CSE 559A: Computer Vision

Fall 2020: T-R: 11:30-12:50pm @ Wrighton 300 / Zoom

Instructor: Ayan Chakrabarti (ayan@wustl.edu).
Course Staff: Adith Boloor, Patrick Williams

http://www.cse.wustl.edu/~ayan/courses/cse559a/

Oct 20, 2020
• Reminder: Recitation this Friday for PSET 2.
• PSET 2 due next Tuesday.
• PSET 1 solutions available.
• We will be shifting to purely online lectures from next week.
• From next week, we’ll have office hours with me from Tue, Thu: 1-1:30pm right after lecture.
PHOTOMETRIC STEREO

- Last time: estimating normals and albedo from image.
- Going from normals to a depth map by solving a least squares problem.
- Solving in the Fourier domain: Frankot-Chellappa
NORMALS TO DEPTH

$$Z = \arg \min_Z \|g_x - f_x * Z\|^2 + \|g_y - f_y * Z\|^2 + \lambda R(Z)$$

Version 2: Use conjugate gradient.

- Allows us to use different weights for different pixels.
NORMALS TO DEPTH

\[
Z = \arg\min_Z \sum_n w[n](g_x[n] - (f_x \ast Z)[n])^2 + \sum_n w[n](g_y - (f_y \ast Z)[n])^2 + \lambda R(Z)
\]

Version 2: Use conjugate gradient.

- Allows us to use different weights for different pixels.
  - Set \( w[n] \) to 0 for masked out pixels.
  - Set \( w[n] \) to \( (\hat{n}_z[n])^2 \) everywhere else.
    - Accounts for the fact that we got gradients by dividing by \( \hat{n}_z \).
    - Smaller values will be noisier.
NORMALS TO DEPTH

\[ Z = \arg \min_Z \sum_n w[n](g_x[n] - (f_x * Z)[n])^2 + \sum_n w[n](g_y - (f_y * Z)[n])^2 + \lambda R(Z) \]

\[ Z = \arg \min_Z Z^T QZ - 2Z^T b + c \]

- Begin with all zeroes guess \( Z_0 \) for \( Z \)
- \( k = 0, r_0 \leftarrow b - QZ_0, \quad p_0 \leftarrow r_0 \)
- Repeat (for say a fixed number of iterations)
  - \( \alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T Q p_k} \)
  - \( Z_{k+1} \leftarrow Z_k + \alpha_k p_k \)
  - \( r_{k+1} \leftarrow r_k - \alpha_k Q p_k \)
  - \( \beta_k \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \)
  - \( p_{k+1} \leftarrow r_{k+1} + \beta_k p_k \)
  - \( k = k + 1 \)
NORMALS TO DEPTH

\[ Z = \arg \min_Z \sum_n w[n](g_x[n] - (f_x * Z)[n])^2 + \sum_n w[n](g_y - (f_y * Z)[n])^2 + \lambda \sum_n ((f_r * Z[n])^2 \]

\[ Z = \arg \min_Z Z^T QZ - 2Z^T b + c \]

We need to figure out:

- What is \(b\) (should be same shape as image)
- How do compute \(Qp\) for a given \(p\) (where \(p\) is same shape as image)

Let \(W\) be a diagonal matrix with values of \(w[n]\). \(F_x, F_y\) and \(F_r\) denote convolutions by \(f_x, f_y\), and \(f_r\).

\[ Z = \arg \min(G_x - F_xZ)^T W(G_x - F_xZ) + (G_y - F_yZ)^T W(G_y - F_yZ) + \lambda(F_rZ)^T (F_rZ) \]

What are \(Q\) and \(b\) ?

\[ Q = F_x^T W F_x + F_y^T W F_y + \lambda F_r^T F_r \]

\[ b = F_x^T W G_x + F_y^T W G_y \]
NORMALS TO DEPTH

\[ Z = \arg \min_Z \sum_n w[n] (g_x[n] - (f_x * Z)[n])^2 + \sum_n w[n] (g_y - (f_y * Z)[n])^2 + \sum_n ((f_r * Z)[n])^2 \]

\[ Z = \arg \min_Z Z^T QZ - 2Z^T b + c \]

- What is \( b \) (should be same shape as image)
- How do compute \( Qp \) for a given \( p \) (where \( p \) is same shape as image)

\[
Q = F_x^T WF_x + F_y^T WF_y + \lambda F_r^T F_r \\
b = F_x^T WG_x + F_y^T WG_y
\]

\[
Qp = ((p * f_x) \times w) * \bar{f}_x + (p * f_y) \times w) * \bar{f}_y + \lambda((p * f_r) * \bar{f}_r)
\]

\( \bar{f} \) mean the flipped versions of \( f \). \( \times \) means element-wise product.

\( * \) can be same convolutions with zero padding

\[
b = (g_x \times w) * \bar{f}_x + (g_y \times w) * \bar{f}_y
\]

- Remember, \( p^T Qp \) is just \( \langle p, Qp \rangle \). So compute \( Qp \), take element-wise product with \( p \), and sum across all pixels.
PHOTOMETRIC STEREO
PHOTOMETRIC STEREO++

- Robust Photometric Stereo
  In the presence of shadows, specular highlights
- Robust Photometric Stereo
  In the presence of shadows, specular highlights
  - Simple option: For each pixel, just drop the darkest $n$ and brightest $n$ pixels.
- Robust Photometric Stereo

  In the presence of shadows, specular highlights
  - Simple option: For each pixel, just drop the darkest $n$ and brightest $n$ pixels.

**Robust Photometric Stereo via Low-Rank Matrix Completion and Recovery**

Wu et al., PAMI 2011 / ACCV 2010
PHOTOMETRIC STEREO++

\[ I_1 = \mathbf{\ell}_1^T(\rho \  \hat{n}) \]
\[ I_2 = \mathbf{\ell}_2^T(\rho \  \hat{n}) \]
\[ I_3 = \mathbf{\ell}_3^T(\rho \  \hat{n}) \]
PHOTOMETRIC STEREO++

Three measurements,

\[ I_1 = \ell_1^T (\rho \quad \hat{n}) \]
\[ I_2 = \ell_2^T (\rho \quad \hat{n}) \]
\[ I_3 = \ell_3^T (\rho \quad \hat{n}) \]
PHOTOMETRIC STEREO++

\[ I_1 = \ell_1^T (\rho \hspace{0.5em} \hat{n}) \]
\[ I_2 = \ell_2^T (\rho \hspace{0.5em} \hat{n}) \]
\[ I_3 = \ell_3^T (\rho \hspace{0.5em} \hat{n}) \]

Three measurements, three images,
PHOTOMETRIC STEREO++

\[ I_1 = \ell_1^T (\rho \ \hat{n}) \]
\[ I_2 = \ell_2^T (\rho \ \hat{n}) \]
\[ I_3 = \ell_3^T (\rho \ \hat{n}) \]

Three measurements, three images, only works for static scenes!
PHOTOMETRIC STEREO++

\[ I_R = \ell_R^T (\rho \ \hat{n}) \]
\[ I_G = \ell_G^T (\rho \ \hat{n}) \]
\[ I_B = \ell_B^T (\rho \ \hat{n}) \]

Multiplex in color !!
PHOTOMETRIC STEREO++

\[ I_R = \ell_R^T (\rho \, \hat{n}) \]
\[ I_G = \ell_G^T (\rho \, \hat{n}) \]
\[ I_B = \ell_B^T (\rho \, \hat{n}) \]

Multiplex in color !!

Three observations, three unknowns ?
\[ I_R = \ell_R^T (\rho \hat{n}) \]
\[ I_G = \ell_G^T (\rho \hat{n}) \]
\[ I_B = \ell_B^T (\rho \hat{n}) \]

Multiplex in color!!

Three observations, three unknowns?
PHOTOMETRIC STEREO++

\[ I_R = \ell_R^T (\rho_R \hat{n}) \]
\[ I_G = \ell_G^T (\rho_G \hat{n}) \]
\[ I_B = \ell_B^T (\rho_B \hat{n}) \]

Multiplex in color!!

Three observations, five unknowns.

(so close ......)
PHOTOMETRIC STEREOP++

Solutions?

\[ I_R = \ell_T^R (\rho_R, \^n) \]
\[ I_G = \ell_T^G (\rho_G, \^n) \]
\[ I_B = \ell_T^B (\rho_B, \^n) \]

Multiplex in color !!

Three observations, five unknowns.

(so close ......)
PHOTOMETRIC STEREO++

Solutions?

- More colors, hyperspectral imaging?

\[
I_R = \ell^T_R (\rho_R \hat{n}) \\
I_G = \ell^T_G (\rho_G \hat{n}) \\
I_B = \ell^T_B (\rho_B \hat{n})
\]

Multiplex in color!!

Three observations, five unknowns.

(so close ......)
Solutions?

- More colors, hyperspectral imaging?

  Every additional channel, adds another unknown.
  (But with narrow wavelength bands, you can assume albedo
  of neighboring bands vary smoothly).

\[
I_R = \ell_R^T (\rho_R \hat{n}) \\
I_G = \ell_G^T (\rho_G \hat{n}) \\
I_B = \ell_B^T (\rho_B \hat{n})
\]

Multiplex in color!!

Three observations, five unknowns.

(so close ....)
Solutions?

- More colors, hyperspectral imaging?
  Every additional channel, adds another unknown.
  (But with narrow wavelength bands, you can assume albedo
  of neighboring bands vary smoothly).

- What if I knew albedo (or even just albedo chromaticity)?

\[
I_R = \ell_T^R (\rho_R, \hat{n}) \\
I_G = \ell_T^G (\rho_G, \hat{n}) \\
I_B = \ell_T^B (\rho_B, \hat{n})
\]

Multiplex in color!!

Three observations, five unknowns.

(so close ......)
PHOTOMETRIC STEREO++

Solutions?

- More colors, hyperspectral imaging?
  Every additional channel, adds another unknown.
  (But with narrow wavelength bands, you can assume albedo
   of neighboring bands vary smoothly).

- What if I knew albedo (or even just albedo chromaticity)?
  - Want to capture shape of object, let's paint it's surface with
    known color paint.

\[
\begin{align*}
I_R &= \ell_R^T (\rho_R \hat{n}) \\
I_G &= \ell_G^T (\rho_G \hat{n}) \\
I_B &= \ell_B^T (\rho_B \hat{n})
\end{align*}
\]

Multiplex in color!!

Three observations, five unknowns.

(so close ......)
PHOTOMETRIC STEREO++

Solutions?

- More colors, hyperspectral imaging?
  Every additional channel, adds another unknown.
  (But with narrow wavelength bands, you can assume albedo
   of neighboring bands vary smoothly).

- What if I knew albedo (or even just albedo chromaticity)

  - Want to capture shape of object, let's paint it's surface with
    known color paint.

\[
\begin{align*}
I_R &= \ell_R^T (\rho_R \hat{n}) \\
I_G &= \ell_G^T (\rho_G \hat{n}) \\
I_B &= \ell_B^T (\rho_B \hat{n})
\end{align*}
\]

Multiplex in color!!

Three observations, five unknowns.

(so close ......)
PHOTOMETRIC STEREO++

Solutions?

- More colors, hyperspectral imaging?
  Every additional channel, adds another unknown.
  (But with narrow wavelength bands, you can assume albedo
   of neighboring bands vary smoothly).

- What if I knew albedo (or even just albedo chromaticity)?

- Want to capture shape of object, let's paint it's surface with
  known color paint.

\[
I_R = \ell_R^T (\rho_R \hat{n}) \\
I_G = \ell_G^T (\rho_G \hat{n}) \\
I_B = \ell_B^T (\rho_B \hat{n})
\]

Multiplex in color!!

Three observations, five unknowns.

(PS: this wouldn't work, not lambertian)

(so close ......)
Solutions?

- More colors, hyperspectral imaging?
  Every additional channel, adds another unknown.
  (But with narrow wavelength bands, you can assume albedo
  of neighboring bands vary smoothly).

- What if I knew albedo (or even just albedo chromaticity)?

  - Want to capture shape of object, let's paint it's surface with
    known color paint.
    (actually, used sometimes ... use powder instead of paint)

\[
\begin{align*}
I_R &= \ell_R^T (\rho_R \hat{n}) \\
I_G &= \ell_G^T (\rho_G \hat{n}) \\
I_B &= \ell_B^T (\rho_B \hat{n})
\end{align*}
\]

Multiplex in color!!

Three observations, five unknowns.

(so close ......)
Solutions?

- More colors, hyperspectral imaging?
  Every additional channel, adds another unknown.
  (But with narrow wavelength bands, you can assume albedo
  of neighboring bands vary smoothly).

- What if I knew albedo (or even just albedo chromaticity)?

  - Want to capture shape of object, let's paint it's surface with
    known color paint.
    (actually, used sometimes ... use powder instead of paint)

  - Pull something with known albedo tightly over object

\[
I_R = \ell_R^T (\rho_R \hat{n}) \\
I_G = \ell_G^T (\rho_G \hat{n}) \\
I_B = \ell_B^T (\rho_B \hat{n})
\]

Multiplex in color!!

Three observations, five unknowns.

(so close ......)
PHOTOMETRIC STEREO++

Solutions?

- More colors, hyperspectral imaging?
  Every additional channel, adds another unknown.
  (But with narrow wavelength bands, you can assume albedo
  of neighboring bands vary smoothly).

- What if I knew albedo (or even just albedo chromaticity)?
  - Want to capture shape of object, let's paint it's surface with
    known color paint.
    (actually, used sometimes ... use powder instead of paint)
  - Pull something with known albedo tightly over object

$$I_B = \ell_B^T (\rho_B \hat{n})$$

Multiplex in color!!

Three observations, five unknowns.

(so close ......)
PHOTOMETRIC STEREO++

The GelSight Benchtop Scanner can quickly capture the surface geometry of almost any material regardless of its optical properties. This capability is due to the GelSight sensor, a small block of elastomer with a reflective coating on one side. The unique properties of the GelSight sensor remove the influence of the optical characteristics of the material on the measurement, thereby ensuring accuracy, repeatability, and consistent performance, even on optically complex surfaces.

SEM-like images: GelSight images are often compared to images from a scanning electron microscope (SEM) because both systems reveal surface structure without being influenced by the surface’s optical characteristics. While SEMs offer amazing resolution, they operate in vacuum, require elaborate sample preparation, and typically only capture a small field of view. For imaging at the micron scale and above, GelSight can capture comparable images with much less time and effort.
The GelSight Benchtop Scanner can quickly capture the surface geometry of almost any object, regardless of its optical properties. This capability is due to the GelSight sensor, a small elastomer with a reflective coating on one side. The unique properties of the GelSight sensor ensure reliable and consistent performance, even on optically complex surfaces.

SEM-like images: GelSight images are often compared to images from a scanning electron microscope (SEM) because both systems reveal surface structure without being influenced by the surface’s optical characteristics. While SEMs offer amazing resolution, they operate in vacuum, require elaborate sample preparation, and typically only capture a small field of view. For imaging at the micron scale and above, GelSight can capture comparable images with much less time and effort.
PHOTOMETRIC STEREO++

The GelSight Benchtop Scanner can quickly capture the surface geometry of almost any object regardless of its optical properties. This capability is due to the GelSight sensor, a small elastomer with a reflective coating on one side. The unique properties of the GelSight sensor influence the optical characteristics of the material on the measurement, thereby ensuring repeatability, and consistent performance, even on optically complex surfaces.

**SEM-like images:** GelSight images are often compared to images from SEM (Secondary Electron Microscopy) because both systems reveal surface structure without being invasive. While SEMs offer amazing resolution, they operate in preparation, and typically only capture a small field of view. For images GelSight can capture comparable images with much less time and effort.

Greek coin, 3rd Century BCE. Geometry rendered with a color texture map.
The GelSight Benchtop Scanner can quickly capture the surface geometry of almost any object regardless of its optical properties. This capability is due to the GelSight sensor, a small elastomer with a reflective coating on one side. The unique properties of the GelSight sensor influence the optical characteristics of the material on the measurement, thereby ensuring repeatability and consistent performance, even on optically complex surfaces.

Raised toner particles on clay-coated paper. Geometry rendered with a color texture map.

Greek coin, 3rd Century BCE. Geometry rendered with a color texture map.
PHOTOMETRIC STEREO++

Solutions?

- More colors, hyperspectral imaging?
  Every additional channel, adds another unknown.
  (But with narrow wavelength bands, you can assume albedo
  of neighboring bands vary smoothly).

- What if I knew albedo (or even just albedo chromaticity)?
  - Want to capture shape of object, let's paint it's surface with
    known color paint.
    (actually, used sometimes ... use powder instead of paint)
  - Pull something with known albedo tightly over object

- Make assumptions: albedo is smooth, shape is smooth,
  solve with a prior / regularizer!

\[
I_R = \ell_R^T (\rho_R \hat{n}) \\
I_G = \ell_G^T (\rho_G \hat{n}) \\
I_B = \ell_B^T (\rho_B \hat{n})
\]

Multiplex in color!!

Three observations, five unknowns.

(so close .......)
PHOTOMETRIC STEREO++

Solutions?

- More colors, hyperspectral imaging?
  Every additional channel, adds another unknown. (But with narrow wavelength bands, you can assume albedo of neighboring bands vary smoothly).

- What if I knew albedo (or even just albedo chromaticity)?

  - Want to capture shape of object, let's paint it's surface with

\[
I_R = \ell_R^T (\rho_R \, \hat{n}) \\
I_G = \ell_G^T (\rho_G \, \hat{n}) \\
I_B = \ell_B^T (\rho_B \, \hat{n})
\]

Multiplex in color!!

Three observations, five unknowns.

( so close ......)
GENERAL SHADING

Single Point Light Source at infinity

\[ I = \ell^T (\rho \hat{n}) \]
MULTIPLE POINT LIGHT SOURCES, all at infinity

\[ I = \ell^T (\rho \, \hat{n}) \]
GENERAL SHADING

Multiple Point Light Sources, all at infinity, on at the same time (not PS)

\[ I = \ell^T (\rho \hat{n}) \]
GENERAL SHADING

Multiple Point Light Sources, all at infinity, on at the same time (not PS)

\[ I = \ell_1^T(\rho \, \hat{n}) + \ell_2^T(\rho \, \hat{n}) + \ell_3^T(\rho \, \hat{n}) \]
GENERAL SHADING

General Illumination from Infinity

\[ \hat{n} \]
GENERAL SHADING

General Illumination from Infinity

From infinity implies that the angles of light to the point won't depend on the point's position.
From infinity implies that the angles of light to the point won't depend on the point's position.

Two points with the same surface normal and same albedo will reflect the same light.
GENERAL SHADING

Lambertian Calibration Target
GENERAL SHADING

Lambertian Calibration Target

Environment as seen on a Glass Sphere
For known albedo, gives us total irradiance for every normal.

\[ L_o(\theta_o, \phi_o) = K_c \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \]

Lambertian Calibration Target
GENERAL SHADING

\[ L_o(\theta_o, \phi_o) = K_c \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \]

Lambertian Calibration Target
GENERAL SHADING

\[ L_o(\theta_o, \phi_o) = K_t \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \]

\[ L_o(\theta_o, \phi_o) = K_c \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \]

Lambertian Calibration Target
GENERAL SHADING

\[ L_o(\theta_o, \phi_o) = K_t \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \]

Lambertian Calibration Target

\[ L_o(\theta_o, \phi_o) = K_c \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \]
\[ L_o(\theta_o, \phi_o) = K_t \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \]

Instead of \langle n, l \rangle, now have a 'lookup table' for each normal.

\[ L_o(\theta_o, \phi_o) = K_c \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \]
GENERAL SHADING

\[ L_o(\theta_o, \phi_o) = K_t \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \]

Instead of \( \langle n, l \rangle \), now have a 'lookup table' for each normal.

Cumbersome, so sometimes approximated using "spherical harmonics"

\[ L_o(\theta_o, \phi_o) = K_c \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \]

Lambertian Calibration Target
**GENERAL SHADING**

\[
L_o(\theta_o, \phi_o) = K_t \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i
\]

Instead of \(<n,l>\), now have a 'lookup table' for each normal.

Cumbersome, so sometimes approximated using "spherical harmonics"

\[
I = \rho \ \hat{n}^T L \ \hat{n}
\]

\[
L_o(\theta_o, \phi_o) = K_c \int L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i
\]

Lambertian Calibration Target
Can apply the same idea to handle non-lambertian shading
GENERAL SHADING

Can apply the same idea to handle non-lambertian shading

- Put sphere of same material as object in to the scene.
  (assume constant BRDF on target)

- Assume both light and camera far away from the object.

- Then same normal on both sphere and object will produce the same intensity.
NATURAL SHADING

Want this to work on natural images taken in natural illumination ....
NATURAL SHADING

Want this to work on natural images taken in natural illumination ....

- General unknown illumination environment
- General unknown shape
- General unknown albedo
  Object possibly not Lambertian
SIRFS
shape, illumination, and reflectance from shading

minimize $g(R) + f(Z) + h(L)$
We assume vectors are column vectors.

\[ p' = [x, y] \] implies a 2-D row vector (of size $1 \times 2$)

\[ p = [x, y]^T = \begin{bmatrix} x \\ y \end{bmatrix} \] implies a 2-D column vector (of size $2 \times 1$)
Remember, the pinhole camera

\[(x, y, z) \Rightarrow \left(-f \frac{x}{z}, -f \frac{y}{z}\right)\]
GEOMETRY

The division is annoying, makes projection non-linear.

Can no longer use matrices / linear operations to relate co-ordinates.

But we like matrix operations!

Solution: Homogeneous Co-ordinates

\[ 3D \ (x, y, z) \Rightarrow \left( -f \frac{x}{z}, -f \frac{y}{z} \right) \ 2D \]
HOMOGENEOUS CO-ORDINATES

Book-keeping trick!

- 2D Cartesian Co-ordinates: \((x, y)\)
- 2D Homogeneous Co-ordinates: \((\alpha x, \alpha y, \alpha)\)
- Cartesian to Homogeneous: \((x, y) \rightarrow (ax, ay, \alpha)\)
  - When \(\alpha = 1\), this is called “augmented”: \((x, y, 1)\)
- Homogeneous to Cartesian: \((x', y', \alpha) \rightarrow \left(\frac{x'}{\alpha}, \frac{y'}{\alpha}\right)\)
- A whole family of homogeneous co-ordinates map to the same cartesian co-ordinate
  - Over-parameterization of a 2D point
  - Denote this equality by \(\sim\): \((\alpha_1 x, \alpha_1 y, \alpha_1) \sim (\alpha_2 x, \alpha_2 y, \alpha_2)\)
- Space of 2D Homogeneous co-ordinates denoted as \(\mathbb{P}^2 = \mathbb{R}^3 - (0, 0, 0)\)
- Note that \((x, y, 0)\) is defined. In cartesian co-ordinates, it is the point at infinity along the line joining \((0, 0)\) to \((x, y)\).
- 3D Homogeneous Co-ordinates: \((x, y, z) \Rightarrow (ax, ay, az, \alpha)\)
HOMOGENEOUS CO-ORDINATES

- Useful way to think about 2-D Homogeneous Co-ordinates $\mathbb{P}^2$

“Rays” in $\mathbb{R}^3$
HOMOGENEOUS CO-ORDINATES

• Useful way to think about 2-D Homogeneous Co-ordinates $\mathbb{P}^2$

```
\[ \begin{align*}
x & \quad \text{(x, y)} \\
\text{y} & \quad \text{plane x = 1} \\
\text{z} & \quad \text{plane y = 1} \\
\end{align*} \]
```

“Rays” in $\mathbb{R}^3$

• Cartesian form is “intersection” with plane $z = 1$.
• $(x, y, 0)$ are forms that are parallel to the $z = 1$ plane, intersect at infinity.
• 3-D Homogeneous Co-ordinates are rays in 4D, intersection with a hyper-plane.
HOMOGENEOUS CO-ORDINATES

\[ \left( -f \frac{x}{z}, -f \frac{y}{z} \right) \Leftarrow (x, y, z) \]
HOMOGENEOUS CO-ORDINATES

\[
\left( -f \frac{x}{z}, -f \frac{y}{z} \right) \leftrightarrow (x, y, z)
\]
\[
\begin{pmatrix}
-f \frac{x}{z}, -f \frac{y}{z}
\end{pmatrix} \leftrightarrow (x, y, z)
\]

\[
P_{2d} = \begin{bmatrix}
P_{3d}
\end{bmatrix}
\]
HOMOGENEOUS CO-ORDINATES

\[
(-f \frac{x}{z}, -f \frac{y}{z}) \leftrightarrow (x, y, z)
\]

\[
P_{2d} = \begin{bmatrix}
P_{3d}
\end{bmatrix}
\]

Homogeneous co-ordinates
HOMOGENEOUS CO-ORDINATES

\[
\begin{pmatrix}
-f \frac{x}{z}, -f \frac{y}{z}
\end{pmatrix} \leftrightarrow (x, y, z)
\]

\[P_{2d} = \begin{pmatrix} \cdots \end{pmatrix} \quad P_{3d}\]

Sizes?

Homogeneous co-ordinates
HOMOGENEOUS CO-ORDINATES

\[
\begin{pmatrix}
-f \frac{x}{z}, & -f \frac{y}{z}
\end{pmatrix} \iff (x, y, z)
\]

\[P_{2d} = \begin{pmatrix} \_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ \\
\end{pmatrix}
\]

\[P_{3d}
\]

Sizes?

Homogeneous co-ordinates
Homogeneous co-ordinates

\[
\begin{pmatrix}
-f \frac{x}{z}, -f \frac{y}{z}
\end{pmatrix} \iff (x, y, z)
\]

\[
P_{2d} = \begin{bmatrix}
3 \times 1
\end{bmatrix}
\]

\[
P_{3d}
\]

Sizes?

Homogeneous co-ordinates
Homogeneous co-ordinates

\[
\begin{pmatrix}
-f \frac{x}{z}, -f \frac{y}{z}
\end{pmatrix} \iff (x, y, z)
\]

\[
P_{2d} = \begin{bmatrix} \text{3x1} \end{bmatrix}
\]

\[
P_{3d} = \begin{bmatrix} \text{4x1} \end{bmatrix}
\]

Sizes?
HOMOGENEOUS CO-ORDINATES

\[
\begin{pmatrix}
-x \frac{x}{z}, & -f \frac{y}{z}
\end{pmatrix}
\iff
(x, y, z)
\]

\[
P_{2d} = \begin{pmatrix}
P_{3d}
\end{pmatrix}
\]

3x1 3x4

4x1

Homogeneous co-ordinates
HOMOGENEOUS CO-ORDINATES

\[
\begin{pmatrix}
-f\frac{x}{z}, -f\frac{y}{z}
\end{pmatrix} \leftrightarrow (x, y, z)
\]

\[
P_{2d} = \begin{bmatrix}
 & & & \\
 & & & \\
 & & & \\
\end{bmatrix} \quad 3x1 \quad \begin{bmatrix}
 & & & \\
 & & & \\
 & & & \\
\end{bmatrix} \quad P_{3d} \quad 4x1
\]

Homogeneous co-ordinates
HOMOGENEOUS CO-ORDINATES

\[ (-f \frac{x}{z}, -f \frac{y}{z}) \leftrightarrow (x, y, z) \]

\[ P_{2d} = \begin{bmatrix} \alpha x \\ \alpha y \\ \alpha z \\ \alpha \end{bmatrix} \]

Homogeneous co-ordinates
HOMOGENEOUS CO-ORDINATES

\[
\left(-f \frac{x}{z}, -f \frac{y}{z}\right) \leftrightarrow (x, y, z)
\]

\[
\begin{bmatrix}
  a \\
  b \\
  c
\end{bmatrix}
= \begin{bmatrix}
  \alpha x \\
  \alpha y \\
  \alpha z \\
  \alpha
\end{bmatrix}
\]

Homogeneous co-ordinates
HOMOGENEOUS CO-ORDINATES

\[
\left(-f \frac{x}{z}, -f \frac{y}{z}\right) \iff (x, y, z)
\]

\[
\begin{bmatrix}
    a \\
    b \\
    c \\
\end{bmatrix} =
\begin{bmatrix}
    \alpha x \\
    \alpha y \\
    \alpha z \\
\end{bmatrix}
\]

Homogeneous co-ordinates

\[
\frac{a}{c} = -f \frac{x}{z} \quad \frac{b}{c} = -f \frac{y}{z}
\]
HOMOGENEOUS CO-ORDINATES

\[
\left(-f \frac{x}{z}, -f \frac{y}{z}\right) \iff (x, y, z)
\]

\[
\begin{bmatrix}
a \\
b \\
c \\
\end{bmatrix} = \begin{bmatrix}
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\end{bmatrix} \begin{bmatrix}
\alpha x \\
\alpha y \\
\alpha z \\
\alpha \\
\end{bmatrix}
\]

Take 5 mins!

Homogeneous co-ordinates

\[
\frac{a}{c} = -f \frac{x}{z} \quad \frac{b}{c} = -f \frac{y}{z}
\]
HOMOGENEOUS CO-ORDINATES

\[\left(-f \frac{x}{z}, -f \frac{y}{z}\right) \iff (x, y, z)\]

\[
\begin{bmatrix}
  a \\
  b \\
  c \\
\end{bmatrix}
= 
\begin{bmatrix}
  \text{\textcolor{green}{\begin{array}{cccc}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0 \\
\end{array}}} \\
  \text{\textcolor{red}{\begin{array}{cccc}
  0 & 0 & 0 & 1 \\
  0 & 0 & 0 & 1 \\
  0 & 0 & 0 & 1 \\
\end{array}}} \\
\end{bmatrix}
\begin{bmatrix}
  \alpha x \\
  \alpha y \\
  \alpha z \\
  \alpha \\
\end{bmatrix}
\]

Homogeneous co-ordinates

\[
\frac{a}{c} = -f \frac{x}{z} \quad \frac{b}{c} = -f \frac{y}{z}
\]
HOMOGENEOUS CO-ORDINATES

\[
\left( -f \frac{x}{z}, -f \frac{y}{z} \right) \iff (x, y, z)
\]

\[
\begin{bmatrix}
a \\
b \\
c
\end{bmatrix}
= \begin{bmatrix}
-f & 0 & 0 & 0 \\
\alpha x & \alpha y & \alpha z & \alpha
\end{bmatrix}
\]

Homogeneous co-ordinates

\[
\frac{a}{c} = -f \frac{x}{z} \quad \frac{b}{c} = -f \frac{y}{z}
\]
HOMOGENEOUS CO-ORDINATES

\[
\begin{bmatrix}
-a \\
b \\
c
\end{bmatrix} = \begin{bmatrix}
-f & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
\alpha x \\
\alpha y \\
\alpha z \\
\alpha
\end{bmatrix}
\]

Homogeneous co-ordinates

\[
\frac{a}{c} = -f \frac{x}{z} \quad \frac{b}{c} = -f \frac{y}{z}
\]
HOMOGENEOUS CO-ORDINATES

\[
\begin{pmatrix}
-\frac{f}{z} x \\
-\frac{f}{z} y
\end{pmatrix} \iff (x, y, z)
\]

\[
\begin{bmatrix}
    a \\
    b \\
    c
\end{bmatrix} =
\begin{bmatrix}
    -f & 0 & 0 & 0 \\
    0 & -f & 0 & 0 \\
    \frac{1}{f} & \frac{1}{f} & \frac{1}{f} & \frac{1}{f}
\end{bmatrix}
\begin{bmatrix}
    \alpha x \\
    \alpha y \\
    \alpha z \\
    \alpha
\end{bmatrix}
\]

Homogeneous co-ordinates

\[
\begin{align*}
a &= -f \frac{x}{z} \\
b &= -f \frac{y}{z}
\end{align*}
\]
HOMOGENEOUS CO-ORDINATES

\[
\left(-f \frac{x}{z}, -f \frac{y}{z}\right) \iff (x, y, z)
\]

\[
\begin{bmatrix}
  a \\
  b \\
  c
\end{bmatrix} =
\begin{bmatrix}
  -f & 0 & 0 & 0 \\
  0 & -f & 0 & 0 \\
  \text{[green]} & \text{[green]} & \text{[green]} & \text{[green]}
\end{bmatrix}
\begin{bmatrix}
  \alpha x \\
  \alpha y \\
  \alpha z \\
  \alpha
\end{bmatrix}
\]

Homogeneous co-ordinates

\[
\frac{a}{c} = -f \frac{x}{z} \quad \frac{b}{c} = -f \frac{y}{z}
\]
HOMOGENEOUS CO-ORDINATES

\[
\left( -f \frac{x}{z}, -f \frac{y}{z} \right) \leftrightarrow (x, y, z)
\]

\[
\begin{bmatrix}
  a \\
  b \\
  c
\end{bmatrix} =
\begin{bmatrix}
  -f & 0 & 0 & 0 \\
  0 & -f & 0 & 0 \\
  0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
  \alpha x \\
  \alpha y \\
  \alpha z \\
  \alpha
\end{bmatrix}
\]

Homogeneous co-ordinates

\[
\frac{a}{c} = -f \frac{x}{z} \quad \frac{b}{c} = -f \frac{y}{z}
\]
HOMOGENEOUS CO-ORDINATES

\[
\left(-f \frac{x}{z}, -f \frac{y}{z}\right) \iff (x, y, z)
\]

\[
\begin{bmatrix}
  a \\
  b \\
  c
\end{bmatrix} =
\begin{bmatrix}
  -f & 0 & 0 & 0 \\
  0 & -f & 0 & 0 \\
  0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
  \alpha x \\
  \alpha y \\
  \alpha z \\
  \alpha
\end{bmatrix}
\]

Homogeneous co-ordinates

\[
\begin{align*}
\frac{a}{c} &= -f \frac{x}{z} \\
\frac{b}{c} &= -f \frac{y}{z}
\end{align*}
\]

Works for all non-zero values of \( \alpha \)
HOMOGENEOUS CO-ORDINATES

- Turned non-linear perspective projection into a linear operation.
- Here’s a different projection matrix:

\[
P_{2d} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad P_{3d}
\]

What does this represent?

\((x, y, z) \rightarrow (?, ?)\)
HOMOGENEOUS CO-ORDINATES

- Turned non-linear perspective projection into a linear operation.
- Here’s a different projection matrix:

\[
P_{2d} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad P_{3d}
\]

What does this represent?

\[(x, y, z) \rightarrow (x, y)\]

Orthographic Projection
HOMOGENEOUS CO-ORDINATES

Perspective Projection

Orthographic Projection
HOMOGENEOUS CO-ORDINATES

Perspective Projection

Orthographic Projection

Preserves parallel lines
Doesn't really correspond to a real camera
HOMOGENEOUS CO-ORDINATES

- Also useful to represent translation, rotation, skew in addition to projection
- Learn to chain together all these operations to:
  - Relate points in 3D to points in image
  - Verify angles, metric lengths from calibration targets, …
  - Relate points in two images from different cameras
HOMOGENEOUS CO-ORDINATES: 2D

Lines

Equation of a line in 2D:

\[ ax + by + c = 0 \]

Let \( p = [\alpha x, \alpha y, \alpha]^T \) be homogeneous co-ordinates of a point \((x, y)\). Then,

\[ l^T p = 0, \quad l = [a, b, c]^T \]

Interestingly, \( l \) is also defined “upto scale”: \( l' = [\beta a, \beta b, \beta c]^T \) describes the same line as \( l \).
HOMOGENEOUS CO-ORDINATES: 2D

Lines

Since $l^T p = 0$ for all points that lie on a line:
HOMOGENEOUS CO-ORDINATES: 2D

Lines

Given two points \( p_1 \) and \( p_2 \), what is the homogeneous vector for the line joining them?

It has to be an \( l \) such that \( l^T p_1 = 0 \) and \( l^T p_2 = 0 \).

Is that sufficient to determine \( l \)?

Yes. Because, only need \( l \) upto scale.

Solution given by: \( l = p_1 \times p_2 \) (Vector Cross-product)

Recap: Writing \( u = [u_1, u_2, u_3]^T = u_1 \hat{i} + u_2 \hat{j} + u_3 \hat{k} \), and \( u = [v_1, v_2, v_3]^T = v_1 \hat{i} + v_2 \hat{j} + v_3 \hat{k} \)

\[
\begin{vmatrix}
\hat{i} & \hat{j} & \hat{k} \\
u_1 & u_2 & u_3 \\
v_1 & v_2 & v_3
\end{vmatrix}
= (u_2 v_3 - u_3 v_2)\hat{i} + (u_3 v_1 - u_1 v_3)\hat{j} + (u_1 v_2 - u_2 v_1)\hat{k}
\]

\[
= [(u_2 v_3 - u_3 v_2), (u_3 v_1 - u_1 v_3), (u_1 v_2 - u_2 v_1)]^T
\]
HOMOGENEOUS CO-ORDINATES: 2D

Lines

Given two lines $l_1$ and $l_2$, what is the homogeneous co-ordinate vector $p$ for the point of their intersection?

Same idea: $l_1^T p = p^T l_1 = 0$ and $p^T l_2 = 0$

$p = l_1 \times l_2$

- Cross product between two points gives us the line between them
- Cross product between two lines gives us the point common to both
- What happens if $l_1$ and $l_2$ are parallel?

Answer: Third co-ordinate of $l_1 \times l_2$ is 0. Point at infinity.
HOMOGENEOUS CO-ORDINATES: 2D

Transformations

- Translation:
  - $x' = x - c_x, y' = y - c_y$
- Express as, $p' = T \ p$ where $T$ is a $3 \times 3$ matrix.
Transformations

- Translation:
  - \( x' = x - c_x, y' = y - c_y \)

\[
\begin{bmatrix}
1 & 0 & -c_x \\
0 & 1 & -c_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

- Verify this works for any scaled version of \( T \) above
- Verify this works for \( p = [\alpha x, \alpha y, \alpha] \), for any \( \alpha \neq 0 \)
Transformations

- Rotation Around the Origin
  - $x' = x \cos \theta - y \sin \theta$, $x \sin \theta + y \cos \theta$

  $p' = \begin{bmatrix} \cos \theta & - \sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} p$

- Rotation around a different point $c_x, c_y$?

  $p' = \begin{bmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & - \sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ 0 & 0 & 1 \end{bmatrix} p$
Transformations

- Euclidean Transformation

\[ p' = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} p \]

- \( R \) is a 2 \times 2 rotation matrix, \( R^T R = I \)
- \( t \) is a 2 \times 1 translation vector
- \( 0^T \) here represents a 1 \times 2 row of two zeros
- Preserves orientation, lengths, areas

If \( R^T R = I \), is \( R \) always of the form:

\[
R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}
\]
HOMOGENEOUS CO-ORDINATES: 2D

Transformations

- Euclidean Transformation

\[ p' = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} p \]

- \( R \) is a 2 \( \times \) 2 rotation matrix, \( R^T R = I \)
- \( t \) is a 2 \( \times \) 1 translation vector
- \( 0^T \) here represents a 1 \( \times \) 2 row of two zeros
- Preserves orientation, lengths, areas

If \( R^T R = I \), is \( R \) always of the form:

\[ R = \begin{bmatrix} -\cos \theta & -\sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \]
HOMOGENEOUS CO-ORDINATES: 2D

Transformations

- Euclidean Transformation
  \[ p' = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} p \]
  - \( R \) is a \( 2 \times 2 \) rotation matrix, \( R^T R = I \)
  - \( t \) is a \( 2 \times 1 \) translation vector
  - \( 0^T \) here represents a \( 1 \times 2 \) row of two zeros
  - Preserves orientation, lengths, areas

- Isometries
  - \( R^T R = I \) can also correspond to reflections
    \[ R = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \]
  - If we allow this in \( R \) above, more general than euclidean
  - Preserves lengths, areas, but not orientation.
Transformations

What about scaling?

Allow uniform scaling $s$ along both co-ordinates:

$$p' = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} p$$

Called a similarity: preserves ratio of lengths, angles.
HOMOGENEOUS CO-ORDINATES: 2D

Transformations

Affine Transformation

\[ p' = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} p \]

where \( A \) is a general invertible \( 2 \times 2 \).

Preserves ratios of areas, parallel lines stay parallel.

Prove that parallel lines stay parallel.

- Consider the homogeneous vector \( q \) for intersection of two lines that are parallel.
- The third co-ordinate of \( q \) is 0, because the lines don’t intersect.
- The affine transform doesn’t change the third co-ordinate.
- Hence, the lines still intersect at infinity after the transformation.
HOMOGENEOUS CO-ORDINATES: 2D

Most general form:

\[ p' = Hp \]

where \( H \) is a general invertible \( 3 \times 3 \) matrix.

- Called a projective transform or **homography**.
- All bets are off! Parallel lines can now intersect. Maps quadrilaterals to quadrilaterals.
- The only certainty you have is that straight lines will remain straight (won’t map straight lines to curves).

Why?

\[ p = \alpha p_1 + (1 - \alpha)p_2 \Rightarrow p' = Hp = \alpha Hp_1 + (1 - \alpha)Hp_2 = \alpha p'_1 + (1 - \alpha)p'_2 \]
HOMOGENEOUS CO-ORDINATES: 2D

Most general form:

\[ p' = Hp \]

where \( H \) is a general invertible \( 3 \times 3 \) matrix.

- Called a projective transform or \textbf{homography}.
- All bets are off! Parallel lines can now intersect. Maps quadrilaterals to quadrilaterals.
- The only certainty you have is that straight lines will remain straight (won’t map straight lines to curves).
- \( H \) is a \( 3 \times 3 \) matrix defined upto scale. So 8 degrees of freedom.

Hierarchy of Transforms

- Translation (2 dof) < Euclidean (3 dof) < Affine (6 dof) < Homography (8 dof)

- Defines mapping of co-ordinates of corresponding points in two images taken from different views:
  - If all corresponding points lie on a plane in the world.
  - If only the camera orientation has changed in two views (camera center is at the same place)