

Universidad Nacional de San Antonio Abad del Cusco
Departamento Académico de Informática
COMPUTACIÓN GRAFICA I
Práctica N° 7

TRANSFORMACION 2D
SHEARING

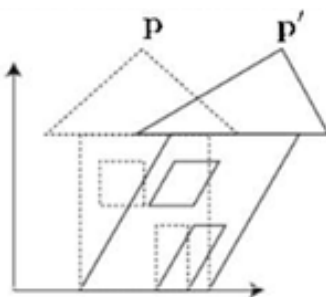
1. OBJETIVO.

- Entender los conceptos geométricos de deformación.
- Implementar y aplicar transformaciones de deformación..

2. BASE TEORICA

2.1. Deformación (Shearing).

La operación de deformación o corte (shearing) en 2D produce una distorsión de la forma del objeto al estirarlo con respecto a un eje.

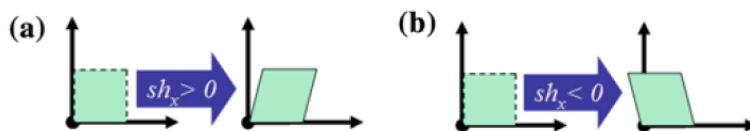


2.2. Deformación a lo largo del eje x.

La deformación se realiza a lo largo del eje x aplicando las siguientes ecuaciones:

$$\begin{aligned}x_2 &= x_1 + sh_x y_1 \\ y_2 &= y_1,\end{aligned}$$

donde sh_x es un factor que controla la deformación a lo largo del eje x. Este factor puede ser mayor o menor que 0. El efecto se muestra en la figura.



En forma matricial, esta operación se expresa como:

$$\underbrace{\begin{bmatrix} x_2 \\ y_2 \end{bmatrix}}_{\vec{p}_2} = \underbrace{\begin{bmatrix} 1 & sh_x \\ 0 & 1 \end{bmatrix}}_{\vec{sh}_x(sh_x)} \underbrace{\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}}_{\vec{p}_1}$$

donde \vec{sh}_x es una matriz de 2×2 que representa la operación de deformación a lo largo del eje x para puntos no homogéneos en el espacio 2D; y sh_x es el factor de

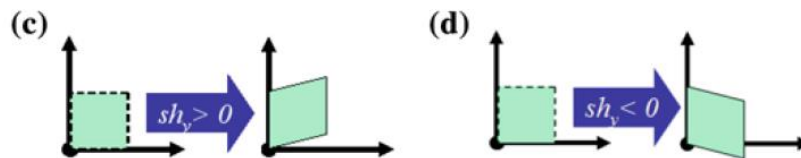
deformación. Tenga en cuenta que todos los puntos en el eje **x** (es decir, que tienen coordenadas **y** de 0) no se ven afectados por la matriz **'Sh_x'**.

2.3. Deformación a lo largo del eje y.

La deformación se realiza a lo largo del eje y aplicando las siguientes ecuaciones:

$$\begin{aligned}x_2 &= x_1, \\ y_2 &= sh_y x_1 + y_1\end{aligned}$$

donde **sh_y** es un factor que controla la deformación a lo largo del eje **y**. Este factor puede ser mayor o menor que 0. El efecto se muestra en la Fig.



En forma matricial, esta operación se expresa como:

$$\underbrace{\begin{bmatrix} x_2 \\ y_2 \end{bmatrix}}_{\vec{p}_2} = \underbrace{\begin{bmatrix} 1 & 0 \\ sh_y & 1 \end{bmatrix}}_{Sh_y(sh_y)} \underbrace{\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}}_{\vec{p}_1}$$

donde **'Sh_y'** es una matriz de 2 × 2 que representa la operación de deformación a lo largo del eje **y** para puntos no homogéneos en el espacio 2D; y **sh_y** es el factor de deformación. Tener en cuenta que todos los puntos en el eje **y** (es decir, que tienen coordenadas **x** de 0) no se ven afectados por la matriz **'Sh_y'**

3. DESARROLLO DE LA PRÁCTICA.

3.1. Deformación de un rectángulo con respecto al eje X.

```
#include<math.h>
#include<stdlib.h>
#include<GL/glew.h>
#include<GL/glut.h>
#include<stdio.h>
using namespace std;
/--deformación de rectángulo con respecto al eje X
void deformaRectanguloX(double x1, double y1, double x2, double y2, double
sh)
{
    int x1d, y1d, x2d, y2d, x3d, y3d, x4d, y4d;
    //--dibujar el rectangulo original
    glBegin(GL_LINE_LOOP);
    glVertex2f(x1, y1);
    glVertex2f(x2, y1);
    glVertex2f(x2, y2);
    glVertex2f(x1, y2);
    glEnd();

    //--deformar cada punto del rectangulo con respecto al eje X
    x1d = x1 + sh*y1;
    y1d = y1;

    x2d = x2 + sh*y1;
    y2d = y1;
```

```

        x3d = x2 + sh*y2;
        y3d = y2;

        x4d = x1 + sh*y2;
        y4d = y2;

        //--dibujar el rectangulo deformado con respecto al eje X
        glBegin(GL_LINE_LOOP);
        glVertex2f(x1d, y1d);
        glVertex2f(x2d, y2d);
        glVertex2f(x3d, y3d);
        glVertex2f(x4d, y4d);
        glEnd();
    }
    //--despliega el gráfico
    void display()
    {
        double x1, y1, x2, y2, sh;
        x1 = 0.0;
        y1 = 0.0;
        x2 = 50.0;
        y2 = 40.0;

        sh = 2;

        glClear(GL_COLOR_BUFFER_BIT); //--establece el color de la ventana
        deformaRectanguloX(x1, y1, x2, y2, sh);
        glFlush(); //--fuerza la ejecución de los comandos de OpenGL
    }

    void myinit()
    {
        glClearColor(1.0, 1.0, 1.0, 1.0); glColor3f(1.0, 0.0, 0.0);
        glPointSize(1.0); //--tamaño de los puntos
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(0.0, 499.0, 0.0, 499.0);
    }

    void main(int argc, char** argv)
    {
        /* Inicializacion GLUT estándar*/
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(500, 500); /* ventana 500x500 pixeles */
        glutInitWindowPosition(0, 0);
        glutCreateWindow("Reflexión"); /* título de la ventana*/
        glutDisplayFunc(display);
        myinit(); /* fija o establece los atributos */
        glutMainLoop(); /* entra a un ciclo de evento */
    }

```

3.2. Deformación de un rectángulo con respecto al eje Y.

```

    //--deformación de rectangulo con respecto al eje Y
    void deformaRectanguloY(double x1, double y1, double x2, double y2, double
sh)
    {
        int x1d, y1d, x2d, y2d, x3d, y3d, x4d, y4d;
        //--dibujar el rectangulo original
        glBegin(GL_LINE_LOOP);

```

```

glVertex2f(x1, y1);
glVertex2f(x2, y1);
glVertex2f(x2, y2);
glVertex2f(x1, y2);
glEnd();

//--deformar cada punto del rectángulo con respecto al eje Y
x1d = x1;
y1d = y1 + sh*x1;

x2d = x2;
y2d = y1+ sh*x2;

x3d = x2;
y3d = y2 + sh*x2;

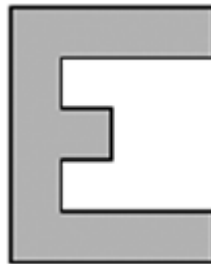
x4d = x1;
y4d = y2 + sh*x1;

//--dibujar el rectángulo deformado con respecto al eje X
glBegin(GL_LINE_LOOP);
glVertex2f(x1d, y1d);
glVertex2f(x2d, y2d);
glVertex2f(x3d, y3d);
glVertex2f(x4d, y4d);
glEnd();
}

```

4. TAREA

- 4.1. Escribir un programa que deforme la siguiente figura teniendo como factores de escalamiento en X=2 y en Y=3



1. BIBLIOGRAFIA

- ✓ Donald Hearn, M. Pauline Baker (2006); Gráficas por Computadora con OpenGL; Pearson Prentice Hall; 3° edición; Madrid.
- ✓ Rimón Elías (2019); Digital Media A problema-solving Aproach for Computer Graphics; eBook. New York
- ✓ Sumanta Guha (2019); Computer Graphics Through OpenGL; Third Edition; Taylor & Francis Group. NW.
- ✓ Gordon V.Scott, Clevenger Jhon (2019). Computer Graphics Programming in OpenGL with C++. Mercury Learning and Information.