Projective Geometry Redux & Image Based Rendering

By Abe Davis
Announcements

• Project 4 is released
  • Due Friday, April 17 by 11:59pm.
  • The project will be done in groups of two, with groups defaulting to the Project 3 groups (though groups can be changed on CMSX)

• New Grading Policy
  • Check email for more information
Today’s Lecture

• Homogeneous Coordinates and Geometry in n-dimensions
  • New slides, following my own derivations, intended to help with confusion I’ve noticed in the first part of the course
  • Mostly re-derives stuff you know, but hopefully with stronger motivation, rigor, and intuition

• Image Based Rendering and Light Fields
  • Not in previous versions of the course, but an active area of work in computer vision with many applications (e.g., AR/VR, film special effects, etc.)
Part 1: Building a Geometry of Points & Views

By Abe Davis

Or “Let’s derive homogeneous coordinates from scratch!”
Motivation

- We may observe geometry in different ways
  - e.g., different views or cameras
- How do we separate our description of geometry from our choice of reference frame?

\[
p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}
\]
Motivation

• We may observe geometry in different ways
  • e.g., different views or cameras
• How do we separate our description of geometry from our choice of reference frame?

\[ p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \]

Multiplying a point by a constant
Motivation

- We may observe geometry in different ways
  - e.g., different views or cameras
- How do we separate our description of geometry from our choice of reference frame?

\[ p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \]

Adding two points
Motivation

- We may observe geometry in different ways
  - e.g., different views or cameras
- How do we separate our description of geometry from our choice of reference frame?

\[
\mathbf{p} = \begin{bmatrix}
p_x \\
p_y \\
p_z
\end{bmatrix}
\]

**Problem:**
Standard representation treats points like vectors from the origin

*Points and vectors are NOT the same thing!*
Relating Points and Vectors

- A point is a unique location
- A vector is the difference between two locations
- We can add vectors to points and to other vectors
- We can scale vectors
Homogeneous Values

- A point is a unique location
- A vector is the difference between two locations
- We can add vectors to points and to other vectors
- We can scale vectors

Homogeneous Points and Vectors:

\[
p_a = \begin{bmatrix} p_a \\ 1 \end{bmatrix} \quad p_b = \begin{bmatrix} p_b \\ 1 \end{bmatrix} \quad v = \begin{bmatrix} v \\ 0 \end{bmatrix}
\]

Set an extra value to 1 for points
And to 0 for vectors

For now, let’s only consider homogeneous values that are 0 or 1
Homogeneous Values

- A point is a unique location
- A vector is the difference between two locations
- We can add vectors to points and to other vectors
- We can scale vectors

Homogeneous Points and Vectors:

\[
\mathbf{p}_a = \begin{bmatrix} p_a \\ 1 \end{bmatrix} \quad \mathbf{p}_b = \begin{bmatrix} p_b \\ 1 \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} v \\ 0 \end{bmatrix}
\]

Set an extra value to 1 for points
And to 0 for vectors

For now, let's only consider homogeneous values that are 0 or 1

- point – point = vector
\[
\begin{bmatrix} p_b \\ 1 \end{bmatrix} - \begin{bmatrix} p_a \\ 1 \end{bmatrix} = \begin{bmatrix} p_b - p_a \\ 0 \end{bmatrix}
\]

- point + vector = point
\[
\begin{bmatrix} p \\ 1 \end{bmatrix} + \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} p + v \\ 1 \end{bmatrix}
\]

- vector + vector = vector
\[
\begin{bmatrix} v_a \\ 0 \end{bmatrix} + \begin{bmatrix} v_b \\ 0 \end{bmatrix} = \begin{bmatrix} v_a + v_b \\ 0 \end{bmatrix}
\]

- vector × constant = vector
\[
c \begin{bmatrix} v_a \\ 0 \end{bmatrix} = \begin{bmatrix} cv_a \\ 0 \end{bmatrix}
\]
Homogeneous Values

- A point is a unique location
- A vector is the difference between two locations
- We can add vectors to points and to other vectors
- We can scale vectors

Homogeneous Points and Vectors:

\[
\begin{align*}
\mathbf{p}_a &= \begin{bmatrix} p_a \\ 1 \end{bmatrix} & \mathbf{p}_b &= \begin{bmatrix} p_b \\ 1 \end{bmatrix} & \mathbf{v} &= \begin{bmatrix} v \\ 0 \end{bmatrix}
\end{align*}
\]

Set an extra value to 1 for points
And to 0 for vectors

Point - Point = Vector
\[
\begin{bmatrix} p_b \\ 1 \end{bmatrix} - \begin{bmatrix} p_a \\ 1 \end{bmatrix} = \begin{bmatrix} p_b - p_a \\ 0 \end{bmatrix}
\]

Point + Vector = Point
\[
\begin{bmatrix} p \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} = \begin{bmatrix} p + \mathbf{v} \\ 1 \end{bmatrix}
\]

Vector + Vector = Vector
\[
\begin{bmatrix} \mathbf{v}_a \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_b \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{v}_a + \mathbf{v}_b \\ 0 \end{bmatrix}
\]

Vector × Constant = Vector
\[
c \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} = \begin{bmatrix} c \mathbf{v}_b \\ 0 \end{bmatrix}
\]

For now, let’s only consider homogeneous values that are 0 or 1
Homogeneous Values

• What should it mean to multiply a point by a constant?
• What should it mean to add points?

Recall: The Problem With Regular Coordinates

Multiplying a point by a constant
With regular coordinates

Adding two points
With regular coordinates
Homogeneous Values

• What should it mean to multiply a point by a constant?
• What should it mean to add points?

\[ p_a = \begin{bmatrix} p_a \\ 1 \end{bmatrix}, \quad p_b = \begin{bmatrix} p_b \\ 1 \end{bmatrix} \]

\[ c p_a = \begin{bmatrix} c p_a \\ c \end{bmatrix} \]

\[ p_a + p_b = \begin{bmatrix} p_a + p_b \\ 2 \end{bmatrix} \]

What does it mean to have a homogeneous value that is not 0 or 1?
Homogeneous Values

- What should it mean to multiply a point by a constant?
- What should it mean to add points?

\[ \mathbf{p}_a = \begin{bmatrix} p_a \\ 1 \end{bmatrix} \quad \mathbf{p}_b = \begin{bmatrix} p_b \\ 1 \end{bmatrix} \]

\[ c \mathbf{p}_a = \begin{bmatrix} c p_a \\ c \end{bmatrix} \]

\[ \mathbf{p}_a + \mathbf{p}_b = \begin{bmatrix} p_a + p_b \\ 2 \end{bmatrix} \]

\[ \alpha \mathbf{p}_a + \beta \mathbf{p}_b = \begin{bmatrix} \alpha p_a + \beta p_b \\ (\alpha + \beta) \end{bmatrix} \]

When \( \beta = -\alpha \), this becomes \( \alpha(p_a - p_b) \), a vector

When \( \beta = 1 - \alpha \) this becomes \( \begin{bmatrix} \alpha p_a + (1 - \alpha) p_b \\ 1 \end{bmatrix} \), a point

When can these operations be combined to get valid points or vectors?
Homogeneous Values

- What should it mean to multiply a point by a constant?
- What should it mean to add points?

$$p_a = \begin{bmatrix} p_a \\ 1 \end{bmatrix} \quad p_b = \begin{bmatrix} p_b \\ 1 \end{bmatrix}$$

$$c \cdot p_a = \begin{bmatrix} c \cdot p_a \\ c \end{bmatrix}$$

$$p_a + p_b = \begin{bmatrix} p_a + p_b \\ 2 \end{bmatrix}$$

$$\alpha p_a + \beta p_b = \begin{bmatrix} \alpha p_a + \beta p_b \\ (\alpha + \beta) \end{bmatrix}$$

When $\beta = -\alpha$, this becomes $\alpha(p_a - p_b)$, a vector

When $\beta = 1 - \alpha$ this becomes $\begin{bmatrix} \alpha p_a + (1 - \alpha)p_b \\ 1 \end{bmatrix}$, a point

When can these operations be combined to get valid points or vectors?
Homogeneous Values Coordinates

- Homogeneous values keep track of how much our choice of origin has influenced our coordinates.
- We can correct for the influence on a point by dividing all coordinates by the homogeneous value.
Barycentric Coordinates, Homogenization, & Center of Mass

• If we homogenize the weighted sum of $k$ points, we get their center of mass

\[ p = \frac{\sum \alpha_i p_i}{\sum \alpha_i} \quad \text{Weighted sum of points (weights given by alphas)} \]

\[ c = \frac{\sum m_i p_i}{\sum m_i} \quad \text{Equation for center of mass (masses given by m’s)} \]
Homogeneous Coordinates & Projection

• How to project an n-dimensional vector onto an image plane?
Homogeneous Coordinates & Projection

• How to project an n-dimensional vector onto an image plane?

$p = \begin{bmatrix} v_1 \\ 1 \end{bmatrix}$  \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$

What is the projection $p_u$ of $p$ onto the image plane?

Where $u$ is the unit vector perpendicular to the image plane

$\begin{align*}
\mathbf{p}_u &= \begin{bmatrix} \mathbf{v} \\ \mathbf{u}^\top \mathbf{v} \end{bmatrix} \\
&= \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \\
&= \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \\
&= \mathbf{u}
\end{align*}$

In homogeneous coordinates
Homogeneous Coordinates & Projection

• How to project an n-dimensional vector onto an image plane?

Where \( \mathbf{u} \) is the unit vector perpendicular to the image plane.

In homogeneous coordinates:

\[
\mathbf{p}_u = \begin{bmatrix} \mathbf{v} \\ \mathbf{u}^\top \mathbf{v} \end{bmatrix}
\]

How do we express this as a matrix?

\[
\mathbf{P} \mathbf{p} = \begin{bmatrix} \mathbf{v} \\ \mathbf{u}^\top \mathbf{v} \end{bmatrix}
\]

\[
\begin{bmatrix}
\mathbf{I}_{n \times n} & \mathbf{0}_{n \times 1} \\
\mathbf{u}^\top & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{v} \\
1
\end{bmatrix}
= \begin{bmatrix} \mathbf{v} \\ \mathbf{u}^\top \mathbf{v} \end{bmatrix}
\]

Projection onto image plane defined by \( \mathbf{u} \).
Similar triangles used to compute $p_u$ (Extra Slide)
Homogeneous Coordinates & Translation

• 3D translation is not linear in regular 3D coordinates

\[
\begin{bmatrix}
0 \\
\vdots \\
0
\end{bmatrix} =
\begin{bmatrix}
0 \\
\vdots \\
0
\end{bmatrix}
\]

In regular coordinates, no matrix can take the origin away from the origin...
Homogeneous Coordinates & Translation

- Translation is not linear in regular coordinates

\[
\begin{bmatrix}
x_x & y_x & z_x \\
x_y & y_y & z_y \\
x_z & y_z & z_z
\end{bmatrix}
\begin{bmatrix}
P_x \\
P_y \\
P_z
\end{bmatrix}
= P_x x + P_y y + P_z z
\]
Homogeneous Coordinates & Translation

- Translation is not linear in regular coordinates

\[
\begin{bmatrix}
    x_x & y_x & z_x & t_x \\
    x_y & y_y & z_y & t_y \\
    x_z & y_z & z_z & t_z \\
    0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    p_x \\
    p_y \\
    p_z \\
    1
\end{bmatrix} = p_xx + p_yy + p_zz + t
\]
Translation & Rotation: Vectors vs Points

• Points rotate and translate
• Vectors rotate but do not translate
  • Consider the surface normal of an object
  • If we translate the object, the surface normal direction does not change

\[ v = p_n - p_s \]
Homogeneous Coordinates & Translation

• Translating vectors (e.g., surface normals)

\[
\begin{bmatrix}
  x_x & y_x & z_x & t_x \\
  x_y & y_y & z_y & t_y \\
  x_z & y_z & z_z & t_z \\
  0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
  n_x \\
  n_y \\
  n_z \\
  0 \\
\end{bmatrix} = n_xx + n_yy + n_zz
\]
Homogeneous Coordinates: Putting It All Together

\[
\begin{bmatrix}
  x_x & y_x & z_x \\
  x_y & y_y & z_y \\
  x_z & y_z & z_z \\
  \text{u}^T
\end{bmatrix}
\begin{bmatrix}
  t_x \\
  t_y \\
  t_z \\
  0 \\
\end{bmatrix}
\begin{bmatrix}
  v_x \\
  v_y \\
  v_z \\
  1 \\
\end{bmatrix}
= \begin{bmatrix}
  v_x x + v_y y + v_z z + t \\
  \text{u}^T v
\end{bmatrix}
\]
Part 2: Image-Based Rendering

By Abe Davis
Light Fields & Image-Based Rendering

With Stick Figures!
Traditional Photography

Photographer

Scene

User
Traditional Photography

Photographer  Scene  User
What would be the simplest, most naïve, brute force approach to give the viewer control of the camera?
CAPTURE ALL THE IMAGES!
“Light Field” Photography
“Light Field” Photography
We can’t capture all the images
Light Field Photography

Captured Views

Synthesized Views

Scene
Light Field Photography

Captured Views

Synthesized Views

Scene
Sampling and Reconstruction

Camera parameters (e.g. position, orientation, focus, depth of field...)

Images

How do we synthesize this image?
Sampling and Reconstruction

- How do we sample?
- What space do we use to represent our data?
- How do we Interpolate in that space?
- How do we extract images from that space?
Sampling and Reconstruction

• How do we sample?
• What space do we use to represent our data?
• How do we Interpolate in that space?
• How do we extract images from that space?
Sampling and Reconstruction

• How do we sample?
• What space do we use to represent our data?
• How do we Interpolate in that space?
• How do we extract images from that space?
Sampling and Reconstructing Rays
Sampling and Reconstructing Rays
Sampling and Reconstructing Rays

Sample ≈ Pixel ≈ 1 Ray of Light
Captured Images $\rightarrow$ Ray Space $\rightarrow$ New Images

Interpolation happens here
How should we parameterize light?

Light ray = f(?)
Light

• Radiance:
  – $R(\text{position}, \text{angle})$
  – How many dimensions?
Light

• Radiance:
  – \( R(\text{position}, \text{angle}) \)
  – Position = \((x, y, z)\)
  – Angle = \((\text{theta}, \text{phi})\)
  – 5 dimensions
What is a good parameterization for light?

• The Light Field
What is a good parameterization for light?

• The Light Field
  – Unobstructed light
  – Each ray defined by intersection with 2 planes

\[ L(u,v,s,t) \]

**Figure 1:** The light slab representation.
What is a good parameterization for light?

- The Light Field
  - Unobstructed light
  - Each ray defined by intersection with 2 planes

\[ L(u, v, s, t) \]

**Figure 1:** The light slab representation.
The Light Field

Figure 1: The light slab representation.
Ray Space

Viewpoint → Image Plane → Plane

s

u
Ray Space

Each ray (pixel) is a single point
Ray Space
Ray Space
Ray Space
Ray Space
Ray Space

u

s

s

u
Ray Space
Ray Space
Ray Space
Ray Space
Projection Mapping

Captured Image

Requested Image
Projection Mapping

Captured Image

Requested Image

3D Geometry
Projection Mapping

Captured Image

Requested Image

3D Geometry
Projection Mapping

Captured Image

Requested Image

3D Geometry
Projection Mapping

- Captured Image
- Requested Image
- 3D Geometry
Ray Space
Capture Strategies

Camera Array

Lytro

Rely more on sampling

Stereo

Rely more on geometry
What happens when we don’t know geometry?
What happens when we don’t know geometry?
What happens when we don’t know geometry?
What happens when we don’t know geometry?
What happens when we don’t know geometry?
What happens when we don’t know geometry?
Ray Space
Ray Space
Ray Space

Camera Array

Lytro
Capture Strategies

Camera Array

Lytro
Rely more on sampling

Stereo
Rely more on geometry
Specialized Devices

Scene

Camera Array

Lytro