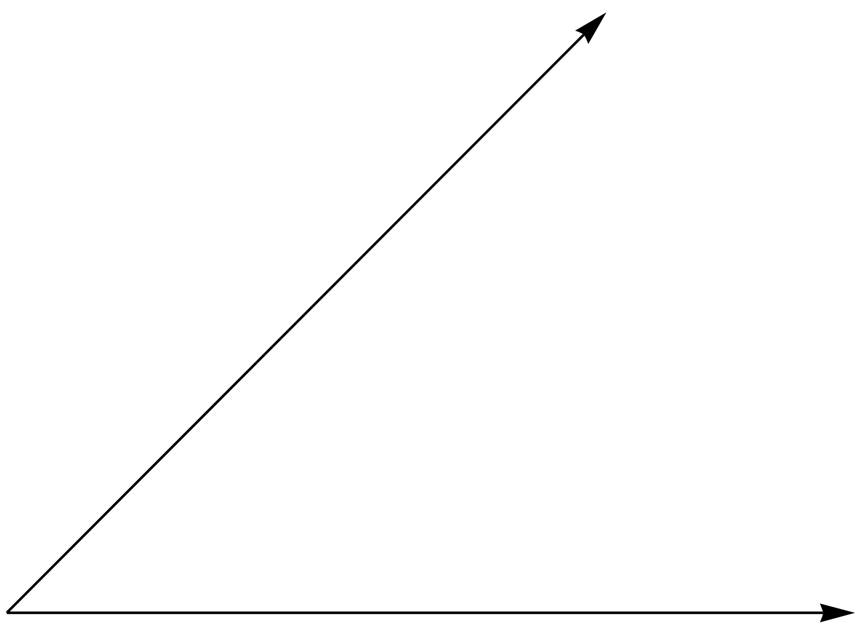
Хувиргалтын матрицын  зэрэглэл нь векторыг хувиргах  хэр олон хэмжээс байгааг хэлж өгдөг.

# Шугаман хамаарал

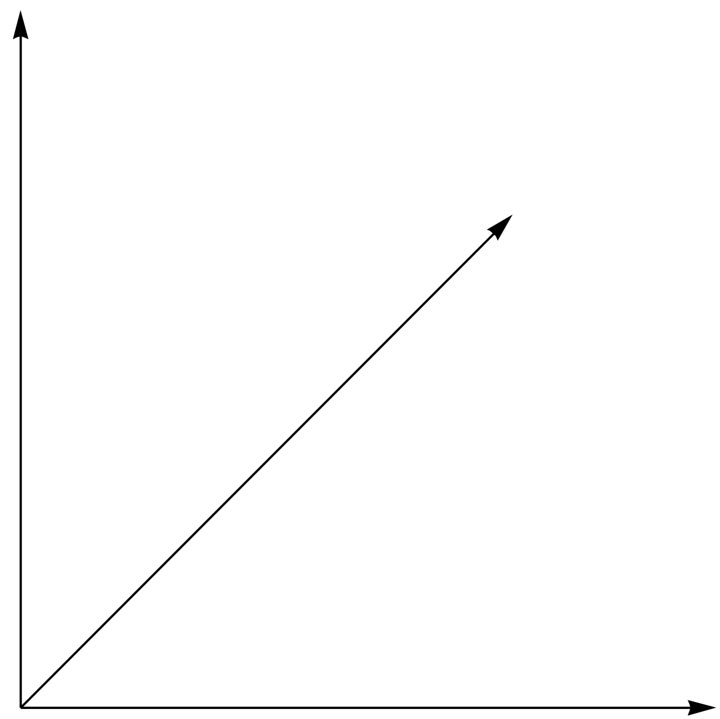
- $\mathbf{v_1}, \ldots, \mathbf{v_n}$ векторууд байна гэж үзье
- Хэрэв $\mathbf{v}_1$ -ыг бусад векторууд $\mathbf{v}_2 \ldots \mathbf{v}_n$ -ийн шугаман хослолоор илэрхийлвэл $\mathbf{v}_1$ нь бусад векторууд дээрх linearly *dependent болно*.
  - $\mathbf{v}_1$ –ийн чиглэл нь $\mathbf{v}_2 \ldots \mathbf{v}_n$ векторуудын чиглэлийн хос лолооор илэрхийлэгдэнэ. (Жнь: $\mathbf{v}_1 = .7\, \mathbf{v}_2 - .7\, \mathbf{v}_4$)
- Хэрэв векторууд дотор шугаман хамааралта й вектор байхгүй бол векторууд нь шугаман хамааралтай байна.
  - Ерөнхий нөхцөл: Хэрэв вектор тус бүр бусад вектор уудтайгаа перпендикуляр байвал $\mathbf{v_1}, \ldots, \mathbf{v_n}$ вектору уд нь үргэлж шугаман хамааралтай байна.

# Шугаман хамаарал

## Шугаман хамаарал



## Шугаман бус хамаарал



# Матрицын эгнээ

- Баганан/мөрөн эгнээ

$$\begin{aligned}\text{col-rank}(\mathbf{A}) &= \text{ the maximum number of linearly independent column vectors of } \mathbf{A}\\ \text{row-rank}(\mathbf{A}) &= \text{ the maximum number of linearly independent row vectors of } \mathbf{A}\end{aligned}$$

  – Баганан эгнээ нь үргэлж мөрөн эгнээтэйгээ тэнцүү байдаг.

- Матрицын эгнээ

$$\text{rank}(\mathbf{A}) \triangleq \text{col-rank}(\mathbf{A}) = \text{row-rank}(\mathbf{A})$$

# Матрицын эгнээ

- Хувиргалтын матрицын хувьд эгнээ нь гаралтын хэмжүүрийг илэрхийлнэ.
- Жн: Хэрэв **A** –ийн эгнээ нь 1 бол хувиргалт нь

**p'=A**p

maps points onto a line.

- 1 эгнээтэй матриц:

$$\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x+y \\ 2x+2y \end{bmatrix}$$

All points get mapped to t he line y=2x

# Матрицын эгнээ

- Хэрэв *m* x *m матрицын эгнээ m бол  дүүрэн эг нээтэй гэж үзнэ.*
  - *m* x 1 векторыг өөр нэг  вектортай  холбоно.
  - Урвуу матрицыг олно.
- Хэрэв эгнээ< *m бол*  “ганц”
  - Хамгийн багадаа нэг хэмжээсээр буурна. Оролт нь юу болохыг  болох үр дүнг харах боломжгүй.
  - Урвуу үүсэхгүй.
- Квадрат бус матрицын хувьд урвуу үүсдэггүй.

# Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- Matrix inverse
- Matrix rank
- Singular Value Decomposition (SVD)
  - Use for image compression
  - Use for Principal Component Analysis (PCA)
  - Computer algorithm

SVD нь 3 матрицын үржвэрээр ма трицыг илэрхийлдэг алгоритм ба матрицын бүтцийг илрүүлэхэд ашиглана.

# Singular Value Decomposition (SVD)

- Матрицыг “үржүүлэгч болгон хуваах”, матрицуудыг үржвэрээр илэрхийлэгдэх хэд хэдэн компьютерийн алгоритм байдаг.
- Хамгийн өргөн хэрэглэгддэг алгоритм нь Singular Value  Decomposition.
- **A** матрицыг  3 матрицын үржвэрээр илэрхийлдэг: $\mathbf{U\Sigma V^T}$
- MATLAB комманд: [U,S,V]=svd(A)

# Singular Value Decomposition (SVD)

$$\mathbf{U\Sigma V^T = A}$$

- **U** болон **V** нь эргүүлэлтийн матриц, **Σ** нь м аштаблах матриц. Жишээлбэл:

$$\underset{U}{\begin{bmatrix} -.40 & .916 \\ .916 & .40 \end{bmatrix}} \times \underset{\Sigma}{\begin{bmatrix} 5.39 & 0 \\ 0 & 3.154 \end{bmatrix}} \times \underset{V^T}{\begin{bmatrix} -.05 & .999 \\ .999 & .05 \end{bmatrix}} = \underset{A}{\begin{bmatrix} 3 & -2 \\ 1 & 5 \end{bmatrix}}$$

# Singular Value Decomposition (SVD)

- 2D дээш:
  - Хэрэв **A** нь $m$ x $n$ бол **U** нь $m$ x $m$, **Σ** нь $m$ x $n$, and **V**$^{\mathbf{T}}$ *нь $n$ x $n$ болно.*
  - (Жич: хэмжээ нь $m$ x $n$ *үржвэрээр үүснэ.* after multiplication)

$$\overset{U}{\begin{bmatrix} -.39 & -.92 \\ -.92 & .39 \end{bmatrix}} \times \overset{\Sigma}{\begin{bmatrix} 9.51 & 0 & 0 \\ 0 & .77 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} = \overset{A}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

# Singular Value Decomposition (SVD)

- **U** болон **V** нь үргэлж эргэлтийн матриц байна.
  - Геометр эргүүлэлт нь матрицаас хамаараад тохи рохгүй ухагдахуун байж болно. Үүнийг “нэгжийн” матриц гэх ба багана тус бүр нь нэгж векторууд байна.
- **Σ** –диогналь матриц
  - Тэг биш оролтуудын тоо = **A**-ын эгнээ
  - Алгоритм нь өндрөөс нам руу эрэмблэгддэг.

$$\overset{U}{\begin{bmatrix} -.39 & -.92 \\ -.92 & .39 \end{bmatrix}} \times \overset{\Sigma}{\begin{bmatrix} 9.51 & 0 & 0 \\ 0 & .77 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} = \overset{A}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

# SVD Аппликейшн

- Бид геометрийн хувиргалтын матриц дахь SVD-ийг авч үзсэн.
- Гэвч зургийн матрицын SVD нь илүү өргөн хэрэглэгддэг.
- Үүнийг ойлгохын тулд SVD нь юу хийдэг талаарх геометр тайлбарыг харах шаардлагатай.

# SVD Аппликейшн

$$\overset{U}{\begin{bmatrix} -.39 & -.92 \\ -.92 & .39 \end{bmatrix}} \times \overset{\Sigma}{\begin{bmatrix} 9.51 & 0 & 0 \\ 0 & .77 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} = \overset{A}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

- Зүүнээс баруун руу үржвэр хэрхэн тооцож байгааг хар:
- **U** матрицын 1 багана **Σ** матрицын эхний утгаар масштаб лагдана.

$$\overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \qquad \overset{A_{partial}}{\begin{bmatrix} 1.6 & 2.1 & 2.6 \\ 3.8 & 5.0 & 6.2 \end{bmatrix}}$$

- **A** матрицын багануудад векторууд үр дүн харагдана.

# SVD Аппликейшн

$$\overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \qquad \overset{A_{partial}}{\begin{bmatrix} 1.6 & 2.1 & 2.6 \\ 3.8 & 5.0 & 6.2 \end{bmatrix}}$$

$$+ \; \overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \qquad \overset{A_{partial}}{\begin{bmatrix} -.6 & -.1 & .4 \\ .2 & 0 & -.2 \end{bmatrix}}$$

$$= \qquad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

- (***U****-ийн i-р багана* )·(***Σ-****ийн i утга*)·(***V****$^T$ –ийн i-р мөр*) үржвэрийг **A**-ийн элементүүдээр үржүүлнэ.

# SVD Аппликейшн

$$\overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \quad \overset{A_{partial}}{\begin{bmatrix} 1.6 & 2.1 & 2.6 \\ 3.8 & 5.0 & 6.2 \end{bmatrix}}$$

$$\overset{A}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

$$\overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \quad \overset{A_{partial}}{\begin{bmatrix} -.6 & -.1 & .4 \\ .2 & 0 & -.2 \end{bmatrix}}$$

- Бид **A** матрицыг ***U***-*ийн баганануудын шугаман хослолтой адил байгуулна.*
- ***U***-*ийн бүх баганы элементүүдээр анхдагч матрицыг байгуулна.*
- Гэвч бодит өгөгдөл дээр бид ихэвчлэн ***U***-ийн эх ний хэдэн баганыг ашиглах ба заримыг нь хаадаг. ( Дээрх жишээнээс $\boldsymbol{A}_{\boldsymbol{partial}}$, нь анхдагч матриц

# SVD Аппликейшн

$$\overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \quad \overset{A_{partial}}{\begin{bmatrix} 1.6 & 2.1 & 2.6 \\ 3.8 & 5.0 & 6.2 \end{bmatrix}}$$

$$\overset{A}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

$$\overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \quad \overset{A_{partial}}{\begin{bmatrix} -.6 & -.1 & .4 \\ .2 & 0 & -.2 \end{bmatrix}}$$

- ***U**-ийн эхний багануудыг өгөгдлийн гол элементүүд  гэж нэрлэдэг.*
- They show the major patterns that can be added to produce the columns of the original matrix
- **V$^T$** мөр нь матрицын баганыг үүсгэхэд гол элементүүд нь хэрхэн холигдож байгааг харуулдаг.

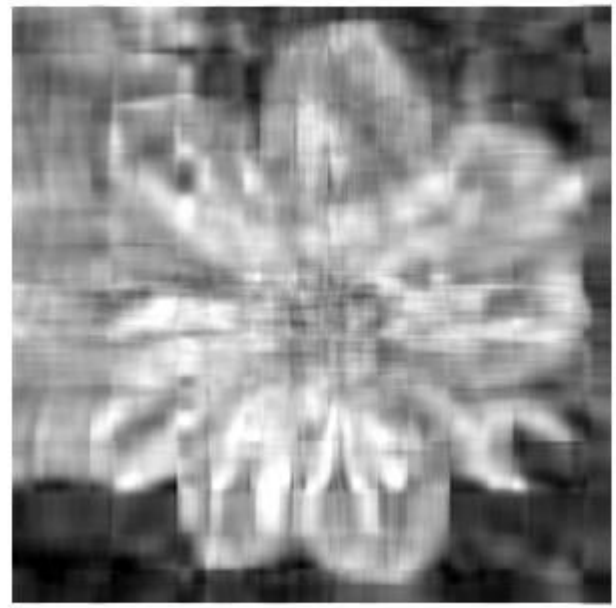
# SVD Аппликейшн

$$\overset{U}{\begin{bmatrix} -.39 & -.92 \\ -.92 & .39 \end{bmatrix}} \times \overset{\Sigma}{\begin{bmatrix} 9.51 & 0 & 0 \\ 0 & .77 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} = \overset{A}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

**Σ** –ийг харвал эхний багана нь хамгийн их утгатай байгааг харж болно.

2 дахь баганы утга нь маш бага утгатай байхад

# SVD Аппликейшн



- Дээрх зургийн хувьд өөрчлөлтийг танихдаа 300 элементийн зөвхөн эхний 10-ыг ашигласан.
- SVD нь зургийг шахах/нягтруулахад хэрэглэнэ.

# Гол элементүүдийн шинжилгээ

$$\underset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \underset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \qquad \underset{A_{partial}}{\begin{bmatrix} 1.6 & 2.1 & 2.6 \\ 3.8 & 5.0 & 6.2 \end{bmatrix}}$$

- ***U*** матрицын багануud нь өгөгдлийн гол элементүүд гэдгийг сана:  анхдагч матрицын баганыг  боловсруулахад үндсэн загварыг нэмнэ.
- Үүний нэг хэрэглээ нь баган бүр нь хуваагдмал өгөгдөл байх матрицыг байгуулахад хэрэглэгдэнэ.
- Тухайн матриц дээр  SVD  ажиллуулан, багана  хоорон дахь ерөнхий загварыг харахын тулд ***U*** матрицын эхний багануудыг  ажиглан хар.
- Үүнийг өгөгдлийн загварын гол элементүүдийн шинжилгээ гэж нэрлэдэг(*Principal Component Analysis* (or PCA)

# Гол элементүүдийн шинжилгээ

$$\overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \qquad \overset{A_{partial}}{\begin{bmatrix} 1.6 & 2.1 & 2.6 \\ 3.8 & 5.0 & 6.2 \end{bmatrix}}$$

- Ихэвчлэн боловсруулагдаагүй өгөгдлүүд олон тодорхойгүй хэв загвартай байдаг.
- PCA бидэнд өгөгдлийн боловсруулагдаагүй ан хдагч хэлбэрийг ашиглахын оронд гол элемент үүдийн загвар дээр нь жин тогтоох хэлбэрээр өгөгдлийг дүрслэх боломж олгодог.
- Загвар тус бүрд жин тогсооноор загвар хооронд ын ялгаа нь болох “утгыг” илэрхийлж чадна.
- Энэ багахан хэмжээний дүрслэл нь машин сурга лт, бусад алгоритмуудад ашиг тустай.

# Агуулга

- Vectors and matrices
  - Basic Matrix Operations
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- Matrix inverse
- Matrix rank
- Singular Value Decomposition (SVD)
  - Use for image compression
  - Use for Principal Component Analysis (PCA)
  - Computer algorithm

Компьютер SVD –ыг маш хурдан тооцоолдог. Энэ талаар сонирхогчдод алгоритмын товч зөвөлгөө өгнө.

# Хавсралт: SVD-ыг хэрхэн тооцоолдог вэ?

- Eigenvector ашиглан SVD-ыг тооцоолох компьютер алгоритм
  – Дараах материал нь SVD-г "шидэт хар хайрцаг"-аа бага болохыг харуулсан. Гэвч та энэ хичээл дээр SVD-г шидэт хар хайрцаг мэт үзвэл өмнөх хичээлээ судлах хэрэгтэй юм.

# Өөрийн вектор (Eigenvector) тодорхойлолт

- **A** квадрат матриц байна гэж үзье. Ax= λx хэлбэрээр x вектор and λ скалярыг олно.
- Өөрөөр хэлбэр вектор хаана байгааг олно. Хэрэв бид тэдгээрийг **A** матрицаар хувиргавал чиглэл нь өөрчлөгдөхгүй зөвхөн масштаб өөрчлөгдөнө.
- Эдгээр векторуудыг eigenvectors гэж нэрлэдэг. (Германаар матрицын "Өөрийн вектор" (self ve ctor), Масштаблах нөхцөл λ-г өөрийн утга (eige nvalues) гэж нэрлэдэг.
- *m* x *m матриц нь* λ нь 0 биш байх ≤ *m* eige nvectors –тай байна.

# Өөрийн векторыг олох

- Компьютер Ax= λx томъёоны x-ыг олохдоо давталтын алгоритм ашиглана :
  - x=random unit vector
  - while(x hasn't converged)
    - x=Ax
    - normalize x
- x нь өөрийн векторт хурдан давхцана.
- Зарим энгийн өөрчлөлтүүд дээр өөрийн векторыг олоход энэ алгоритмыг ашиглана.

# SVD олох

- Квадрат матрицын хувьд өөрийн векторууды г олдог бол SVD-ыг бүх матрицын хувьд олж болно.
- svd(A) олоход  компьютер дараах үйлдлийг хийнэ:
  - $AA^T$-ын өөрийн векторыг авах (квадрат матриц бай на).
    - **U** матрицын баганууд өөрийн векторууд юм.
    - Өөрийн утгууд нь ганц тоон утга байна.( **Σ-**ын оролт).
    - **V** матрицын баганууд нь өөрийн вектор ( эсвэл $\mathbf{V}^T$ мөр)

# SVD олох

- Зарчим: SVD нь хурдан, том матрицад зор иулагдсан.
- Өргөн хэрэглэгддэг
- SVD –г тооцоолох өөр олон төрлийн алгоритм ууд байдаг.
  - MATLAB-ын svd() комманд нь бидний тооцоолол д үр ашигтай, шаардлагатай олон сонголтуудтай.

SVD-ын талаарх дэлгэрэнгүй тайлбарыг эндээс авна уу:
http://www.ams.org/samplings/feature-column/fcarc-svd

# Бид энэ хичээлээр

- Vectors and matrices
  - Basic Matrix Operations
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- Matrix inverse
- Matrix rank
- Singular Value Decomposition (SVD)
  - Use for image compression
  - Use for Principal Component Analysis (PCA)
  - Computer algorithm