The View Matrix

Lecture 24

Robb T. Koether

Hampden-Sydney College

Wed, Oct 30, 2019

Outline

Outline

The View Matrix

Definition (The View Matrix)

The view matrix is the matrix that transforms the world coordinates into the eye coordinates.

- The function `lookAt()` creates the view matrix.
- The parameters to `lookAt()` are
  - The eye point `eye`,
  - The look point `look`,
  - The up vector `up`.

The View Matrix

- In earlier versions of OpenGL, the libraries maintained the modelview matrix on the modelview stack.
- The top matrix was the product of the model matrix and the view matrix.
- It was critically important to push the view matrix first, followed by the various model matrices.
- The function `gluLookAt()`, in the `glu` library, created the view matrix and pushed it onto the modelview stack.
- Now all of that is handled by the programmer.

The `lookAt()` Function

The `setView()` Function

```
void setView()
{
    GLfloat yaw_r = yaw * DEG_TO_RAD;
    GLfloat pitch_r = pitch * DEG_TO_RAD;

    eye[0] = look[0] + eye_dist * sinf(yaw_r) * cosf(pitch_r);
    eye[1] = look[1] + eye_dist * sinf(pitch_r);
    eye[2] = look[2] + eye_dist * cosf(yaw_r) * cosf(pitch_r);

    view = lookAt(eye, look, up);

    glUniformMatrix4fv(view_loc, 1, GL_FALSE, view);
    glUniform3fv(eye_loc, 1, eye);
}
```

The lookAt() Function

The setView() Function

```
void setView()
{
    GLfloat yaw_r = yaw * DEG_TO_RAD;
    GLfloat pitch_r = pitch * DEG_TO_RAD;

    eye = look + vec3(eye_dist * sinf(yaw_r) * cosf(pitch_r),
                      eye_dist * sinf(pitch_r),
                      eye_dist * cosf(yaw_r) * cosf(pitch_r));

    view = lookAt(eye, look, up);

    glUniformMatrix4fv(view_loc, 1, GL_FALSE, view);
    glUniform3fv(eye_loc, 1, eye);
}
```

Outline

The **uvw** Eye Coordinate System

- In the eye coordinate system, the "eye" is
  - Located at the origin
  - Looking in the negative z-direction.

The **uvw** Eye Coordinate System

- In the eye coordinate system, the "eye" is
  - Located at the origin
  - Looking in the negative z-direction.
- The view matrix actually moves the entire scene in front of the eye, which is always at the origin, always looking down the negative z-axis.

The **uvw** Eye Coordinate System

- In the eye coordinate system, the "eye" is
  - Located at the origin
  - Looking in the negative z-direction.
- The view matrix actually moves the entire scene in front of the eye, which is always at the origin, always looking down the negative z-axis.
- But it is more intuitive to think of the view matrix as moving the eye from the origin to the eye position.

The **uvw** Eye Coordinate System

- In the eye coordinate system, the "eye" is
  - Located at the origin
  - Looking in the negative z-direction.
- The view matrix actually moves the entire scene in front of the eye, which is always at the origin, always looking down the negative z-axis.
- But it is more intuitive to think of the view matrix as moving the eye from the origin to the eye position.
- The two transformations are inverses of each other.

The **uvw** Eye Coordinate System

- Let the vectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ be unit vectors in a RHS, expressed in world coordinates, located at the eye position, and oriented so that the eye is looking along $-\mathbf{w}$ towards the look point.

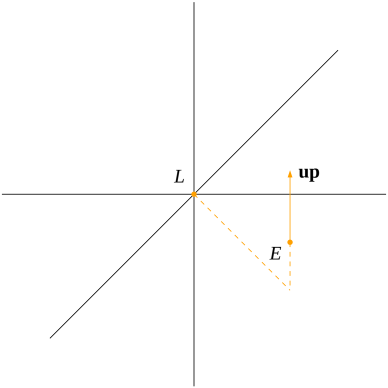
The **uvw** Eye Coordinate System

- Let the vectors **u**, **v**, and **w** be unit vectors in a RHS, expressed in world coordinates, located at the eye position, and oriented so that the eye is looking along $-\mathbf{w}$ towards the look point.
- We will calculate **u**, **v**, and **w** from `eye` (E), `look` (L), and `up` (**up**).

The **uvw** Eye Coordinate System

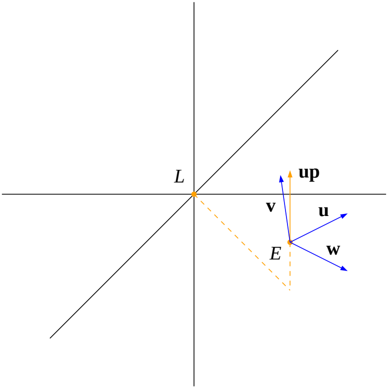
- Let the vectors **u**, **v**, and **w** be unit vectors in a RHS, expressed in world coordinates, located at the eye position, and oriented so that the eye is looking along $-\mathbf{w}$ towards the look point.
- We will calculate **u**, **v**, and **w** from `eye` (E), `look` (L), and `up` (**up**).
- These vectors are key to defining the view matrix.

The **uvn** Eye Coordinate System



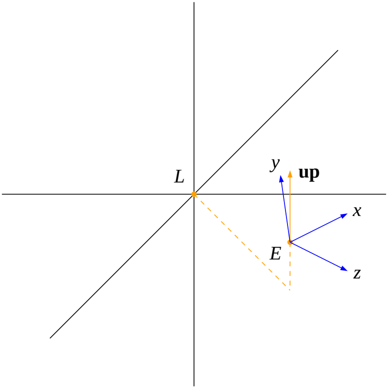
- Given the eye point E, the look point L, and the up vector **up**.

The **uvn** Eye Coordinate System



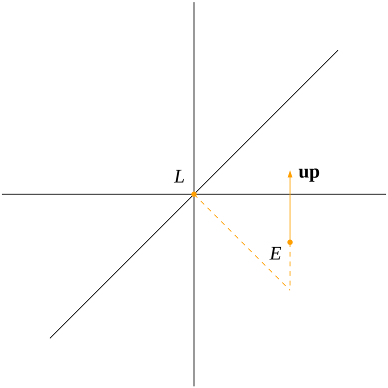
- We need to determine the basic unit vectors **u**, **v**, and **w** of the eye coordinate system.

The **uvn** Eye Coordinate System



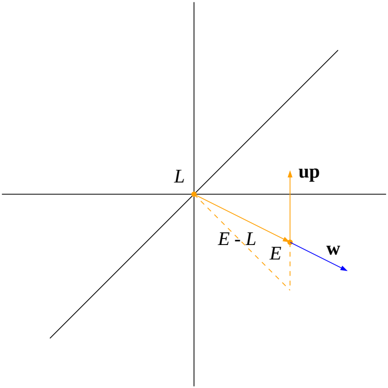
- They correspond to the *x*-, *y*-, and *z*-axes of that system.

The **uvn** Eye Coordinate System



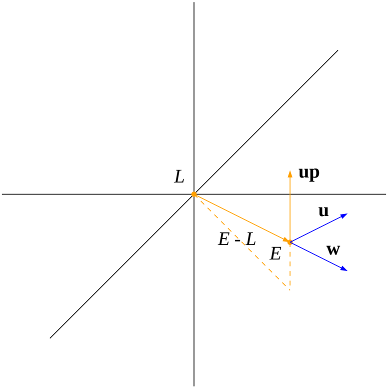
- Let E be the eye position, L the look point, and **up** the up vector.

The **uvn** Eye Coordinate System



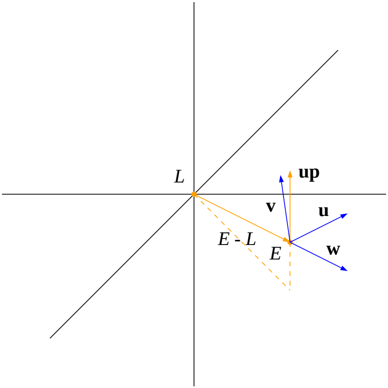
- Define $\mathbf{w}' = E - L$ and $\mathbf{w} = \dfrac{\mathbf{w}'}{|\mathbf{w}'|}$.

The **uvn** Eye Coordinate System



- The vector **u** must be perpendicular to **w** and **up**.
- Define $\mathbf{u}' = \mathbf{up} \times \mathbf{w}$ and $\mathbf{u} = \dfrac{\mathbf{u}'}{|\mathbf{u}'|}$.

The **uvn** Eye Coordinate System



- We cannot assume that **up** is perpendicular to **w**.
- Therefore, let **v** be the unit vector $\mathbf{v} = \mathbf{w} \times \mathbf{u}$.

Outline

The View Matrix

- Let the vectors **i**, **j**, and **k** be the basic unit vectors in the eye coordinate system.
- The transformation to the eye coordinate system is determined by the world vectors **u**, **v**, and **w**.
- The view matrix `view` must transform **u**, **v**, **w** into **i**, **j**, **k**:

$$\mathtt{view} \cdot \mathbf{u} = \mathbf{i}$$
$$\mathtt{view} \cdot \mathbf{v} = \mathbf{j}$$
$$\mathtt{view} \cdot \mathbf{w} = \mathbf{k}$$

The View Matrix

- That is,

$$\texttt{view} \cdot \mathbf{u} = \begin{pmatrix} v_{11} & v_{12} & v_{13} & a \\ v_{21} & v_{22} & v_{23} & b \\ v_{31} & v_{32} & v_{33} & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ u_z \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{i},$$

- That is,

$$
\begin{aligned}
(v_{11}, v_{12}, v_{13}) \cdot \mathbf{u} &= 1, \\
(v_{21}, v_{22}, v_{23}) \cdot \mathbf{u} &= 0, \\
(v_{31}, v_{32}, v_{33}) \cdot \mathbf{u} &= 0.
\end{aligned}
$$

- That is,

$$(v_{11}, v_{12}, v_{13}) \cdot \mathbf{u} = 1,$$
$$(v_{21}, v_{22}, v_{23}) \cdot \mathbf{u} = 0,$$
$$(v_{31}, v_{32}, v_{33}) \cdot \mathbf{u} = 0.$$

- Recall that

$$\mathbf{u} \cdot \mathbf{u} = 1,$$
$$\mathbf{v} \cdot \mathbf{u} = 0,$$
$$\mathbf{w} \cdot \mathbf{u} = 0.$$

The View Matrix

- And,

$$\mathtt{view} \cdot \mathbf{v} = \begin{pmatrix} v_{11} & v_{12} & v_{13} & a \\ v_{21} & v_{22} & v_{23} & b \\ v_{31} & v_{32} & v_{33} & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \mathbf{j},$$

- That is,

$$\begin{aligned}
(v_{11}, v_{12}, v_{13}) \cdot \mathbf{v} &= 0, \\
(v_{21}, v_{22}, v_{23}) \cdot \mathbf{v} &= 1, \\
(v_{31}, v_{32}, v_{33}) \cdot \mathbf{v} &= 0.
\end{aligned}$$

- That is,

$$(v_{11}, v_{12}, v_{13}) \cdot \mathbf{v} = 0,$$
$$(v_{21}, v_{22}, v_{23}) \cdot \mathbf{v} = 1,$$
$$(v_{31}, v_{32}, v_{33}) \cdot \mathbf{v} = 0.$$

- Recall that

$$\mathbf{u} \cdot \mathbf{v} = 0,$$
$$\mathbf{v} \cdot \mathbf{v} = 1,$$
$$\mathbf{w} \cdot \mathbf{v} = 0.$$

- And,

$$\texttt{view} \cdot \mathbf{w} = \begin{pmatrix} v_{11} & v_{12} & v_{13} & a \\ v_{21} & v_{22} & v_{23} & b \\ v_{31} & v_{32} & v_{33} & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w_x \\ w_y \\ w_z \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \mathbf{k},$$

- That is,

$$
\begin{aligned}
(v_{11}, v_{12}, v_{13}) \cdot \mathbf{w} &= 0, \\
(v_{21}, v_{22}, v_{23}) \cdot \mathbf{w} &= 0, \\
(v_{31}, v_{32}, v_{33}) \cdot \mathbf{w} &= 1.
\end{aligned}
$$

- That is,

$$\begin{aligned}(v_{11}, v_{12}, v_{13}) \cdot \mathbf{w} &= 0,\\ (v_{21}, v_{22}, v_{23}) \cdot \mathbf{w} &= 0,\\ (v_{31}, v_{32}, v_{33}) \cdot \mathbf{w} &= 1.\end{aligned}$$

- Recall that

$$\begin{aligned}\mathbf{u} \cdot \mathbf{w} &= 0,\\ \mathbf{v} \cdot \mathbf{w} &= 0,\\ \mathbf{w} \cdot \mathbf{w} &= 1.\end{aligned}$$

The View Matrix

- Therefore, the view matrix will be of the form

$$\mathbf{V} = \begin{pmatrix} u_x & u_y & u_z & a \\ v_x & v_y & v_z & b \\ w_x & w_y & w_z & c \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

with a, b, and c to be determined (the translation).

The View Matrix

- To determine a, b, and c, use that fact that $\mathbf{V}$ also transforms E to the origin:
$$\mathbf{V}E = O.$$
- That is,
$$\texttt{view} \cdot E = \begin{pmatrix} u_x & u_y & u_z & a \\ v_x & v_y & v_z & b \\ w_x & w_y & w_z & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} e_x \\ e_y \\ e_z \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = O.$$

The View Matrix

- When we multiply, we get

$$\begin{pmatrix} u_x & u_y & u_z & a \\ v_x & v_y & v_z & b \\ w_x & w_y & w_z & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} e_x \\ e_y \\ e_z \\ 1 \end{pmatrix} = \begin{pmatrix} u_x e_x + u_y e_y + u_z e_z + a \\ v_x e_x + v_y e_y + v_z e_z + b \\ w_x e_x + w_y e_y + w_z e_z + c \\ 1 \end{pmatrix}.$$

The View Matrix

- Thus,

$$\begin{aligned} a &= -(u_x e_x + u_y e_y + u_z e_z) = -\mathbf{u} \cdot \mathbf{e} \\ b &= -(v_x e_x + v_y e_y + v_z e_z) = -\mathbf{v} \cdot \mathbf{e} \\ c &= -(w_x e_x + w_y e_y + w_z e_z) = -\mathbf{w} \cdot \mathbf{e} \end{aligned}$$

  where $\mathbf{e} = E - O$.

The View Matrix

- Therefore, the matrix created by `lookAt()` is

$$\texttt{view} = \begin{pmatrix} u_x & u_y & u_z & -\mathbf{u}\cdot\mathbf{e} \\ v_x & v_y & v_z & -\mathbf{v}\cdot\mathbf{e} \\ w_x & w_y & w_z & -\mathbf{w}\cdot\mathbf{e} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The View Matrix

- Verify that `view` transforms the points

$$
\begin{aligned}
E &\to (0,0,0)\\
E + \mathbf{u} &\to (1,0,0)\\
E + \mathbf{v} &\to (0,1,0)\\
E + \mathbf{w} &\to (0,0,1)
\end{aligned}
$$

The `lookAt()` Function in `vmath.h`

The `lookAt()` Function in `vmath.h`

```
mat4 lookAt(const vec3& eye, const vec3& look,
    const vec3& up)
{
    vec3 w = normalize(look - eye);
    vec3 upN = normalize(up);
    vec3 u = normalize(cross(w, upN));
    vec3 v = cross(u, w);
    mat4 M = mat4(
        vec4(u[0], v[0], -w[0], 0),
        vec4(u[1], v[1], -w[1], 0),
        vec4(u[2], v[2], -w[2], 0),
        vec4(0, 0, 0, 1));

    return M * translate(-eye);
}
```

Outline

Finding the View Matrix

Example (Finding the View Matrix)

- Let $E = (16, 15, 12)$, $L = (0, 0, 0)$, and **up** $= (0, 1, 0)$.
- Then

Finding the View Matrix

Example (Finding the View Matrix)

- Let $E = (16, 15, 12)$, $L = (0, 0, 0)$, and $\mathbf{up} = (0, 1, 0)$.
- Then

$$\mathbf{w}' = E - L = (16, 15, 12)$$

Finding the View Matrix

Example (Finding the View Matrix)

- Let $E = (16, 15, 12)$, $L = (0, 0, 0)$, and **up** $= (0, 1, 0)$.
- Then

$$\mathbf{w}' = E - L = (16, 15, 12)$$

$$\mathbf{w} = \frac{\mathbf{w}'}{|\mathbf{w}'|} = \tfrac{1}{25}(16, 15, 12)$$

Finding the View Matrix

Example (Finding the View Matrix)

- Let $E = (16, 15, 12)$, $L = (0, 0, 0)$, and $\mathbf{up} = (0, 1, 0)$.
- Then

$$\mathbf{w}' = E - L = (16, 15, 12)$$

$$\mathbf{w} = \frac{\mathbf{w}'}{|\mathbf{w}'|} = \tfrac{1}{25}(16, 15, 12) = (0.64, 0.60, 0.48)$$

Finding the View Matrix

Example (Finding the View Matrix)

- Let $E = (16, 15, 12)$, $L = (0, 0, 0)$, and $\mathbf{up} = (0, 1, 0)$.
- Then

$$\mathbf{w}' = E - L = (16, 15, 12)$$

$$\mathbf{w} = \frac{\mathbf{w}'}{|\mathbf{w}'|} = \tfrac{1}{25}(16, 15, 12) = (0.64, 0.60, 0.48)$$

$$\mathbf{u}' = \mathbf{up} \times \mathbf{w} = (0.48, 0.00, -0.64)$$

Finding the View Matrix

Example (Finding the View Matrix)

- Let $E = (16, 15, 12)$, $L = (0, 0, 0)$, and $\mathbf{up} = (0, 1, 0)$.
- Then

$$\mathbf{w}' = E - L = (16, 15, 12)$$

$$\mathbf{w} = \frac{\mathbf{w}'}{|\mathbf{w}'|} = \tfrac{1}{25}(16, 15, 12) = (0.64, 0.60, 0.48)$$

$$\mathbf{u}' = \mathbf{up} \times \mathbf{w} = (0.48, 0.00, -0.64)$$

$$\mathbf{u} = \frac{\mathbf{u}'}{|\mathbf{u}'|} = \tfrac{1}{0.80}(0.48, 0.00, -0.64)$$

Finding the View Matrix

Example (Finding the View Matrix)

- Let $E = (16, 15, 12)$, $L = (0, 0, 0)$, and $\mathbf{up} = (0, 1, 0)$.
- Then

$$\mathbf{w}' = E - L = (16, 15, 12)$$
$$\mathbf{w} = \frac{\mathbf{w}'}{|\mathbf{w}'|} = \tfrac{1}{25}(16, 15, 12) = (0.64, 0.60, 0.48)$$
$$\mathbf{u}' = \mathbf{up} \times \mathbf{w} = (0.48, 0.00, -0.64)$$
$$\mathbf{u} = \frac{\mathbf{u}'}{|\mathbf{u}'|} = \tfrac{1}{0.80}(0.48, 0.00, -0.64) = (0.60, 0.00, -0.80)$$

Finding the View Matrix

Example (Finding the View Matrix)

- Let $E = (16, 15, 12)$, $L = (0, 0, 0)$, and $\mathbf{up} = (0, 1, 0)$.
- Then

$$\begin{aligned}
\mathbf{w}' &= E - L = (16, 15, 12) \\
\mathbf{w} &= \frac{\mathbf{w}'}{|\mathbf{w}'|} = \tfrac{1}{25}(16, 15, 12) = (0.64, 0.60, 0.48) \\
\mathbf{u}' &= \mathbf{up} \times \mathbf{w} = (0.48, 0.00, -0.64) \\
\mathbf{u} &= \frac{\mathbf{u}'}{|\mathbf{u}'|} = \tfrac{1}{0.80}(0.48, 0.00, -0.64) = (0.60, 0.00, -0.80) \\
\mathbf{v} &= \mathbf{w} \times \mathbf{u} = (-0.48, 0.80, -0.36)
\end{aligned}$$

Example (Finding the View Matrix)

- To summarize,

$$\mathbf{u} = (+0.60, +0.00, -0.80)$$
$$\mathbf{v} = (-0.48, +0.80, -0.36)$$
$$\mathbf{w} = (+0.64, +0.60, +0.48)$$

Finding the View Matrix

Example (Finding the View Matrix)

- Also

$$\mathbf{e} = E - O = (16, 15, 12).$$

- So

$$\begin{aligned} \mathbf{e} \cdot \mathbf{u} &= 0 \\ \mathbf{e} \cdot \mathbf{v} &= 0 \\ \mathbf{e} \cdot \mathbf{w} &= 25 \end{aligned}$$

Finding the View Matrix

Example (Finding the View Matrix)

- Therefore, the view matrix is

$$\mathbf{V} = \begin{pmatrix} +0.60 & +0.00 & -0.80 & 0 \\ -0.48 & +0.80 & -0.36 & 0 \\ +0.64 & +0.60 & +0.48 & -25 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Outline

Homework

Homework

- The Red Book, p. 220.
- See Transformation Matrix.
- See Camera Transformation.