CSE 559A: Computer Vision

Fall 2019: T-R: 11:30-12:50pm @ Hillman 60

Instructor: Ayan Chakrabarti (ayan@wustl.edu).
Course Staff: Arushee Agrawal, Annie Lee, Jiahao Li, Xiaochen Zhou

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

Sep 26, 2019
• Reminder: PSET 2 out.
  ■ Get started early!
• Start thinking about final project: Proposal due soon.
PHOTOMETRIC STEREO++

- 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++

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

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

$\ell_R$, $\ell_G$, $\ell_B$

$\rho_R$, $\rho_G$, $\rho_B$

$\hat{n}$

$I_R = \ell_R^T(\rho_R, \hat{n})$

$I_G = \ell_G^T(\rho_G, \hat{n})$

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

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.

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

Multiplex in color!!

Three observations, five unknowns.

(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)
  - 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 ......)
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 soft elastomer with a reflective coating on one side. The unique properties of the GelSight sensor reduce the influence of the optical characteristics of the material on the measurement, thereby ensuring high 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
    sensor that can measure light (IR) in space (and time).

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

Multiplex in color!!

Three observations, five unknowns.

(so close ......)
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

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

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

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

Cumbersome, so sometimes approximated using "spherical harmonics"

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

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

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

\[ Z \quad \text{shape / depth} \quad \rightarrow \quad S(Z, L) \quad \text{log-shading of } Z \text{ and } L \quad \rightarrow \quad L \quad \text{illumination} \]

\[ R \quad \text{log-reflectance} \quad \rightarrow \quad I = R + S(Z, L) \quad \text{Input log-image} \]

minimize \[ Z, R, L \]

\[ g(R) + f(Z) + h(L) \]

Suggested Reading:

Barron & Malik,
"Shape, Reflectance, and Illumination from Shading," PAMI 2015.
Remember, the pinhole camera

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

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

- 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
HOMOGENEOUS CO-ORDINATES

Book-keeping trick!

- 2D Cartesian Co-ordinates: \((x, y)\)
- 2D Homogeneous Co-ordinates: \((\alpha x, \alpha y, \alpha)\)
- Cartesian to Homogeneous: \((x, y) \rightarrow (\alpha x, \alpha y, \alpha)\)
  - 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
  - \textit{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: \((x, y, z) \Rightarrow (\alpha x, \alpha y, \alpha z, \alpha)\)
HOMOGENEOUS CO-ORDINATES

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

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

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

Homogeneous co-ordinates

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

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?

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

**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

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

"Rays" in $\mathbb{R}^3$

\[
\begin{align*}
\text{X-axis} & \quad \text{Y-axis} \\
\text{Z-axis} & \quad \text{Z-axis}
\end{align*}
\]
HOMOGENEOUS CO-ORDINATES

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

"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.
QUICK WORD ABOUT NOTATION

- 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 \( 2 \times 1 \))
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 \( l \).
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.
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:
  - $x' = x - c_x, y' = y - c_y$
  
  $$p' = \begin{bmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ 0 & 0 & 1 \end{bmatrix} p$$

- 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.
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.