ШУТИС, Мэдээлэл Холбооны Технологийн Сургууль

F.CS209 Компьютерийн график

Лекц 3 - Шугаман алгебр ба Компьютерийн график - 2

Dr. Juan Carlos Niebles
Stanford AI Lab

Prof. Fei--FeiLi
Stanford Vision Lab

Компьютерийн
Ухааны салбар
Х.Хулан

Homogeneous систем

- Нэгэн төрлийн систем

$$\begin{cases} ax \\ by \end{cases}$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = 0, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = 0, \\ \dots\dots\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = 0. \end{cases} \qquad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

Homogeneous систем

- Ерөнхийдөө, матриц үржвэр нь векторын компонентуудыг 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! ☹

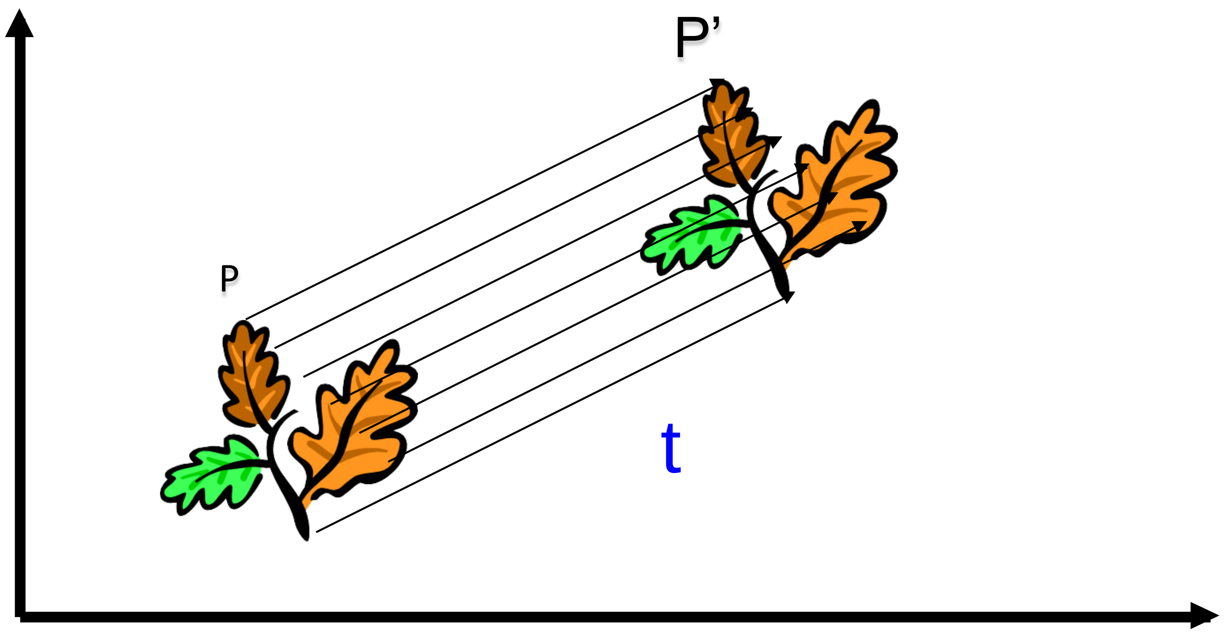
$$\begin{bmatrix} x \\ y \\ 7 \end{bmatrix} \Rightarrow \begin{bmatrix} x/7 \\ y/7 \\ 1 \end{bmatrix}$$

Homogeneous систем

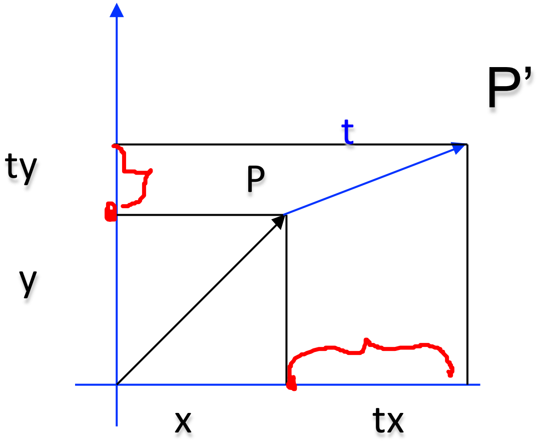
$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

– Дээрх маягаар эргүүлэлт, масштаблах, хазайлгах хувиргалтуудыг хийнэ.
– Үүнийг “homogeneous coordinates” гэж нэрлэдэг.

2D Хувиргалт



Homogeneous координатыг ашигласан 2D хувиргалт



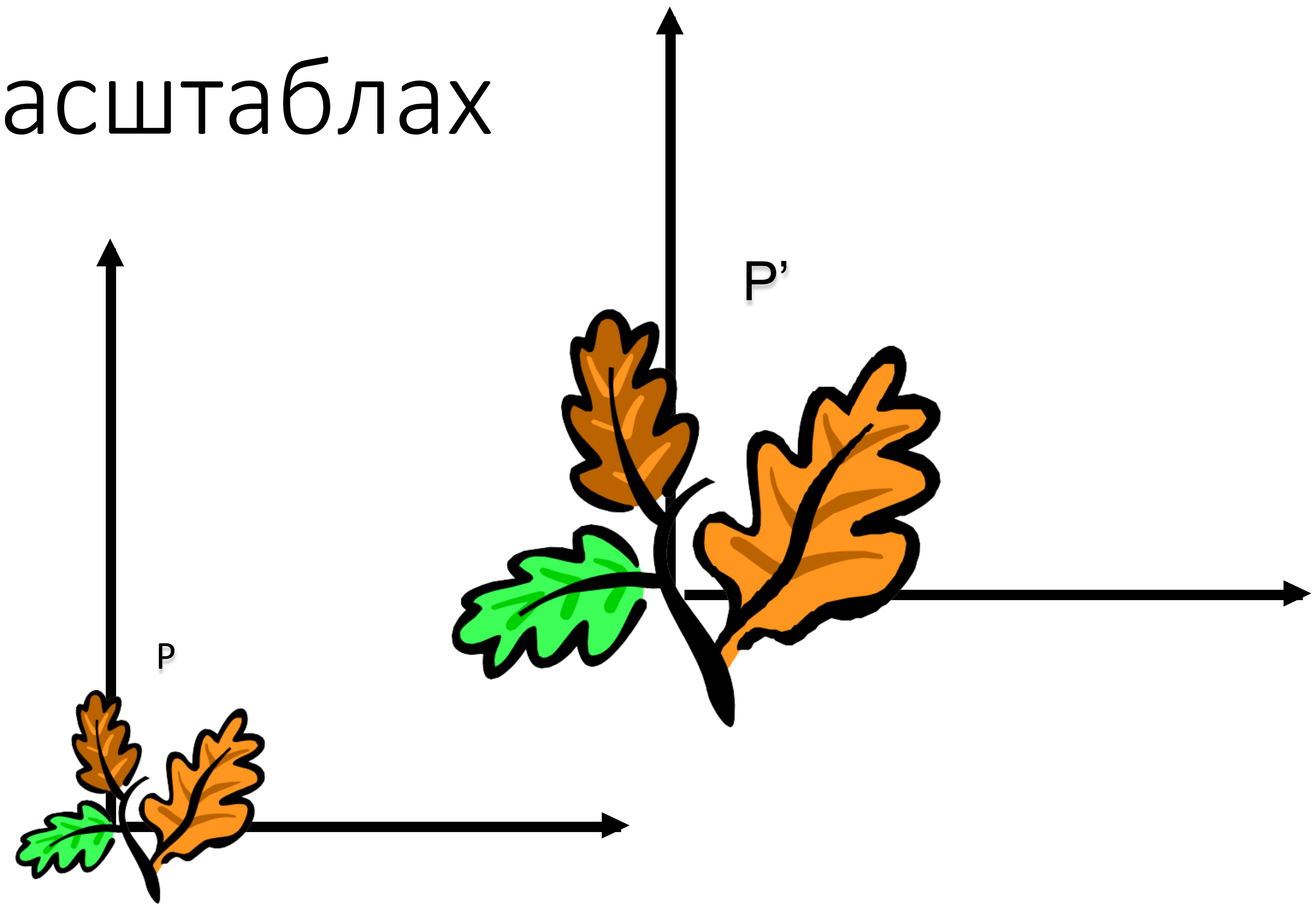
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

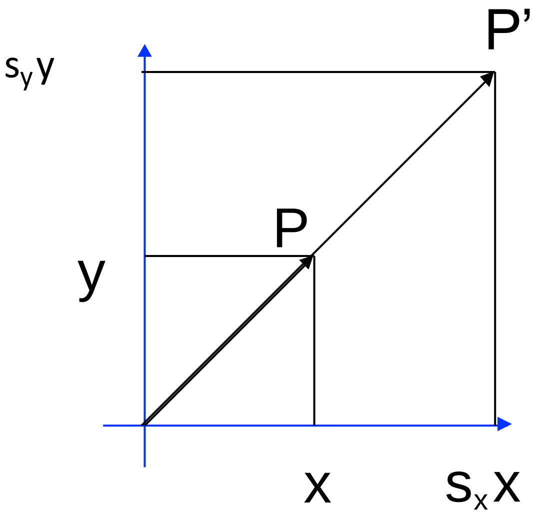
$$P' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \boxed{t_x} \\ 0 & 1 & \boxed{t_y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

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

Масштаблах



Масштаблах томъёо



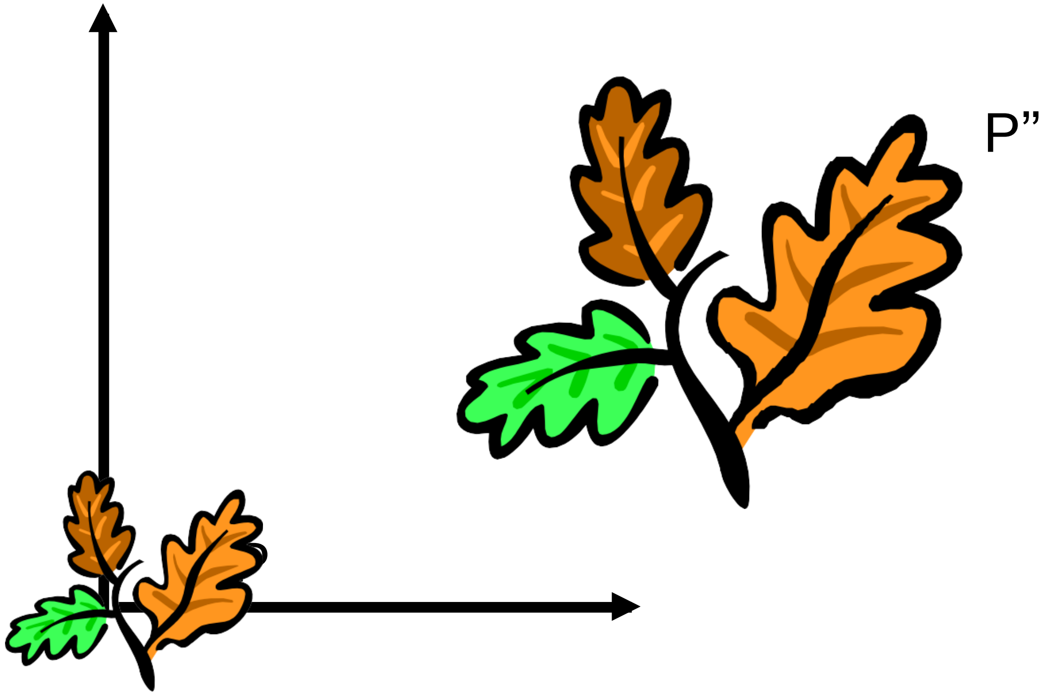
$$\mathbf{P} = (x, y) \to \mathbf{P}' = (s_x x, s_y y)$$

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

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

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

Масштаблах болон хувиргах



P’=S·P P”=T·P’

P’’=T · P’=T ·(S · P)= T ·S ·P = A · P

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

Хувиргалтын матрицын урвуу үйлдэл

Matrix Inverse – Хувиргалтын матрицын урвуу үйлдэл

N эрэмбийн квадрат матриц A-ийн хувьд $A \cdot A^{-1} = A^{-1} \cdot A = E$ *тэнцэтгэл биелэгдэж байвал* A^{-1} *-ийг A-ийн урвуу матриц гэнэ. Үл бөхөх буюу* $\Delta = detA \neq 0$ *байвал A матриц урвуутай ба урвуу матрицыг*

$$A^{-1} = \begin{pmatrix} \frac{A_{11}}{\Delta} & \frac{A_{21}}{\Delta} & \dots & \frac{A_{n1}}{\Delta} \\ \frac{A_{12}}{\Delta} & \frac{A_{22}}{\Delta} & \dots & \frac{A_{n2}}{\Delta} \\ \dots & \dots & \dots & \dots \\ \frac{A_{1n}}{\Delta} & \frac{A_{2n}}{\Delta} & \dots & \frac{A_{nn}}{\Delta} \end{pmatrix}$$

Жишээ нь:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

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

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

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

$A \cdot B \neq B \cdot A$

Матриц үйлдэл

Псевдо урвуу

- Хэрэв A ба B нь мэдэгдэж байгаа AX = B матриц тэгшитгэлээс 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 алиныг нь агуулдаг).
- Эсвэл уг матрицын урвуу огт гарахгүй байж болно.

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

- $\mathbf{v_1, \ldots, v_n}$ векторууд байна гэж үзье

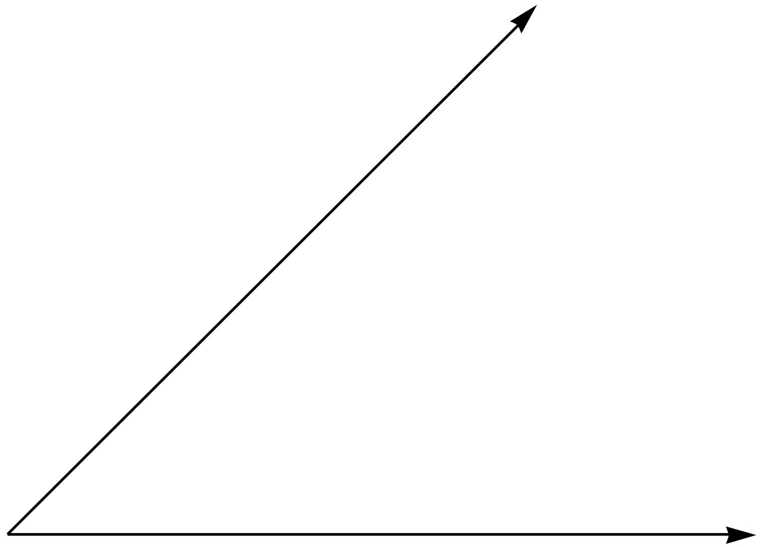
  Хэрэв $\mathbf{v}_1$ -ыг бусад векторууд $\mathbf{v}_2 \ldots \mathbf{v}_\mathrm{n}$ -ийн шугаман хослолоор илэрхийлвэл $\mathbf{v}_1$ нь бусад векторууд дээрх шугаман хамааралтай болно.

  $\mathbf{v}_1$ –ийн чиглэл нь $\mathbf{v}_2 \ldots \mathbf{v}_\mathrm{n}$ векторуудын чиглэлийн хослолооор илэрхийлэгдэнэ. (Жнь: $\mathbf{v}_1 = 7\,\mathbf{v}_2 - 6\,\mathbf{v}_4$)

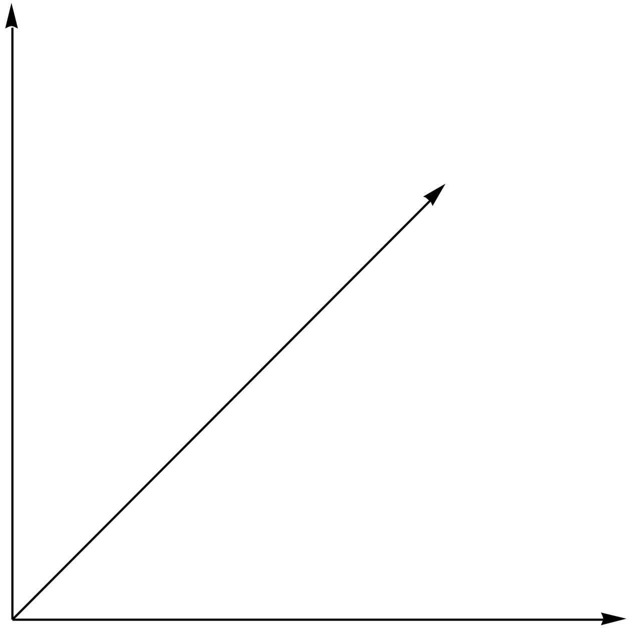
  - Ерөнхий нөхцөл: Хэрэв вектор тус бүр бусад векторуудтайгаа перпендикуляр байвал $\mathbf{v_1, \ldots, v_n}$ векторууд нь үргэлж шугаман хамааралтай байна.

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'=Ap

- 1 эгнээтэй матриц:

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

All points get mapped to the line y=2x

Матрицын эгнээ

- Хэрэв *m* x *m матрицын эгнээ m бол дүүрэн эгнээтэй 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 матрицыг илэрхийлдэг алгоритм ба матрицын бүтцийг илрүүлэхэд ашиглана.

Singular Value Decomposition (SVD)
Сингуляр утгаар задлах

- Матрицыг “үржвэр болгон задлах” буюу матрицуудыг үржвэрээр илэрхийлэх 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 команд: [U,S,V]=svd(A)

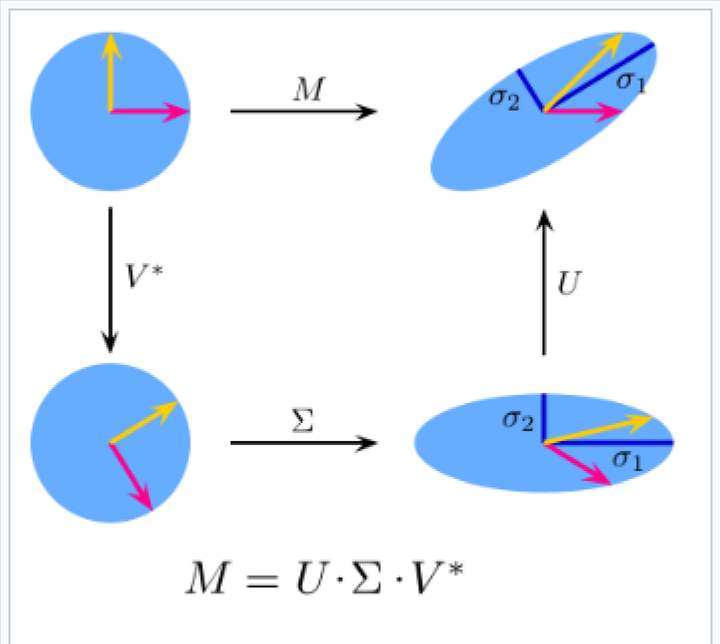
Singular Value Decomposition (SVD)

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

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

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

Singular Value Decomposition (SVD)



цээш:

2 2 2 3

эрэв **A** нь *m* x *n* бол **U** нь *m* x *m*, **Σ** нь *m* x *n*, and

3 3

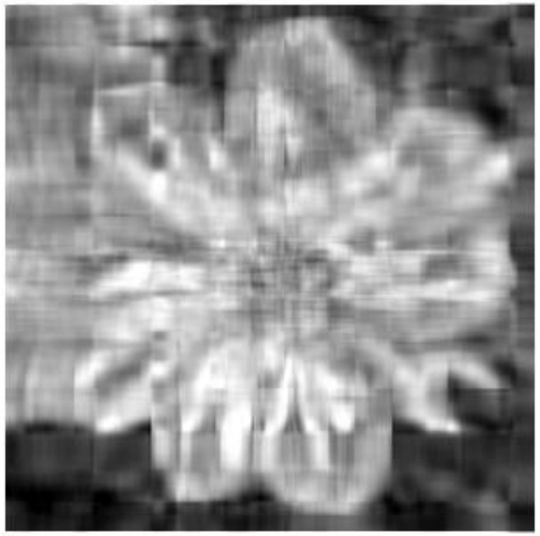
нь *n* x *n болно.*

ич: хэмжээ нь *m* x *n үржвэрээр үүснэ.*)

2x 3

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

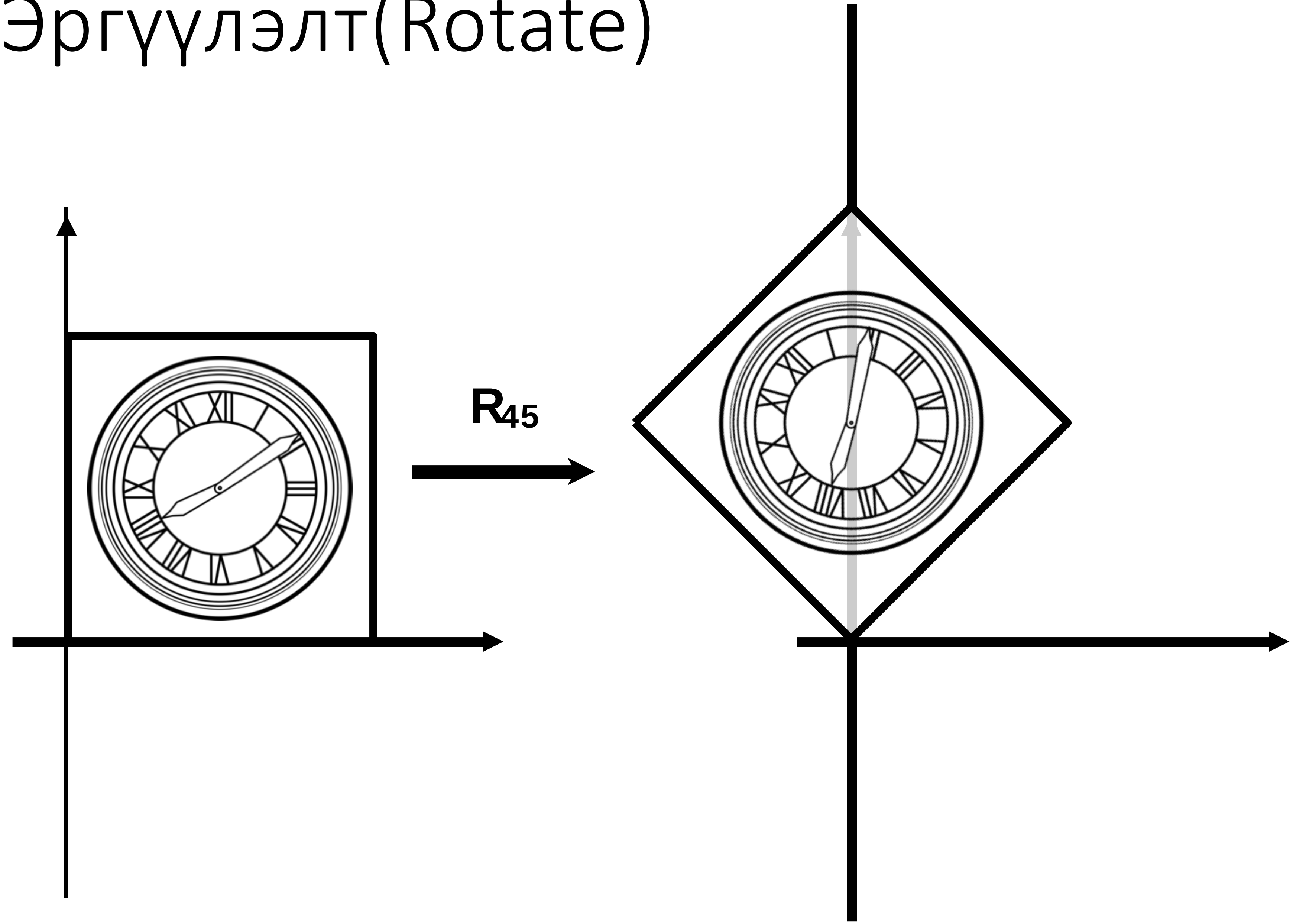
SVD Аппликейшн



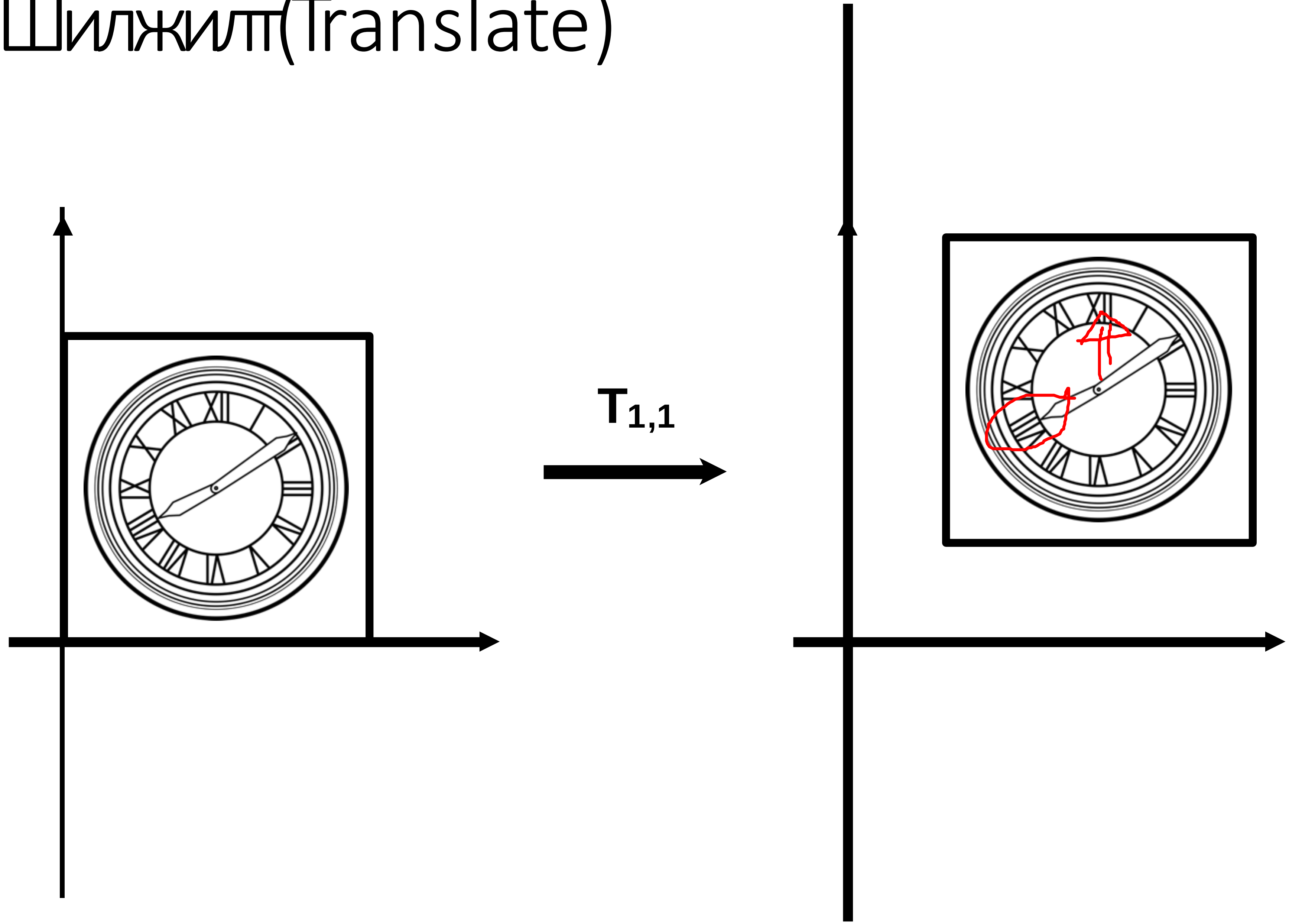
- Дээрх зургийн хувьд өөрчлөлтийг танихдаа 300 элементийн зөвхөн эхний 10-ыг ашигласан.
- SVD нь зургийг шахах/нягтруулахад хэрэглэнэ.

Үндсэн хувиргалтууд

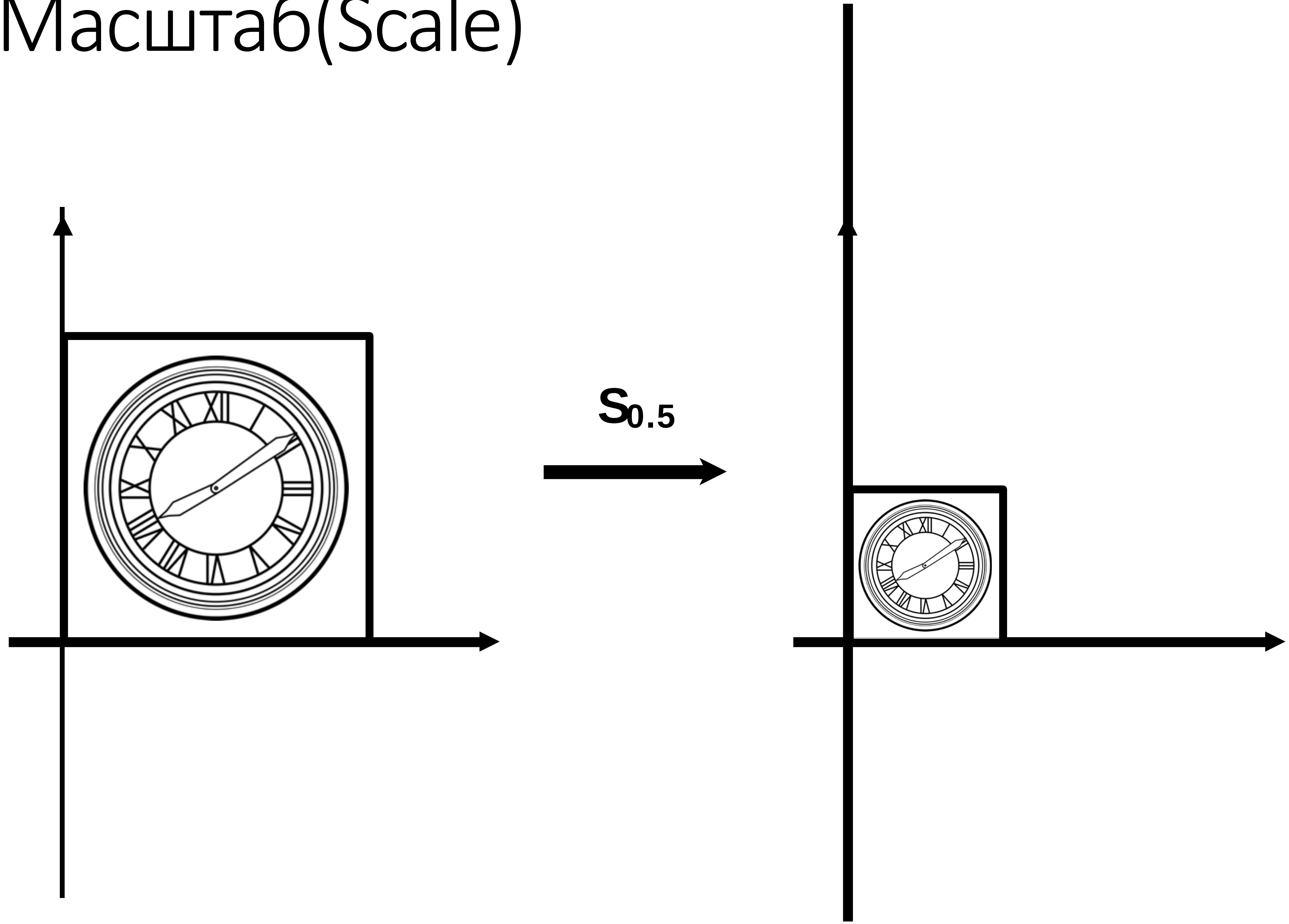
Эргүүлэлт(Rotate)



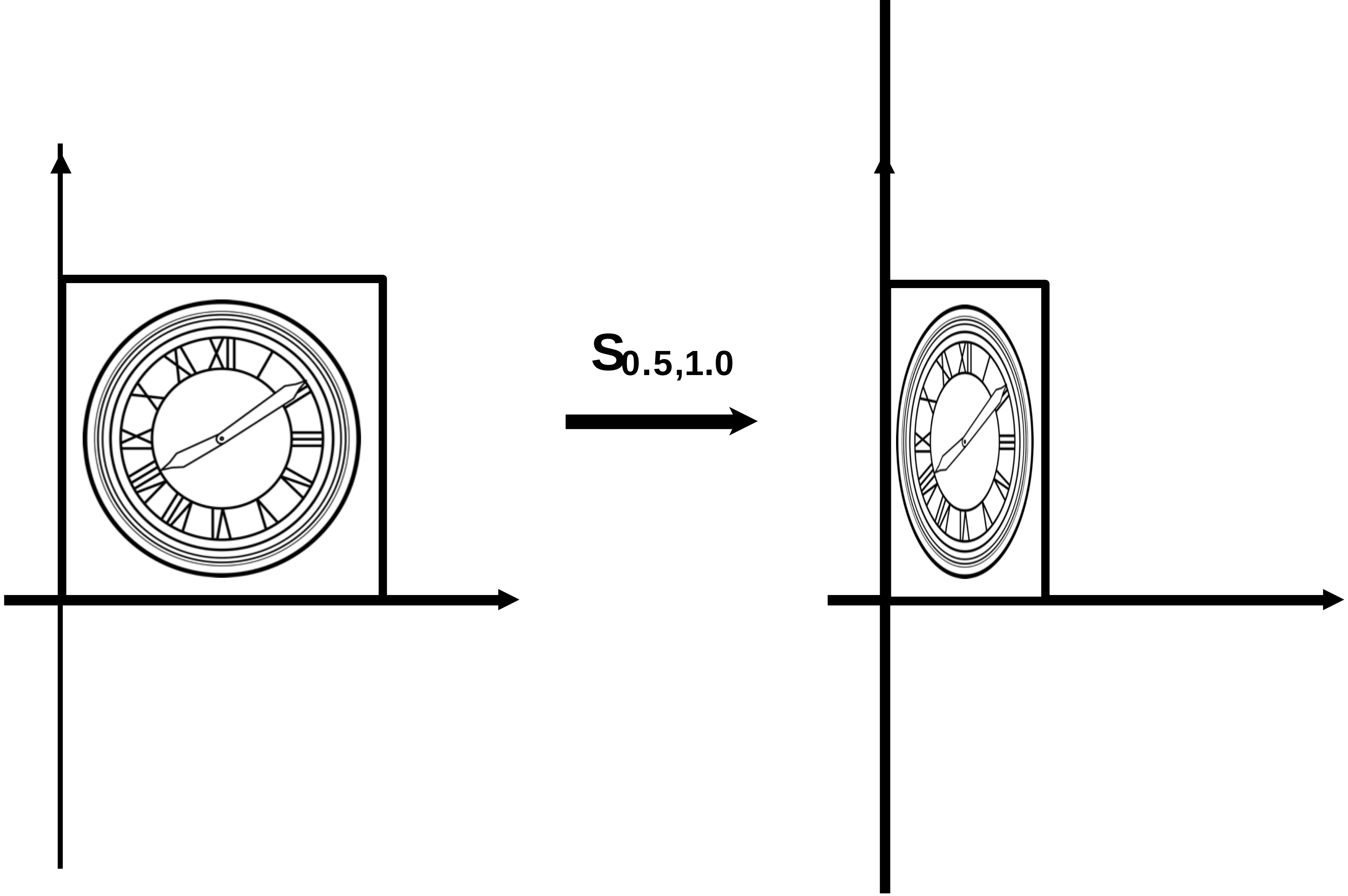
Шилжилт(Translate)



Масштаб(Scale)

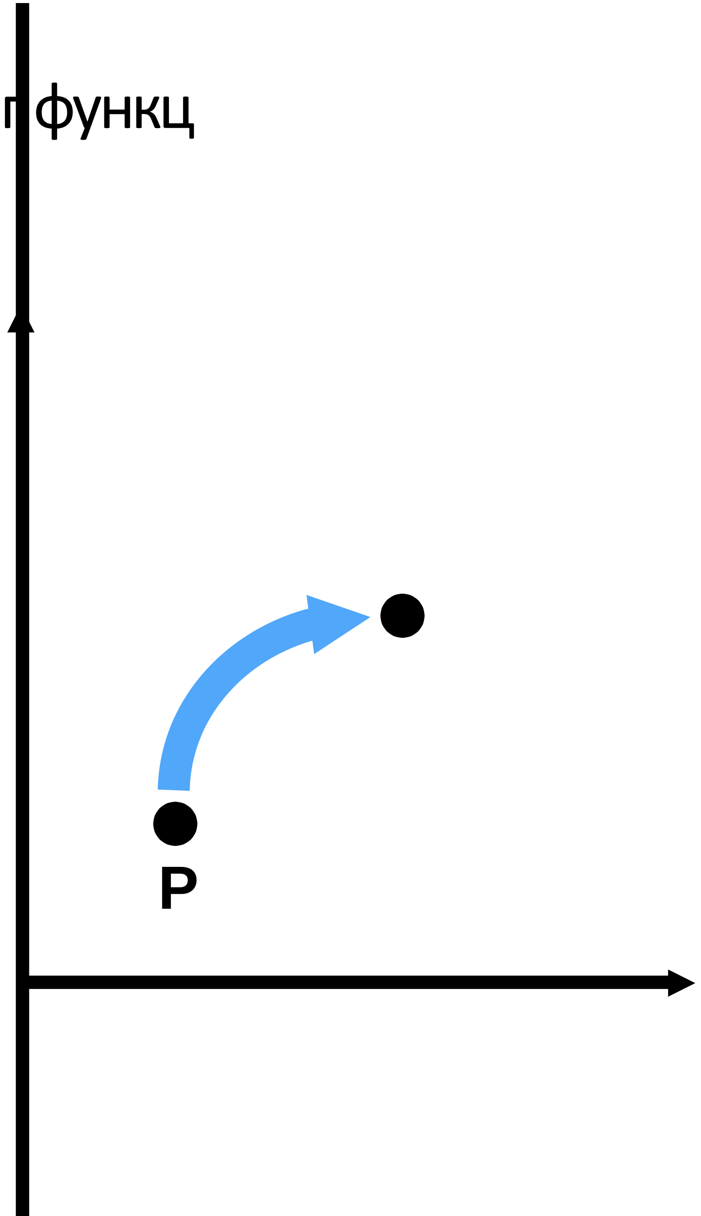


Масштаб (Non-Uniform)



Хувиргалт(Transform) гэж юу вэ?

- Тодорхой цэг дээр үйлдэл хийдэг функц
- $(x', y', z') = F(x, y, z)$
- $P' = F(P)$



Хувиргалтыг судлах нь?

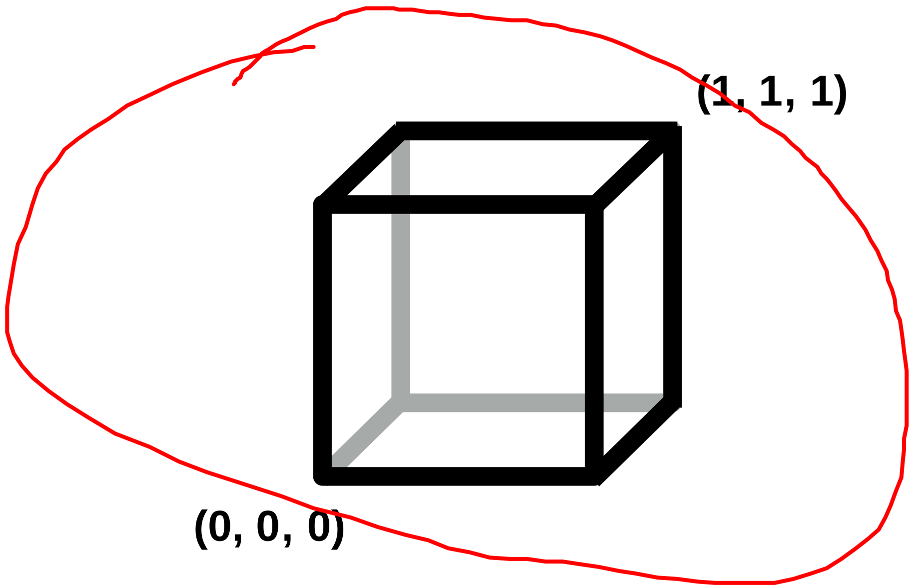
Загварчлах

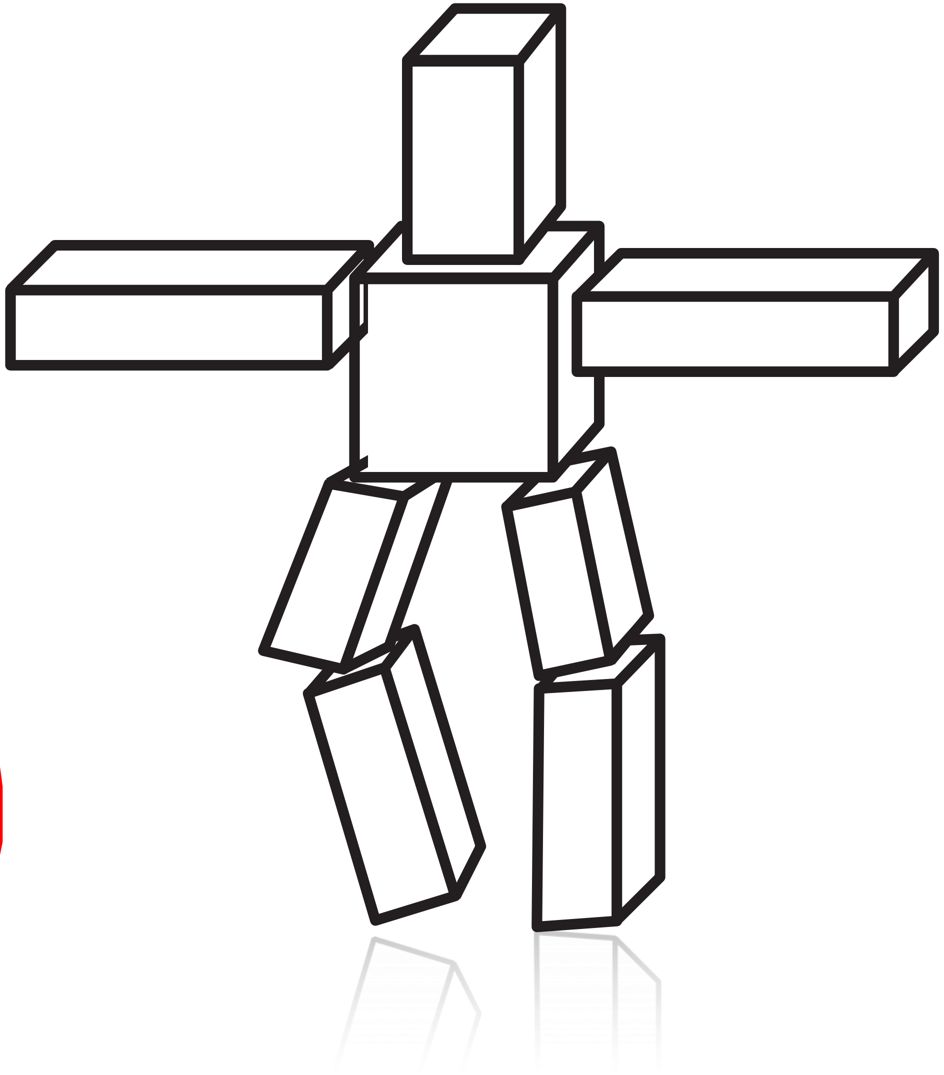
- **Convenient координатад дүрсийг тодорхойлно.**
  - **Объектийн олон хувилбарыг үүсгэх боломжтой.**
- **Шаталсан дэлгэцийг үр дүнтэй дүрсэлнэ**

Харагдац

- **World coordinates to camera coordinates**
- **Parallel / perspective projections from 3D to 2D**

Cube Man (Instancing Hierarchy)





Хичээлийн агуулга

Хувиргалтууд, түүнийг хэрхэн хэрэглэх

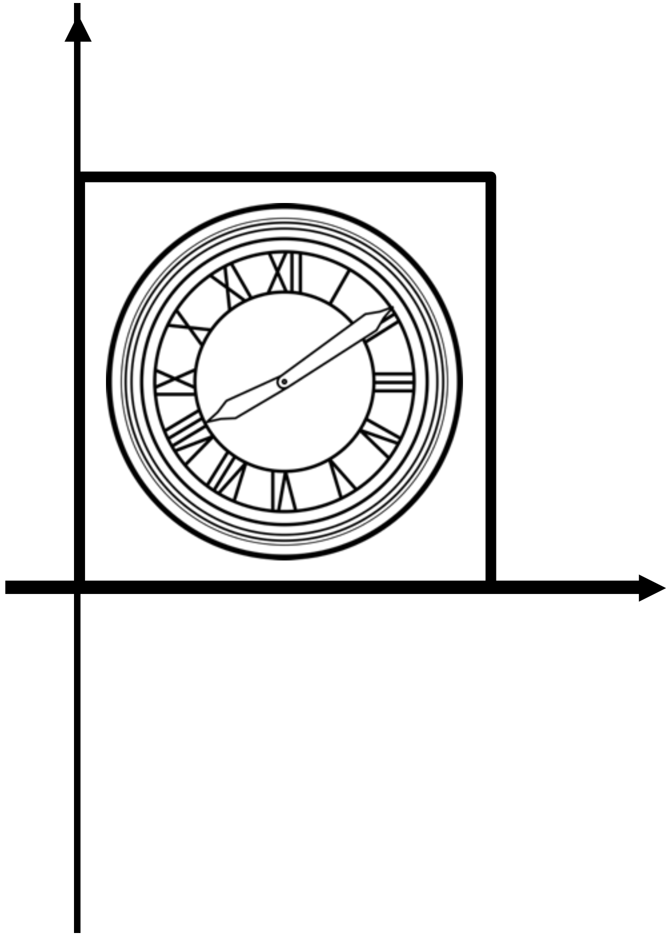
- **Төрөл: rotate, translate, scale, …**
  - **Coordinate frames**
  - **Олон тооны хувиргалтууд байгуулах**
  - **Шаталсан хувиргалт(Hierarchical transforms)**
- **Perspective projection**

Хэрхэн хэрэгжүүлэх?

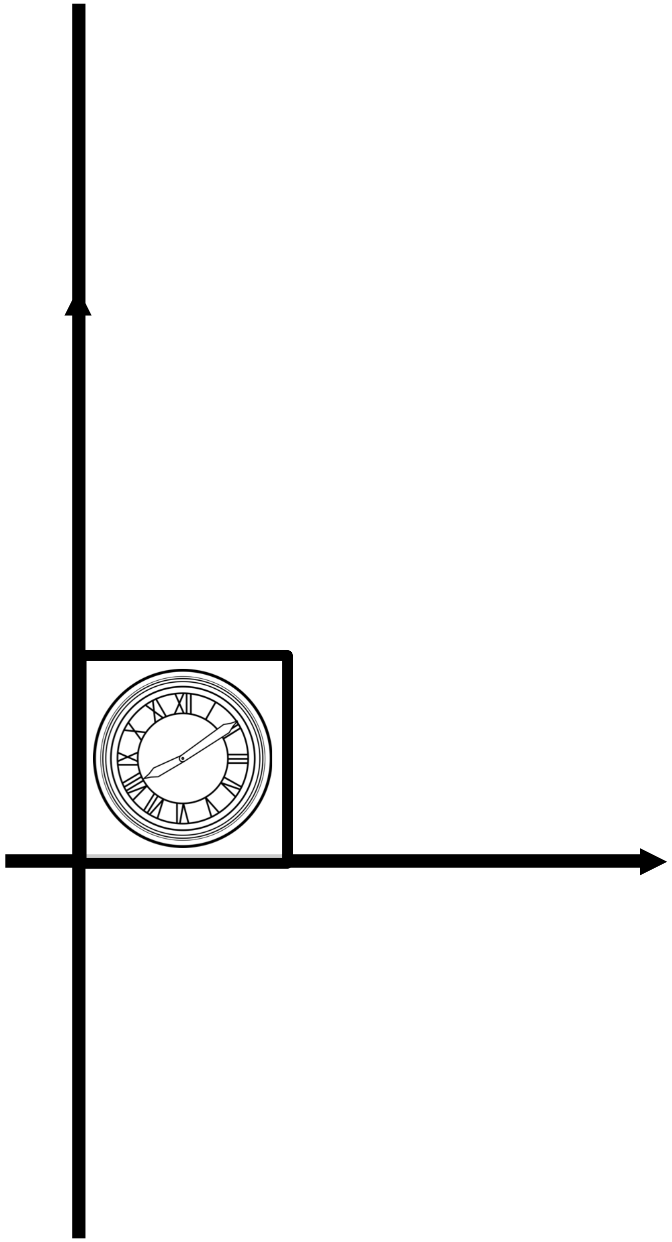
- **Хувиргалтыг матрицаар дүрслэх**
- **Homogeneous координат**

Шугаман хувиргалт = Матриц

Масштаблах хувиргалт



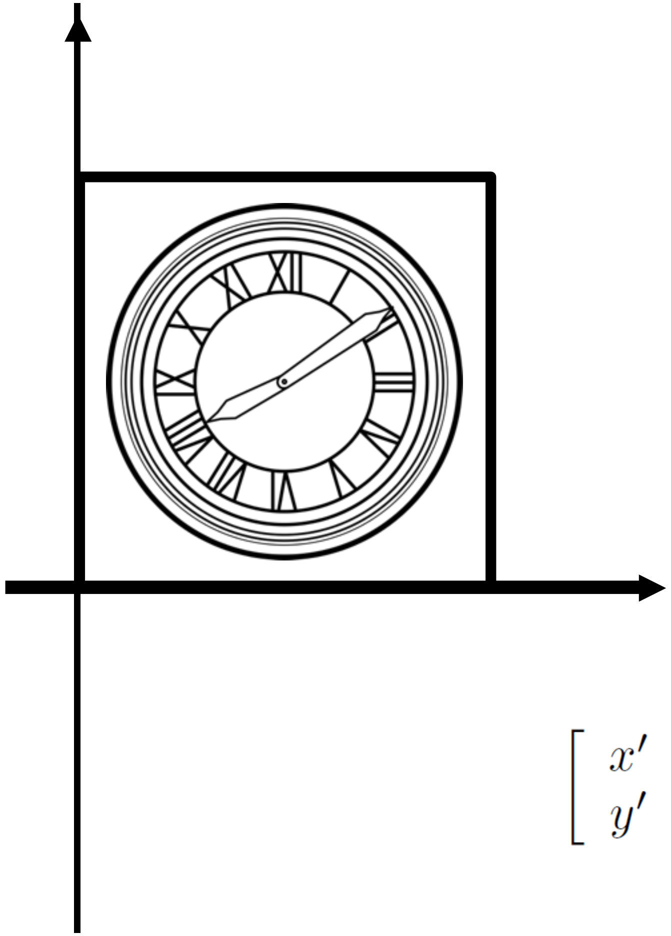
$\xrightarrow{s_{0.5}}$

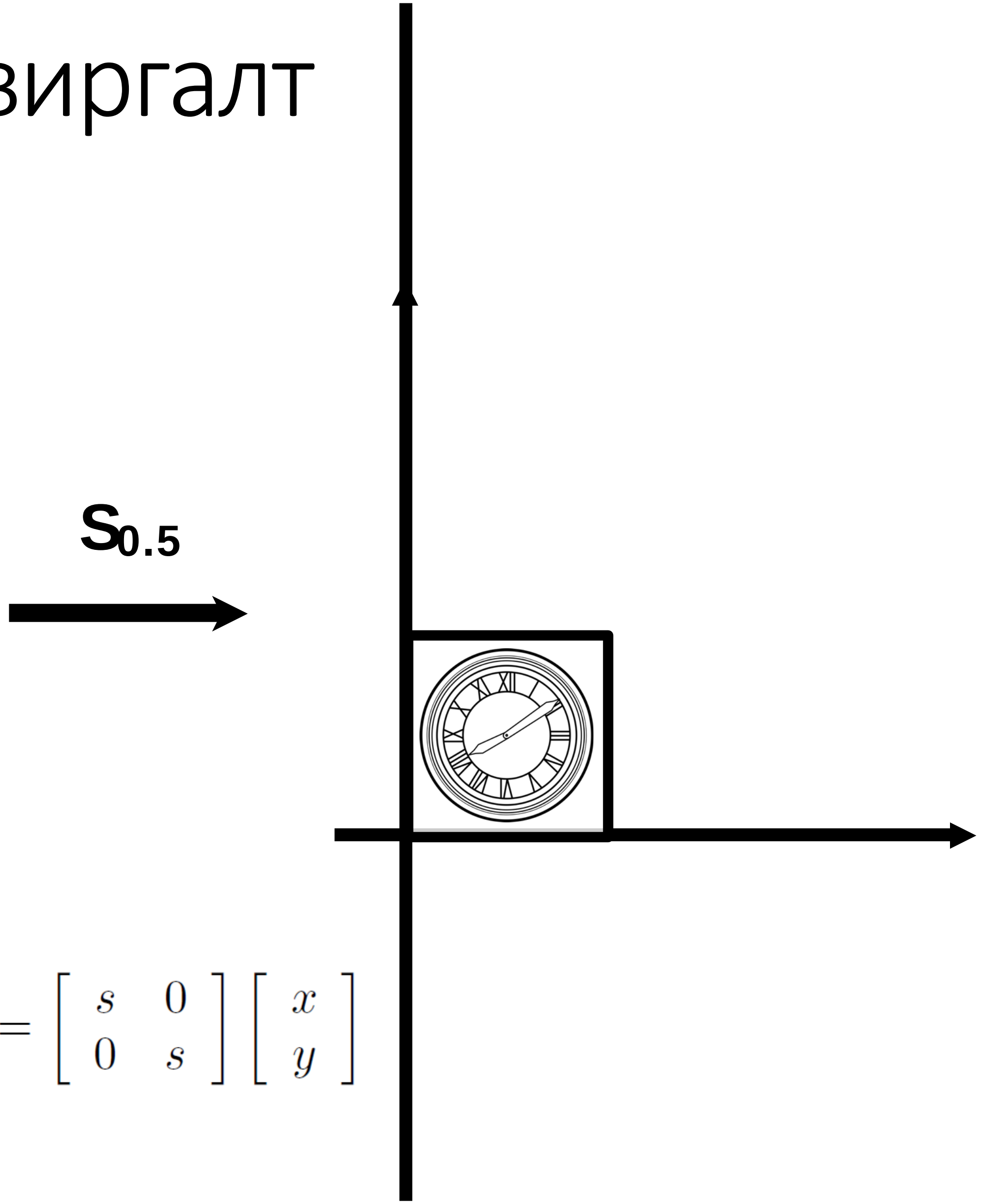


$$x' = sx$$

$$y' = sy$$

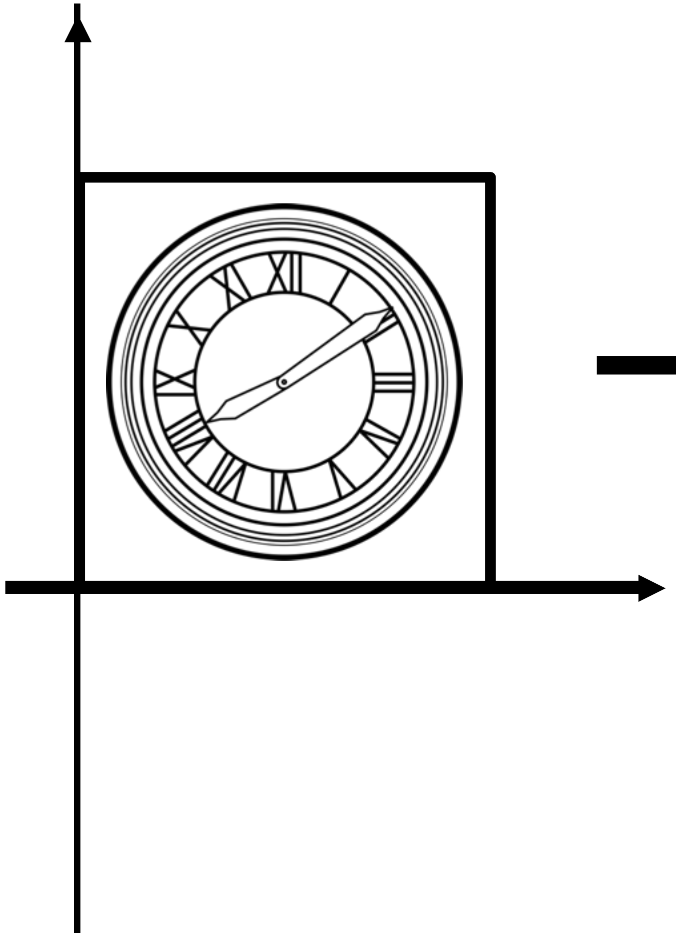
Масштаблах хувиргалт

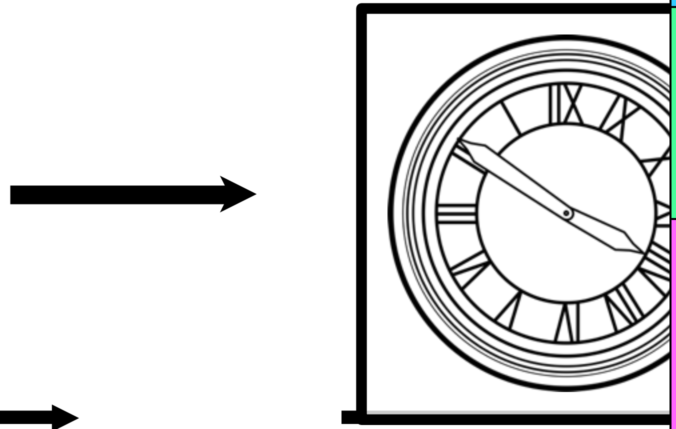




$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Толин матриц(Reflection Matrix)



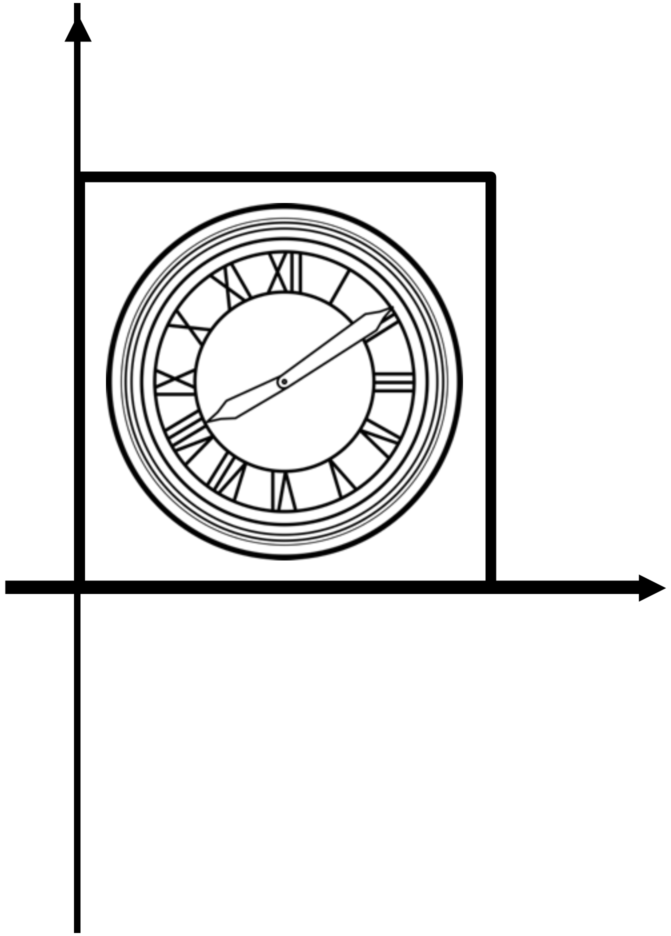


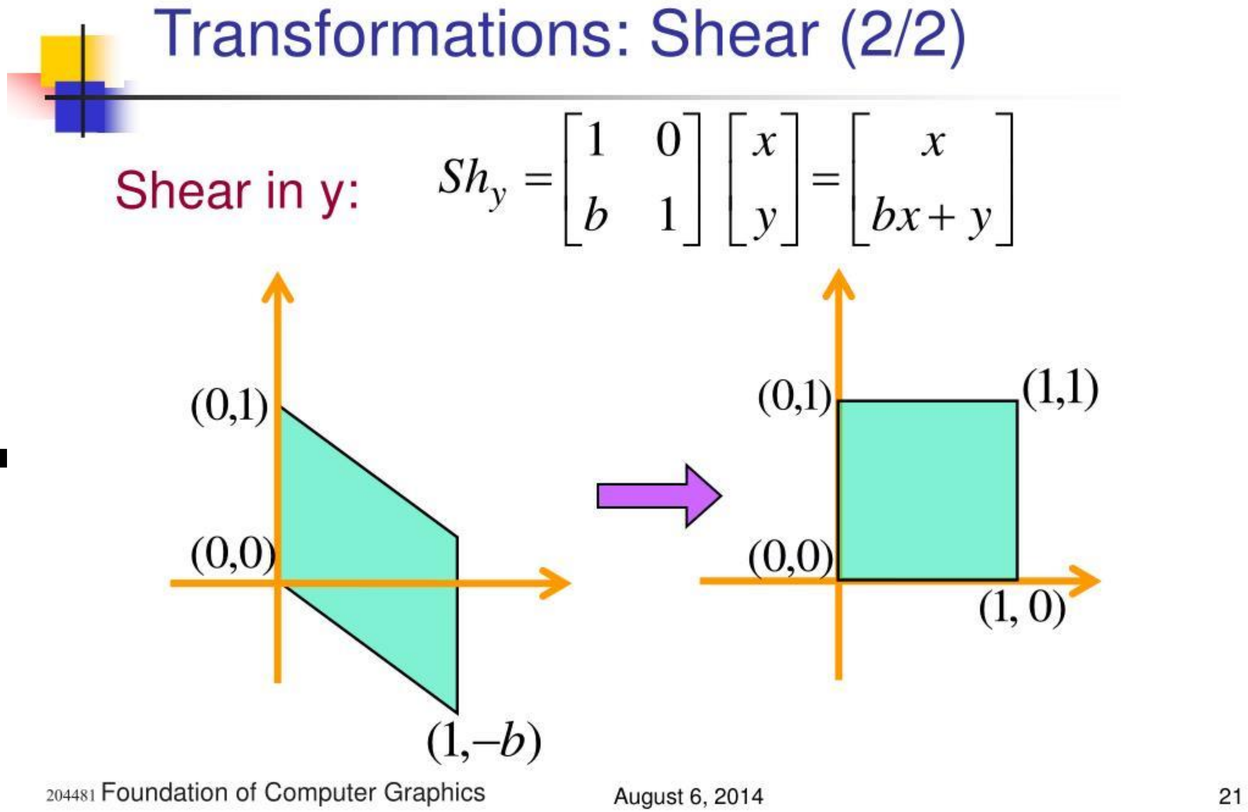
$x' = ??$

$y' = ??$

TYPE OF REFLECTION	Matrix to be multiplied
Reflection about the x- axis	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Reflection about the y-axis	$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$
Reflection about the line y = x	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Reflection about the line y = -x	$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$
Reflection about the origin	$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$

Налуу матриц (Shear Matrix)





$x' = ??$

$y' = ??$

Шугаман хувиргалт = Матриц

$$x' = a\,x + b\,y$$
$$y' = c\,x + d\,y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{x}' = \mathbf{M}\ \mathbf{x}$$

2 хэмжээст координатын систем

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} a \\ c \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

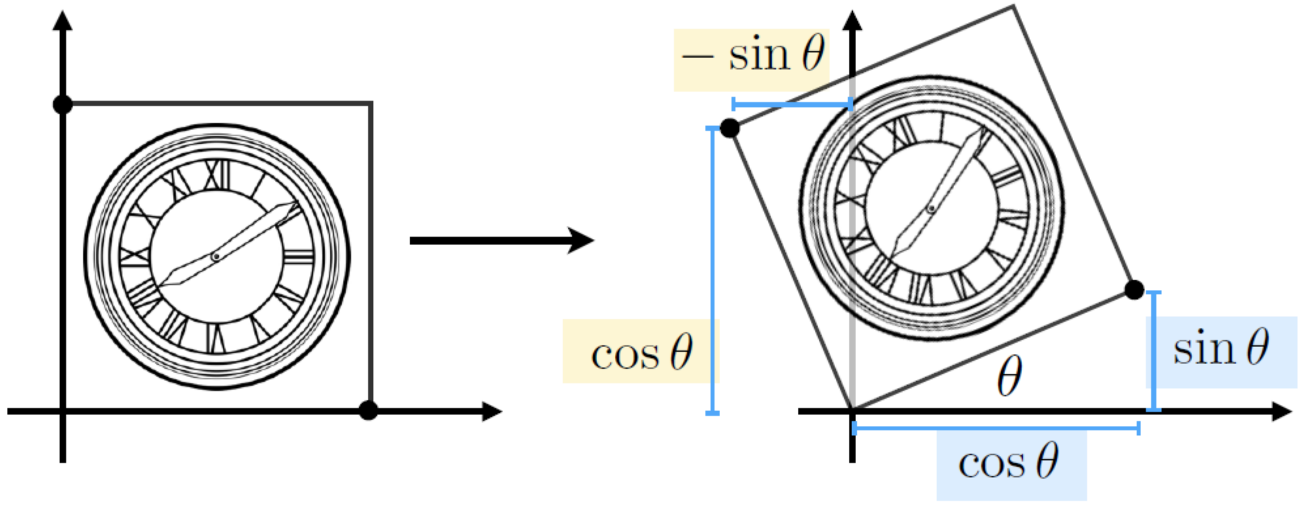
$$\begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Матрицын баганыг координатын системийн x, y тэнхлэг рүү хөрвүүлнэ

Эргэлтийн матриц (Rotation Matrix)

$\frac{1}{2}$

$\frac{\sqrt{0}}{2}$



$\cos 0^\circ = 1$

0 sin 0

30° $\sin 30^\circ$ $\frac{\sqrt{1}}{2}$

45° $\sin 45^\circ$ $\frac{\sqrt{2}}{2}$

60° $\frac{\sqrt{3}}{2}$

1

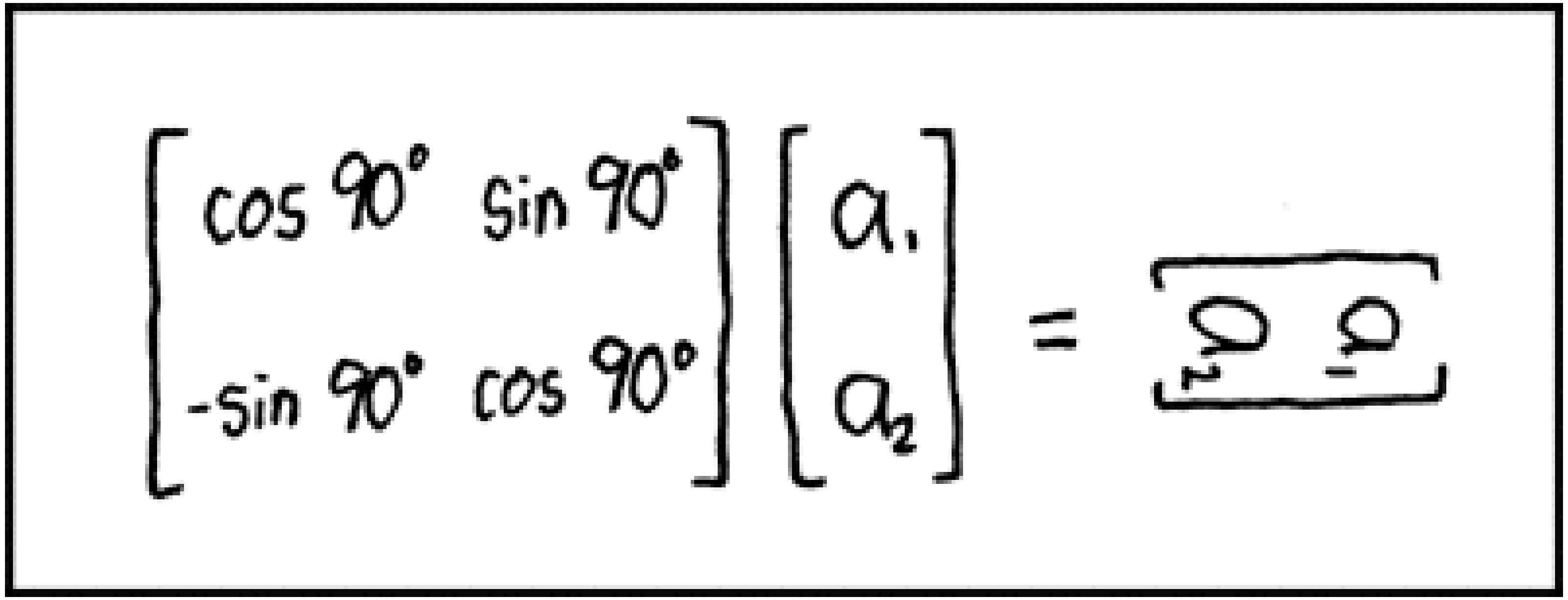
$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

90° $\frac{\sqrt{4}}{2} = 1$

2D Rotation Matrix: Another Way

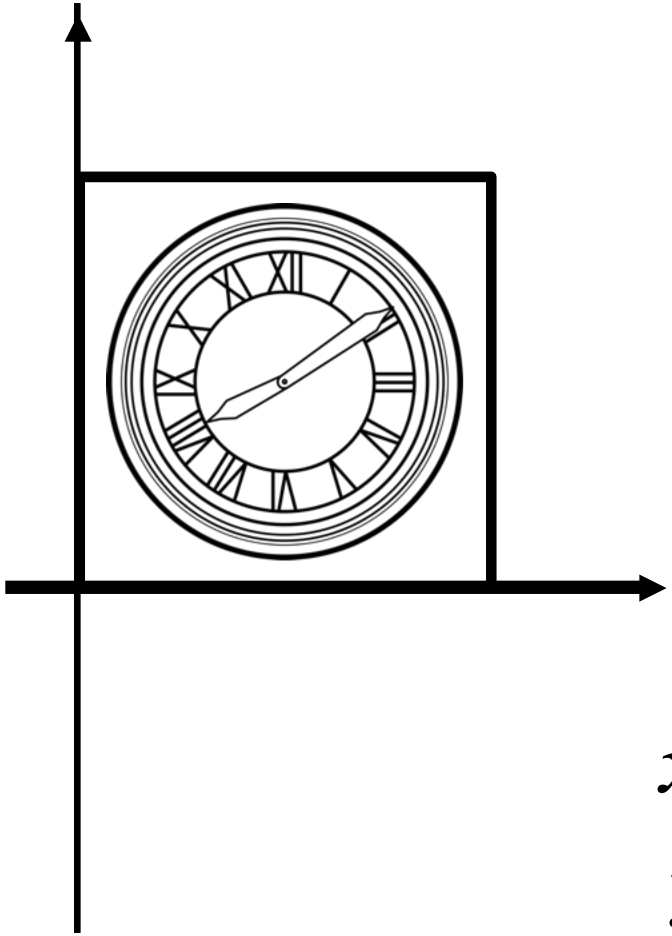
xkcd

A WEBCOMIC OF ROMANCE, SARCASM, MATH, AND LANGUAGE.

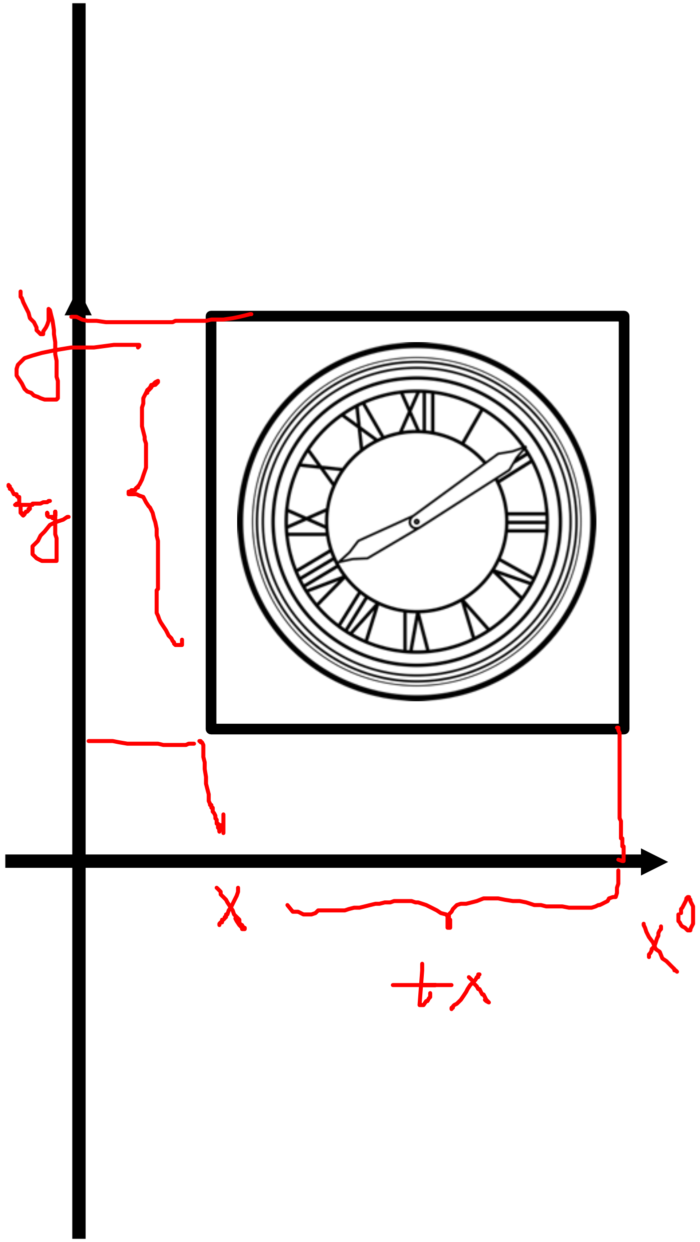


http://xkcd.com/184/

Хувиргалт??



$\xrightarrow{\mathbf{T_{1,1}}}$



$$x^0 = x + t_x$$

$$y^0 = y + t_y$$

Шийдэл: Homogenous координат

Гуравдагч координатыг нэмэх (*w-координат*)

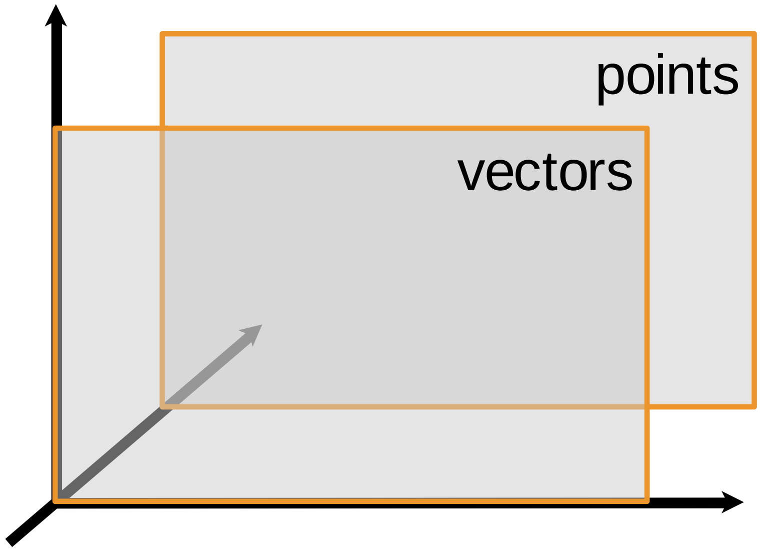
- **2D цэг = (x, y, 1)ᵀ**
- **2D вектор = (x, y, 0)ᵀ**

Хувиргалтын матриц дүрслэл

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix}$$

Homogenous Координат

Геометрийн тайлбар : R^3 дахь 2 хавтгай



Affine Хувиргалтууд

Affine map = linear map + translation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

homogenous координатын хэрэглээ:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

2D Хувиргалтууд

Scale

$$\mathbf{S}(s_x, s_y) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotation

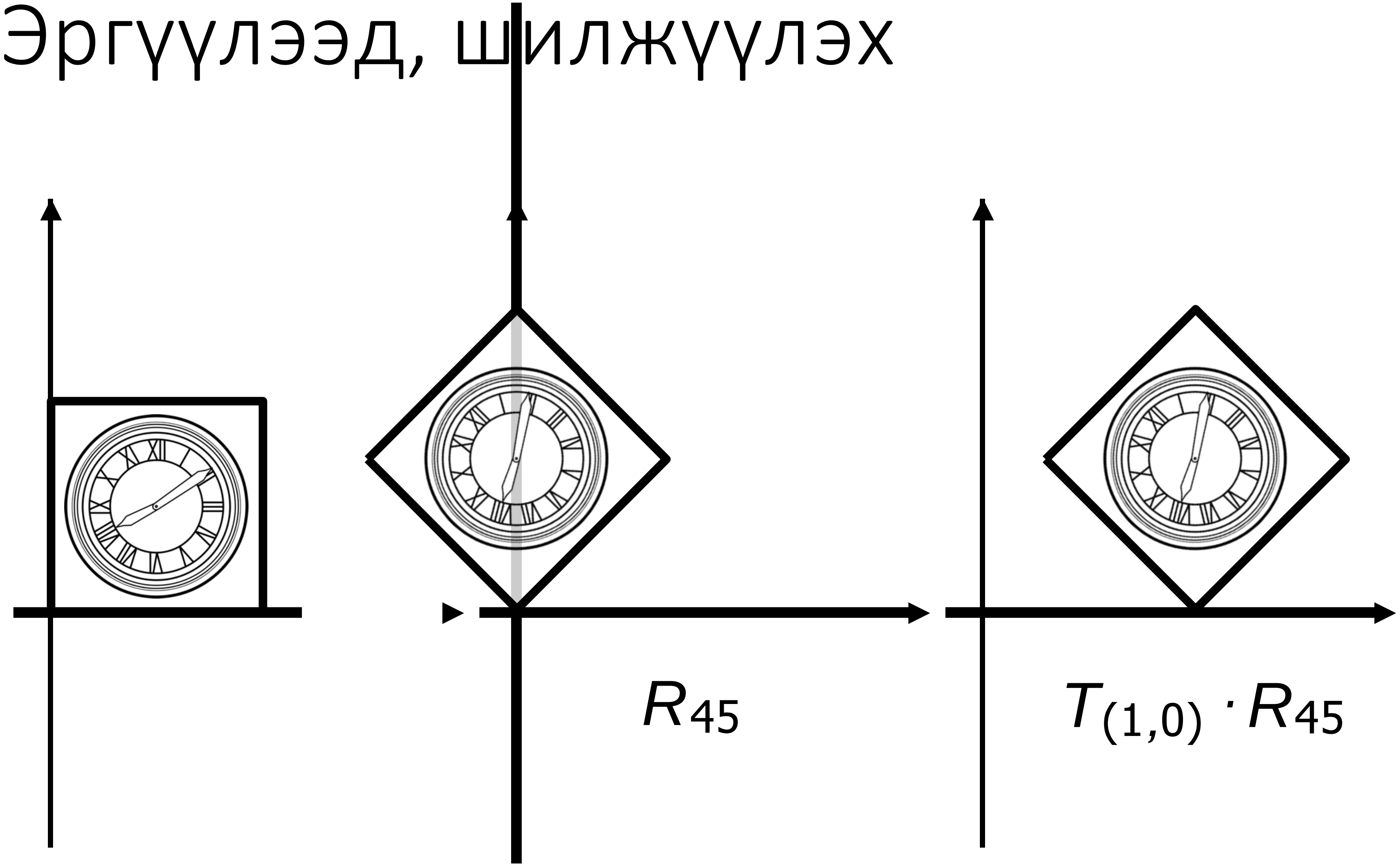
$$\mathbf{R}(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Translation

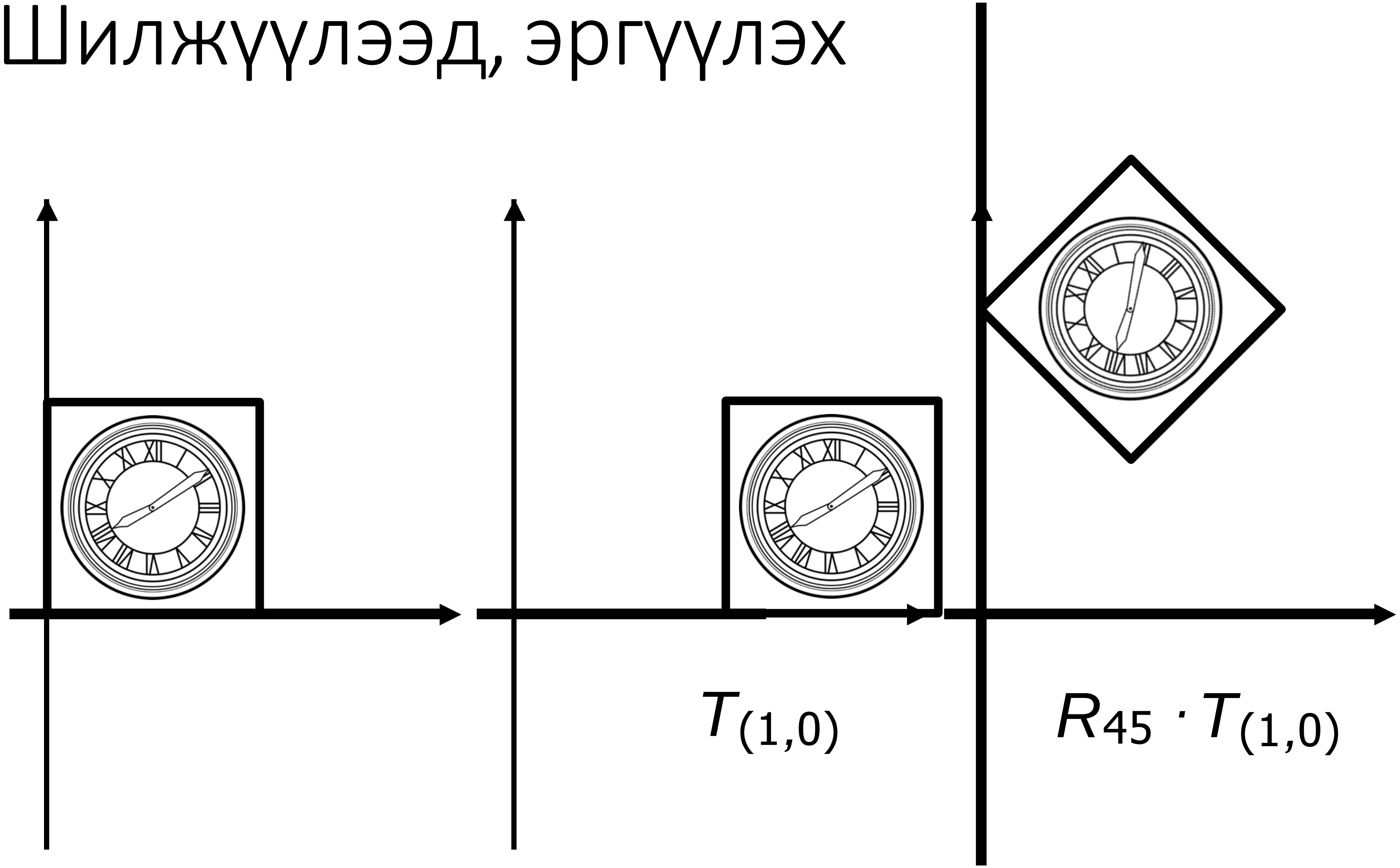
$$\mathbf{T}(t_x, t_y) = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

Хувиргалтыг зохиох

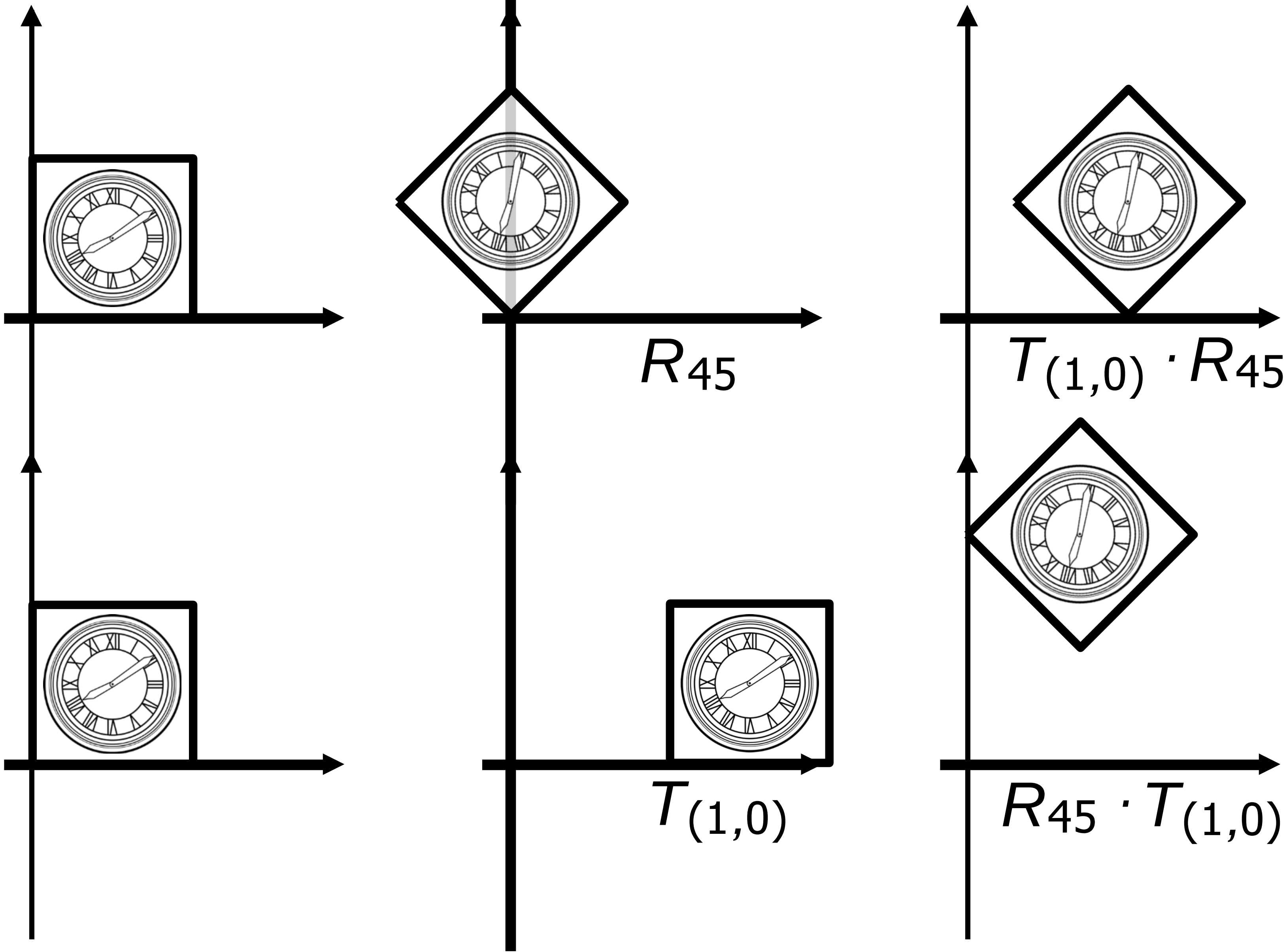
Эргүүлээд, шилжүүлэх



Шилжүүлээд, эргүүлэх



Хувиргалтыг дараалуулах шалтгаан



Хувиргалтыг зохиох

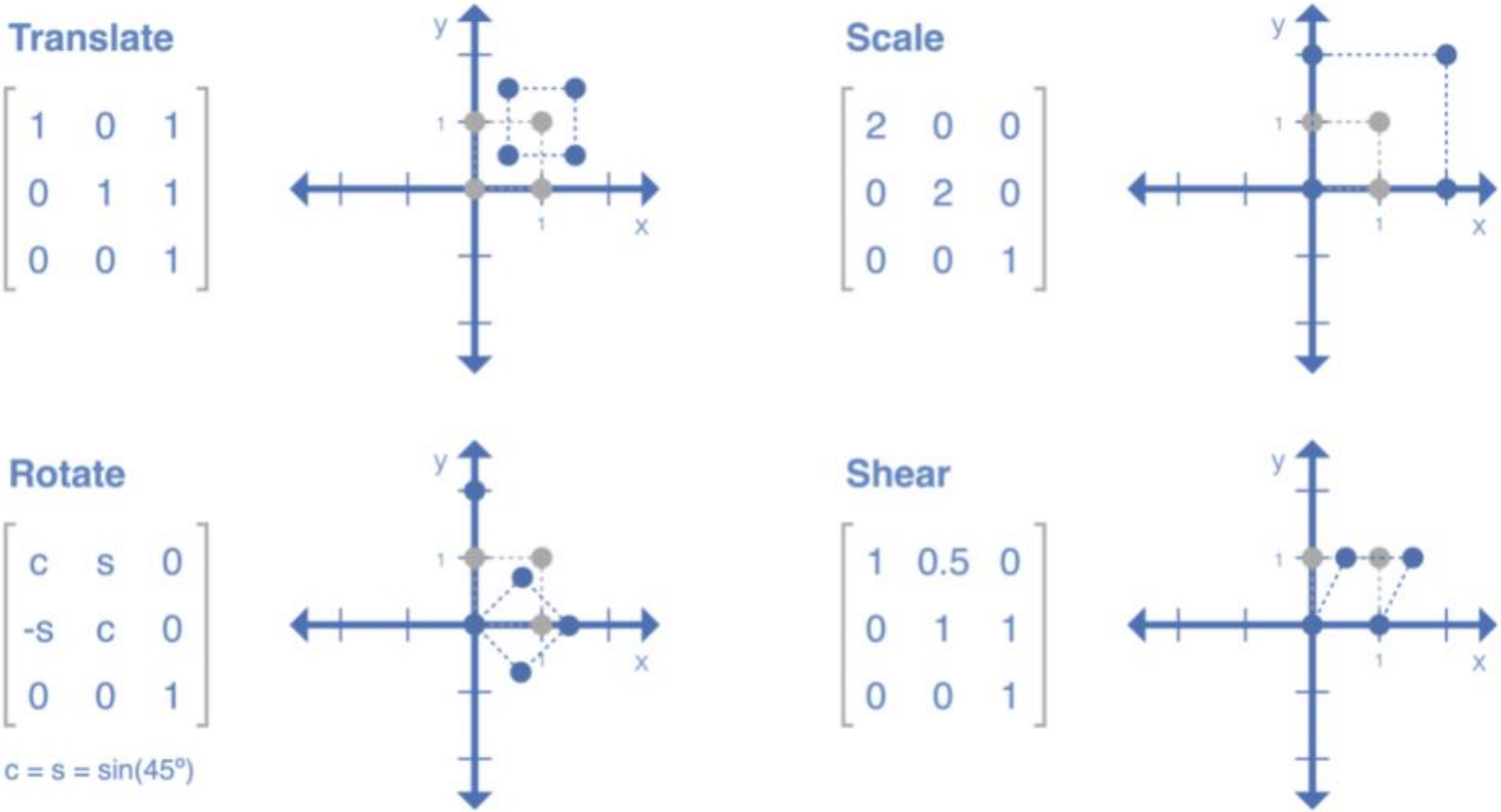
Affine хувиргалтын дараалал A_1, A_2, A_3, …

- **Дараалал маш чухал**

$$A_n(\ldots A_2(A_1(\mathbf{x}))) = \mathbf{A}_n \cdots \mathbf{A}_2 \cdot \mathbf{A}_1 \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Нийлмэл хувиргалтыг илэрхийлэгдэх нэг матрицыг гарган авахын тулд *n матрицыг өмнө нь үржүүлнэ.*

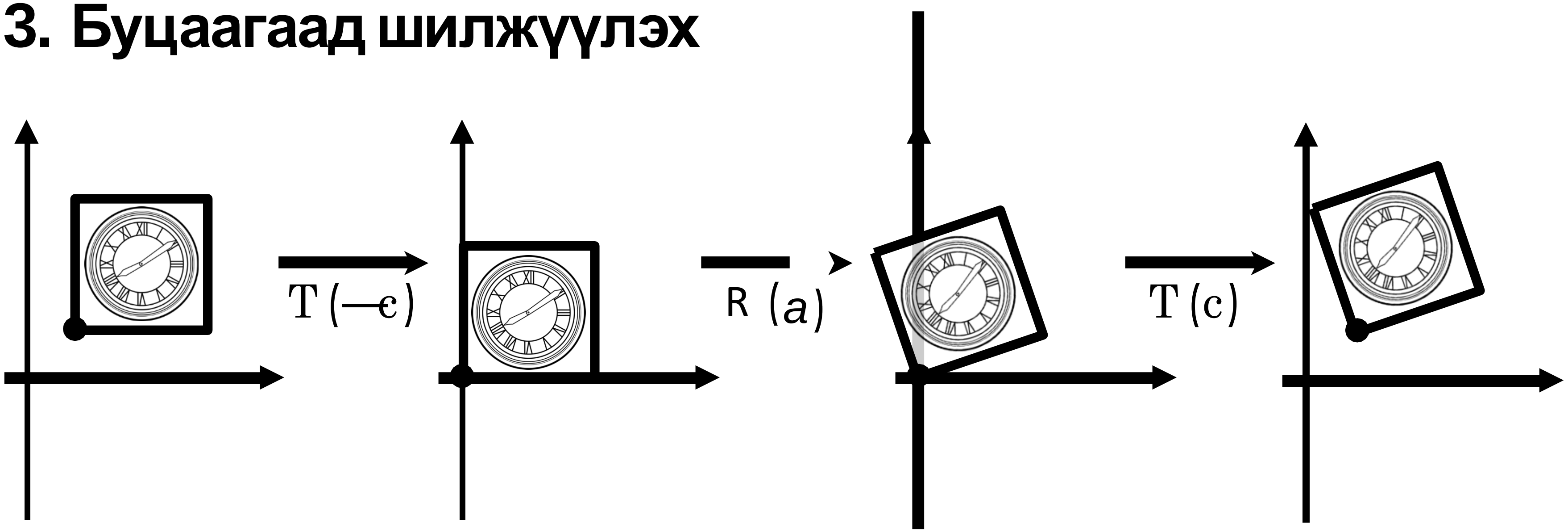
Affine хувиргалтын матрицууд



Нийлмэл матрицыг задлах

Өгөгдсөн цэгийн эргэн тойронд хэрхэн эргүүлэх вэ?

1. **Төвөөс нь мужид шилжүүлнэ**
2. **Эргүүлэлт хийх**
3. **Буцаагаад шилжүүлэх**



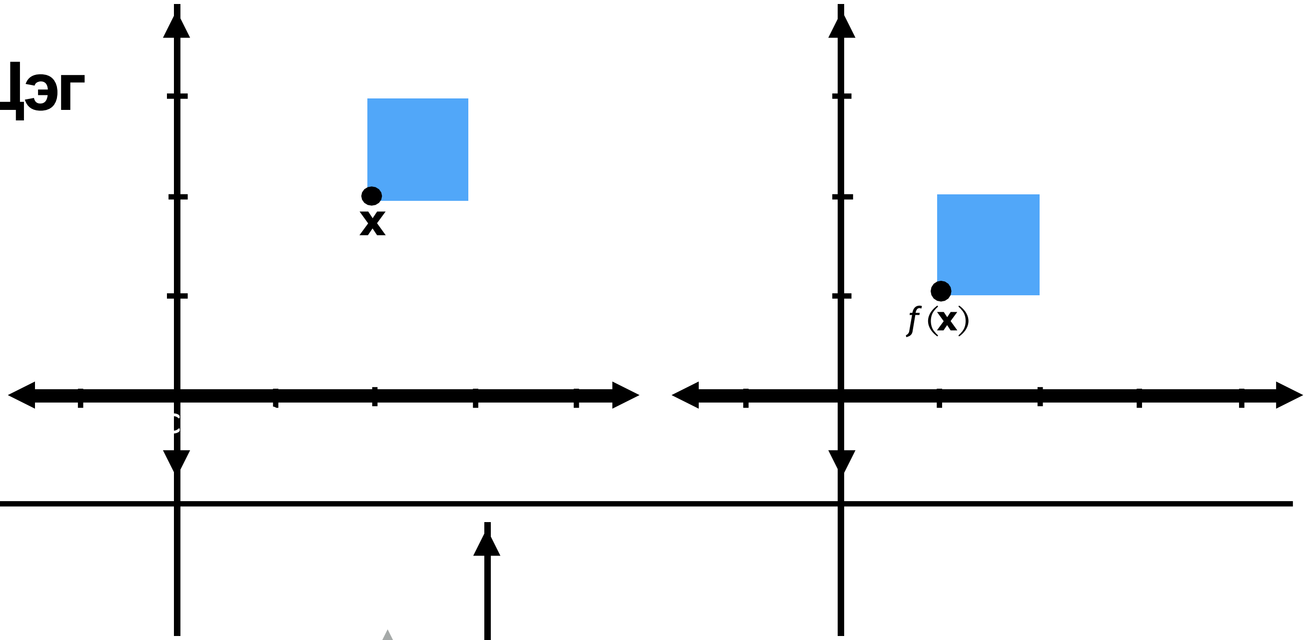
Матриц илэрхийлэл?

$$\mathbf{T}(\mathbf{c}) \cdot \mathbf{R}(\alpha) \cdot \mathbf{T}(-\mathbf{c})$$

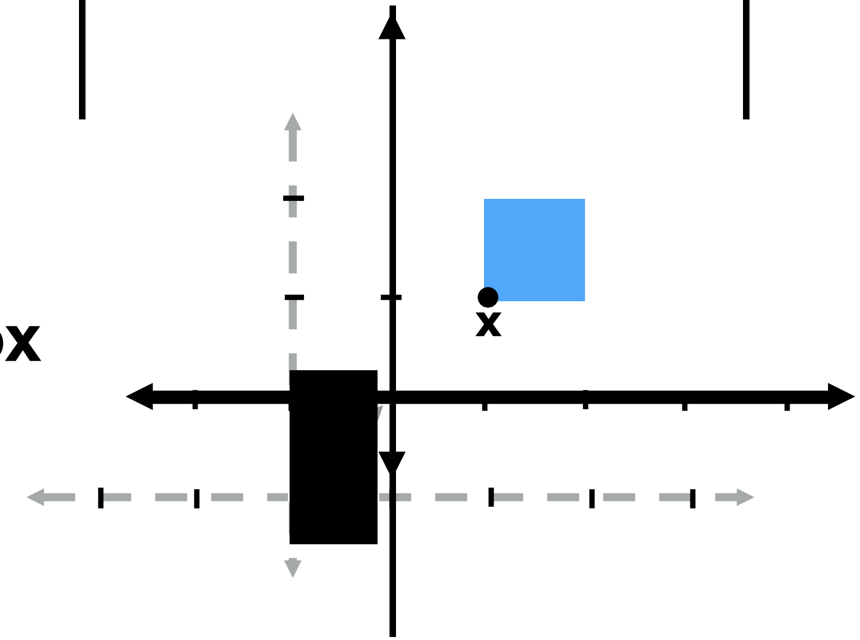
Координатын систем

Координат системийн хувиргалтууд

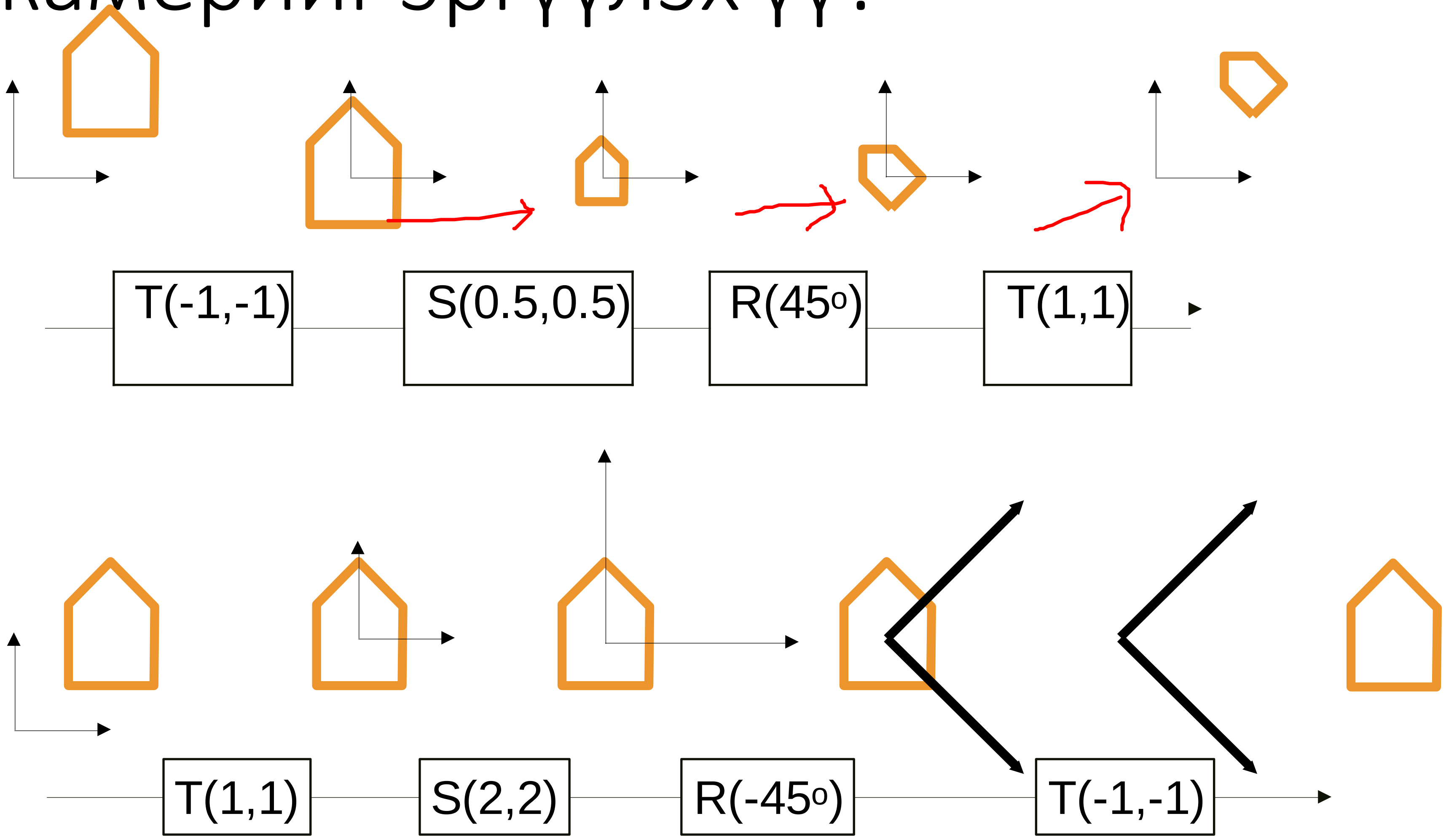
Тайлбар 1: Цэг шилжүүлэх хувиргалт



Тайлбар 2: Координатын системээ өөрчлөх



Объектийг эргүүлэх үү эсвэл камерийг эргүүлэх үү?



3D Хувиргалтууд

3D Хувиргалтууд

Дахин homogeneous координатыг ашиглах:

- **3D цэг = $(x, y, z, 1)^T$**
- **3D вектор = $(x, y, z, 0)^T$**

Use 4×4 matrices for affine хувиргалтын 4×4 матрицыг ашиглана

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

3D Хувиргалтууд

Scale

$$\mathbf{S}(s_x, s_y, s_z) = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Translation

$$\mathbf{T}(t_x, t_y, t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Coordinate Change (Frame-to-world)

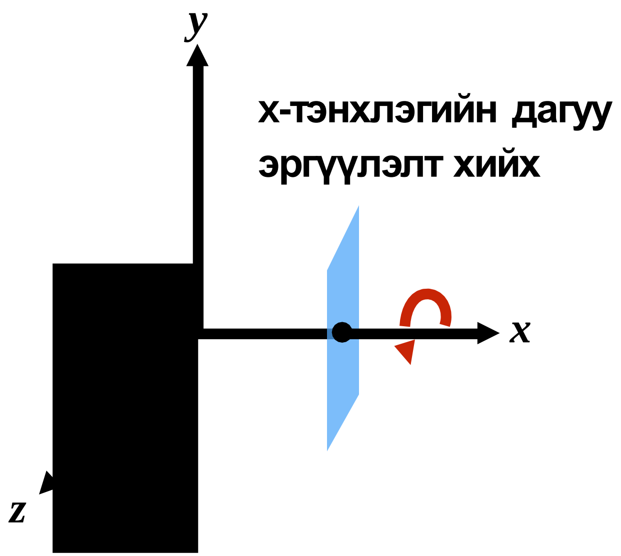
3D Хувиргалтууд

x-, y-, z-тэнхлэгийн дагуу эргүүлэлт хийх

$$\mathbf{R}_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R}_y(\alpha) = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R}_z(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

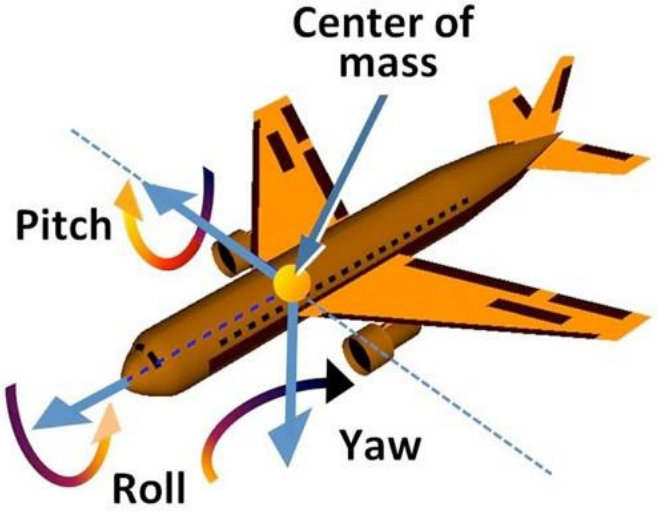


3D Эргүүлэлт

R_x, R_y, R_z–с ямар нэгэн 3D эргүүлэлт хийх боломжтой юу?

$$\mathbf{R}_{xyz}(\alpha,\beta,\gamma) = \mathbf{R}_x(\alpha)\,\mathbf{R}_y(\beta)\,\mathbf{R}_z(\gamma)$$

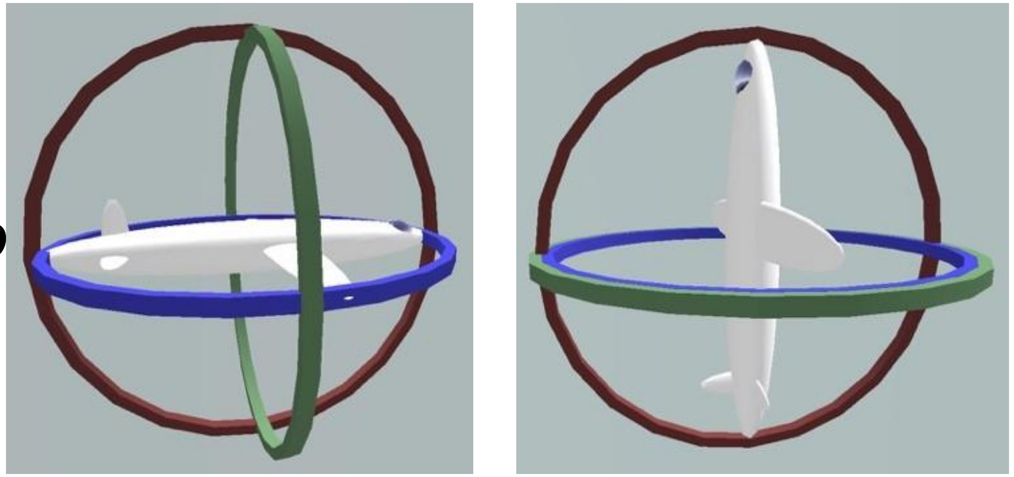
- ***Euler өнцөг гэж нэрлэдэг***
- **Нислэгийн симуляторуудад ашигладаг: эргэлдэх, далайц, хазайлт**



3D Эргүүлэлтүүд

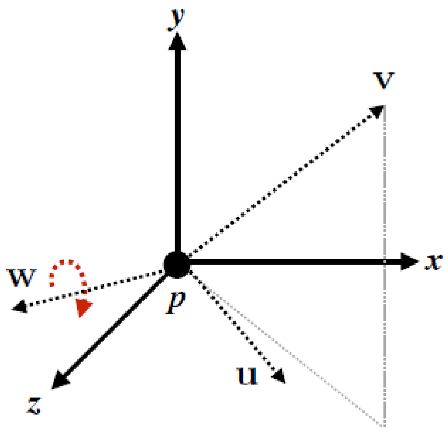
R_x, R_y, R_z–с ямар нэгэн 3D эргүүлэлт хийх боломжтой юу?

- ***Euler өнцөг гэж нэрлэдэг***
- **Нислэгийн симуляторуудад ашигладаг: эргэлдэх, далайц, хазайлт**
- **Асуудал:**
  Гимбалын түгжээ



Дурын өнцөгөөр 3D Эргүүлэлт хийх

Construct orthonormal frame transformation F with p, u, v, w, where p and w match the rotation axis



($F\ R_z(\theta)\ F^{-1}$) хувиргалтыг хэрэгжүүлэх

Тайлбар(аль алинд нь хүчинтэй):

- **Z тэнхлэг рүү зөөж, эргүүлээд байрлалд нь буцаана**
- **Шинэ координатын фрейм дэх w-тэнхлэгийн эргүүлэлт**

Родригугийн эргэлтийн томъёо

N тэнхлэгийн дагуу α өнцгөөр эргүүлэлт

$$\mathbf{R}(\mathbf{n},\alpha) = \cos(\alpha)\,\mathbf{I} + (1-\cos(\alpha))\,\mathbf{n}\mathbf{n}^T + \sin(\alpha)\underbrace{\begin{pmatrix} 0 & n_z & -n_y \\ -n_z & 0 & n_x \\ n_y & -n_x & 0 \end{pmatrix}}_{\mathbf{N}}$$

Энэхүү томъёог хэрхэн батлах вэ?

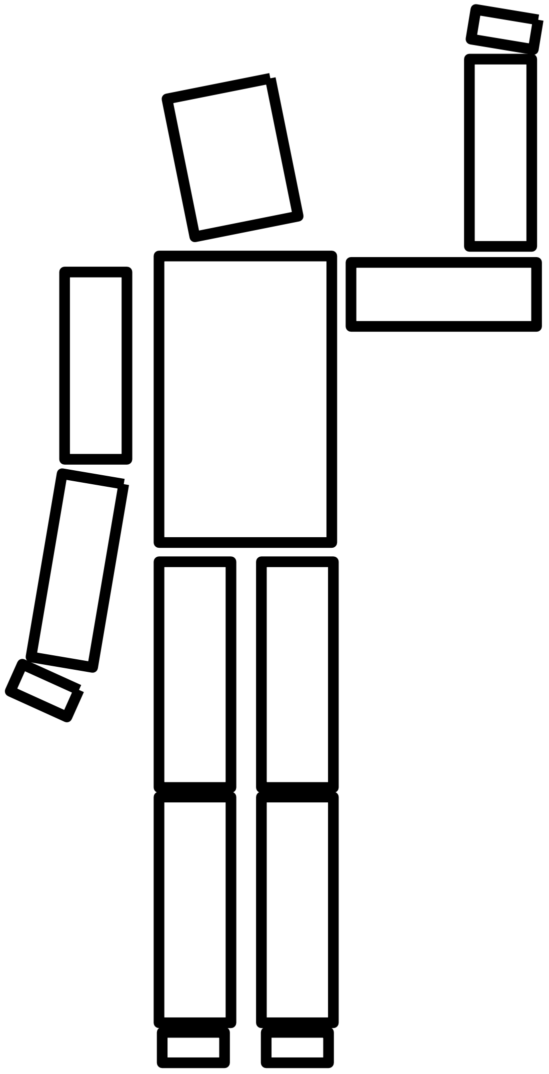
- **N матриц солбисон үрвэрийг тооцоолдог:** $\mathbf{N}\,x = \mathbf{n} \times x$
- **orthonormal систем гэж төсөөлвөл**

$$\begin{aligned} \mathbf{R}\mathbf{n} &= \mathbf{n} \\ \mathbf{R}\mathbf{e}_1 &= \cos\alpha\,\mathbf{e}_1 + \sin\alpha\,\mathbf{e}_2 \\ \mathbf{R}\mathbf{e}_2 &= -\sin\alpha\,\mathbf{e}_1 + \cos\alpha\,\mathbf{e}_2 \end{aligned}$$

Шаталсан хувиргалтууд

Араг яс – Шугаман дүрслэл

Толгой
Их бие
баруун гарны дээд хэсэг
баруун гарны доод хэсэг
зүүн гарны дээд хэсэг
зүүн гарны доод хэсэг
Баруун хөлний дээд хэсэг
Баруун хөлний доод хэсэг
Зүүн хөлний дээд хэсэг
Зүүн хөлний доод хэсэг



Шугаман дүрслэл

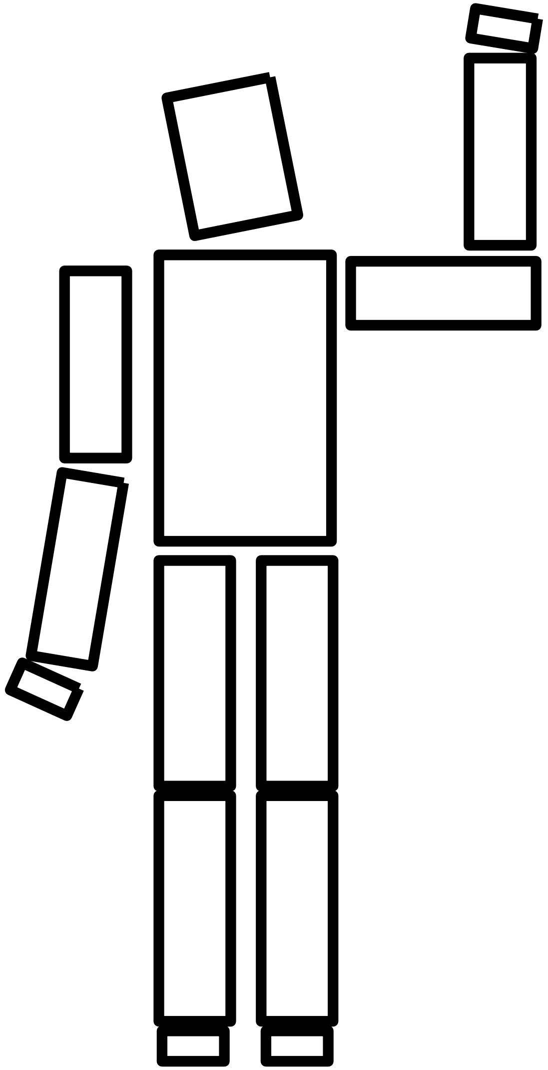
Дүрс тус бүр өөрсдийн хувиргалтуудтай холбоотой.

Нэг засвар олон хувиргалтуудыг шаардаж болно.

Жнь: гарыг өргөхийн тулд гарын бүх хэсгүүдийн хувиргалтыг шинэчлэх шаардлагатай

Араг яс – Шугаман дүрслэл

- **Бие**
  - **Их бие**
    - **Толгой**
    - **Баруун гар**
      - **Гарын дээд хэсэг**
      - **Гарын доод хэсэг**
    - **Зүүн гар**
      - **Гарын дээд хэсэг**
      - **Гарын доод хэсэг**
    - **Баруун хөл**
      - **Хөлний дээд хэсэг**
      - **Хөлний доод хэсэг**
    - **Зүүн хөл**
      - **Хөлний дээд хэсэг**
      - **Хөлний доод хэсэг**



Шаталсан дүрслэл

Бүлэглэсэн дүрслэл (мод)

- Бүлэг тус бүр дэд бүлэг эсвэл дүрсийг агуулна
- Бүлэг тус бүр эцэг бүлэгт хамаарах хувиргалтуудад хамаарна
- Навч-зангилаан дээрх хувиргалт нь цэцэг зангилаанаас навч руу шилжих замууд дээрх бүх шилжилтийг холбох явдал юм.
- Бүлгийн хувиргалтыг өөрчлөх нь бүх хэсгүүдэд нөлөөлдөг.
  - Зөвхөн нэг зангилаан дээр дээд түвшиний засвар хийхийг зөвшөөрдөг
  - Жнь: зүүн гарыг өргөхийн тулд тухайн бүлэгт зөвхөн нэг өөрчлөлтийг шаардана

Араг яс – Шугаман дүрслэл

```
translate(0, 10);
    drawTorso();
        pushmatrix();  // push a copy of transform onto stack
            translate(0, 5);  // right-multiply onto current transform
            rotate(headRotation);  // right-multiply onto current transform
            drawHead();
        popmatrix();  // pop current transform off stack
        pushmatrix();
            translate(-2, 3);
            rotate(rightShoulderRotation);
            drawUpperArm();
            pushmatrix();
                translate(0, -3);  rotate
                (elbowRotation);  draw
                LowerArm();  pushmat
                rix();
                    translate(0, -3);
                    rotate(wristRotation);
                    drawHand();
                popmatrix();
            popmatrix();                    popmatrix(
        );
        ....
```

right hand

right lower arm group

right arm group