

Fall 2015

# CSCI 420: Computer Graphics



## Exercise 2. Simulate a Roller Coaster



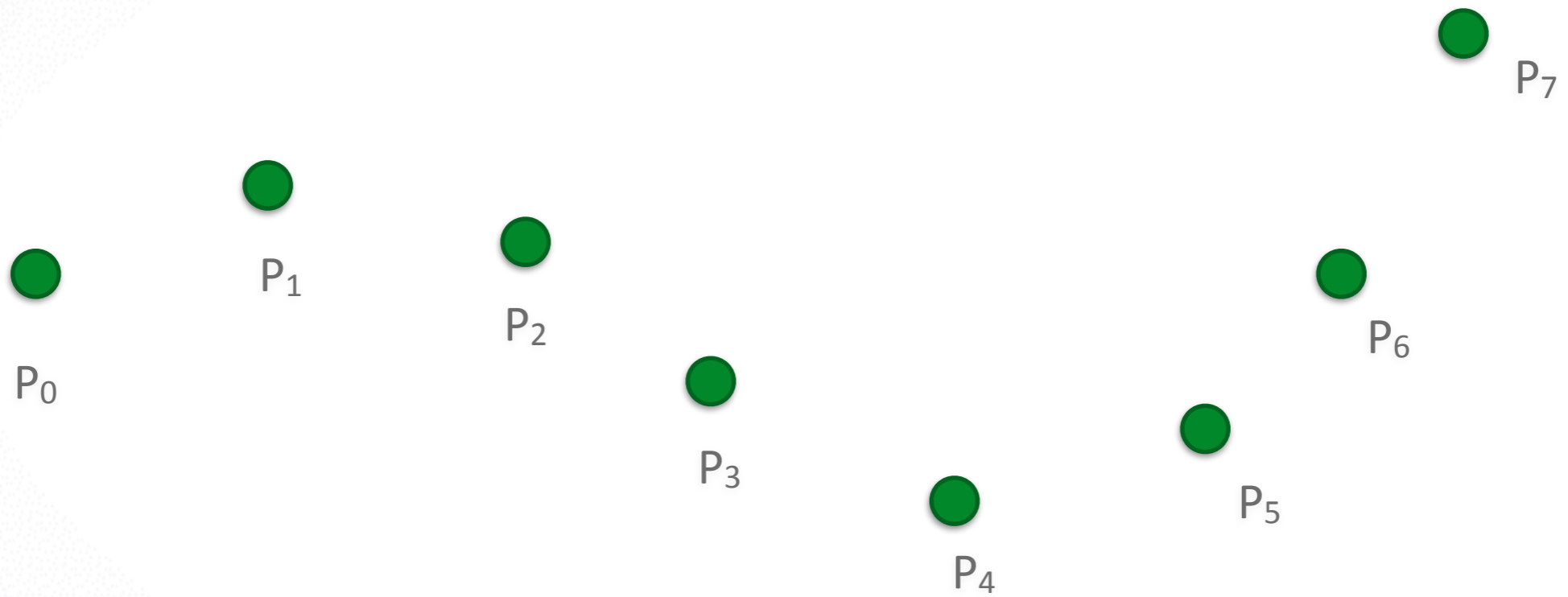
Hao Li

<http://cs420.hao-li.com>

# Level 1: Roller Coaster

- Splines
- Texture maps and their use in OpenGL
- Camera manipulations--the use of transformations to create realistic first-person movement

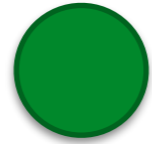
# Level 1: Spline



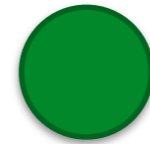
# Level 1: Spline



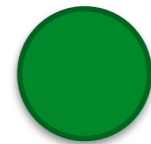
P<sub>0</sub>



P<sub>1</sub>



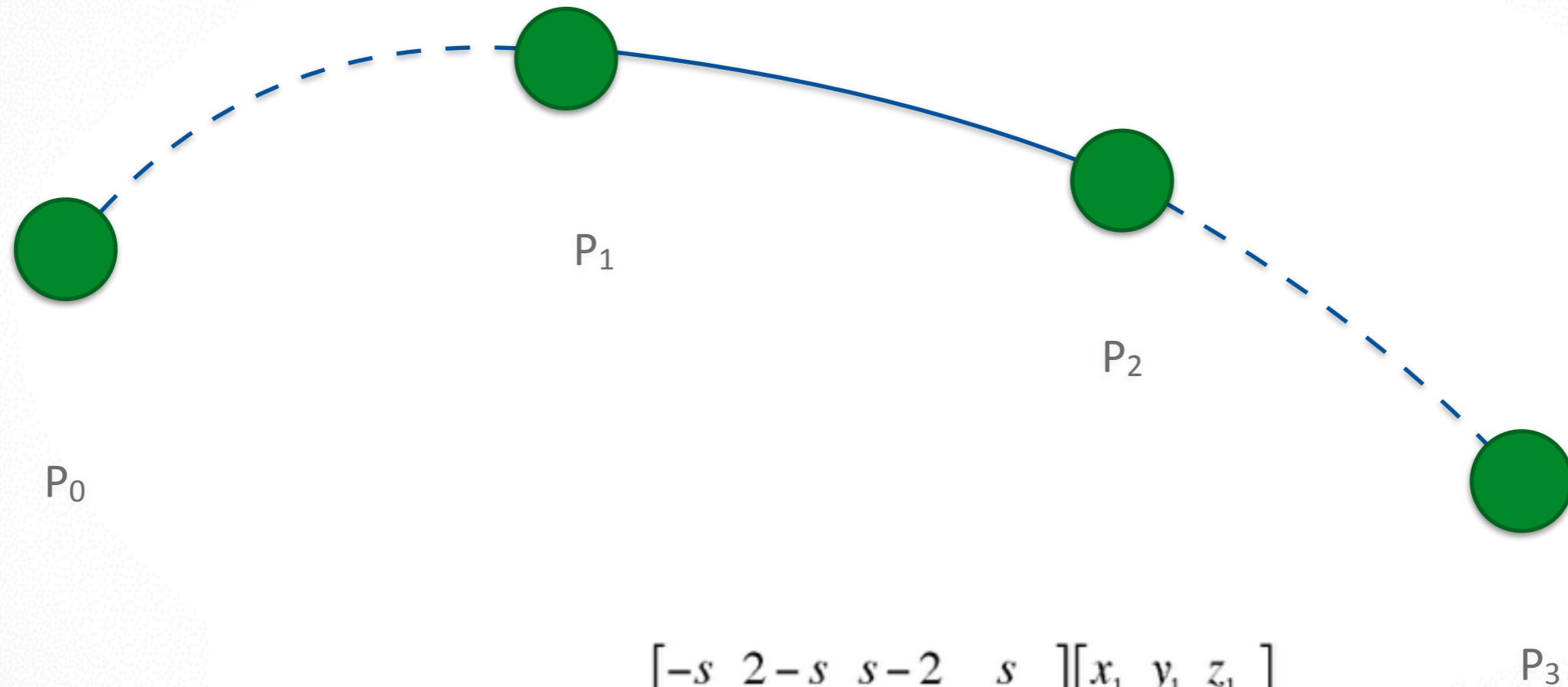
P<sub>2</sub>



P<sub>3</sub>

Catmull-Rom Spline

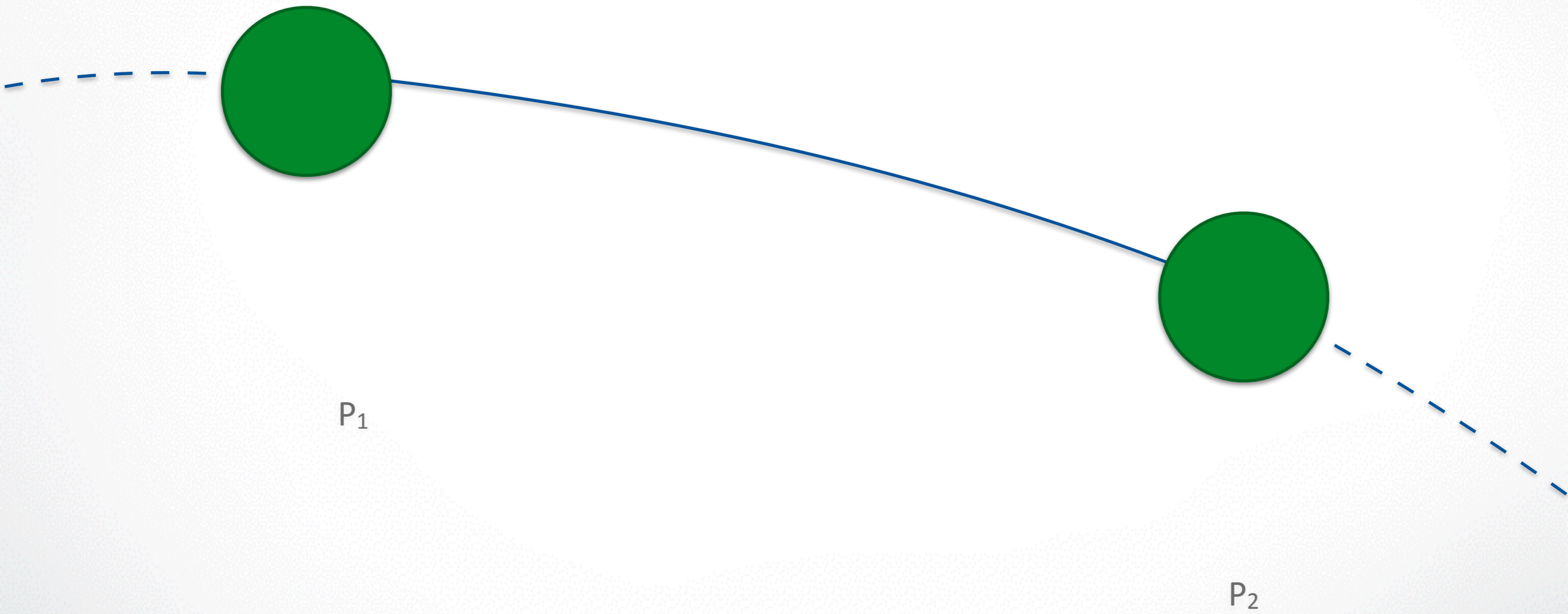
# Level 1: Spline



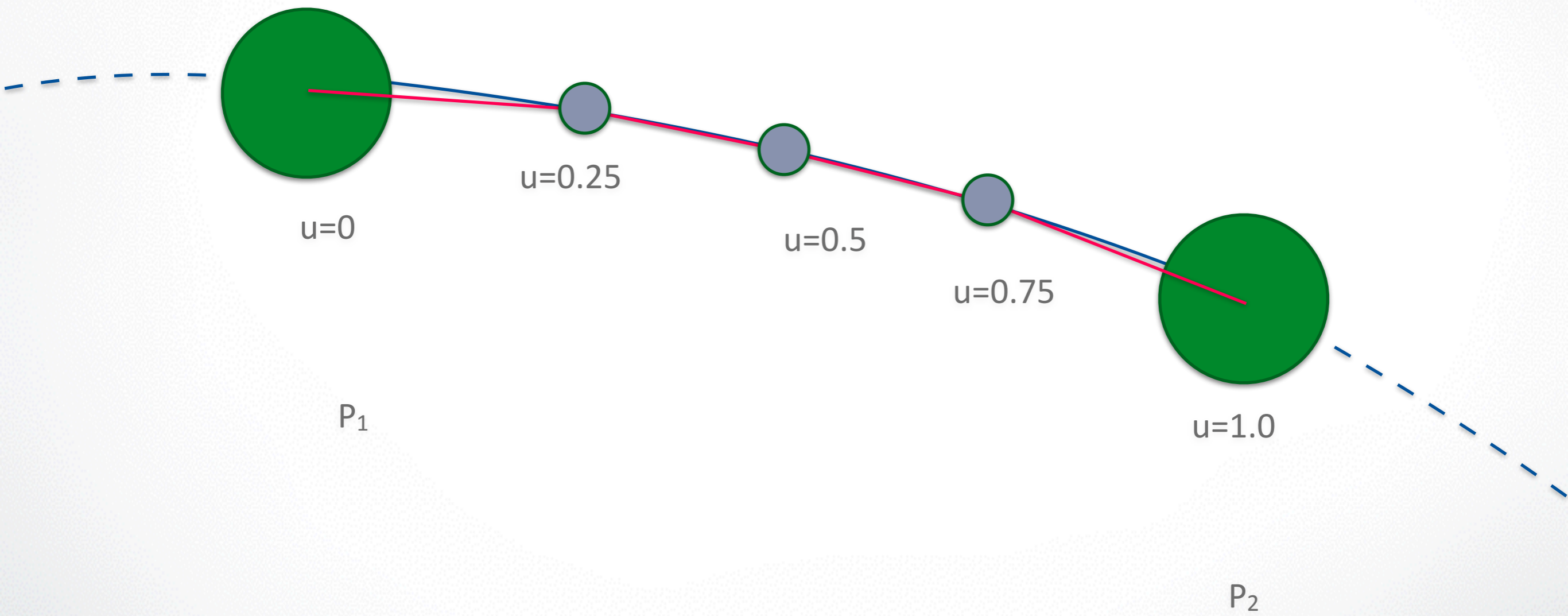
$$[x \ y \ z] = [u^3 \ u^2 \ u \ 1] \begin{bmatrix} -s & 2-s & s-2 & s \\ 2s & s-3 & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{bmatrix}$$

basis                      control matrix

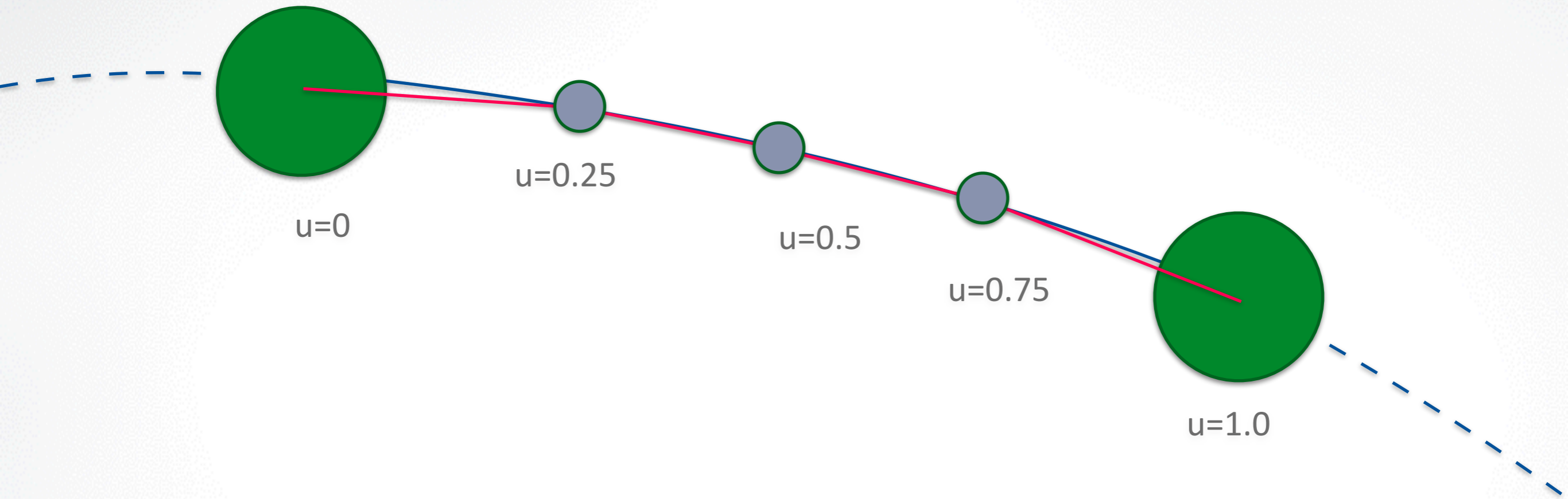
# Level 1: Spline



# Level 1: Spline



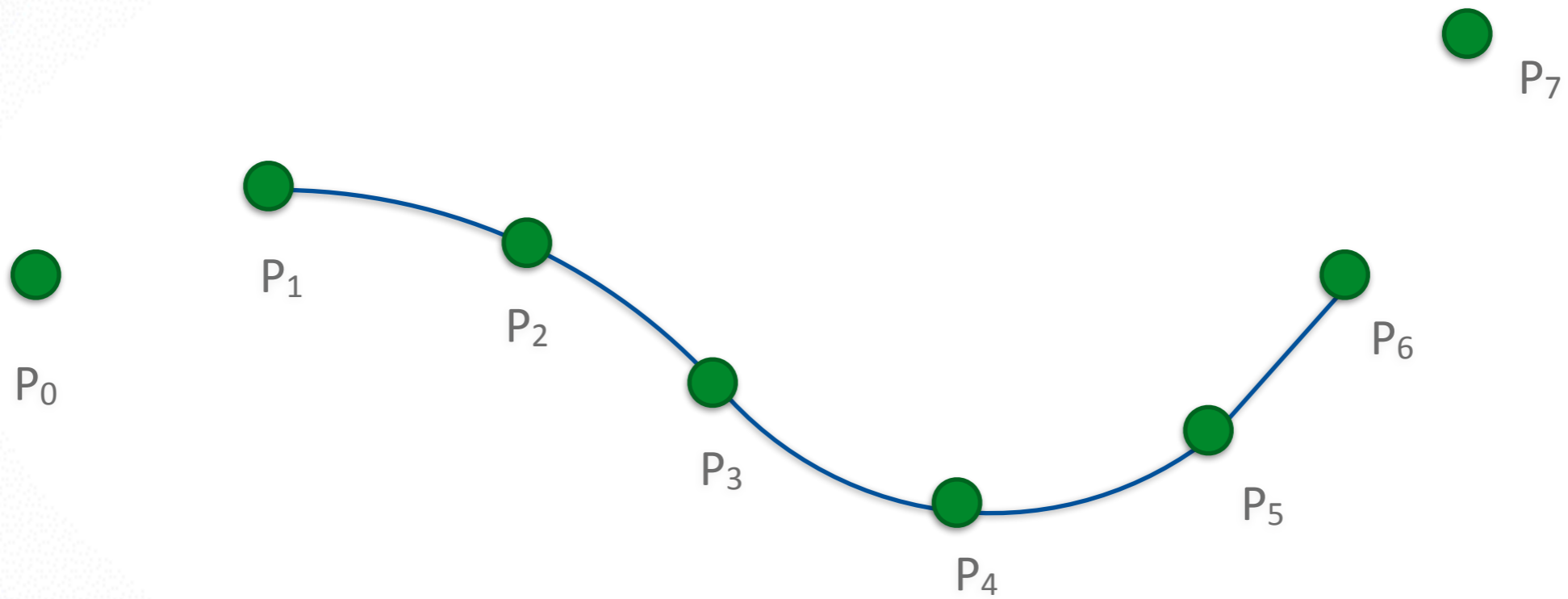
# Level 1: Spline



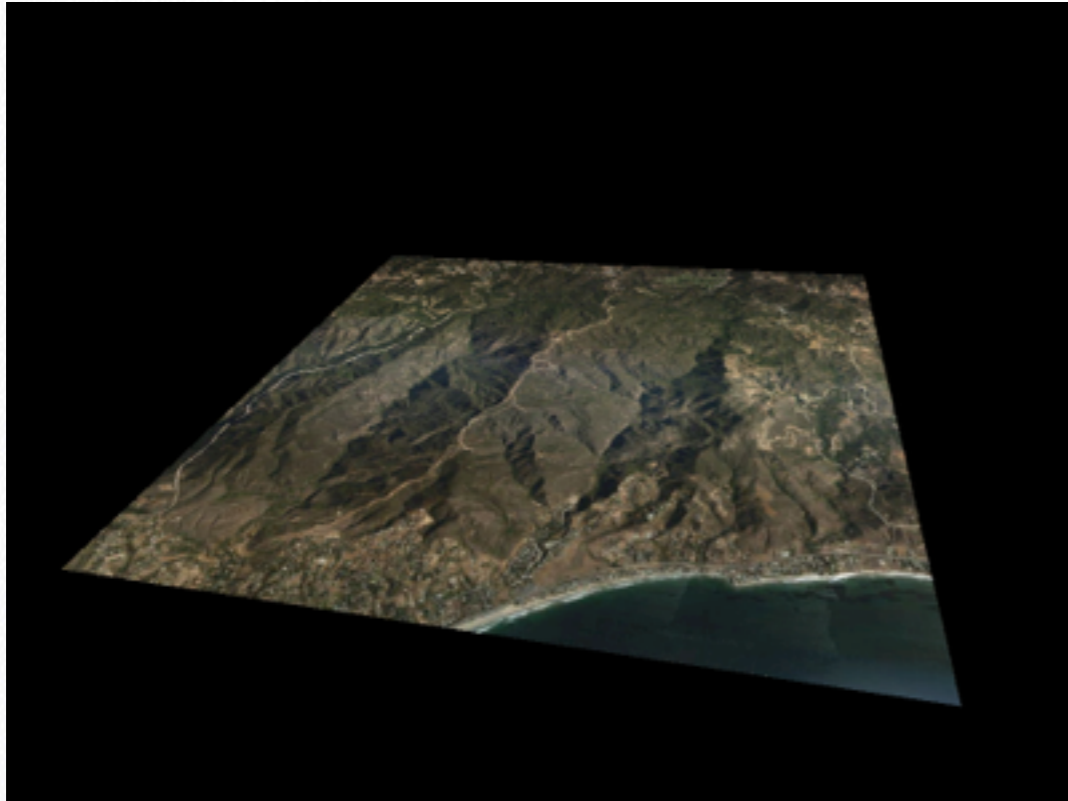
- $u = 0.0, 0.001, 0.002, \dots, 0.999, 1.0$ .
- connect consecutive points with lines



# Level 1: Spline



# Level 2: Ground



Plane



Height Field

# Level 2: Ground



# Level 2: Ground

```
GLuint texture[1];

void texload(int i, char *filename)
{
    Pic* img;
    img = jpeg_read(filename, NULL);
    glBindTexture(GL_TEXTURE_2D, texture[i]);
    glTexImage2D(GL_TEXTURE_2D,
                0,
                GL_RGB,
                img->nx,
                img->ny,
                0,
                GL_RGB,
                GL_UNSIGNED_BYTE,
                &img->pix[0]);
    pic_free(img);
}

void myinit()
{
    glGenTextures(1, texture);
    texload(0, "fileName.jpg");
}
```

# Level 2: Ground

```
void display()
{
    .....

    glEnable(GL_TEXTURE_2D);
    glBindTexture(GL_TEXTURE_2D, texture[0]);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);

    glBegin(GL_POLYGON);
    glTexCoord2f(1.0, 0.0);
    glVertex3f(1.0, 1.0, 1.0);
    glTexCoord2f(0.0, 0.0);
    glVertex3f(-1.0, 1.0, 1.0);
    glTexCoord2f(0.0, 1.0);
    glVertex3f(-1.0, 1.0, -1.0);
    glTexCoord2f(1.0, 1.0);
    glVertex3f(1.0, 1.0, -1.0);
    glEnd();
    glDisable(GL_TEXTURE_2D);

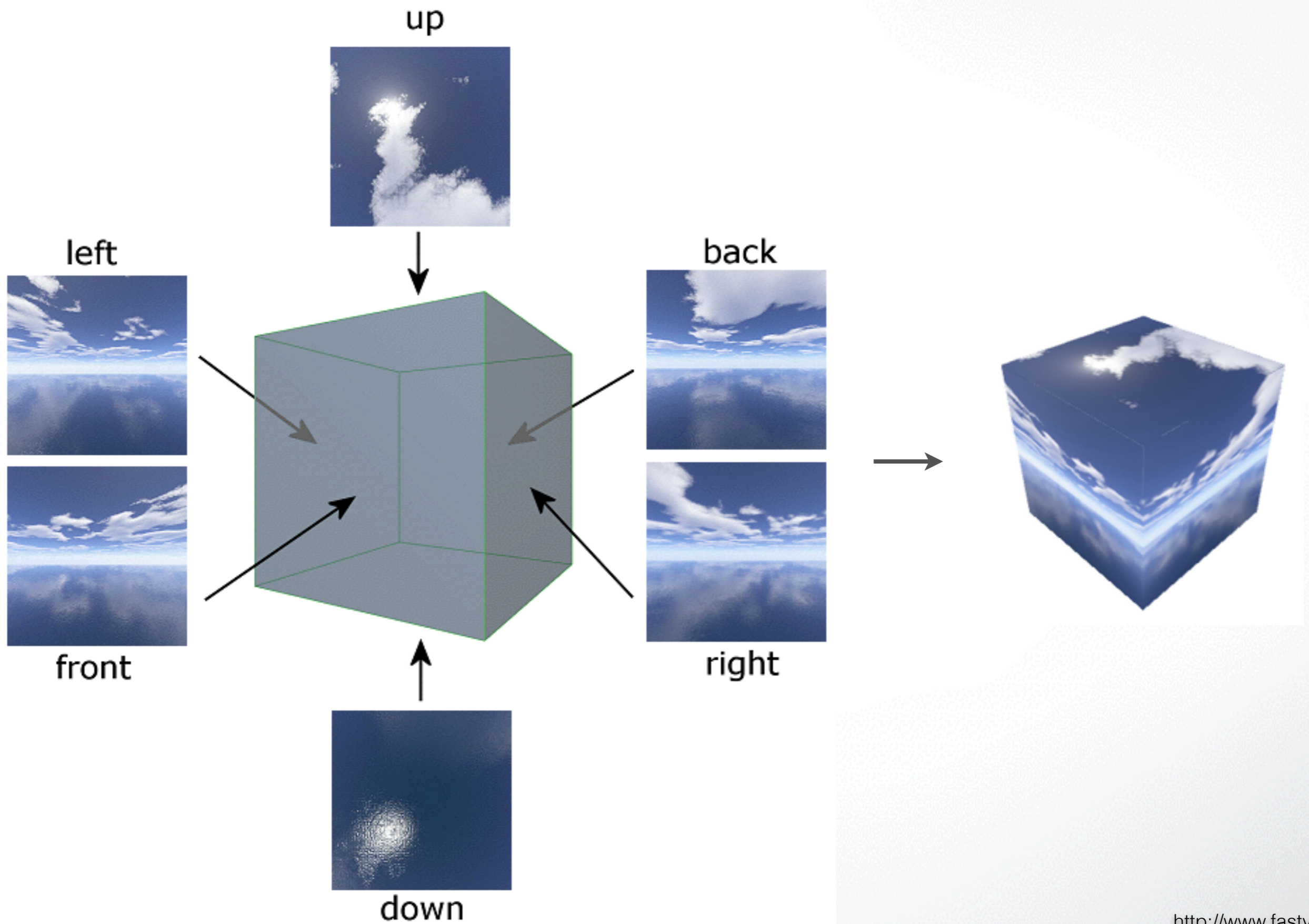
    .....
}
```

# Level 3: Sky

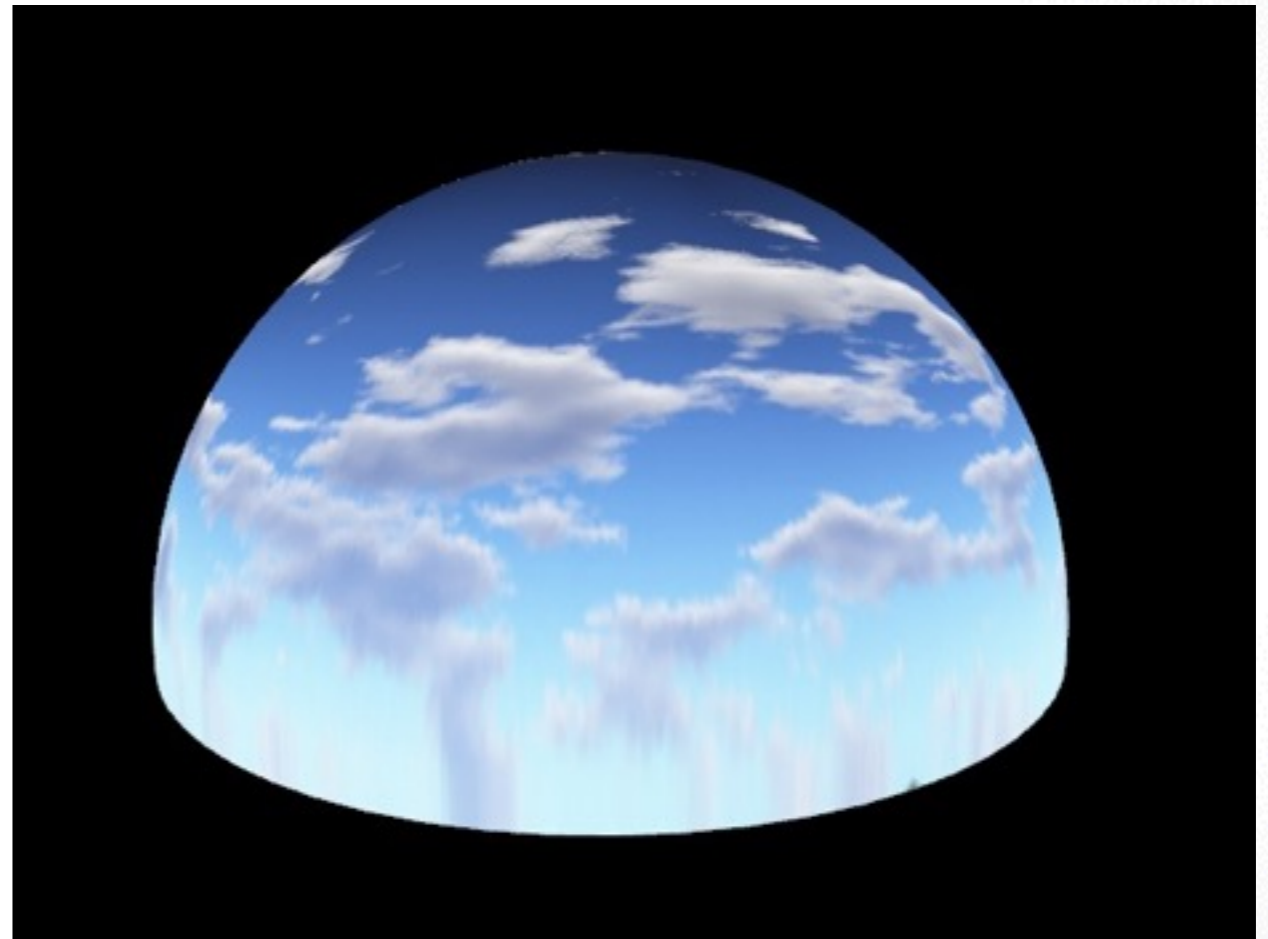


Sky cube texture

# Level 3: Sky



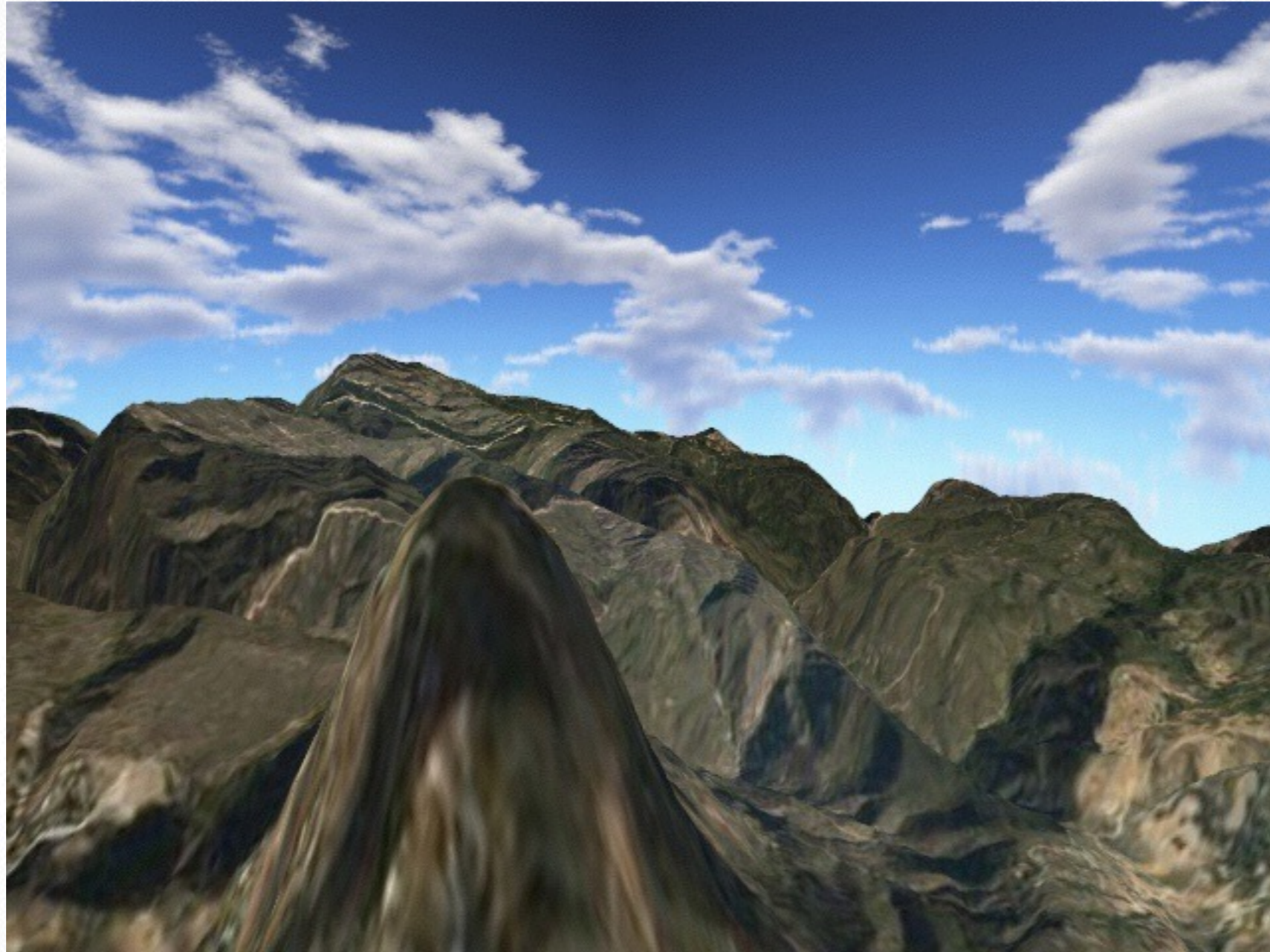
# Level 3: Sky



Sky dome texture

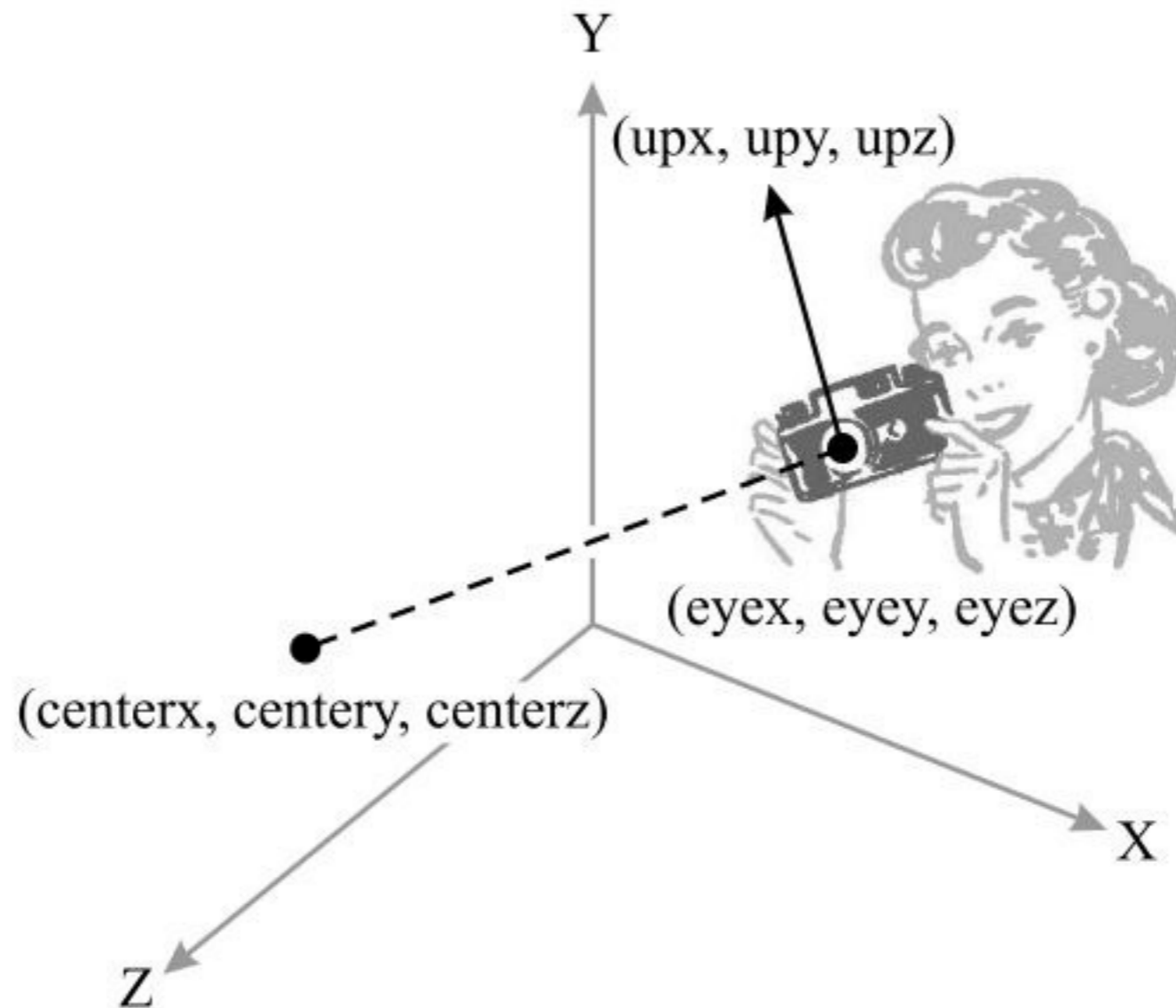


# Level 3: Sky



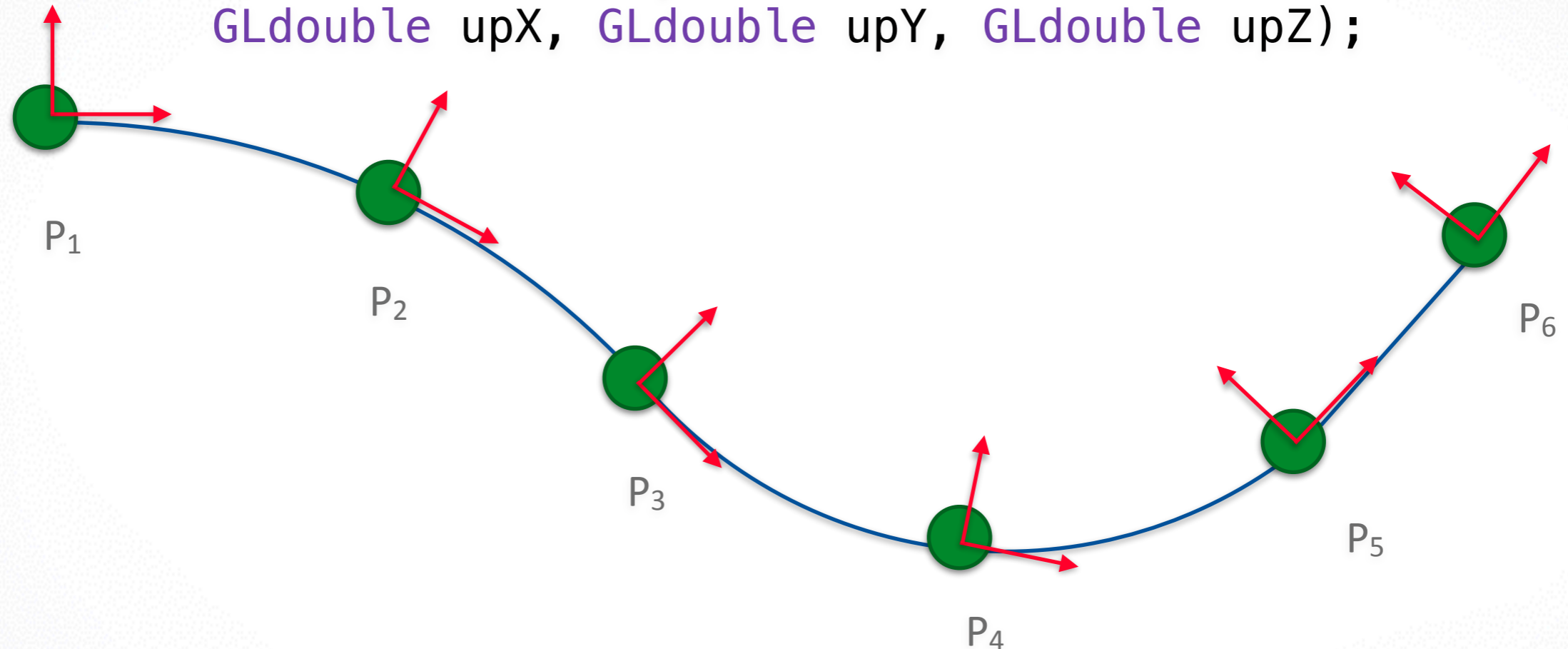
# Level 4: The Ride

```
void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ,  
              GLdouble centerX, GLdouble centerY, GLdouble centerZ,  
              GLdouble upX, GLdouble upY, GLdouble upZ);
```



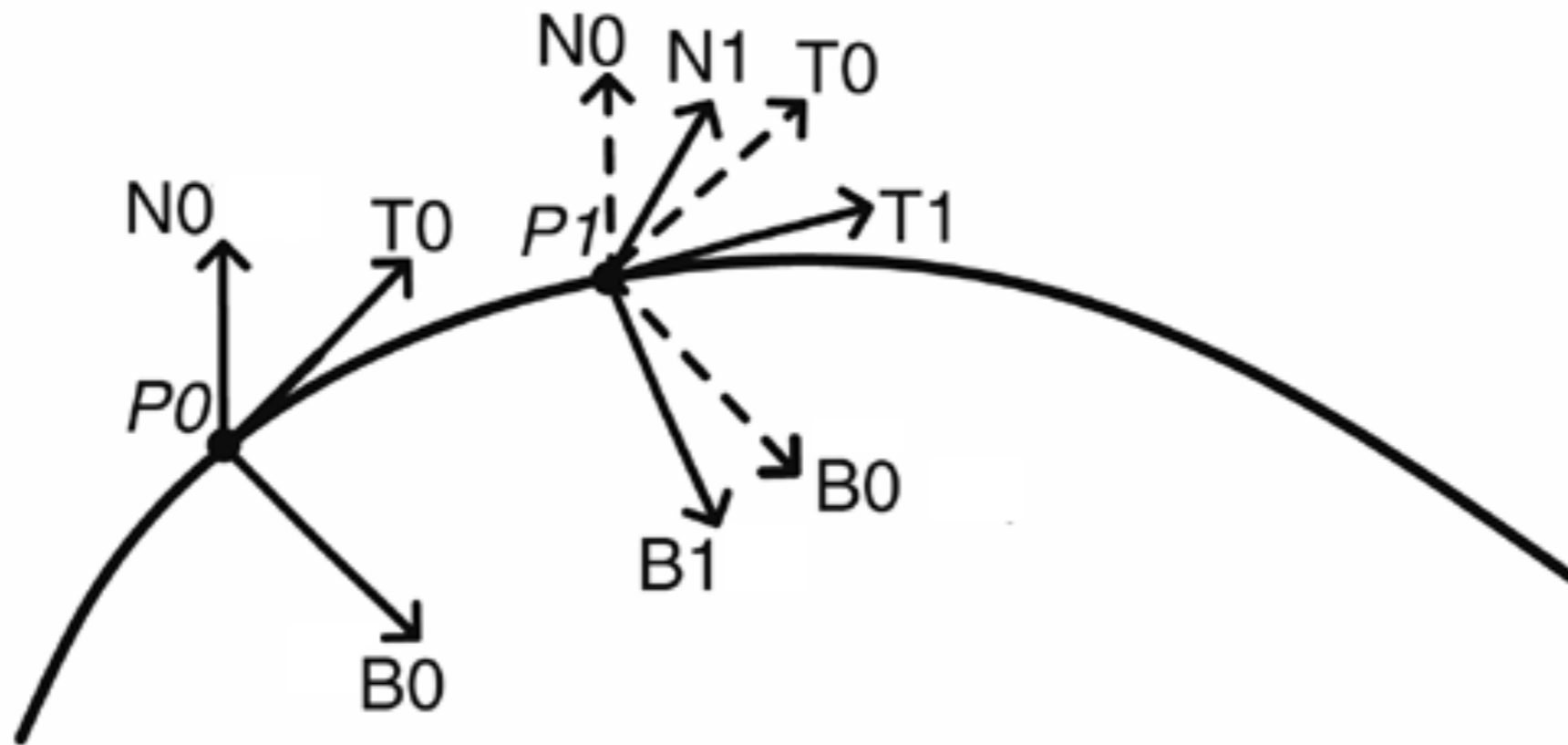
# Level 4: The Ride

```
void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ,  
              GLdouble centerX, GLdouble centerY, GLdouble centerZ,  
              GLdouble upX, GLdouble upY, GLdouble upZ);
```



- eye:  $p(u)$
- tangent (orientation):  $t(u) = \text{unit}(p'(u)) = \text{unit}([3u^2 \ 2u \ 1 \ 0] M C)$ .
- center = eye +  $a$ \*tangent
- How to compute up vector?

# Level 4: The Ride

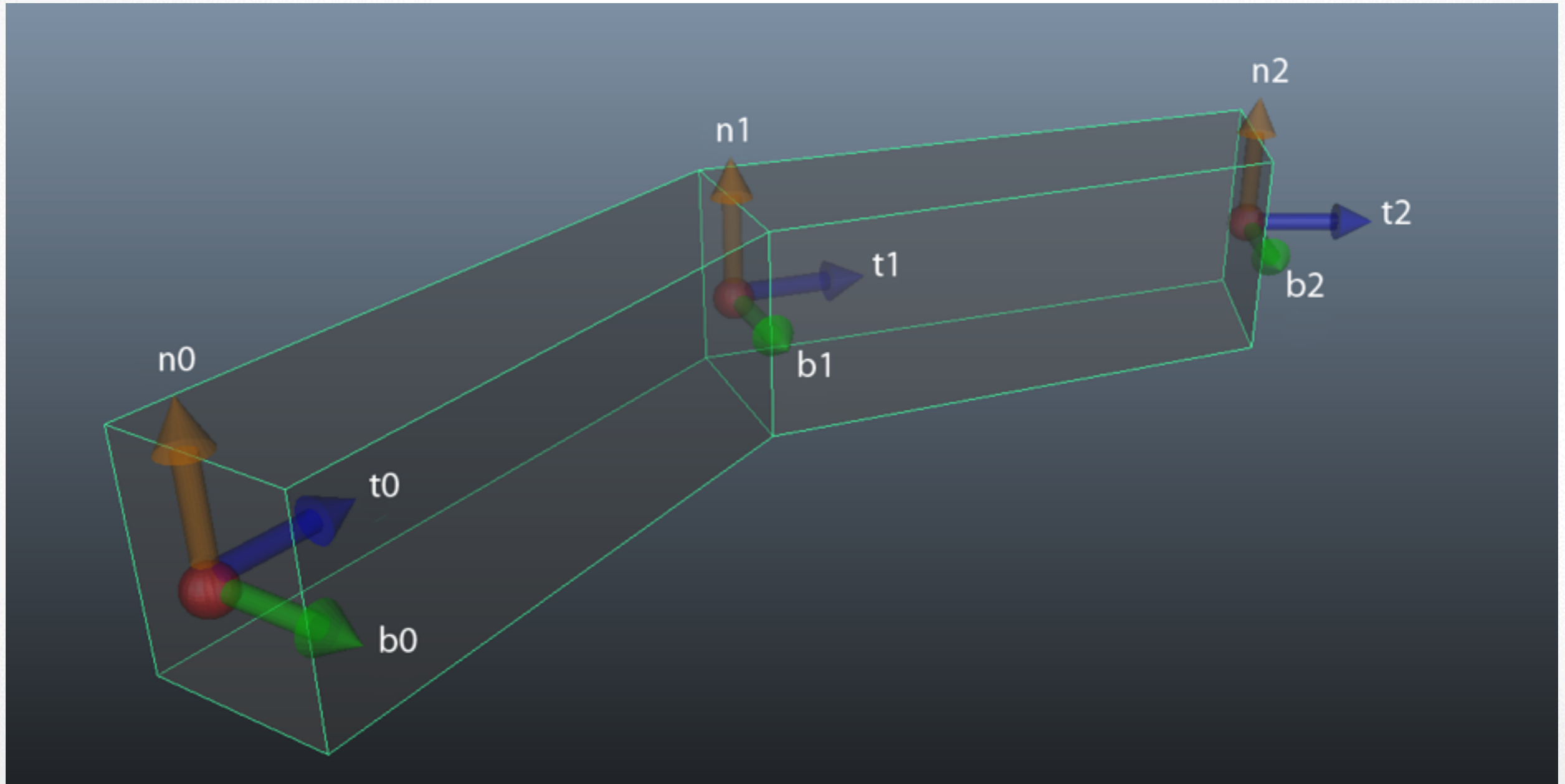


Computing a reference frame from the previous frame

# Level 4: The Ride

- Establish a local coordinate system for each point on the curve  
T, N, B
- Initialization:  $T_0, N_0, B_0$   
 $u = 0 \Rightarrow T_0$   
Pick an arbitrary unit(V).  $N_0 = \text{unit}(T_0 \times V)$  and  $B_0 = \text{unit}(T_0 \times N_0)$ . This guarantees  $T_0, N_0, B_0$  orthonormal to each other.
- Next view:  $T_1, N_1, B_1$   
Move  $u$  ahead, compute  $T_1$  based on new  $u$   
 $N_1 = \text{unit}(B_0 \times T_1)$  and  $B_1 = \text{unit}(T_1 \times N_1)$
- $T_2, N_2, B_2, \dots$
- Update camera “up” vector to be N or B
- Guarantee camera orientation changes continuously

# Level 5: Rail Cross-section



# Demo



# Submission

- Deadline: **Oct 20, 2015 11:59 pm**
- **Start this assignment as soon as you can**
- Upload a .zip compressed file named “Exercise2-YourName.zip” to blackboard
- Include your code with comments
- Include a readme file
- Include JPEG frames or a video



# Contact

- Office Hours on course website
- Emails: [olszewsk@usc.edu](mailto:olszewsk@usc.edu), [lingyu.wei@usc.edu](mailto:lingyu.wei@usc.edu)
- When you sent emails, add “CSCI420” in the title, and please send them to both of us
- Highly recommended to post your questions on blackboard

Enjoy it!

