|  |  |
| --- | --- |
| **Logo  Description automatically generated** | **BÀI THỰC HÀNH ĐỒ HỌA MÁY TÍNH** |
|  | **CÁC PHÉP CHIẾU**  **(Projection Transformation)** |

**Họ và tên Sinh viên:**Phan Trần Nhật Hạ  

**Mã Sinh viên:**102210159     **Nhóm:** 21Nh12

[**1.**     **Giới thiệu**](http://itfdut.ddns.net/mod/resource/tknetlab/LabDHMT-CacPhepChieu.htm#_Toc129273393)

[**2.**     **labProjection.cpp -  Chương trình minh họa các phép chiếu**](http://itfdut.ddns.net/mod/resource/tknetlab/LabDHMT-CacPhepChieu.htm#_Toc129273394)

[**3.**     **labOrthoProject.cpp**](http://itfdut.ddns.net/mod/resource/tknetlab/LabDHMT-CacPhepChieu.htm#_Toc129273395)

[**4.**     **labPersProjection02.cpp**](http://itfdut.ddns.net/mod/resource/tknetlab/LabDHMT-CacPhepChieu.htm#_Toc129273396)

[**5.**     **labPersProjection02.cpp**](http://itfdut.ddns.net/mod/resource/tknetlab/LabDHMT-CacPhepChieu.htm#_Toc129273397)

[**6.**     **BÀI TẬP**](http://itfdut.ddns.net/mod/resource/tknetlab/LabDHMT-CacPhepChieu.htm#_Toc129273398)

**1.    Giới thiệu**

§  Trước khi thực hiện phép chiếu trong OpenGL, cần gọi 2 hàm

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

§  Phép chiếu song song.

§  Phép chiếu phối cảnh (Perspective Projection): Đặc điểm của phép chiếu này là đối tượng càng lùi ra xa thì trông càng nhỏ

void **glFrustum**(GLdouble left, GLdouble right, GLdouble bottom,GLdouble top, GLdouble near, GLdouble far);

**2.    labProjection.cpp -  Chương trình minh họa các phép chiếu**

Xây dựng và hoàn thiện chương trình sử dụng các đoạn lệnh sau:

void myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glMatrixMode(GL\_PROJECTION); // set up projection

glLoadIdentity();

gluLookAt( ... ); // set up camera frame

gluPerspective(fovy, aspect, near, far);

// or glFrustum(...)

// or glOrtho(-3.0, 3.0, -3.0, 3.0, 1.0, 50.0);

glMatrixMode(GL\_MODELVIEW);

drawModel(); // draw everything

glutSwapBuffers();

}

Ví dụ:

*// top left: top view*

glViewport(0, win\_height/2, win\_width/2, win\_height/2);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-3.0, 3.0, -3.0, 3.0, 1.0, 50.0);

gluLookAt(0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -1.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

glCallList(object);

*// top right: right view*

glViewport(win\_width/2, win\_height/2, win\_width/2, win\_height/2);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-3.0, 3.0, -3.0, 3.0, 1.0, 50.0);

gluLookAt(5.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

glCallList(object);

*// bottom left: front view*

glViewport(0, 0, win\_width/2, win\_height/2);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

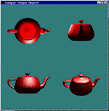
glOrtho(-3.0, 3.0, -3.0, 3.0, 1.0, 50.0);

gluLookAt(0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

glCallList(object);

*// bottom right: rotating perspective view*

glViewport(win\_width/2, 0, win\_width/2, win\_height/2);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(70.0, 1.0, 1, 50);

gluLookAt(0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

glMatrixMode(GL\_MODELVIEW);

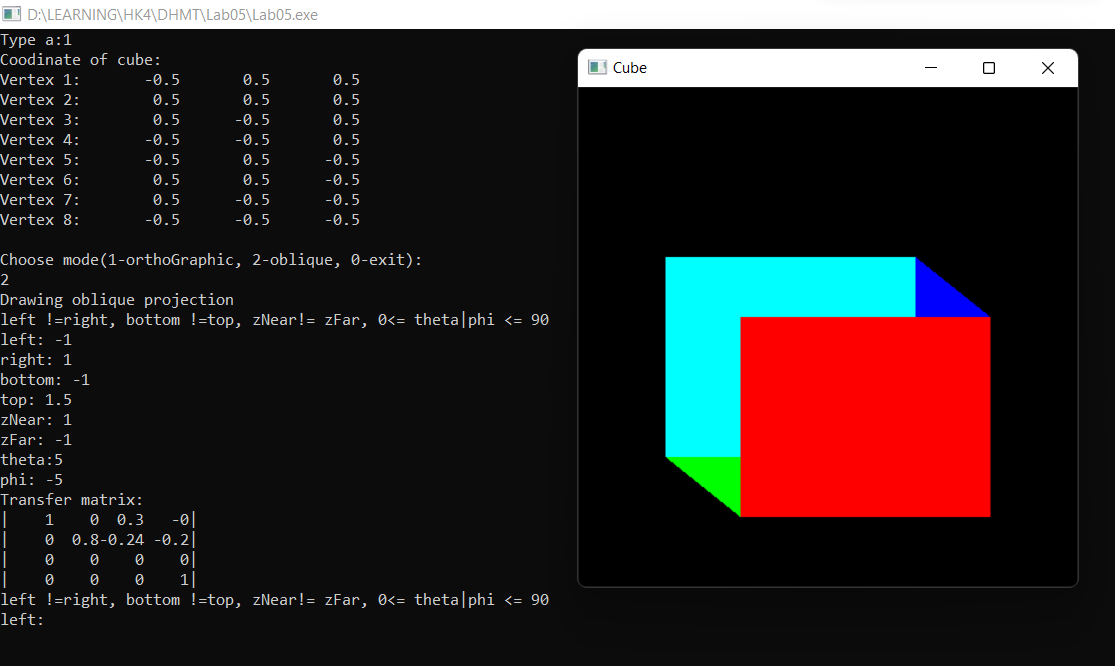
glLoadIdentity();

glRotatef(30.0, 1.0, 0.0, 0.0);

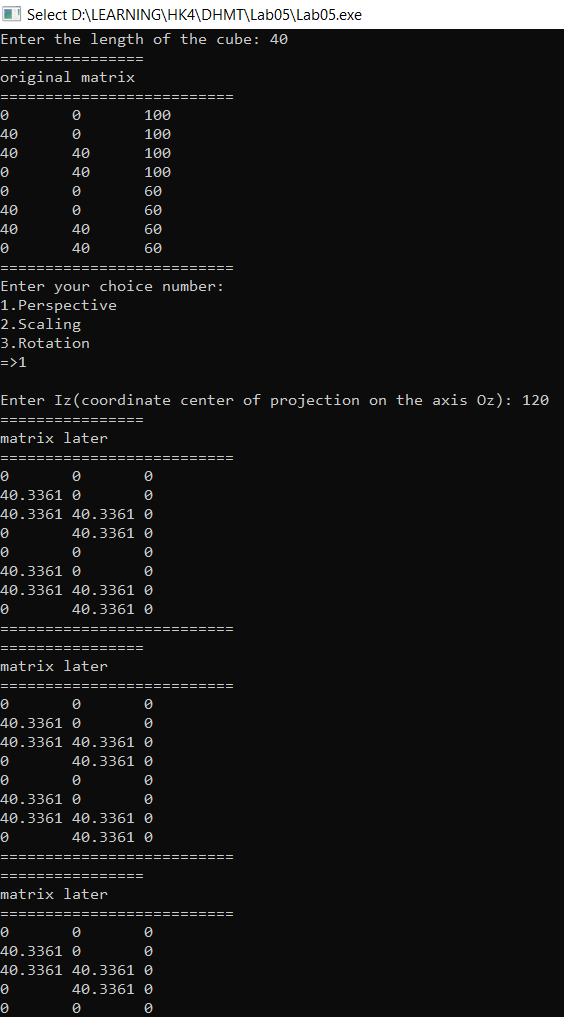
glRotatef(Angle, 0.0, 1.0, 0.0);

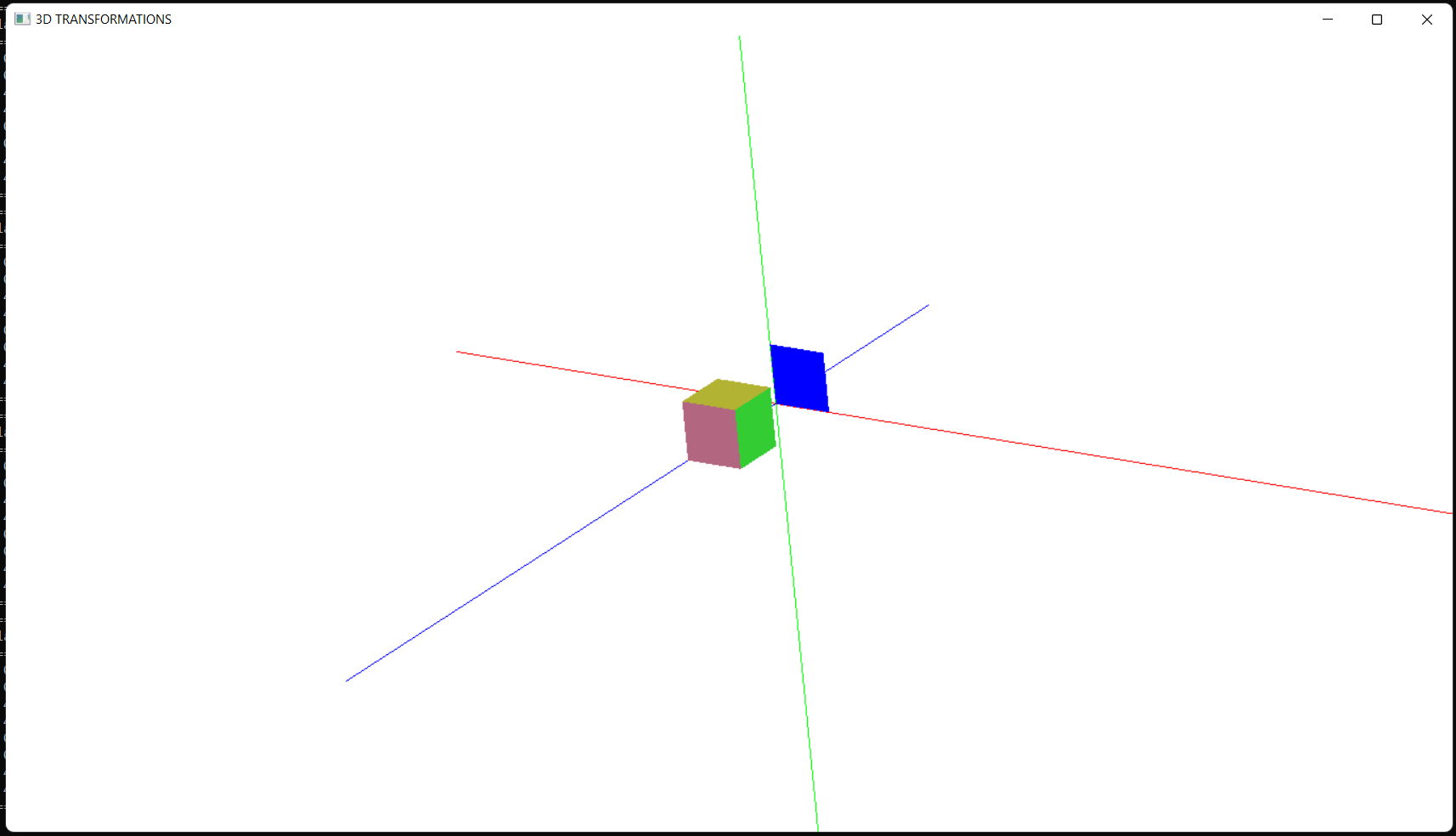
glCallList(object);

**3.    labOrthoProject.cpp**

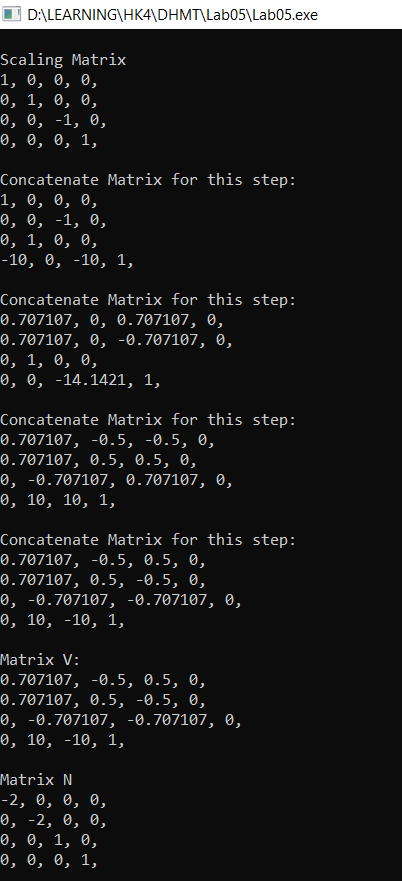
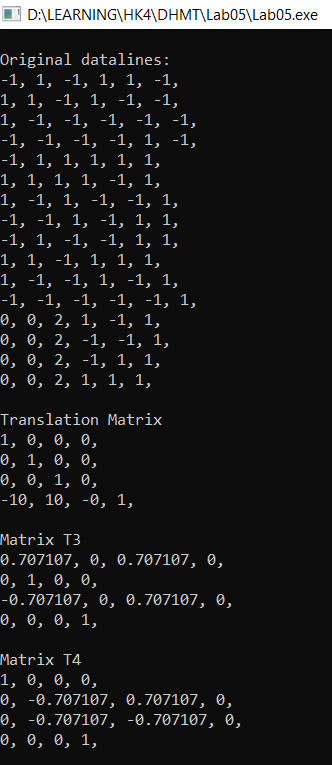


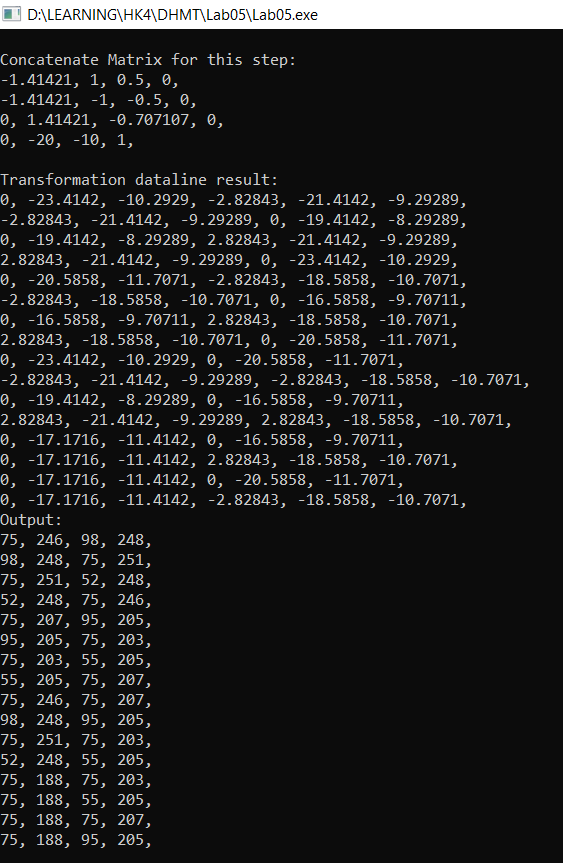
**4.    labPersProjection02.cpp**

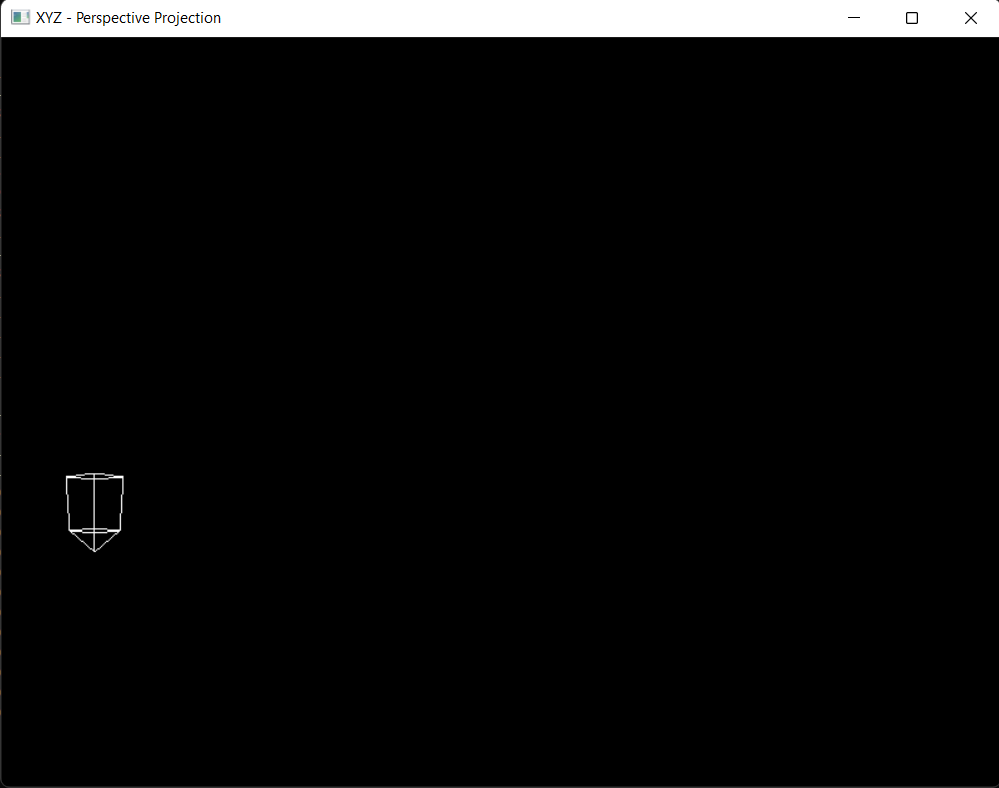


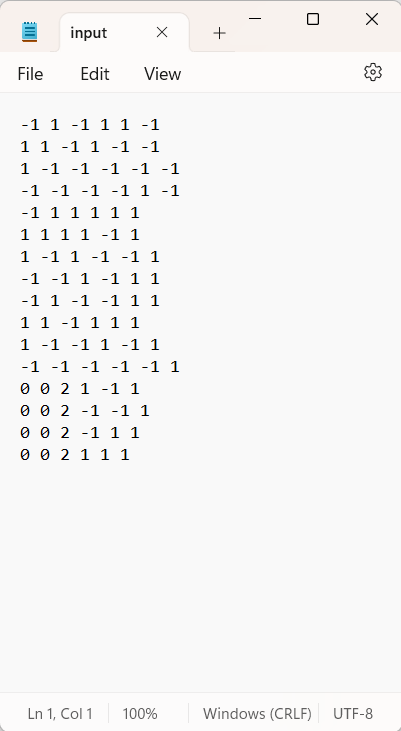


**5.    labPersProjection02.cpp**







**6.    BÀI TẬP**

Cho điểm P(x, y, z) trong hệ tọa độ thế giới thực (world space)

1) Trình bày ma trận biến đổi và xác định điểm Pw(xw, yw, zw, 1) khi thực hiện các phép biến đổi sau:

a. Phép tịnh tiến với dx, dy, dz

b. Phép quay quanh trục Ox, Oy, Oz

c. Phép quay quanh một trục PQ có P(xp, yp, zp), Q(xq, yq, zq)

2) Trình bày ma trận biến đổi và xác định điểm Pc(xc, yc, zc, 1) trong hệ tọa độ camera space khi thực hiện:

a) Phép biến đổi camera đặt tại t = (x1, y1, z1), nhìn vào c = (x2, y2, z2), hướng lên k = (0, 1, 0)]

3) Trình bày ma trận biến đổi và xác định điểm Pclip(xclip, yclip, zclip, wclip) trong clipping/projection space khi thực hiện:

a) Phép chiếu song song

b) Phép chiếu xiên

c) Phép chiếu trực giao

d) Phép chiếu phối cảnh

**BÀI TẬP 1:** Cho điểm P (x, y, z) xác định điểm Pw (xw, yw, zw, 1)

1. *Phép tịnh tiến với dx, dy, dz:*
2. *Phép quay quanh trục Ox, Oy, Oz:*

* Quanh trục Ox một góc θ:
* Quanh trục Oy một góc θ:
* Quanh trục Oz một góc θ:

1. *Phép xoay quanh trục PQ*

P (xp, yp, zp); Q (xq, yq, zq)

* Tịnh tiến PQ sao cho một điểm trùng với góc tọa độ O, ở đây ta tịnh tiến PQ sao cho P trùng với O với ma trận:

[MT]-1 =

* Ta có PQ = l.
* Xoay PQ một góc θ quanh trục Ox để PQ nằm trên mặt Oxz, ta có ma trận:

[MR] θx =

* Xoay PQ một góc α quanh trục Oy để PQ nằm trên tia Oz, ta có ma trận:

[MR] αy =

* Xoay PQ quanh trục Oz một góc ϕ, t có ma trận:

[MR] ϕz  =

* Xoay PQ ngược lại trục Oy một góc -α, ta có ma trận:

[MR] -αy =

* Xoay PQ ngược lại trục Ox một góc - θ, ta có ma trận:

[MR] -θx =

* Tịnh tiến PQ về vị trí ban đầu, ta có ma trận:

[MT] =

* Ma trận biến đổi là tích các ma trận thành phần:

[MR]PQ = [MT]-1. [MR] θx. [MR] αy. [MR] ϕz . [MR] -αy. [MR] -θx. [MT]

**BÀI TẬP 2:** Xác định điểm Pc (xc, yc, zc, 1) trong hệ tọa độ camera space

st = (x1, y1, z1)

c = (x2, y2, z2)

k = (0, 1, 0)

n =

u =

v = n.u = =

M-1 =

Pc = Pw. M-1 =. M-1 =

=> Pc (xc, yc, zc, 1)

**BÀI TẬP 3:** Xác định điểm Pclip(xclip, yclip, zclip, wclip) trong clipping/projection space

1. *Phép chiếu song song:*

[MTILT] =

Trong đó w, h, d là các thông số liên quan đến cửa sổ nhìn và tọa độ của mặt phẳng chiếu, và (x,y,z) là tọa độ của điểm quan sát. Các thông số này cần được thiết lập trước khi thực hiện phép chiếu.

1. *Phép chiếu xiên:*

[MISO] = [MTILT] =

1. *Phép chiếu trực giao:*

Oxy:

[MZ] =

Oxz:

[MY] =

Oyz:

[MX] =

1. Phép chiếu phối cảnh:

[Mproj] =

-----------------------------------------------