Simple Stereo

Shree K. Nayar

Columbia University

Topic: Camera Calibration, Module: Reconstruction II

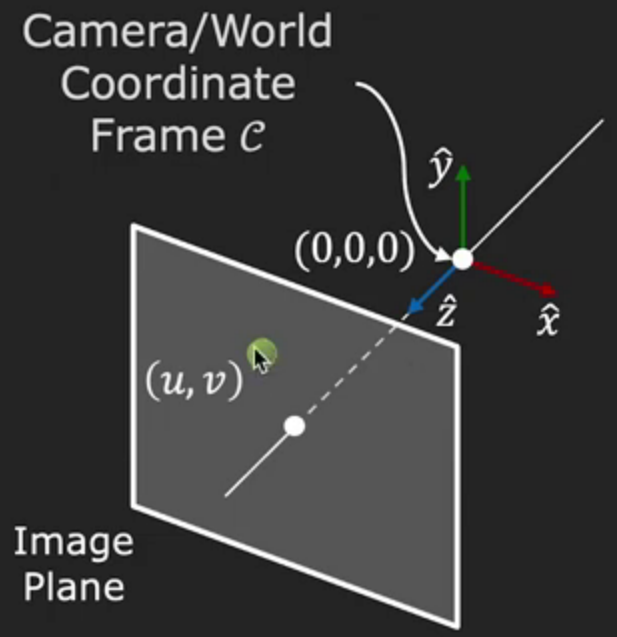
First Principles of Computer Vision

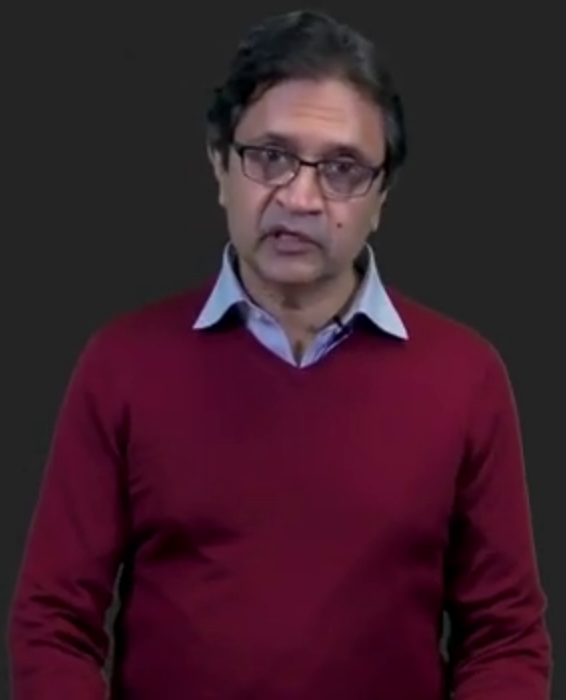
Backward Projection: From 2D to 3D

Given a calibrated camera, can we find the 3D scene point from a single 2D image?

Backward Projection: From 2D to 3D

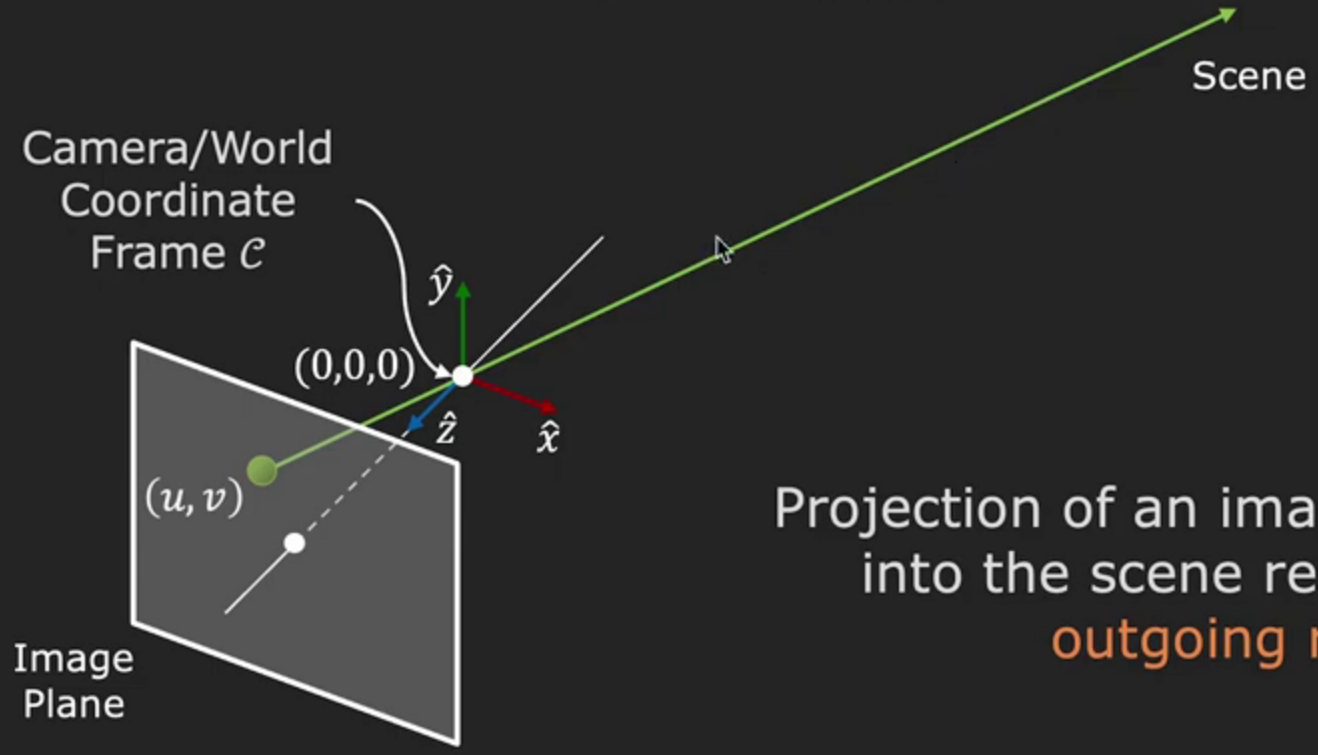
Given a calibrated camera, can we find the 3D scene point from a single 2D image?



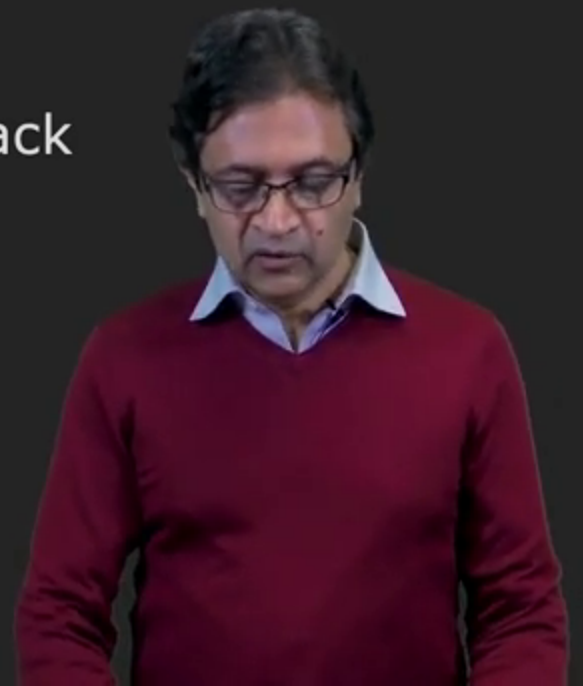


Backward Projection: From 2D to 3D

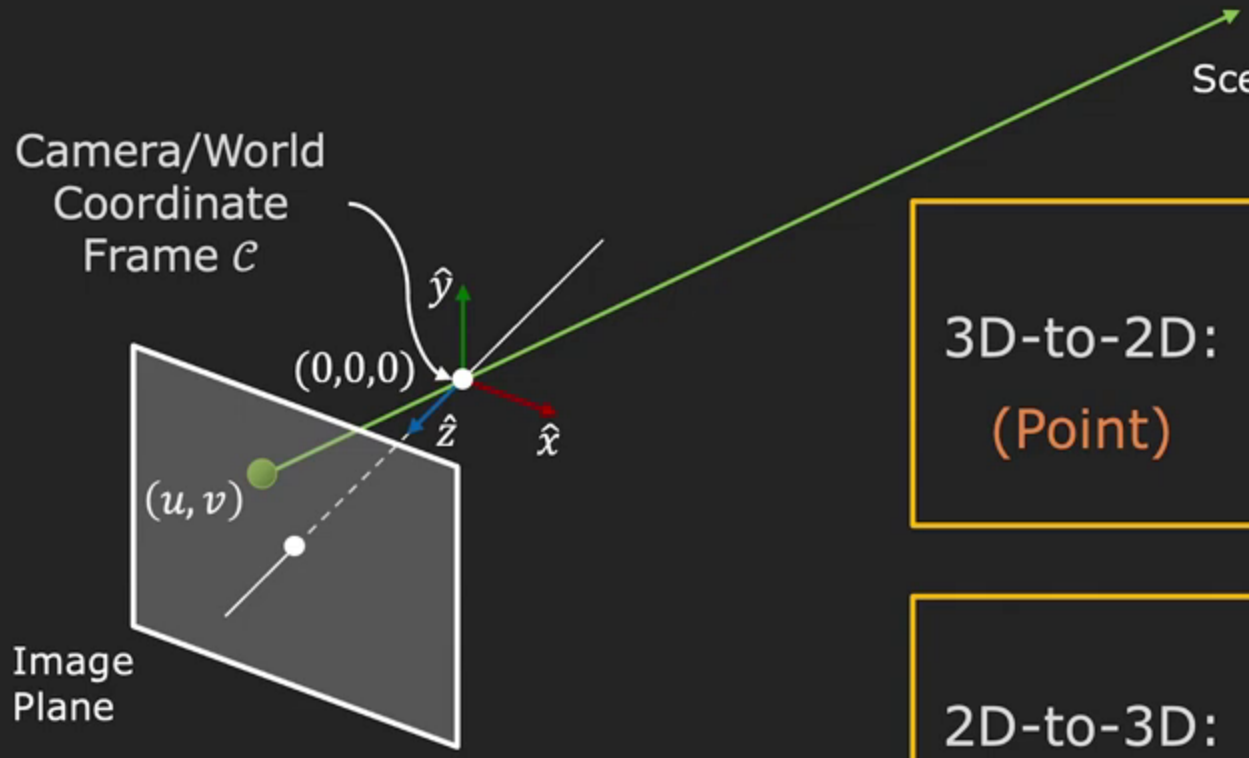
Given a calibrated camera, can we find the 3D scene point from a single 2D image?



Projection of an image point back into the scene results in an outgoing ray.



Computing 2D-to-3D Outgoing Ray



3D-to-2D: (Point)

$$u = f_x \frac{x_c}{z_c} + o_x$$

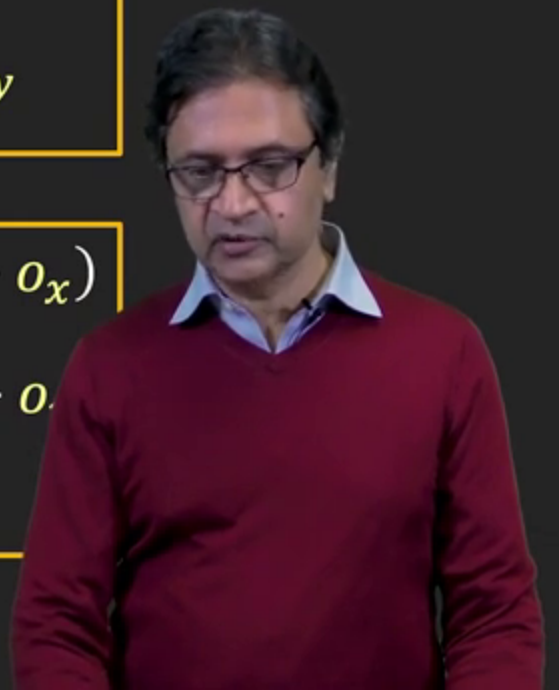
$$v = f_y \frac{y_c}{z_c} + o_y$$

2D-to-3D: (Ray)

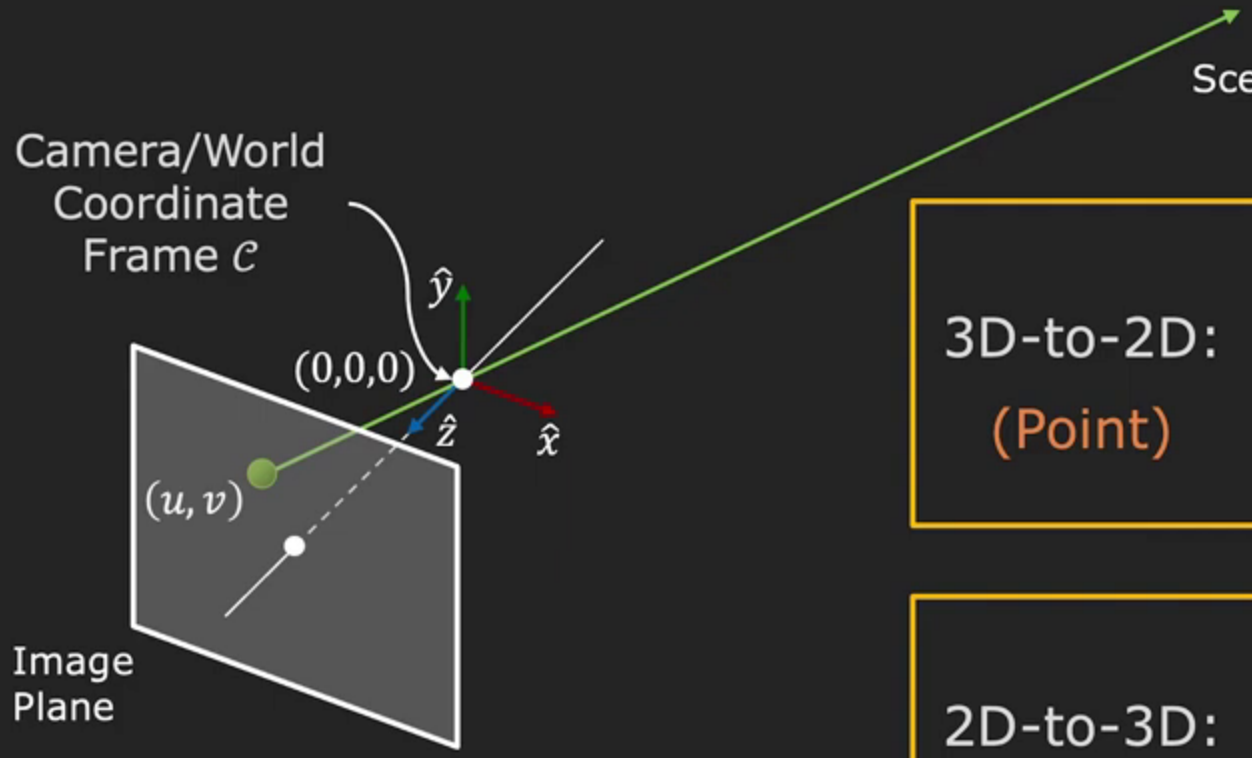
$$x = {}^{z}/_{f_x}\,(u - o_x)$$

$$y = {}^{z}/_{f_y}\,(v - o_y)$$

$$z > 0$$



Computing 2D-to-3D Outgoing Ray



3D-to-2D:
(Point)

$$u = f_x \frac{x_c}{z_c} + o_x$$

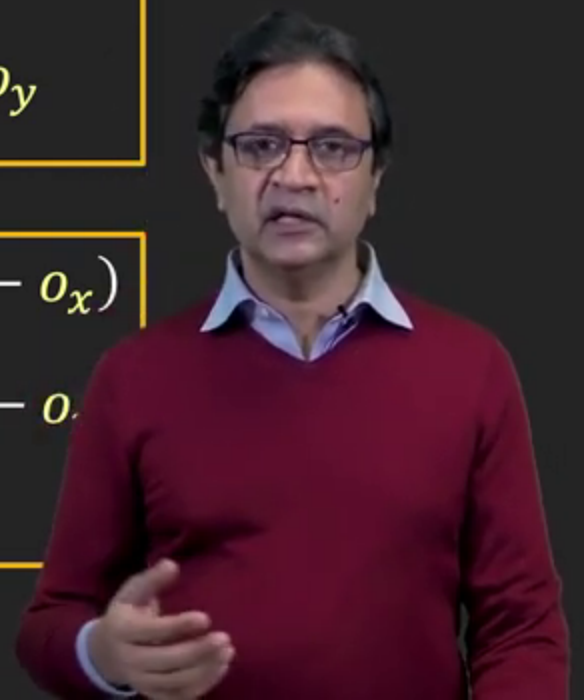
$$v = f_y \frac{y_c}{z_c} + o_y$$

2D-to-3D:
(Ray)

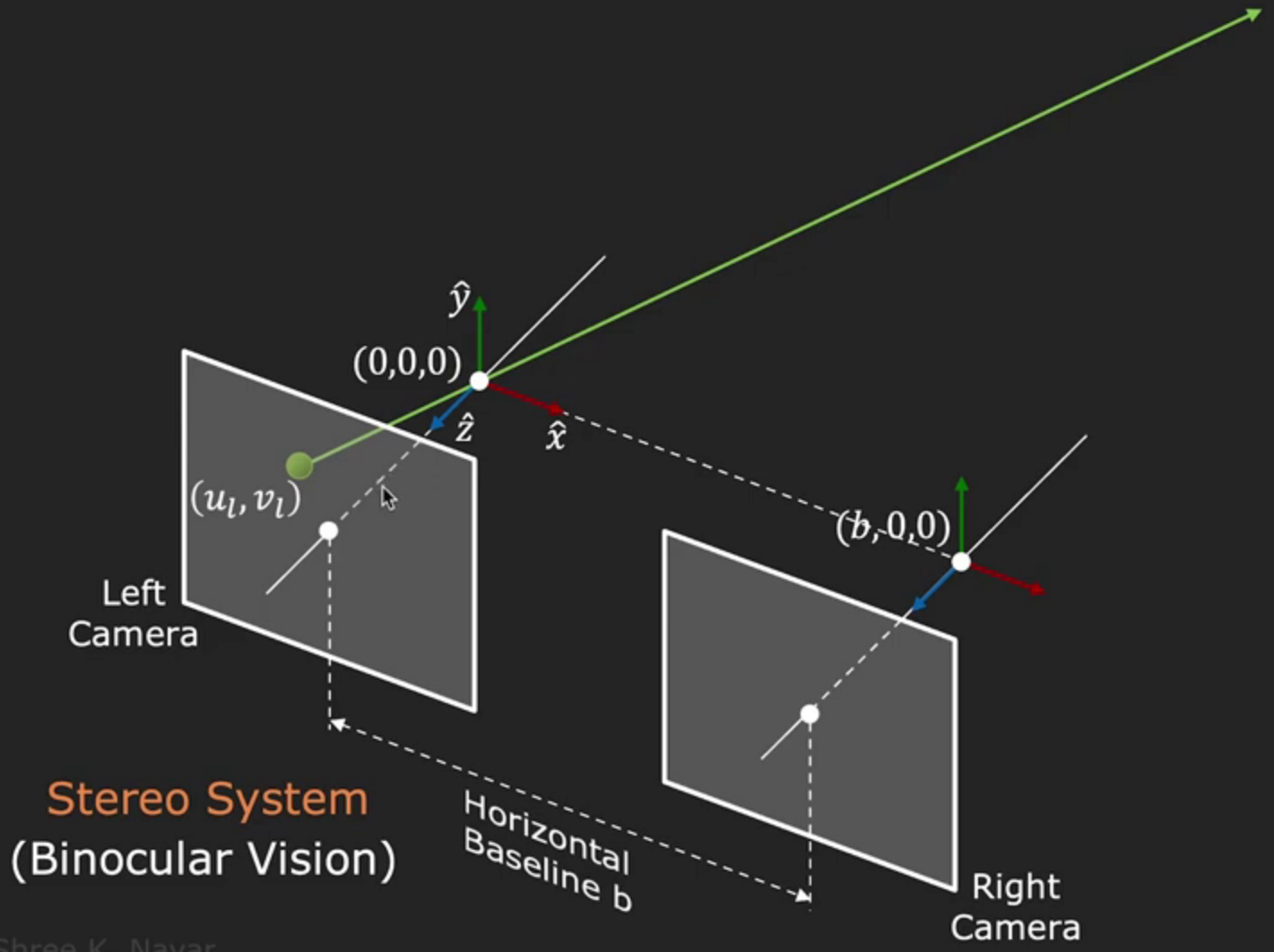
$$x = {}^{z}/_{f_x}\,(u - o_x)$$

$$y = {}^{z}/_{f_y}\,(v - o_y)$$

$$z > 0$$

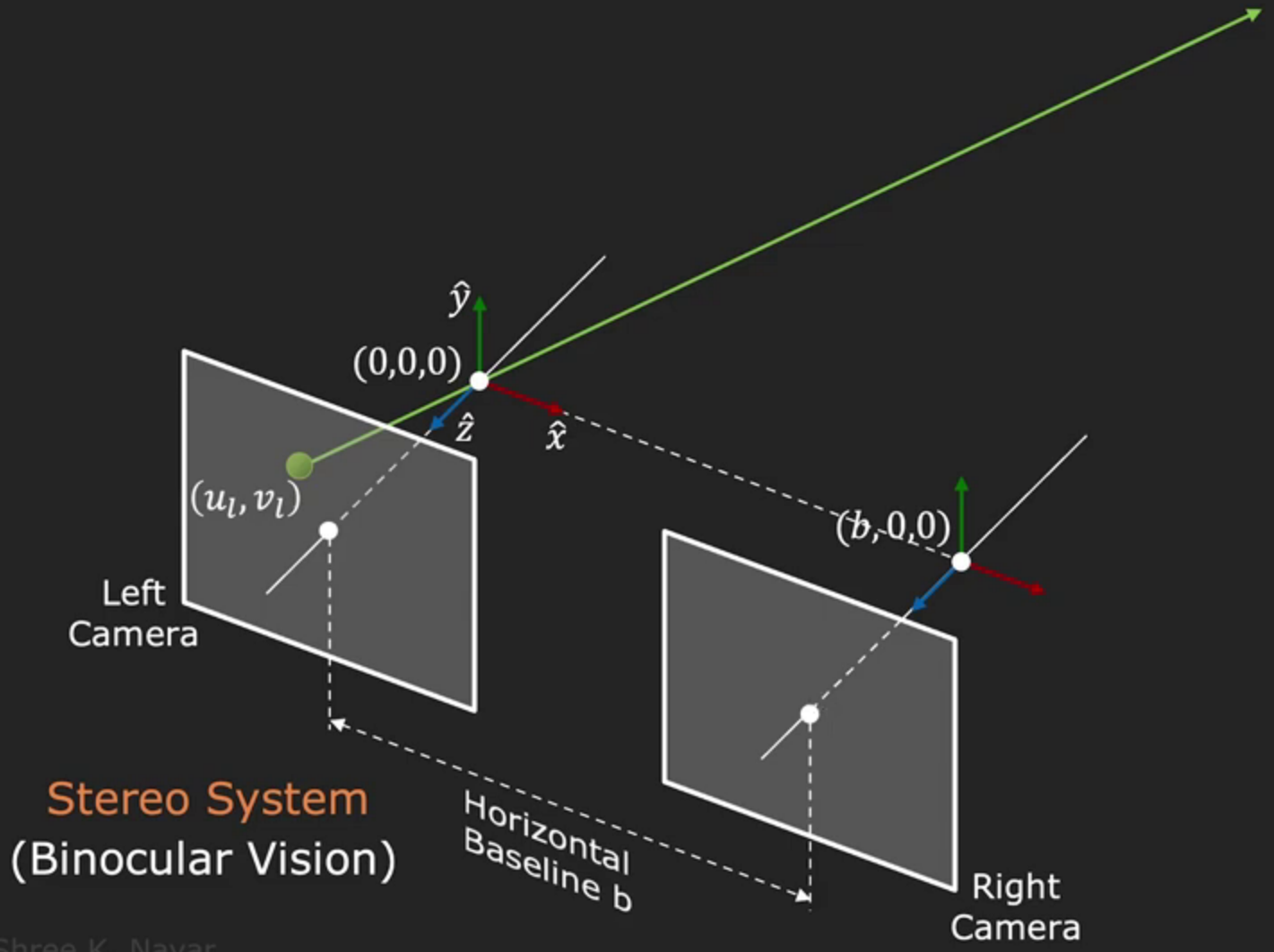


Triangulation using Two Cameras



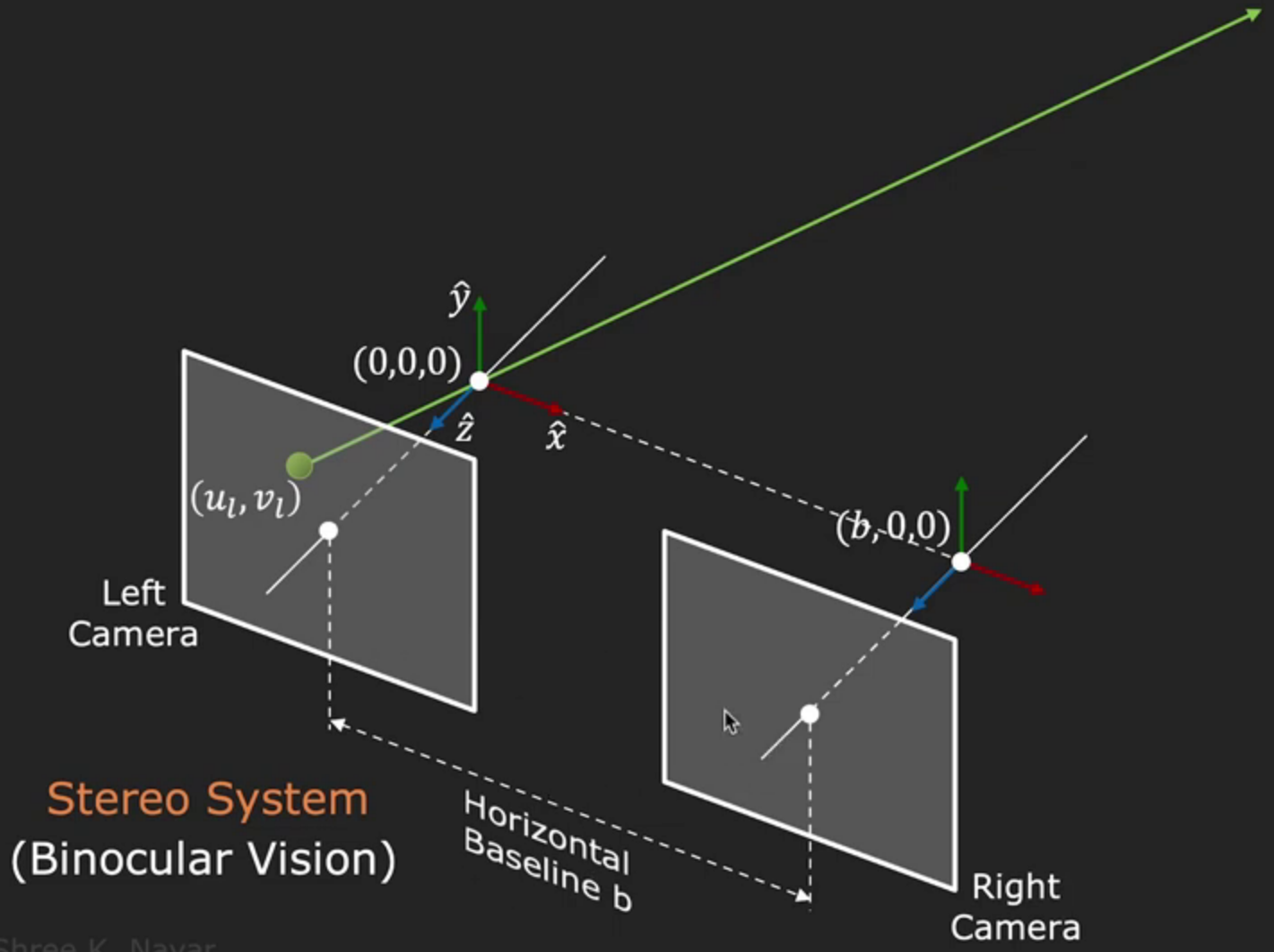
Stereo System
(Binocular Vision)

Triangulation using Two Cameras

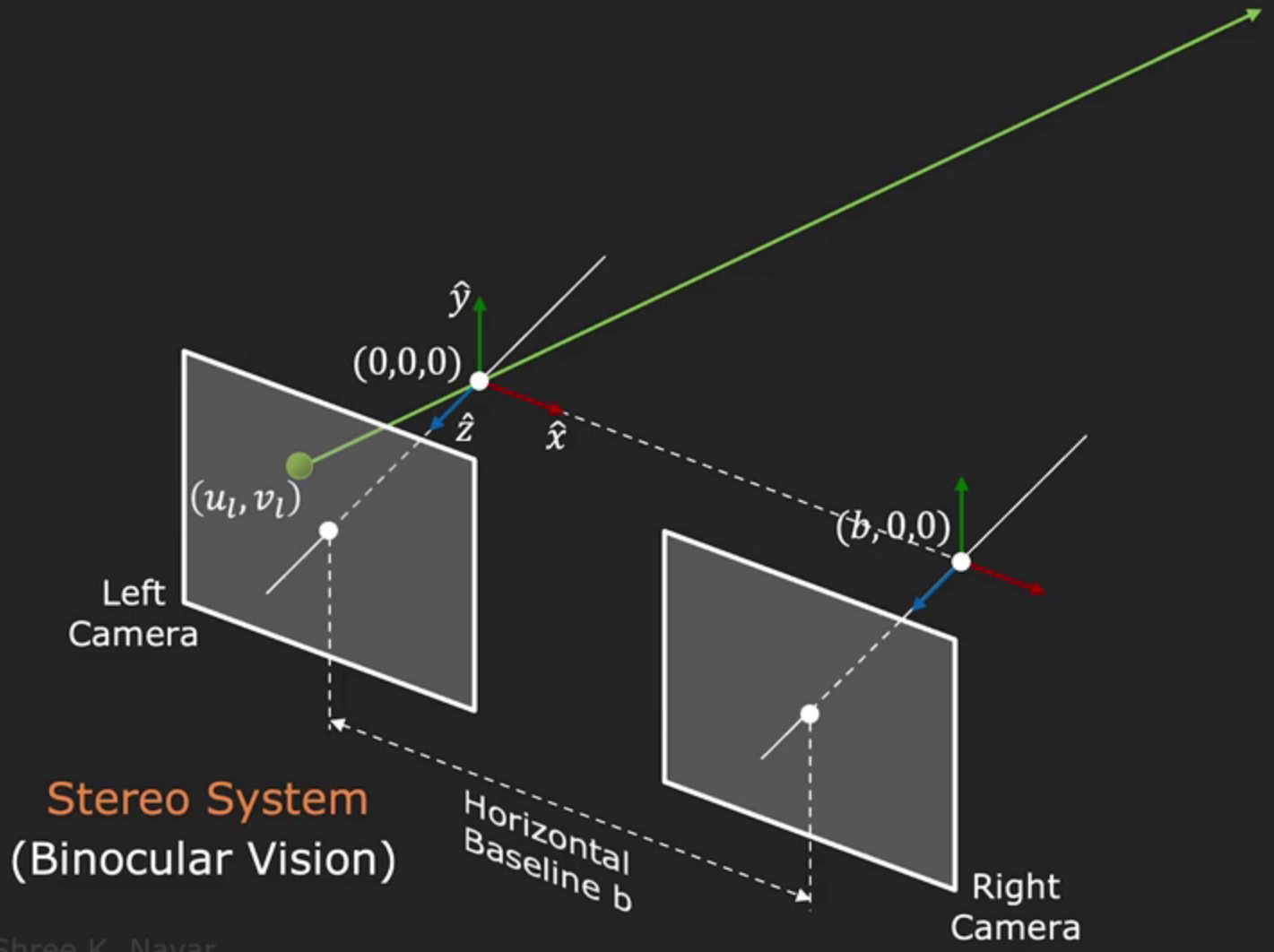


Stereo System
(Binocular Vision)

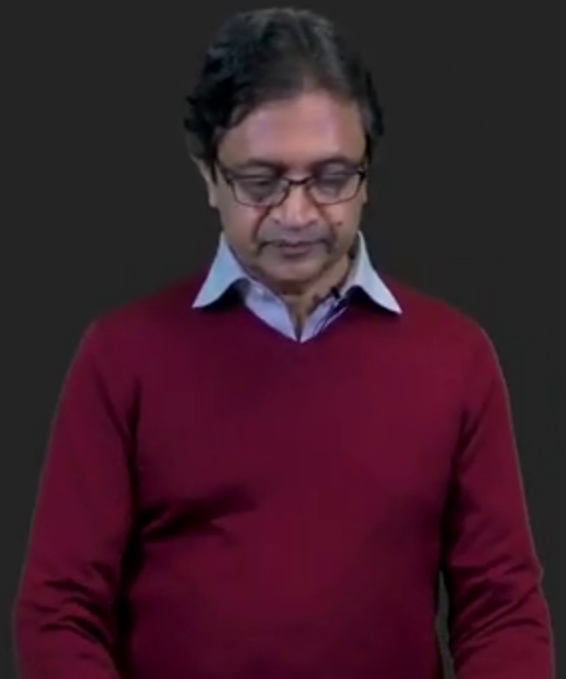
Triangulation using Two Cameras



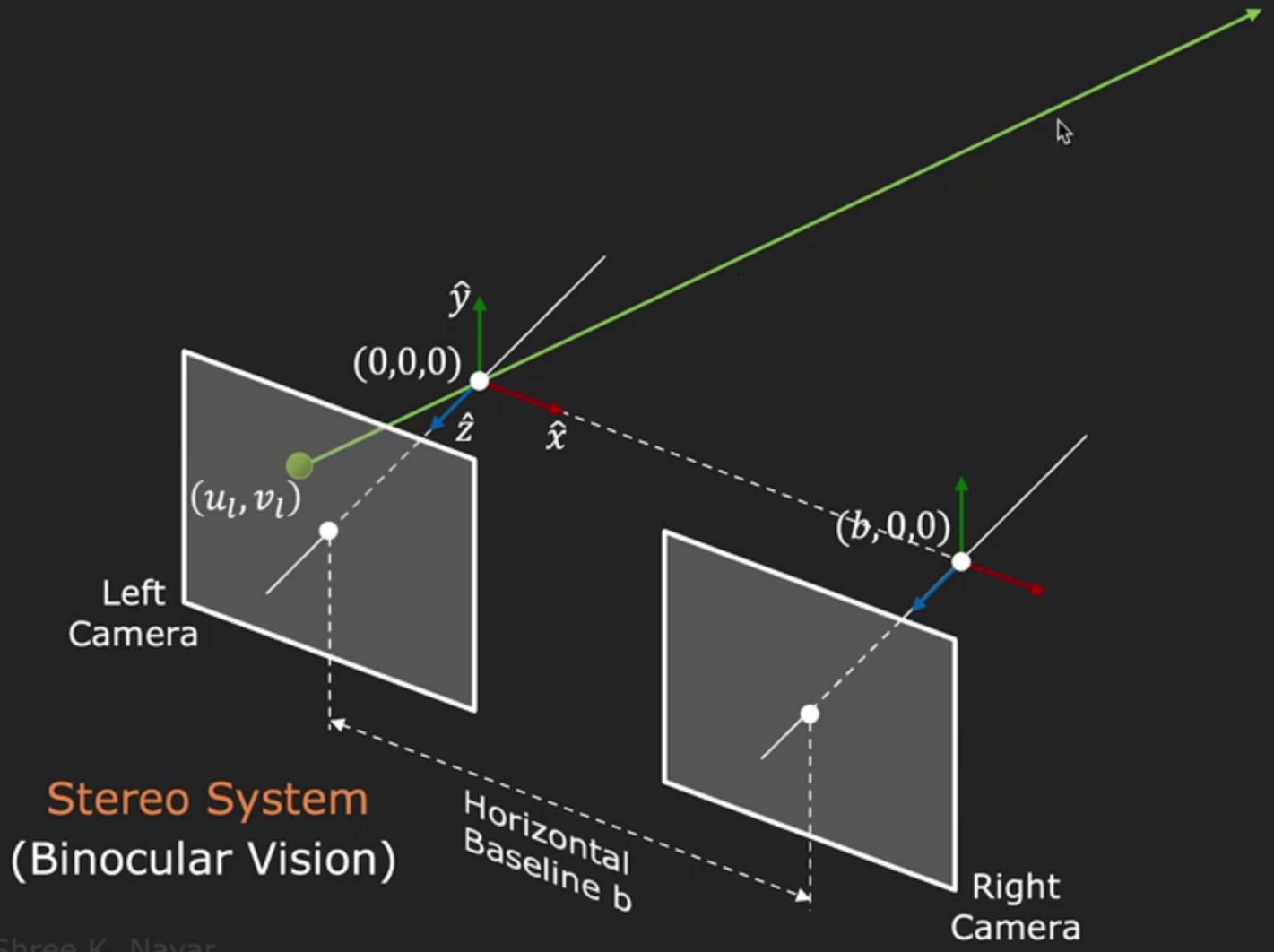
Triangulation using Two Cameras



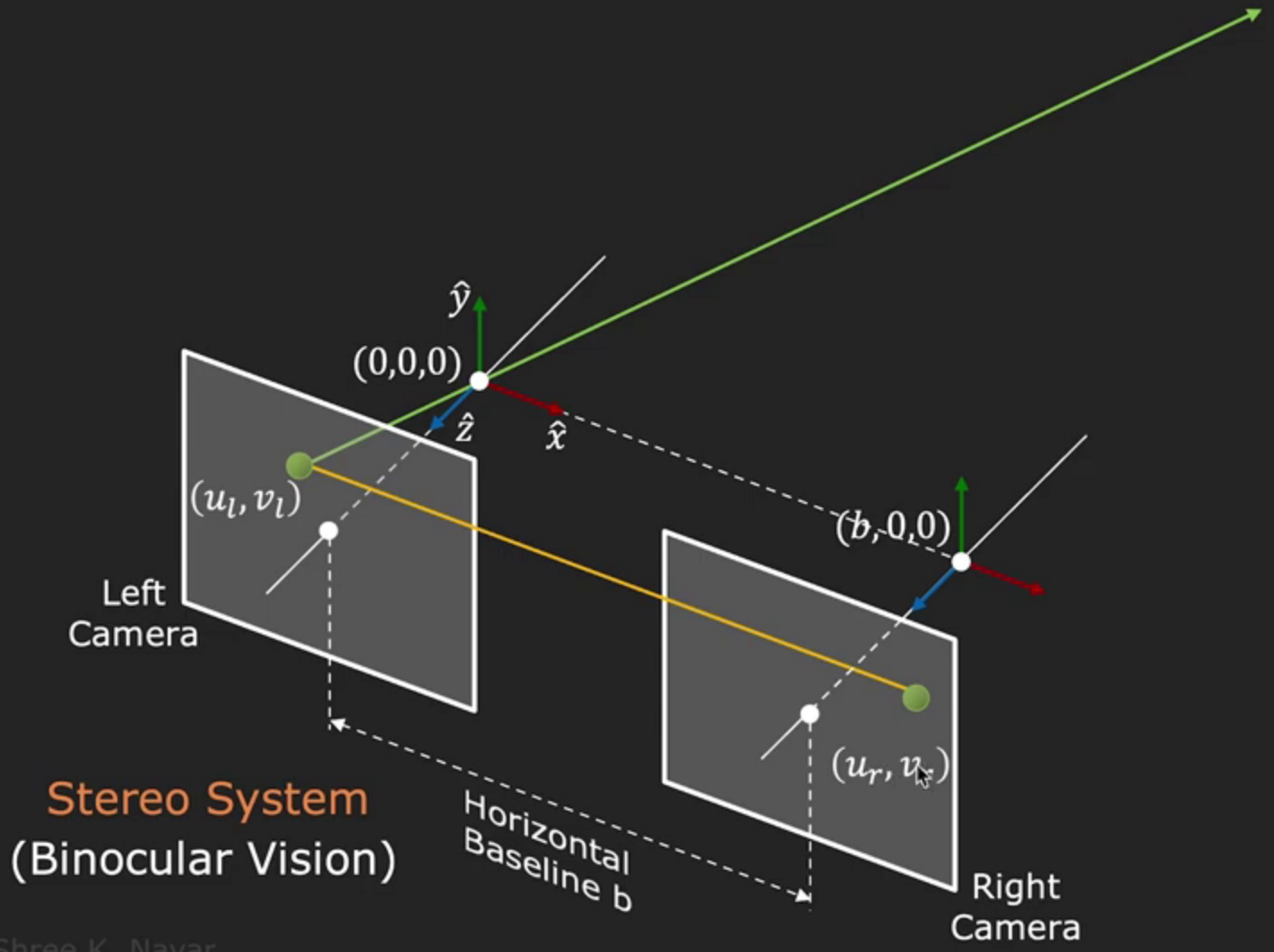
Stereo System
(Binocular Vision)



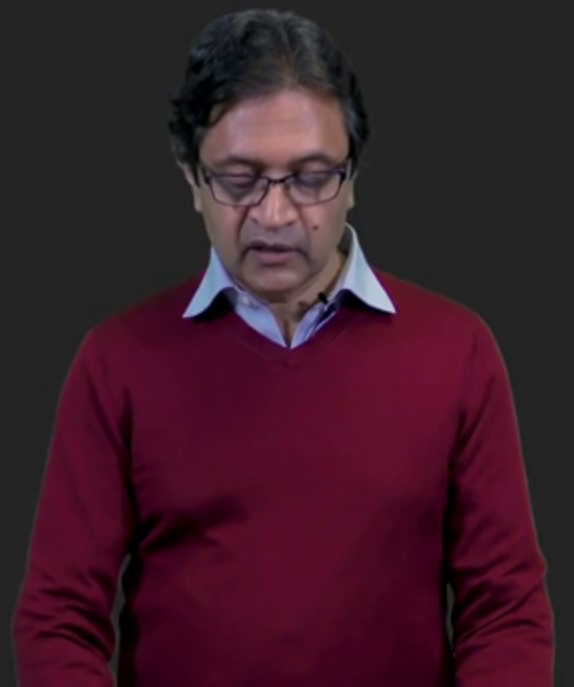
Triangulation using Two Cameras



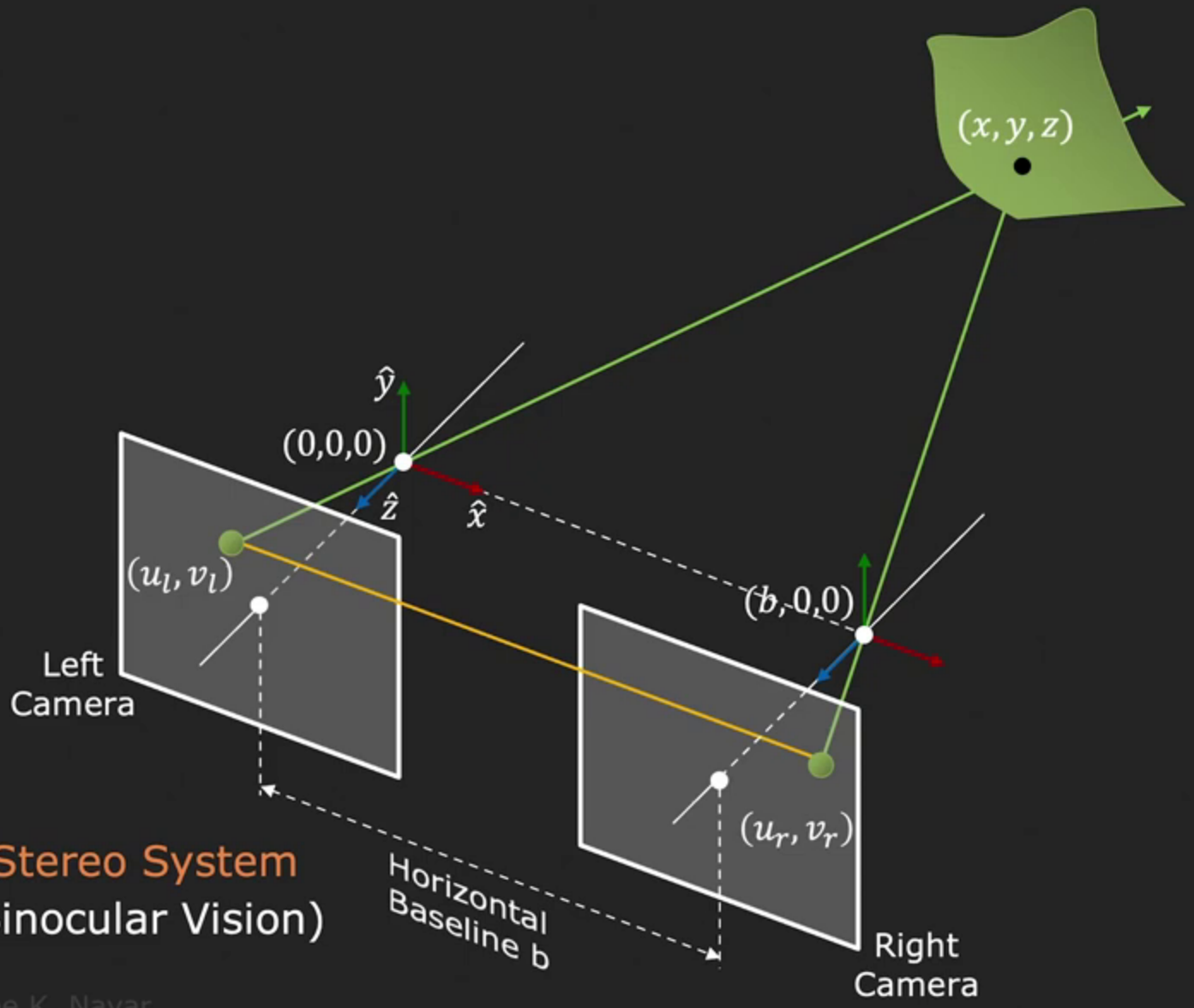
Triangulation using Two Cameras



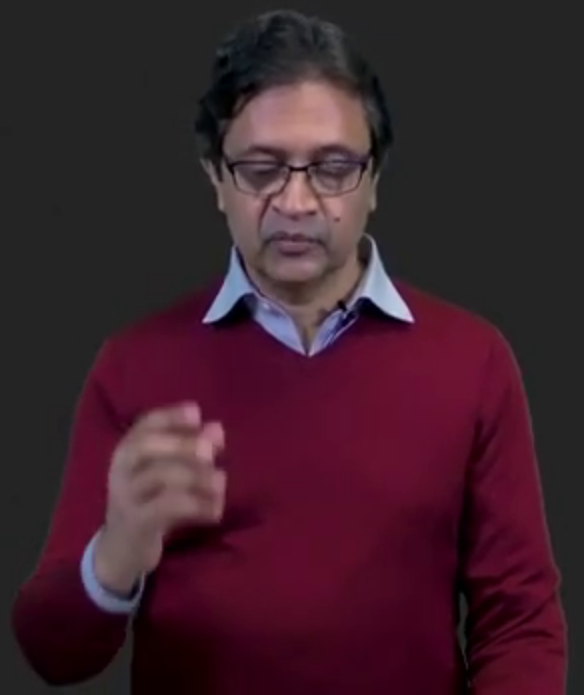
Stereo System
(Binocular Vision)



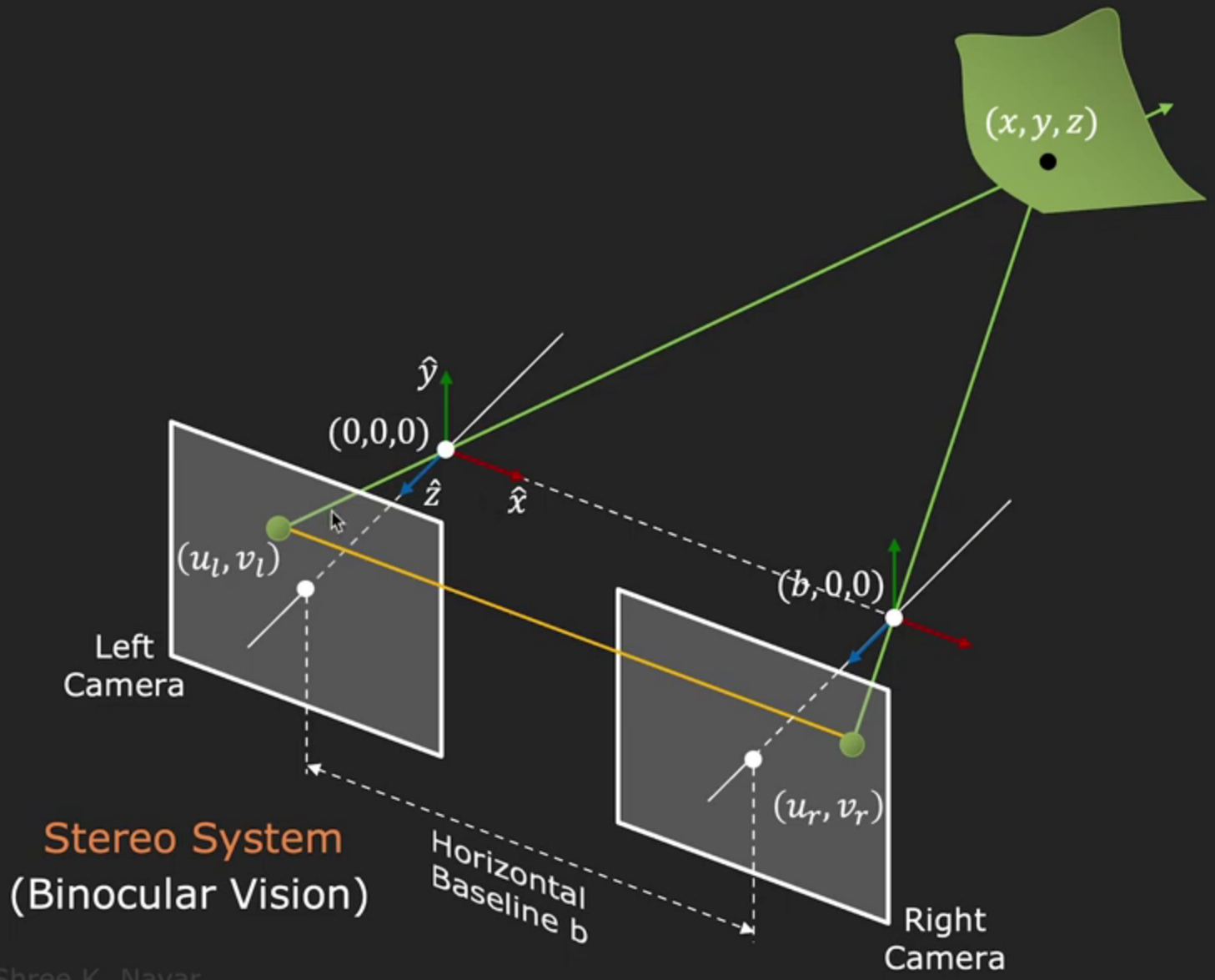
Triangulation using Two Cameras



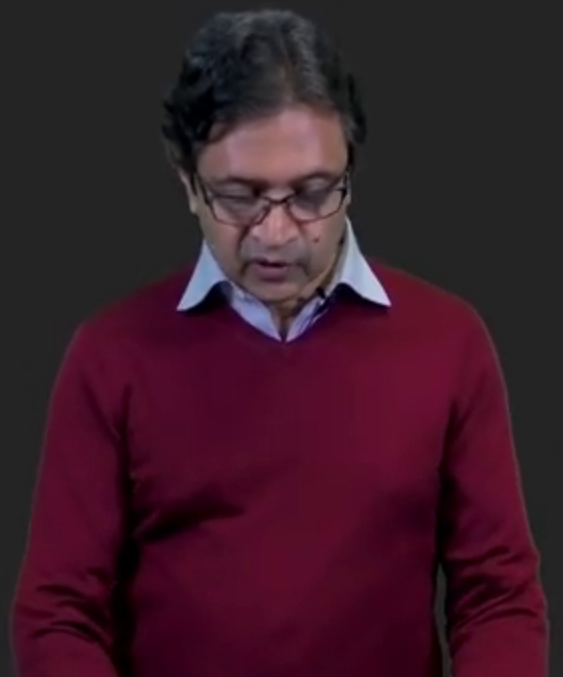
Stereo System
(Binocular Vision)



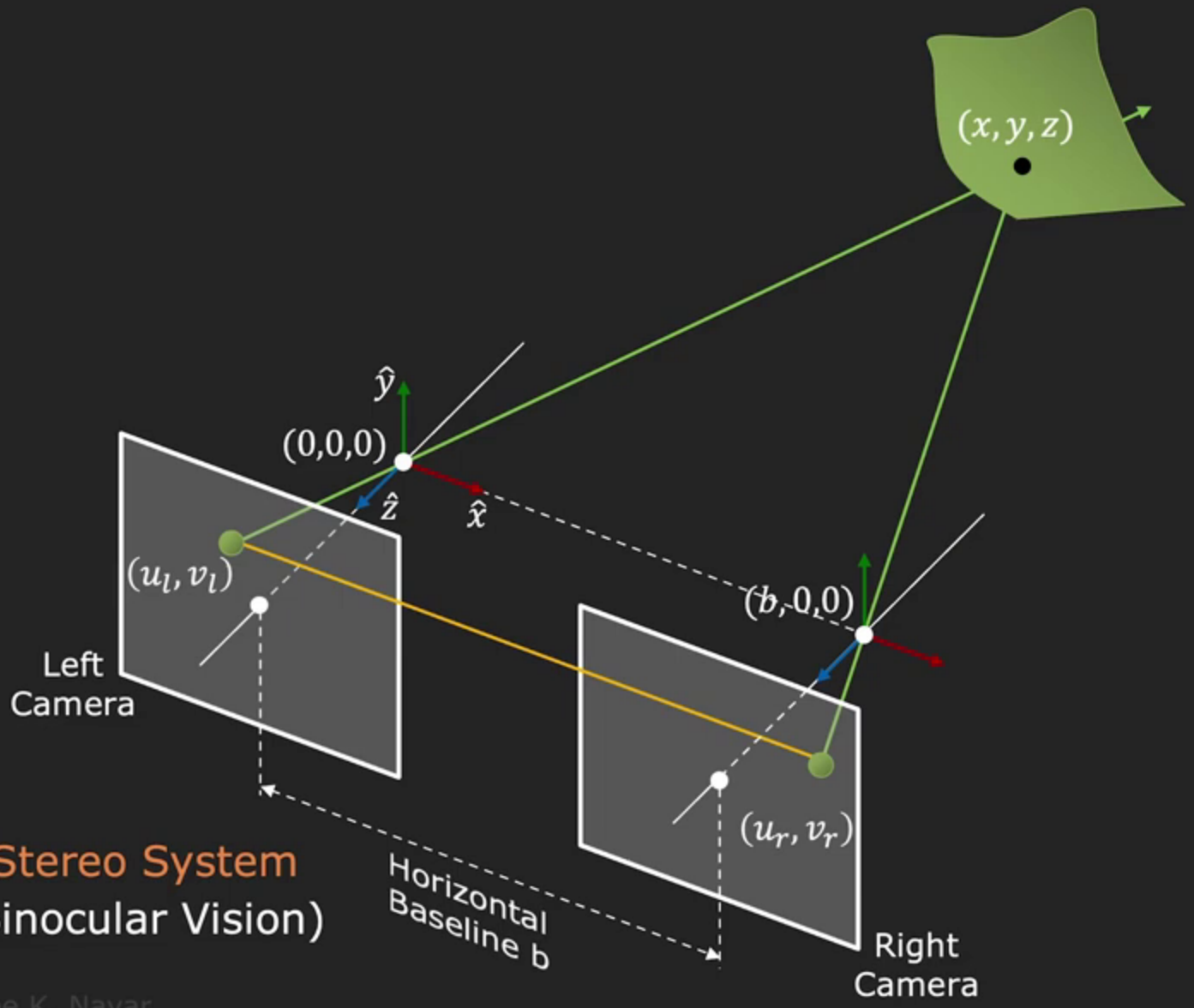
Triangulation using Two Cameras



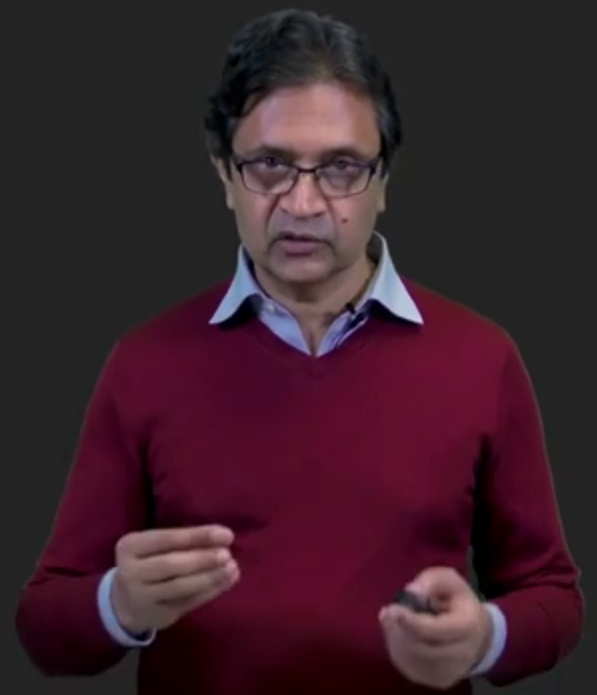
Stereo System
(Binocular Vision)



Triangulation using Two Cameras



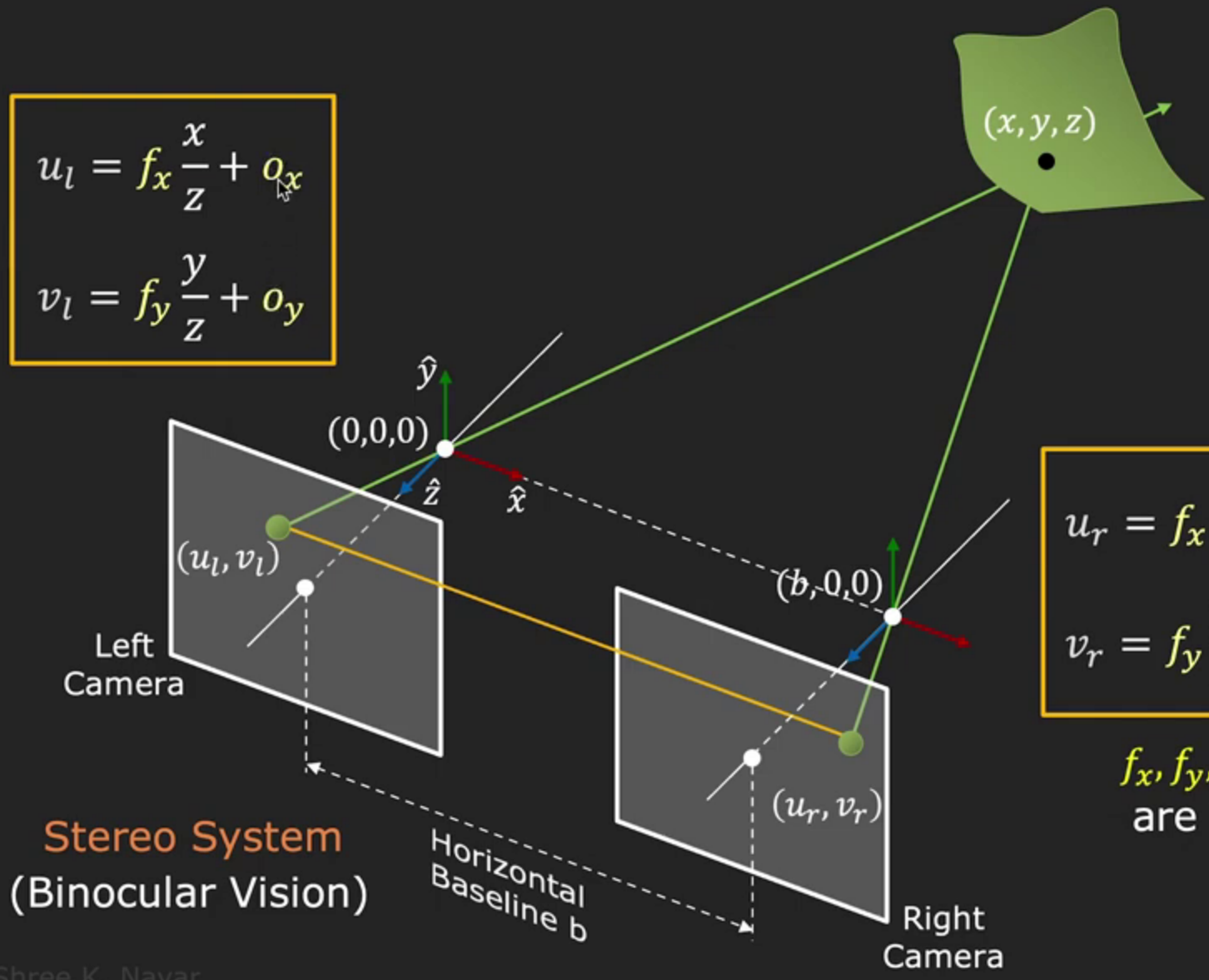
Stereo System
(Binocular Vision)



Triangulation using Two Cameras

$$u_l = f_x \frac{x}{z} + o_x$$

$$v_l = f_y \frac{y}{z} + o_y$$



Stereo System
(Binocular Vision)

$$u_r = f_x \frac{x - b}{z} + o_x$$

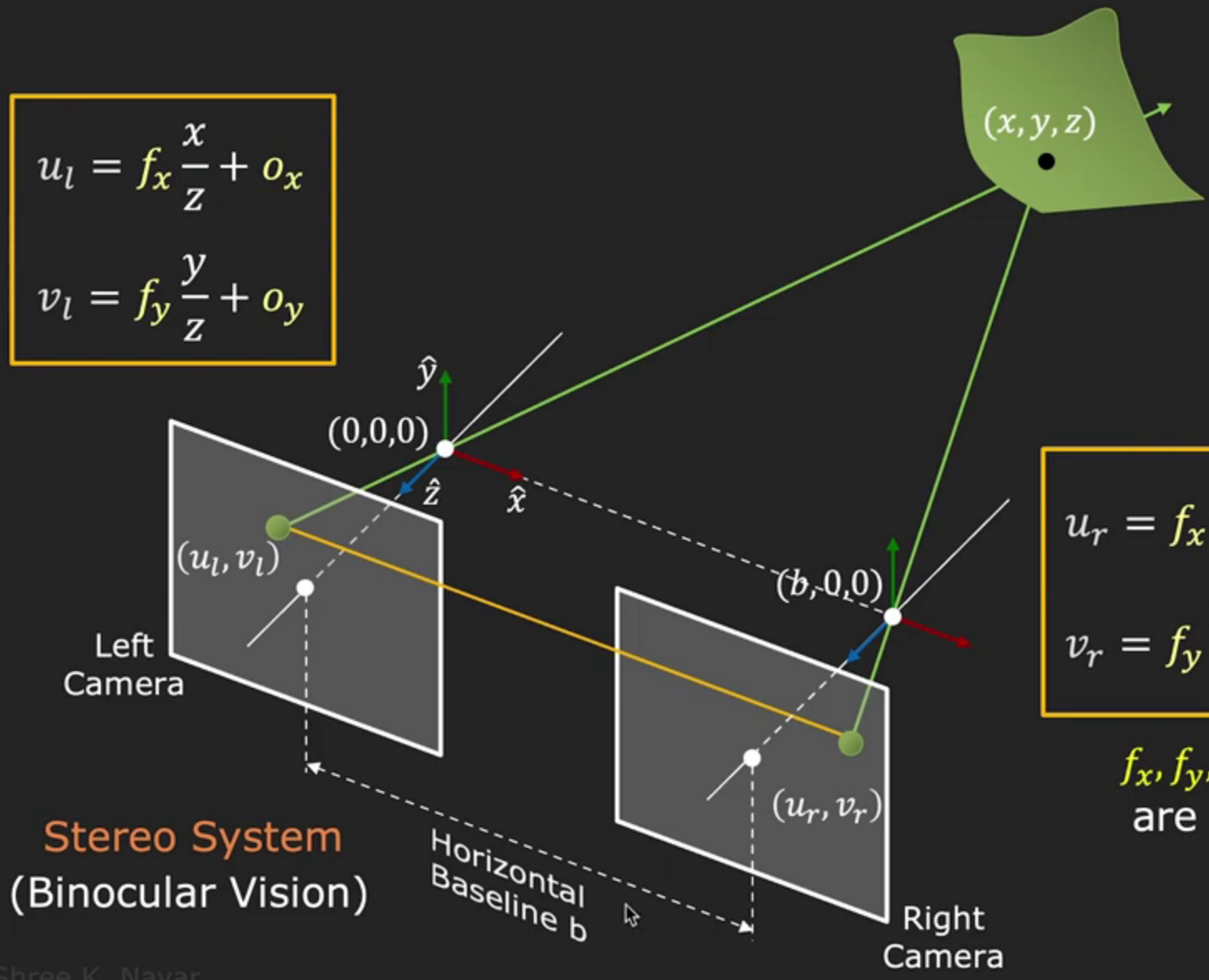
$$v_r = f_y \frac{y}{z} + o_y$$

f_x, f_y, b, o_x, o_y are known.

Triangulation using Two Cameras

$$u_l = f_x \frac{x}{z} + o_x$$

$$v_l = f_y \frac{y}{z} + o_y$$



Stereo System
(Binocular Vision)

$$u_r = f_x \frac{x - b}{z} + o_x$$

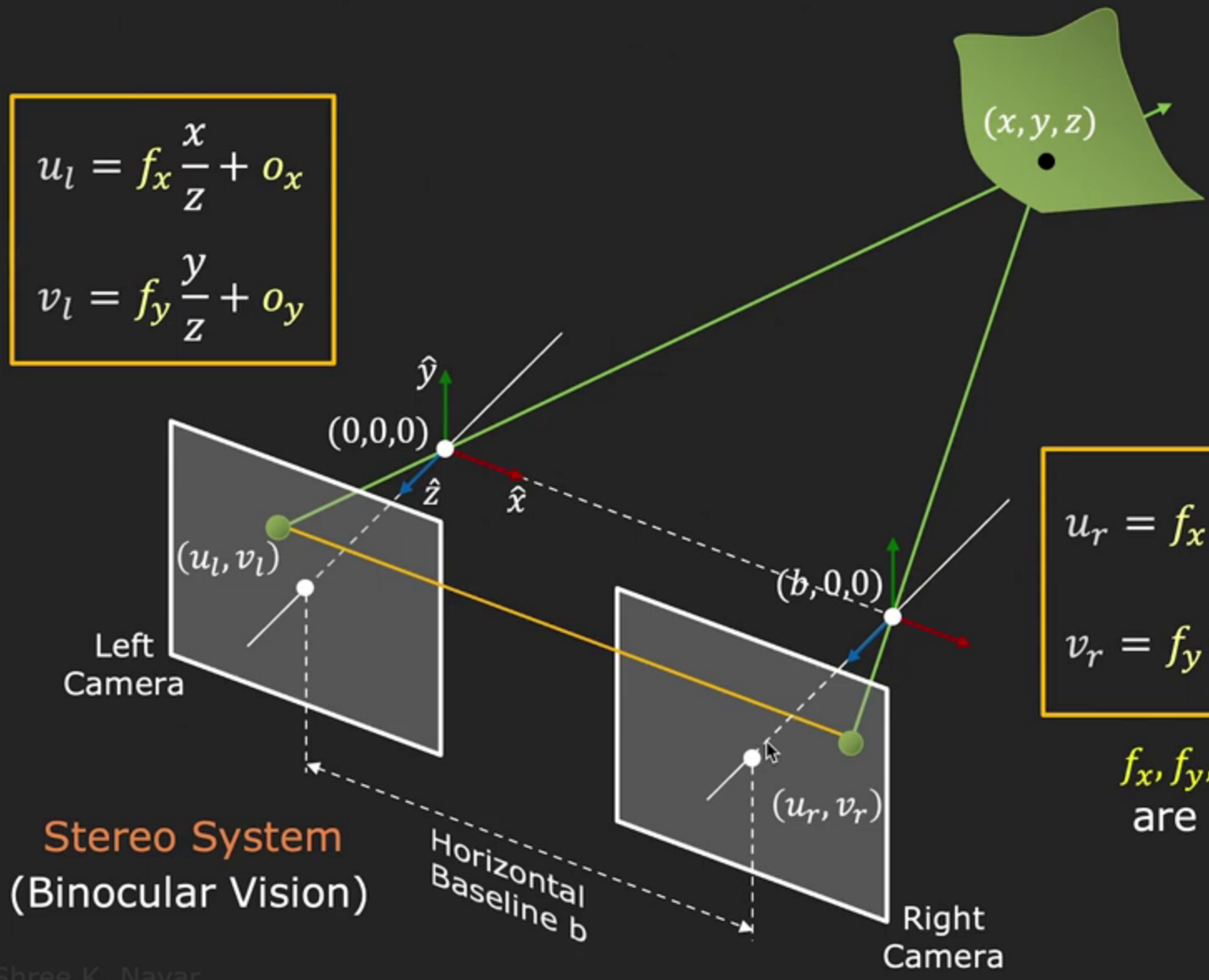
$$v_r = f_y \frac{y}{z} + o_y$$

f_x, f_y, b, o_x, o_y
are known.

Triangulation using Two Cameras

$$u_l = f_x \frac{x}{z} + o_x$$

$$v_l = f_y \frac{y}{z} + o_y$$



Stereo System
(Binocular Vision)

$$u_r = f_x \frac{x - b}{z} + o_x$$

$$v_r = f_y \frac{y}{z} + o_y$$

f_x, f_y, b, o_x, o_y are known.

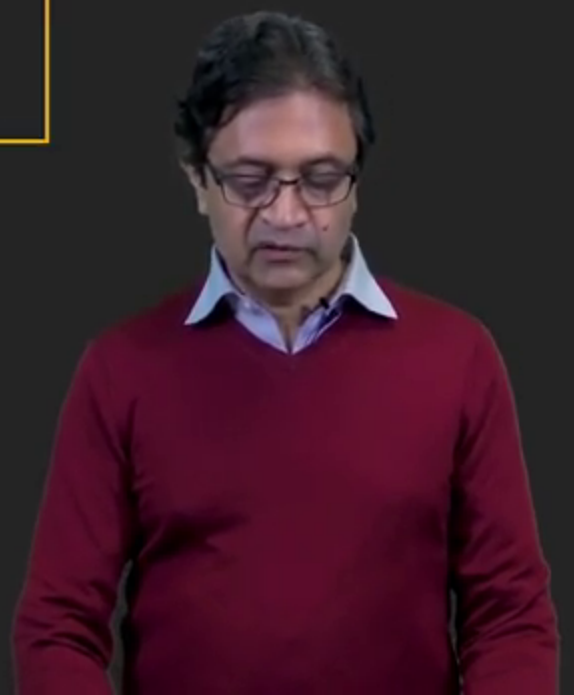
Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$



Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

where $(u_l - u_r)$ is called Disparity.

Depth z is inversely proportional to Disparity.

Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

where $(u_l - u_r)$ is called Disparity.

Depth z is inversely proportional to Disparity.

Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

where $(u_l - u_r)$ is called Disparity.

Depth z is inversely proportional to Disparity.

Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

where $(u_l - u_r)$ is called Disparity.

Depth z is inversely proportional to Disparity.

Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

where $(u_l - u_r)$ is called Disparity.

Depth z is inversely proportional to Disparity.

Disparity is proportional to Baseline.

Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

where $(u_l - u_r)$ is called Disparity.

Depth z is inversely proportional to Disparity.

Disparity is proportional to Baseline.

Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

where $(u_l - u_r)$ is called Disparity.

Depth z is inversely proportional to Disparity.

Disparity is proportional to Baseline.

Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

where $(u_l - u_r)$ is called Disparity.

Depth z is inversely proportional to Disparity.

Disparity is proportional to Baseline.

A Simple Stereo Camera

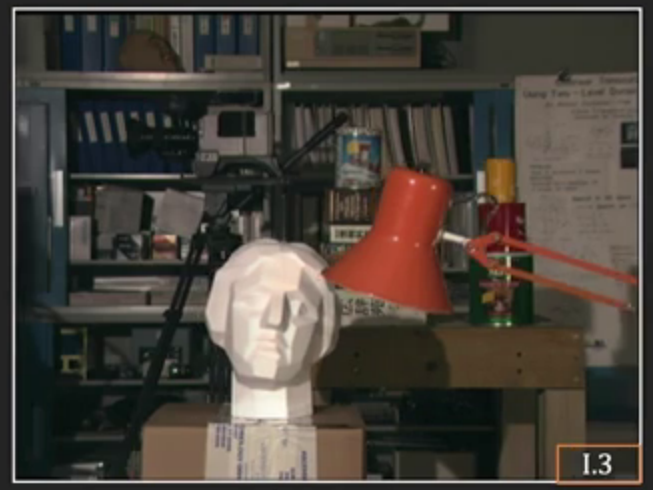
Fujifilm FinePix REAL 3D W3

A Simple Stereo Camera

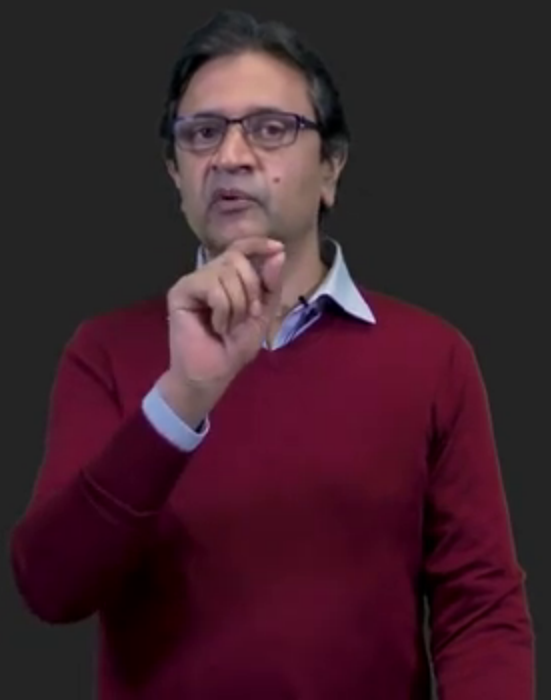
Fujifilm FinePix REAL 3D W3

Stereo Matching: Finding Disparities

Goal: Find the disparity between left and right stereo pairs.

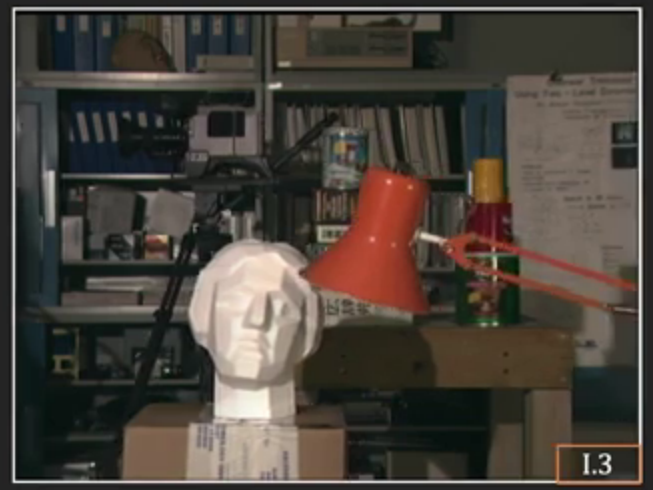


Left/Right Camera Images

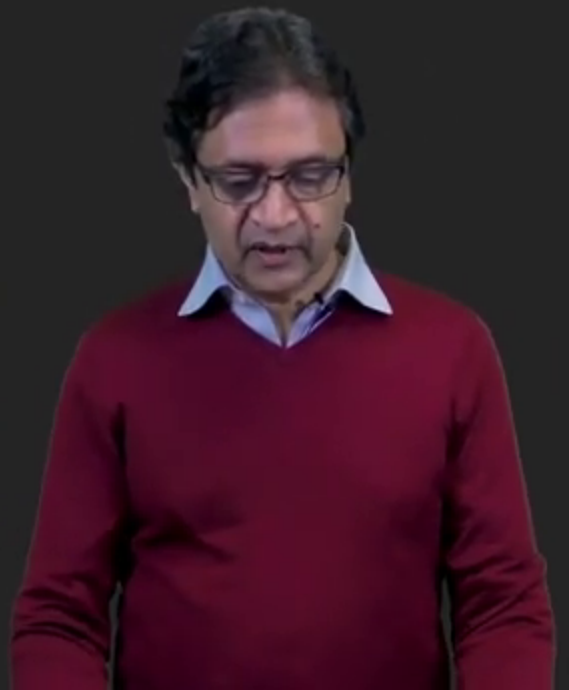


Stereo Matching: Finding Disparities

Goal: Find the disparity between left and right stereo pairs.

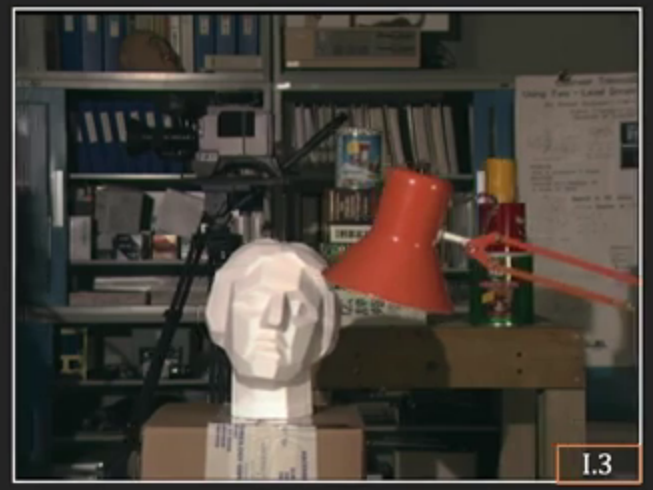


Left/Right Camera Images



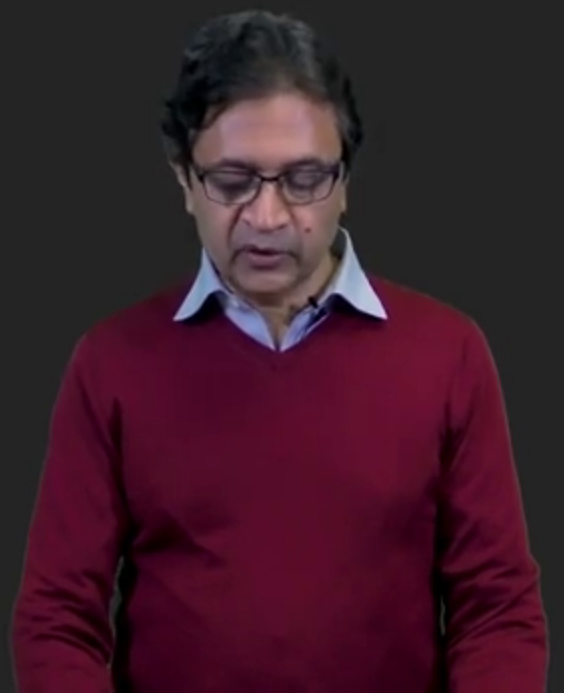
Stereo Matching: Finding Disparities

Goal: Find the disparity between left and right stereo pairs.



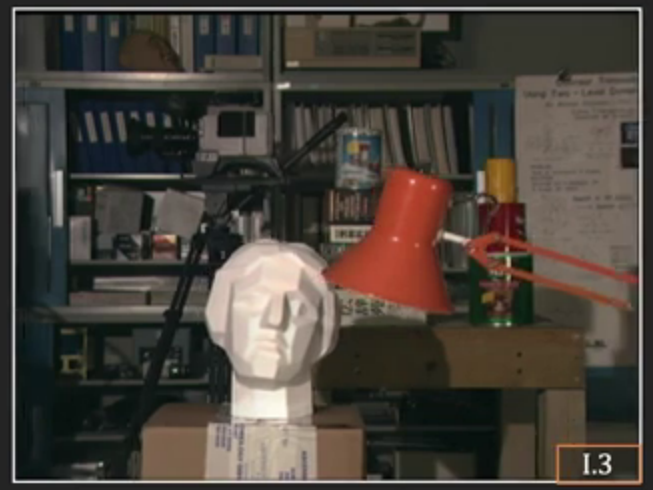
Left/Right Camera Images

Disparity Map (Ground Truth)



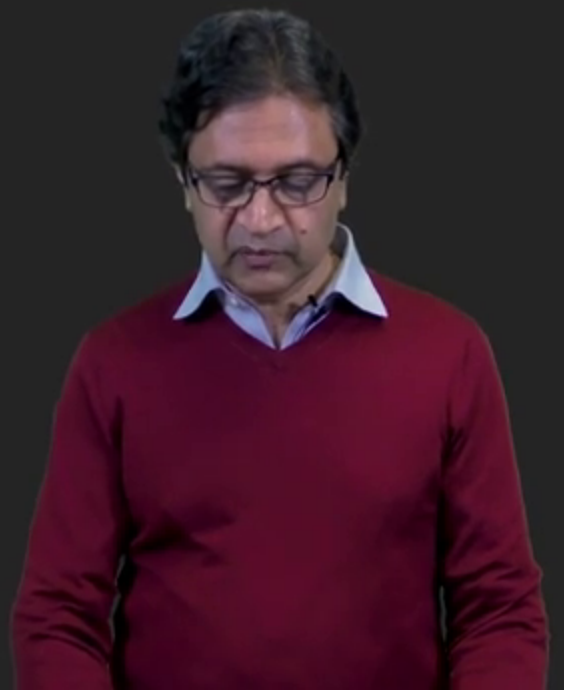
Stereo Matching: Finding Disparities

Goal: Find the disparity between left and right stereo pairs.



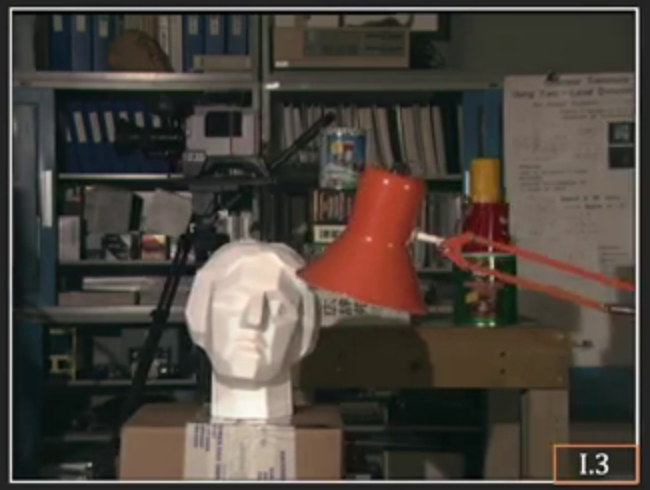
Left/Right Camera Images

Disparity Map (Ground Truth)

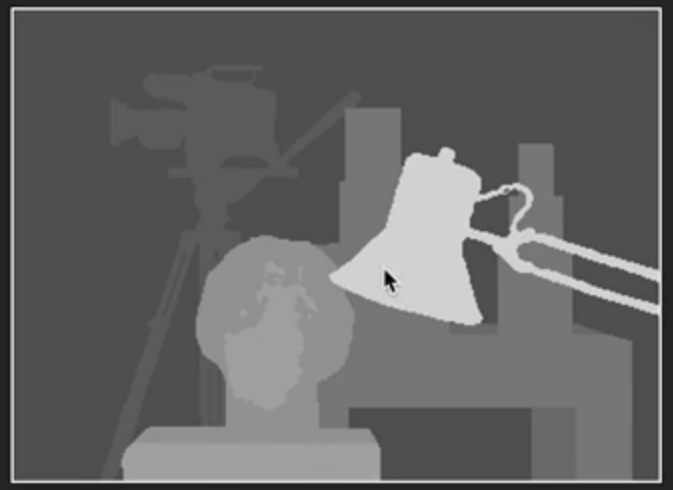


Stereo Matching: Finding Disparities

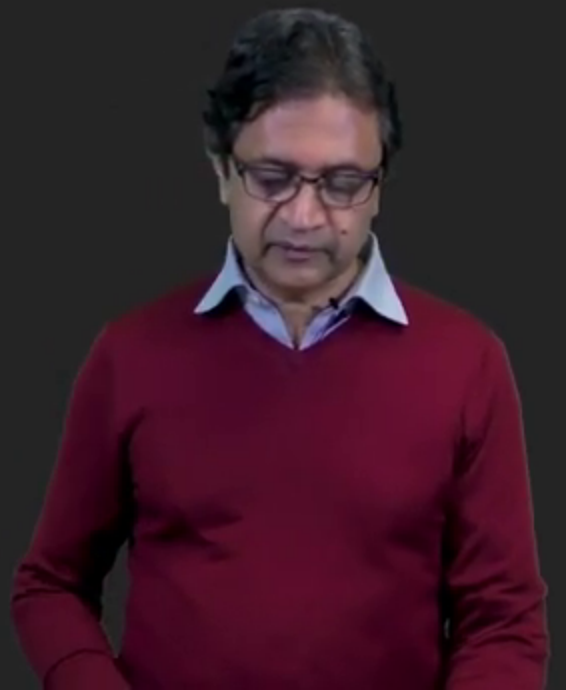
Goal: Find the disparity between left and right stereo pairs.



Left/Right Camera Images

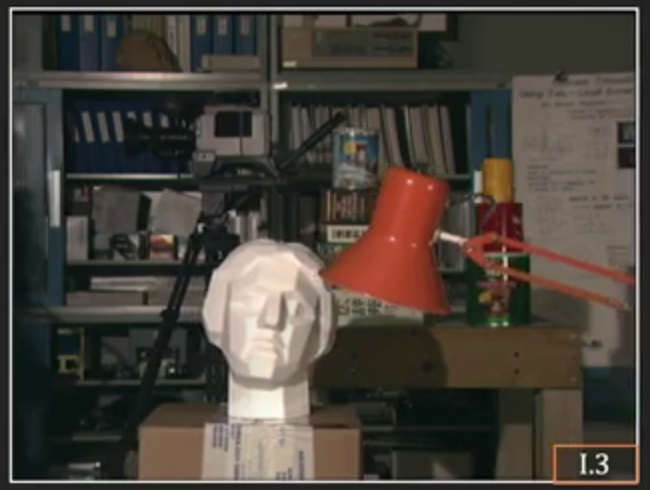


Disparity Map (Ground Truth)



Stereo Matching: Finding Disparities

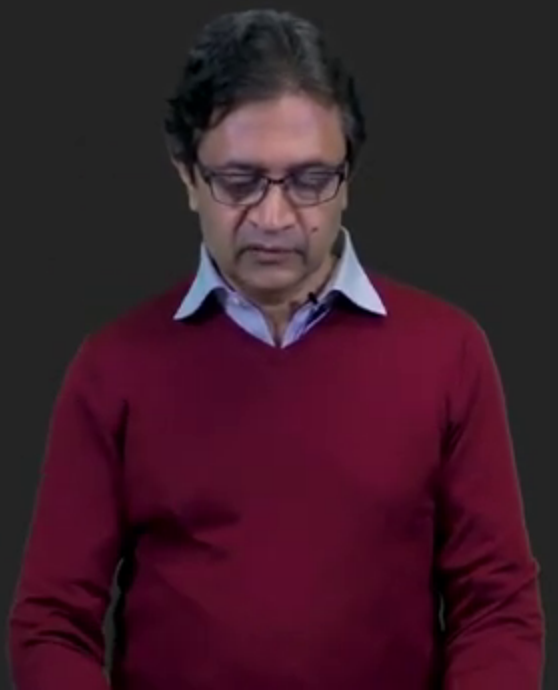
Goal: Find the disparity between left and right stereo pairs.



Left/Right Camera Images

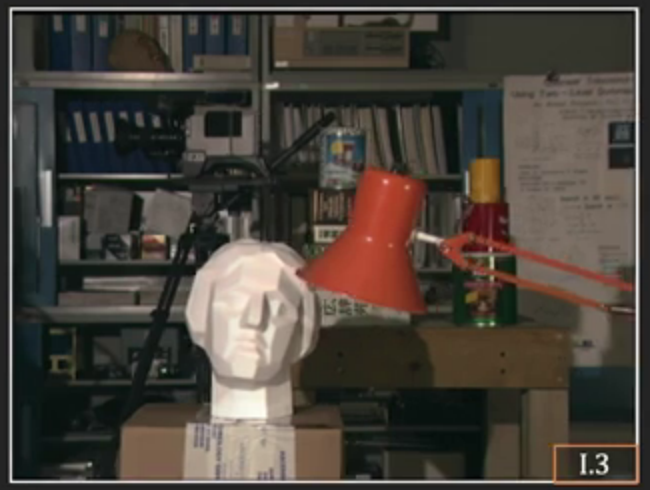
Disparity Map (Ground Truth)

From perspective projection: $v_l = v_r = f_y \frac{y}{z} + o_y$



Stereo Matching: Finding Disparities

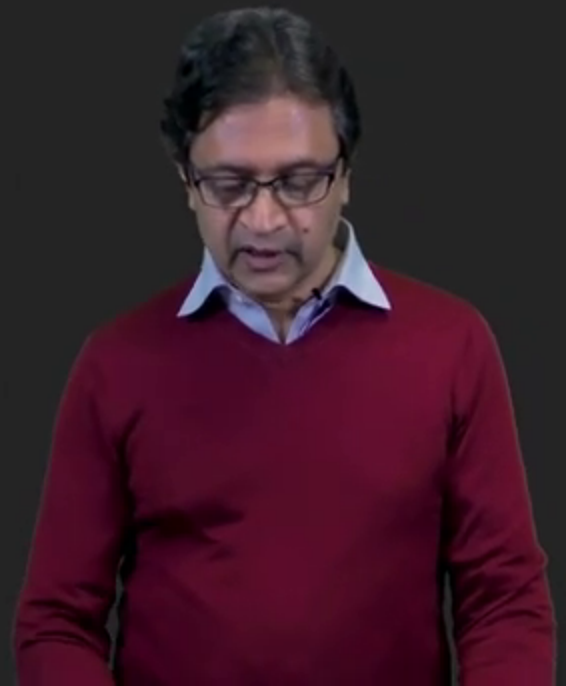
Goal: Find the disparity between left and right stereo pairs.



Left/Right Camera Images

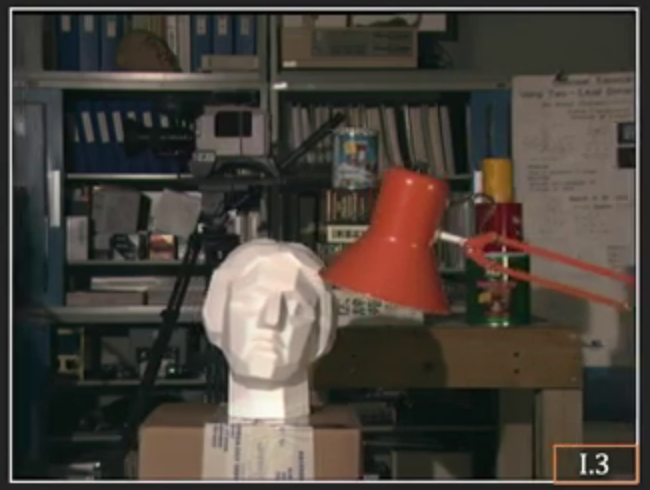
Disparity Map (Ground Truth)

From perspective projection: $v_l = v_r = f_y \frac{y}{z} + o_y$



Stereo Matching: Finding Disparities

Goal: Find the disparity between left and right stereo pairs.

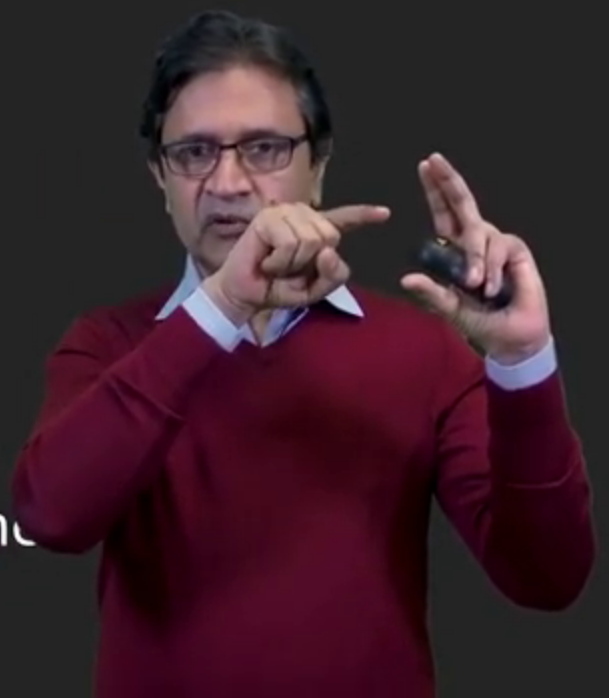


Left/Right Camera Images

Disparity Map (Ground Truth)

From perspective projection: $v_l = v_r = f_y \frac{y}{z} + o_y$

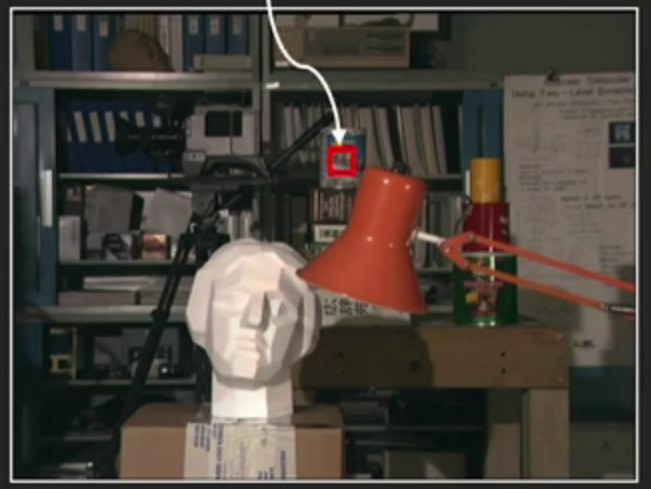
Corresponding scene points lie on the same horizontal scan line



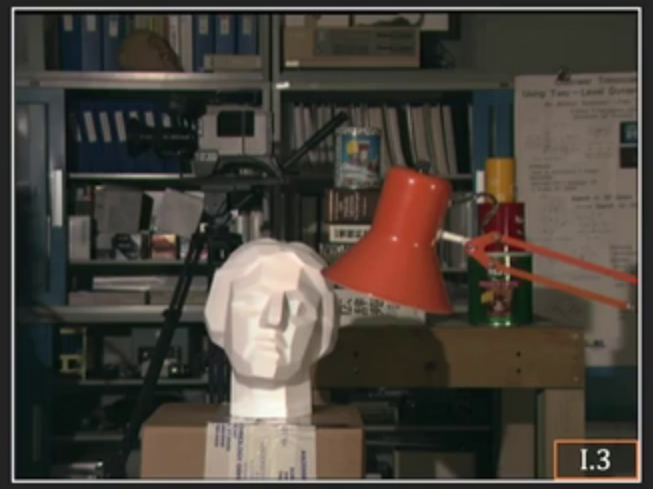
Window Based Methods

Determine Disparity using Template Matching

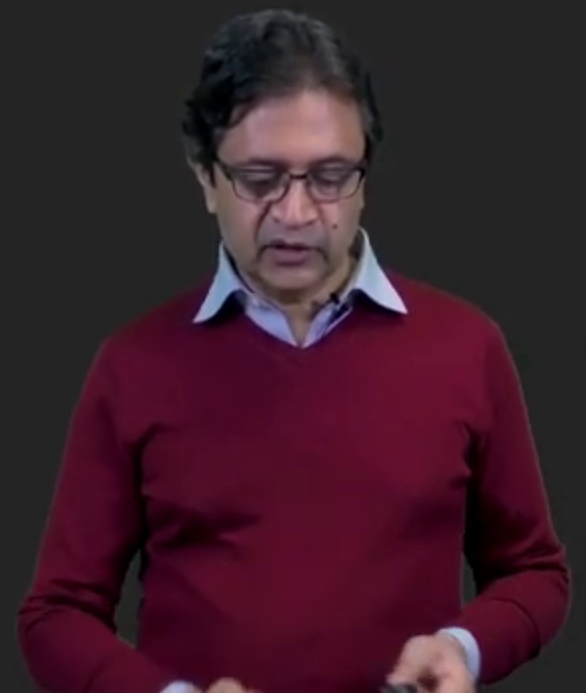
Template Window T



Left Camera Image E_l



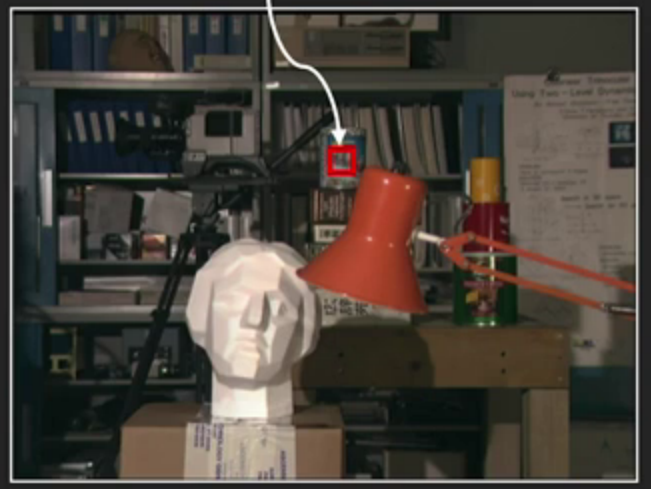
Right Camera Image E_r



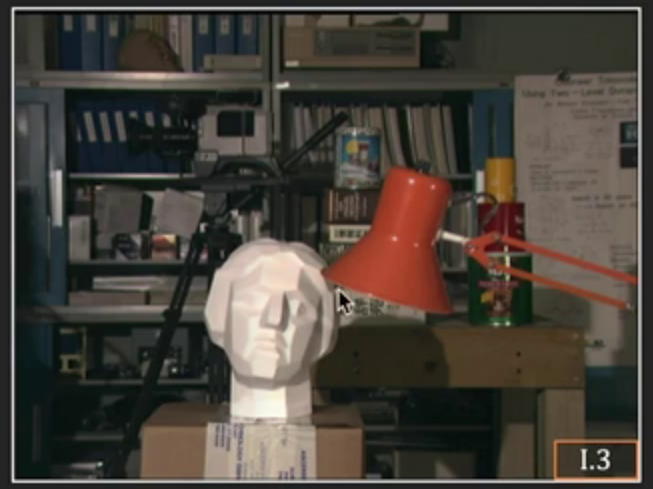
Window Based Methods

Determine Disparity using Template Matching

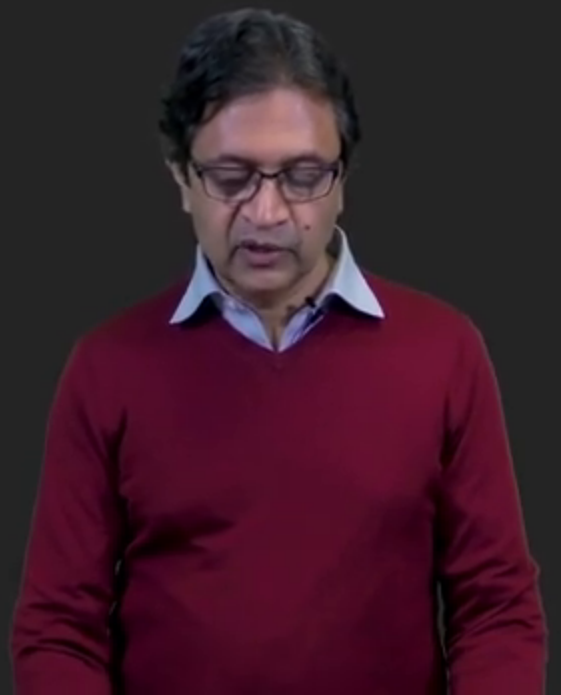
Template Window T



Left Camera Image E_l

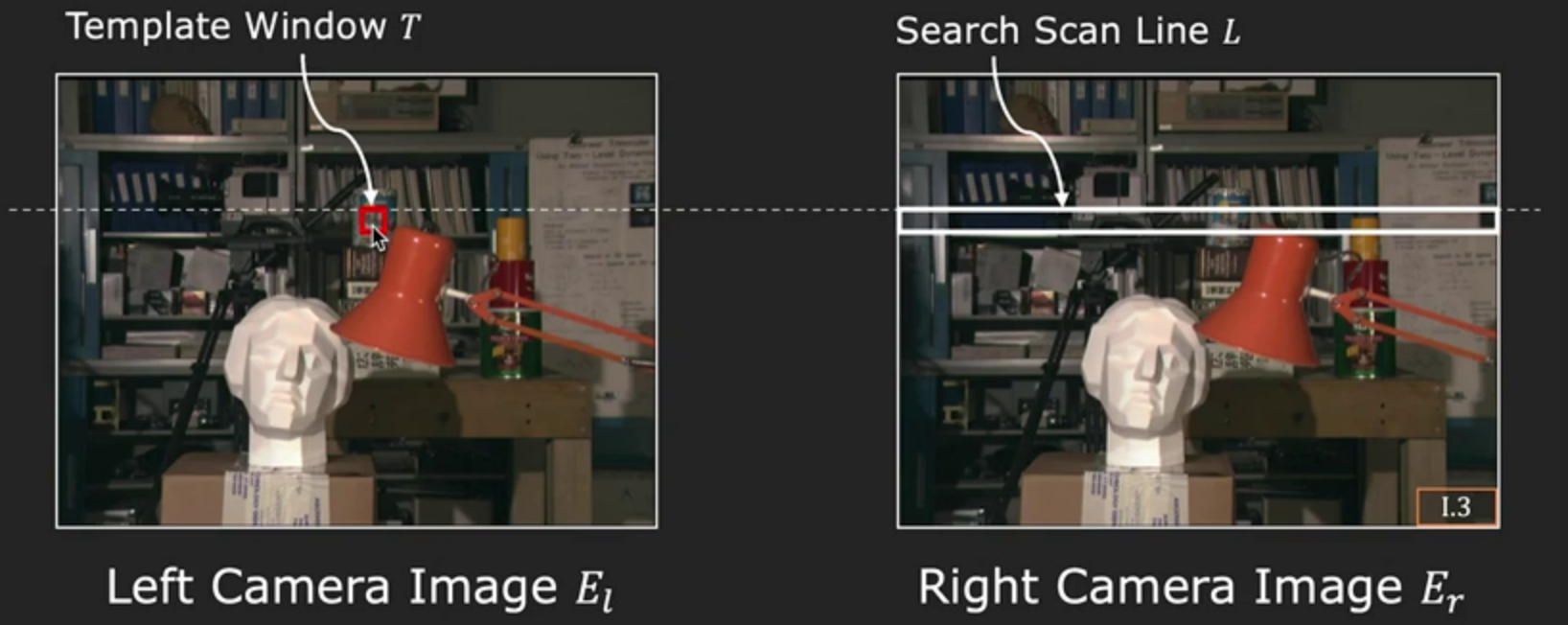


Right Camera Image E_r



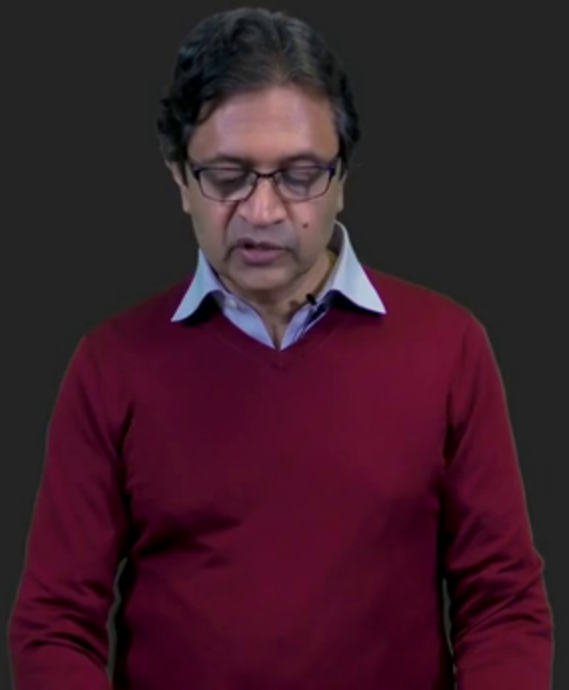
Window Based Methods

Determine Disparity using Template Matching



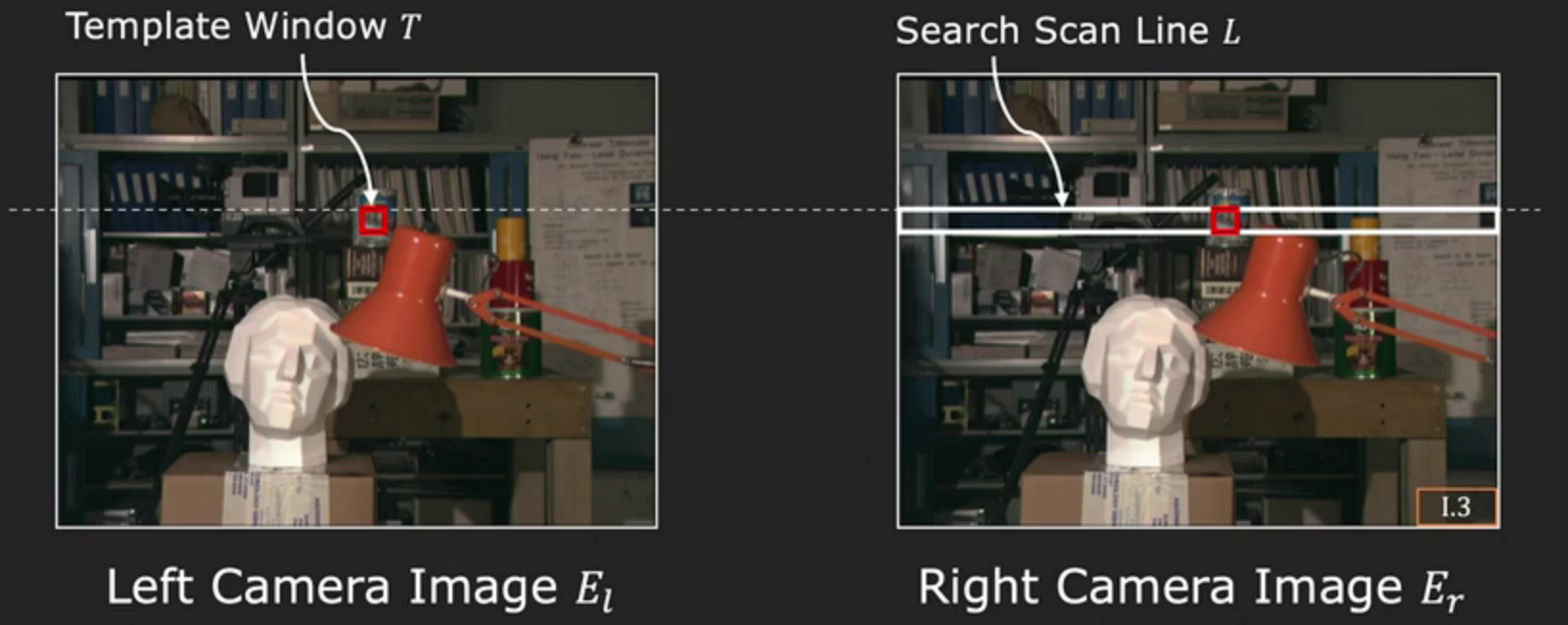
Left Camera Image E_l

Right Camera Image E_r



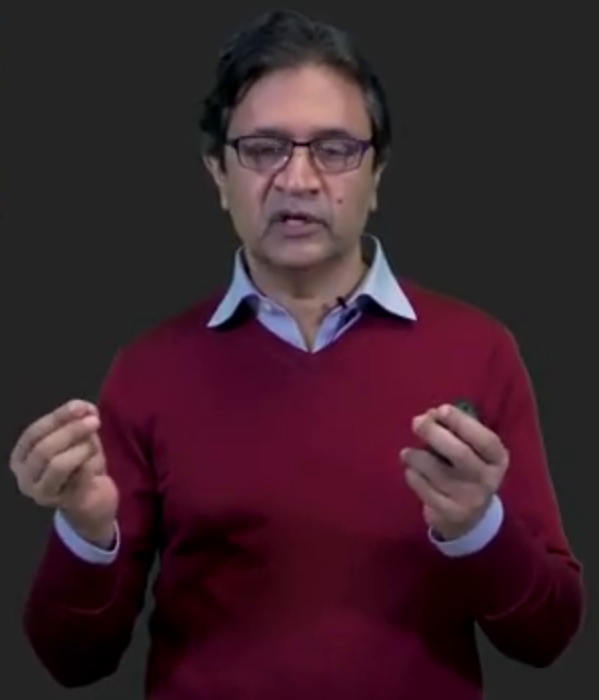
Window Based Methods

Determine Disparity using Template Matching



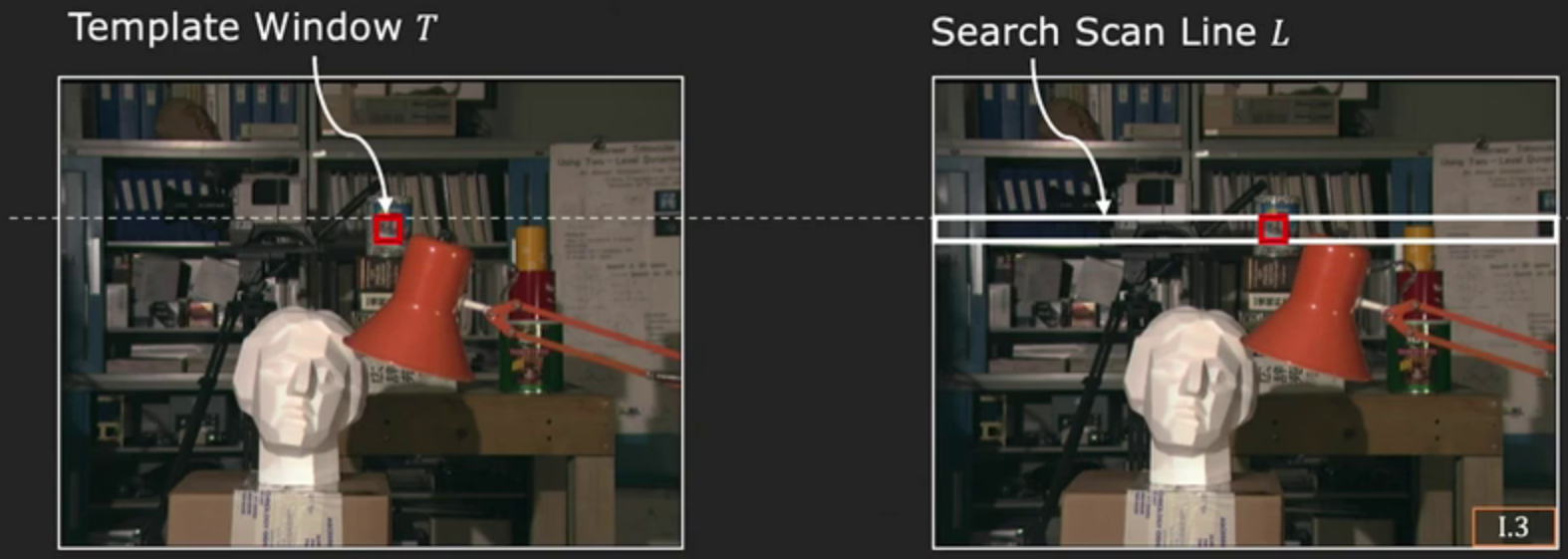
Left Camera Image E_l

Right Camera Image E_r



Window Based Methods

Determine Disparity using Template Matching



Left Camera Image E_l

Right Camera Image E_r

Disparity: $d = u_l - u_r$

Depth: $z = \dfrac{b f_x}{(u_l - u_r)}$

Similarity Metrics for Template Matching

Find pixel $(k, l) \in L$ with Minimum Sum of Absolute Differences:

$$SAD(k,l) = \sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|$$

Similarity Metrics for Template Matching

Find pixel $(k, l) \in L$ with Minimum Sum of Absolute Differences:

$$SAD(k,l) = \sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|$$

Find pixel $(k, l) \in L$ with Minimum Sum of Squared Differences:

$$SSD(k,l) = \sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|^2$$

Similarity Metrics for Template Matching

Find pixel $(k, l) \in L$ with Minimum Sum of Absolute Differences:

$$SAD(k,l) = \sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|$$

Find pixel $(k, l) \in L$ with Minimum Sum of Squared Differences:

$$SSD(k,l) = \sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|^2$$

Find pixel $(k, l) \in L$ with Maximum Normalized Cross-Correlation:

$$NCC(k,l) = \frac{\sum_{(i,j)\in T} E_l(i,j)\, E_r(i+k, j+l)}{\sqrt{\sum_{(i,j)\in T} E_l(i,j)^2 \sum_{(i,j)\in T} E_r(i+k, j+l)^2}}$$

Similarity Metrics for Template Matching

Find pixel $(k, l) \in L$ with Minimum Sum of Absolute Differences:

$$SAD(k,l) = \sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|$$

Find pixel $(k, l) \in L$ with Minimum Sum of Squared Differences:

$$SSD(k,l) = \sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|^2$$

Find pixel $(k, l) \in L$ with Maximum Normalized Cross-Correlation:

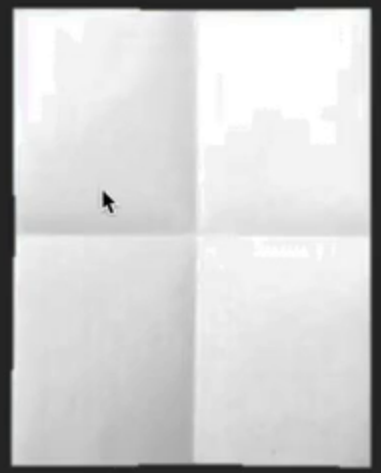
$$NCC(k,l) = \frac{\sum_{(i,j)\in T} E_l(i,j)\, E_r(i+k, j+l)}{\sqrt{\sum_{(i,j)\in T} E_l(i,j)^2 \sum_{(i,j)\in T} E_r(i+k, j+l)^2}}$$

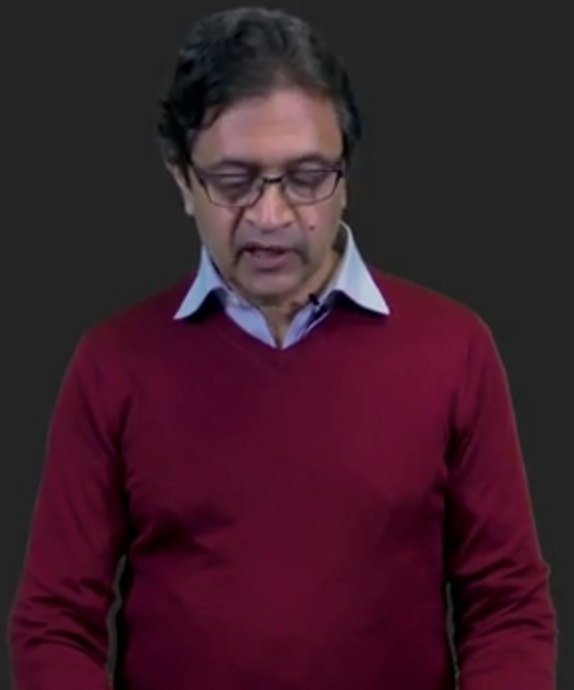
Issues with Stereo Matching

- Surface must have (non-repetitive) texture

Issues with Stereo Matching

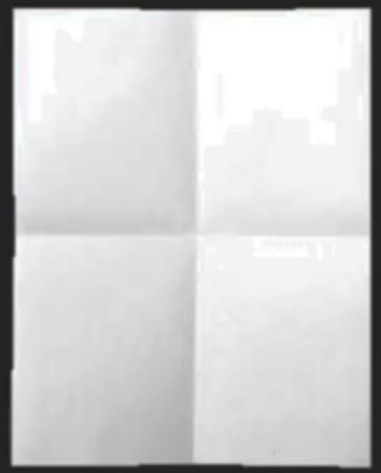
- Surface must have (non-repetitive) texture

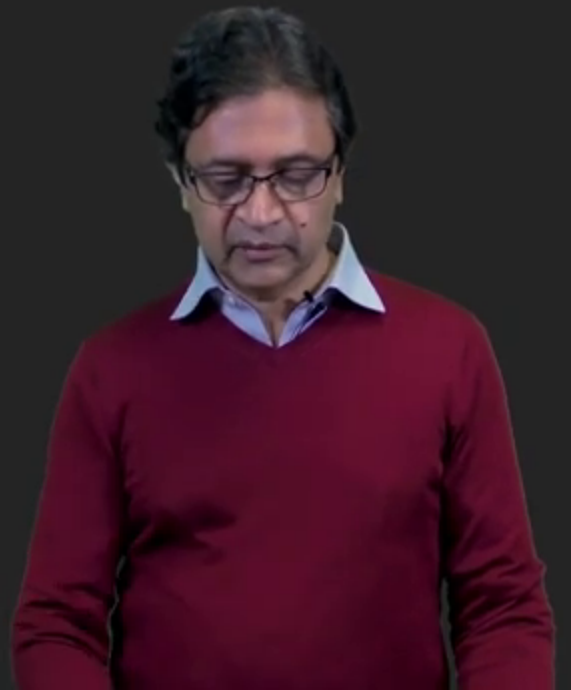




Issues with Stereo Matching

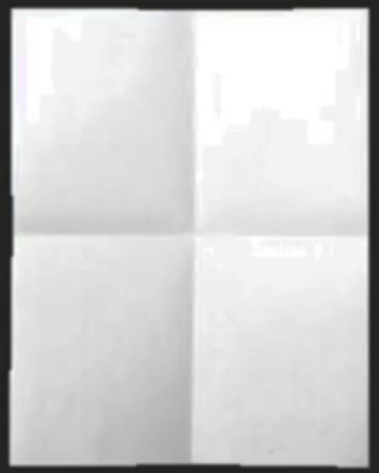
- Surface must have (non-repetitive) texture

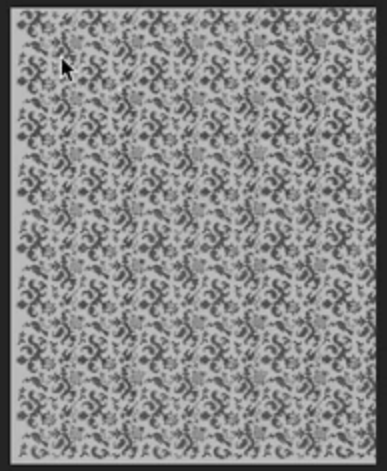


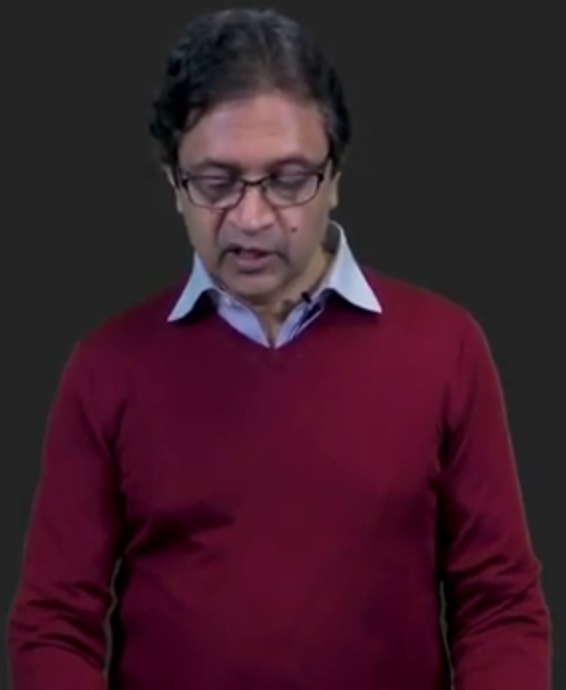


Issues with Stereo Matching

- Surface must have (non-repetitive) texture

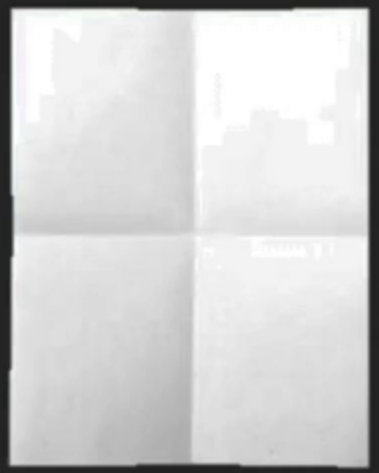


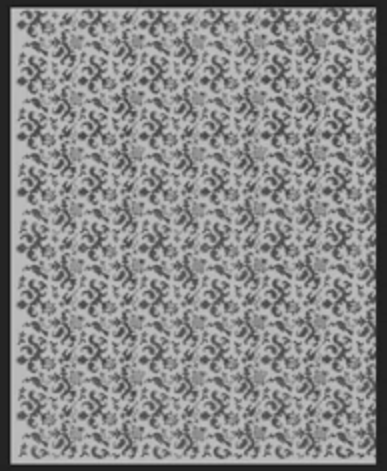


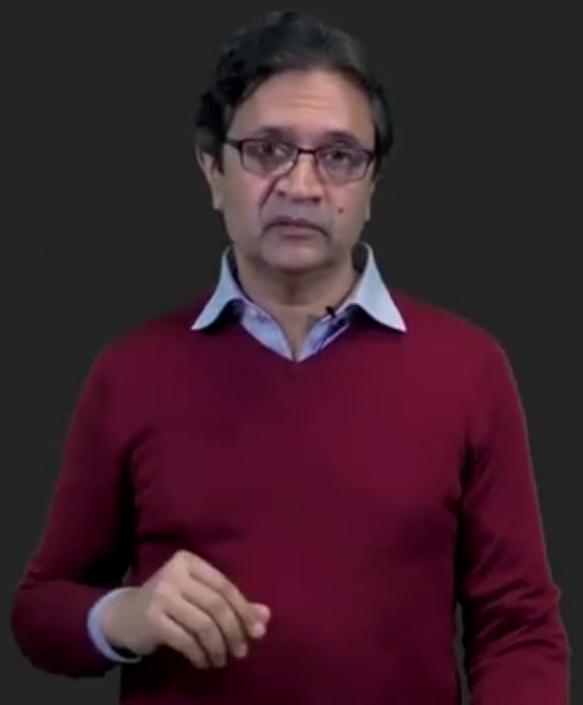


Issues with Stereo Matching

- Surface must have (non-repetitive) texture

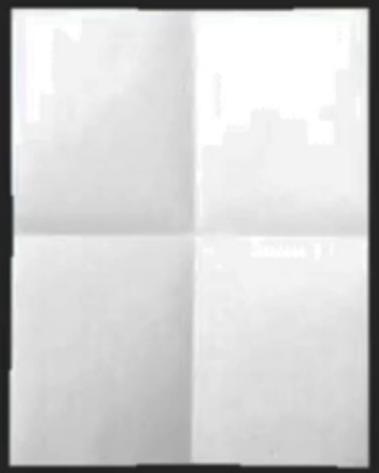


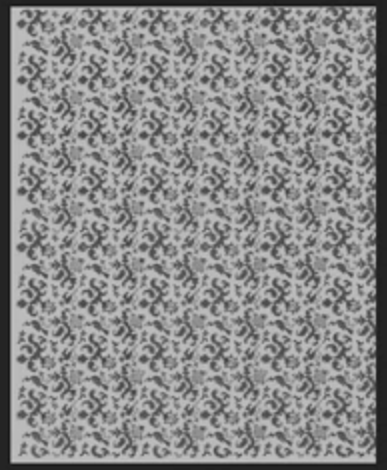




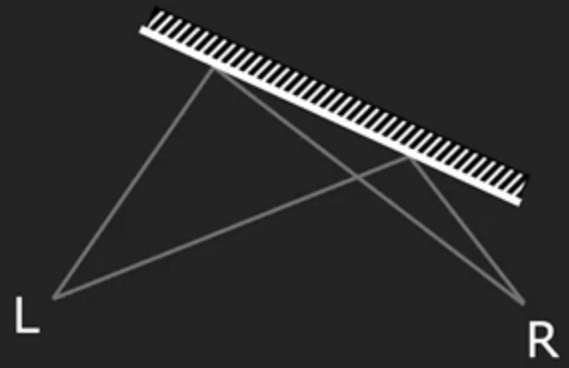
Issues with Stereo Matching

- Surface must have (non-repetitive) texture



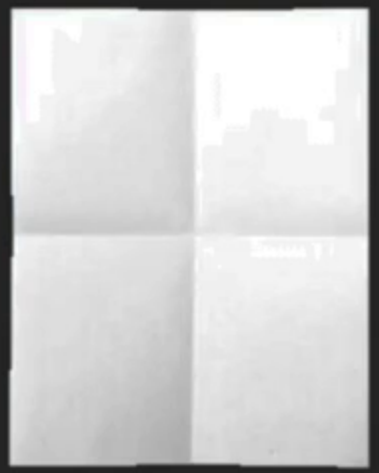


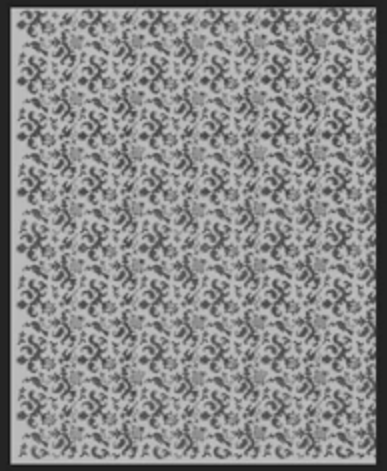
- Foreshortening effect makes matching challenging



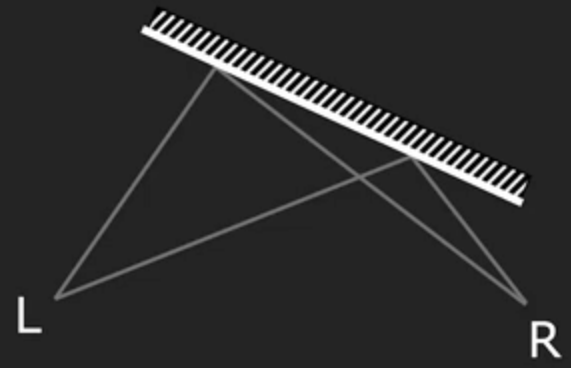
Issues with Stereo Matching

- Surface must have (non-repetitive) texture



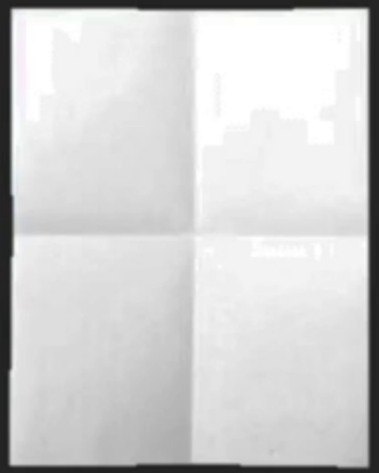


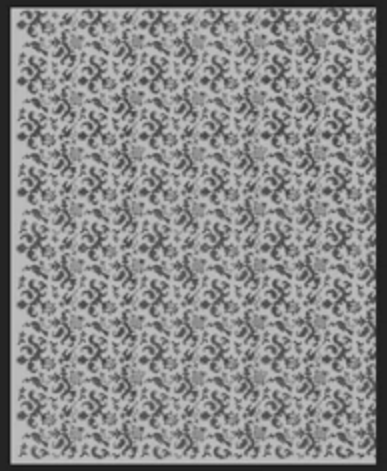
- Foreshortening effect makes matching challenging



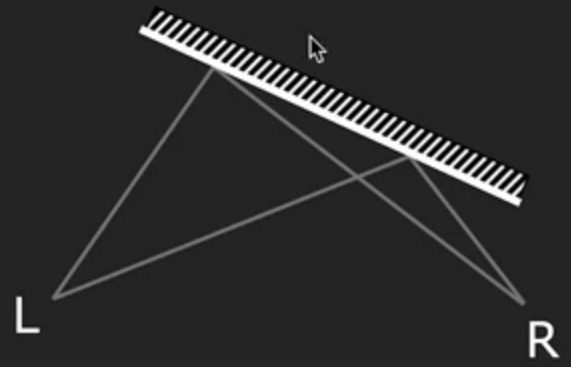
Issues with Stereo Matching

- Surface must have (non-repetitive) texture



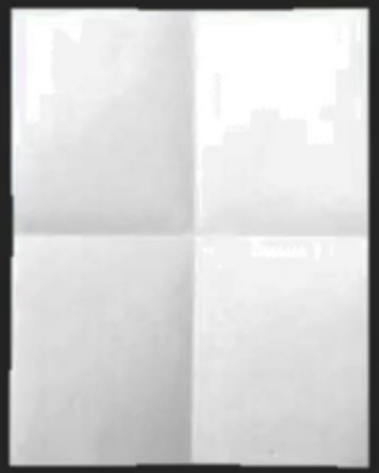


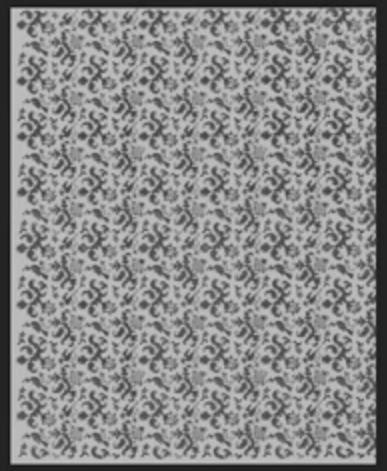
- Foreshortening effect makes matching challenging



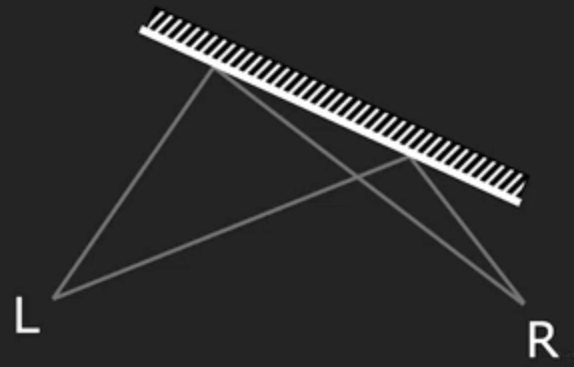
Issues with Stereo Matching

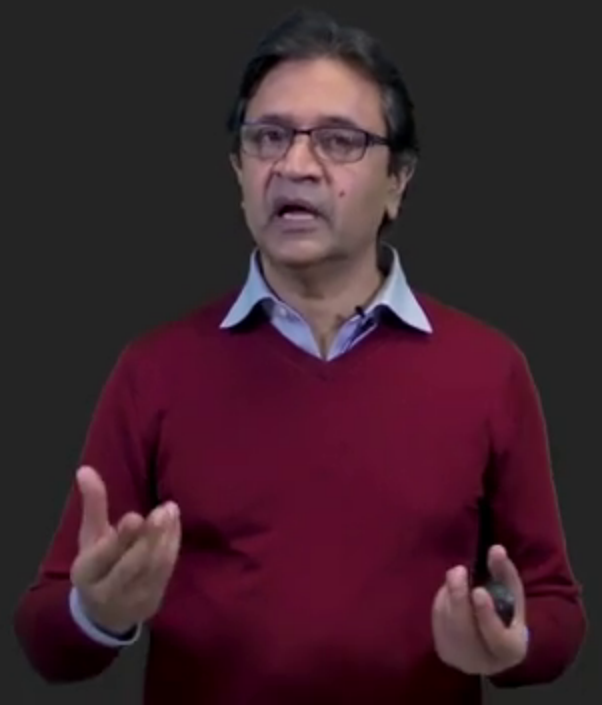
- Surface must have (non-repetitive) texture





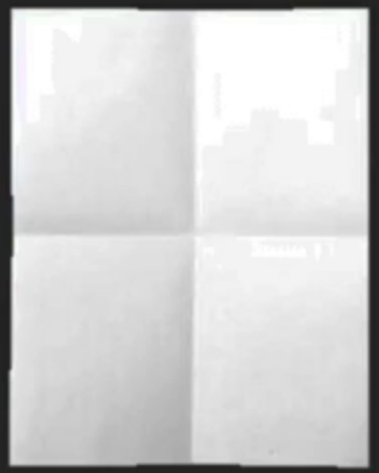
- Foreshortening effect makes matching challenging

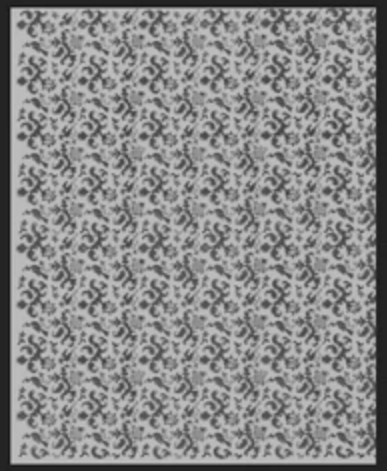




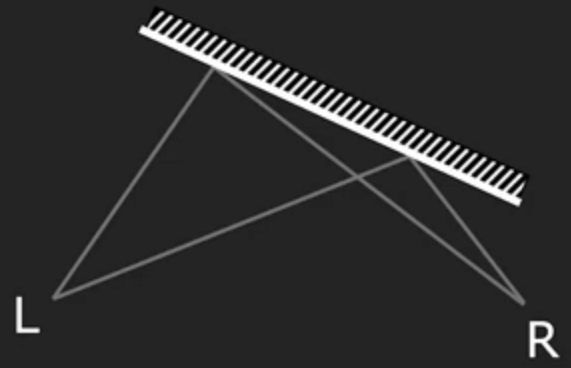
Issues with Stereo Matching

- Surface must have (non-repetitive) texture





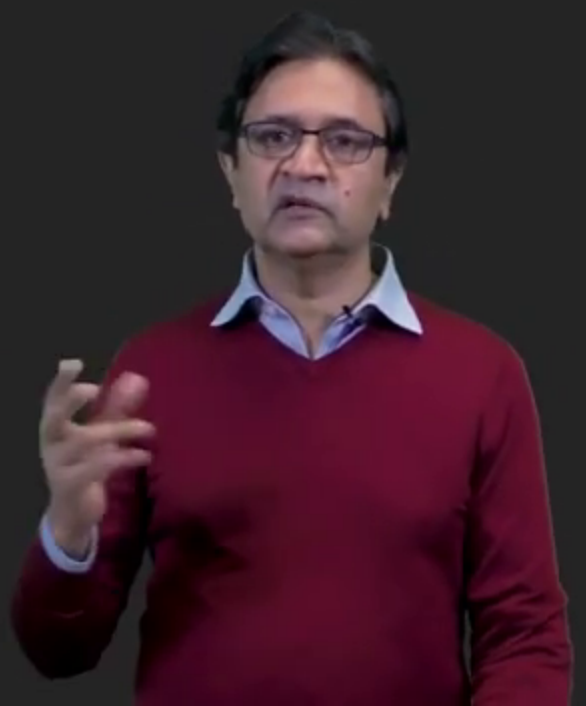
- Foreshortening effect makes matching challenging



How Large Should Window Be?

Window size = 5 pixels
(Sensitive to noise)

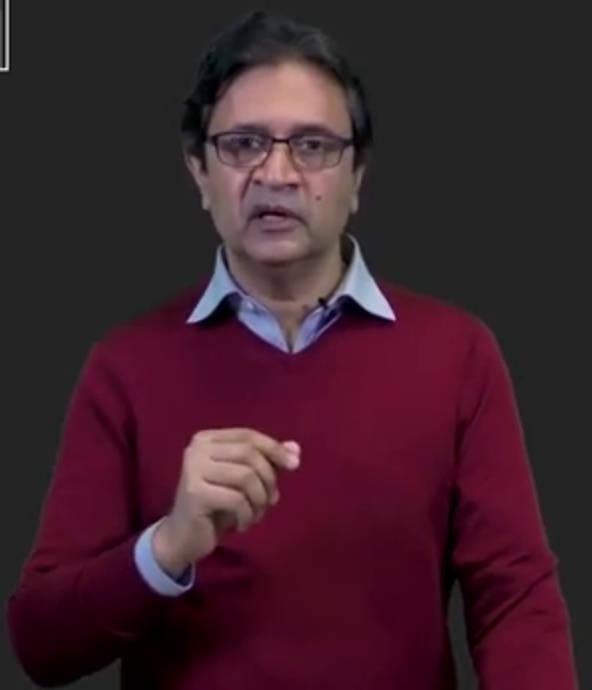
Window size = 30 pixels
(Poor localization)



How Large Should Window Be?

Window size = 5 pixels
(Sensitive to noise)

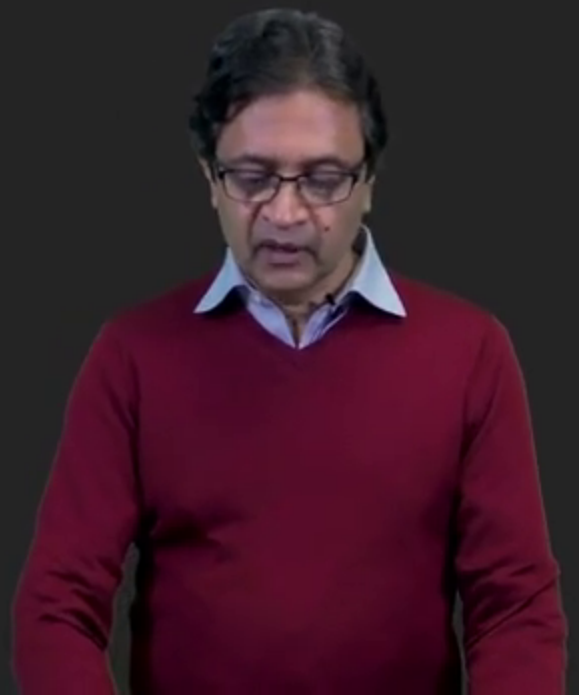
Window size = 30 pixels
(Poor localization)



How Large Should Window Be?

Window size = 5 pixels
(Sensitive to noise)

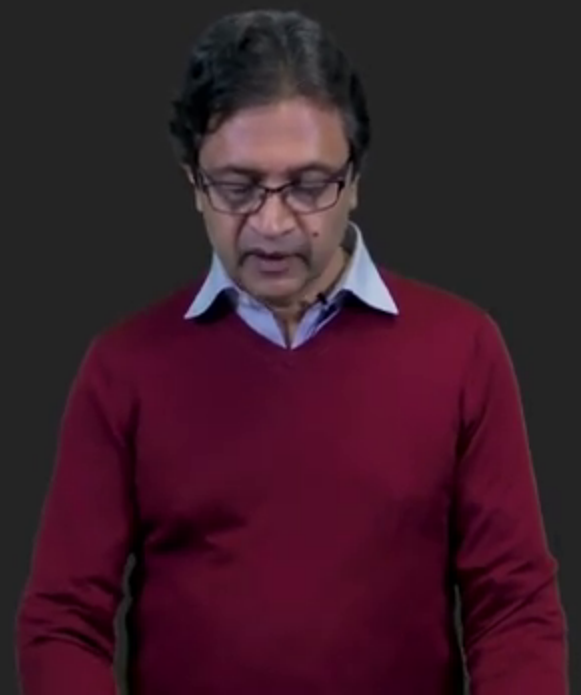
Window size = 30 pixels
(Poor localization)



How Large Should Window Be?

Window size = 5 pixels
(Sensitive to noise)

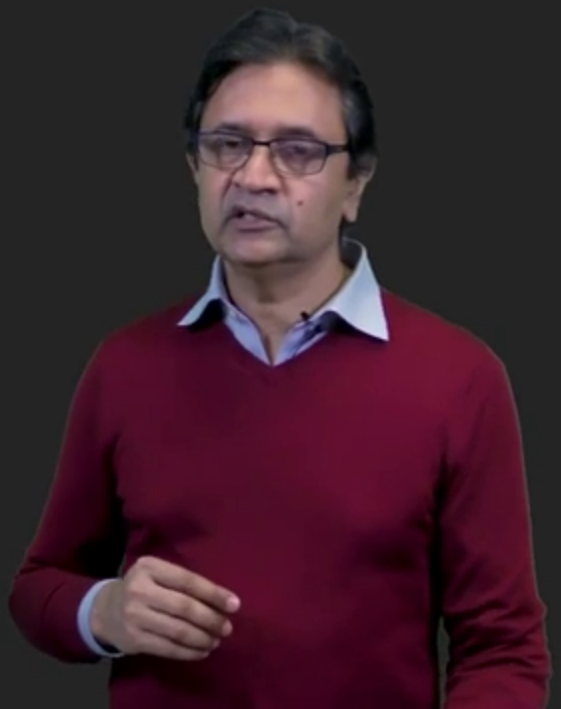
Window size = 30 pixels
(Poor localization)



How Large Should Window Be?

Window size = 5 pixels
(Sensitive to noise)

Window size = 30 pixels
(Poor localization)

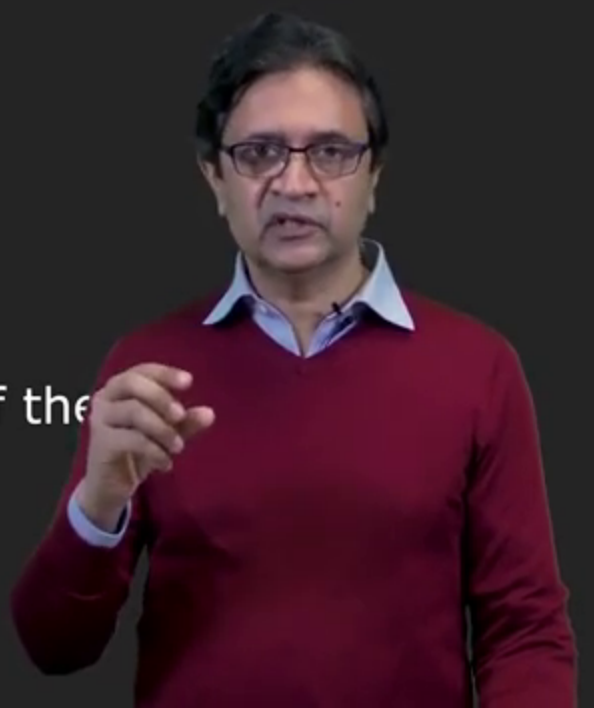


How Large Should Window Be?

Window size = 5 pixels
(Sensitive to noise)

Window size = 30 pixels
(Poor localization)

Adaptive Window Method Solution: For each point, match using windows of multiple sizes and use the disparity that is a result of the best similarity measure (minimize SSD per pixel).

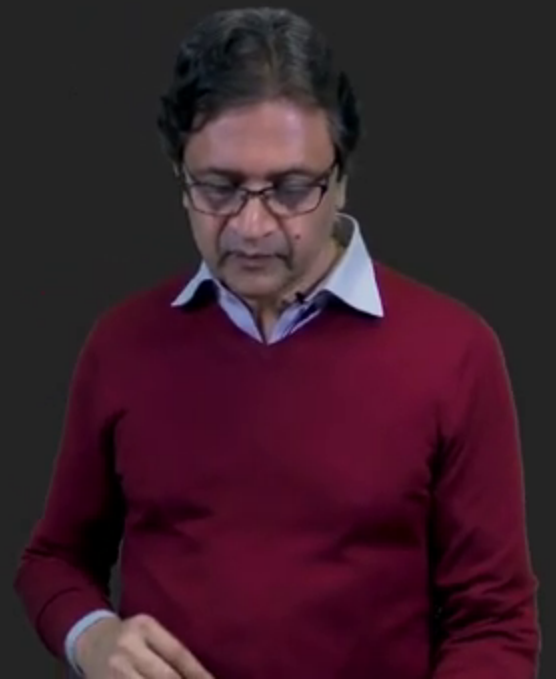


How Large Should Window Be?

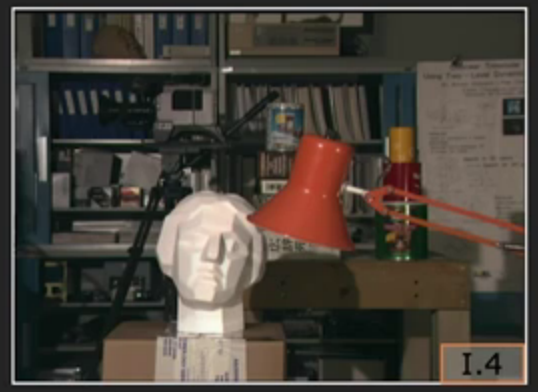
Window size = 5 pixels
(Sensitive to noise)

Window size = 30 pixels
(Poor localization)

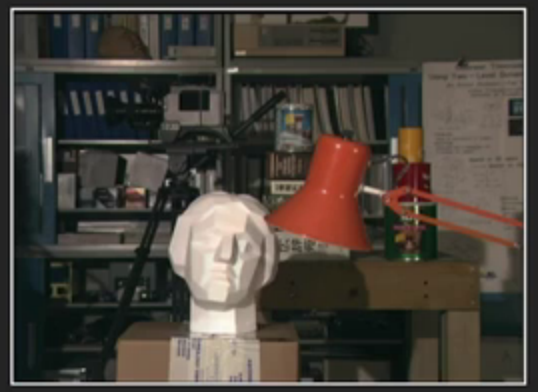
Adaptive Window Method Solution: For each point, match using windows of multiple sizes and use the disparity that is a result of the best similarity measure (minimize SSD per pixel).



Window Based Methods: Results



Left Image



Right Image

Ground Truth

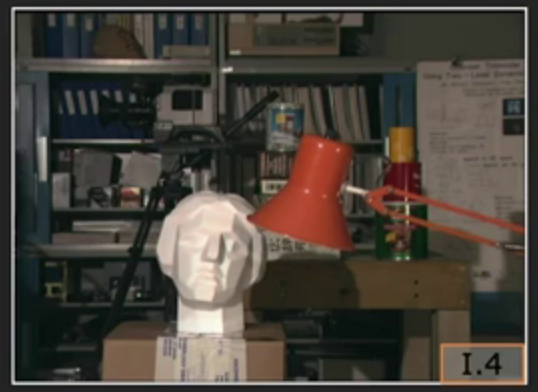
SSD (Window size=21)

SSD – Adaptive Window

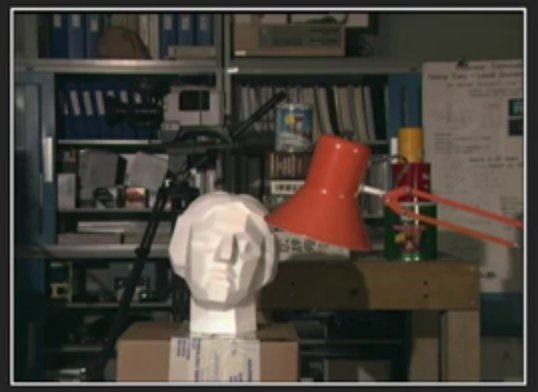
State of the Art

http://vision.middlebury.edu/stereo

Window Based Methods: Results

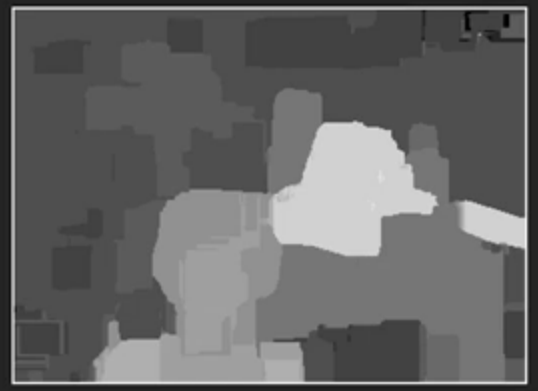


Left Image



Right Image

Ground Truth

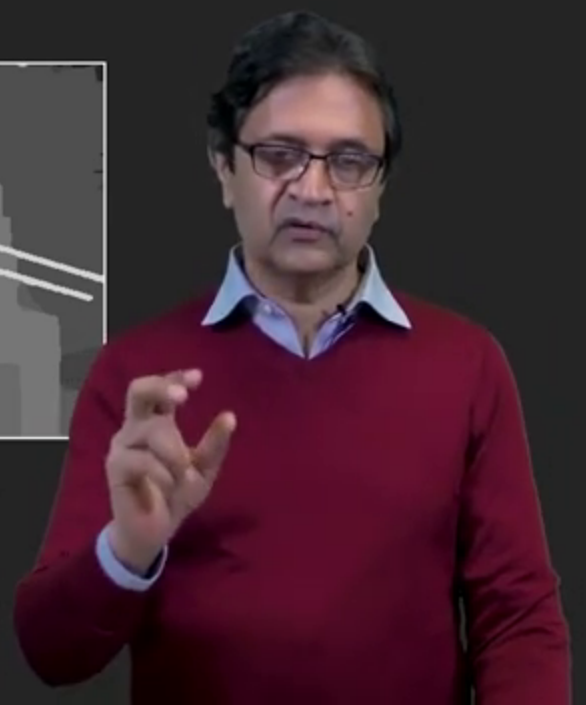


SSD (Window size=21)

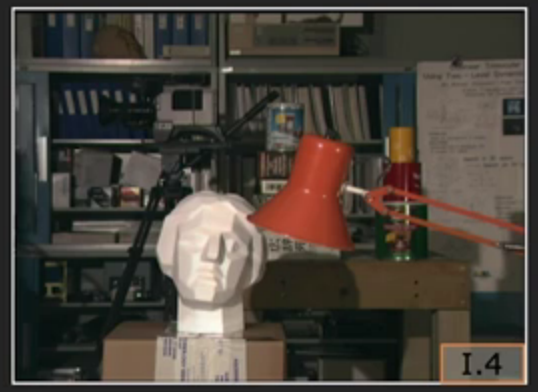
SSD – Adaptive Window

State of the Art

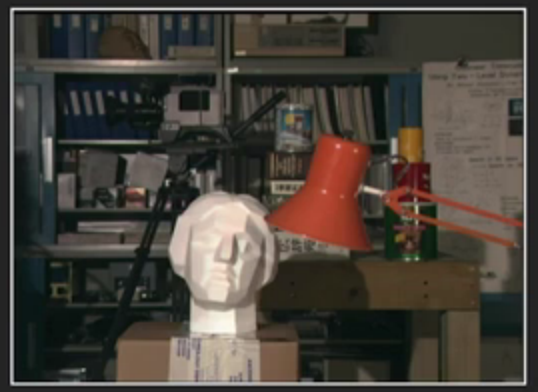
http://vision.middlebury.edu/stereo



Window Based Methods: Results

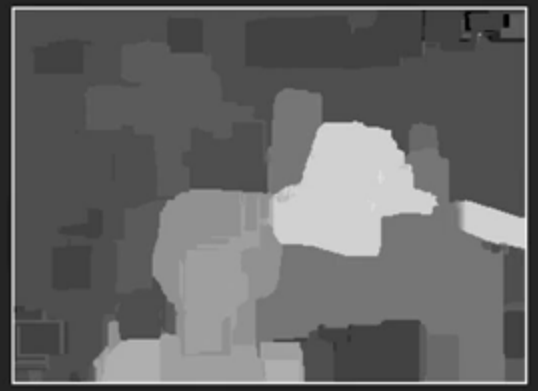


Left Image

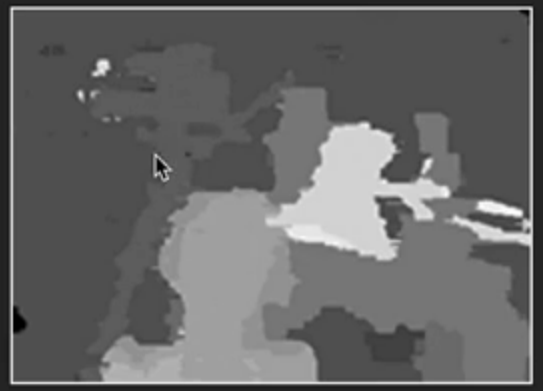


Right Image

Ground Truth



SSD (Window size=21)

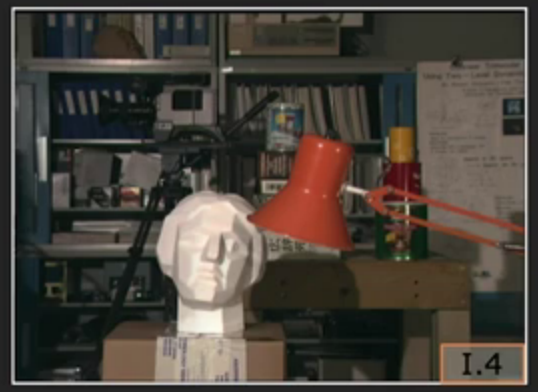


SSD – Adaptive Window

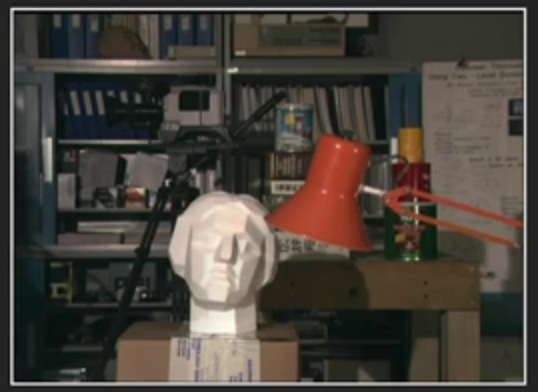
State of the Art

http://vision.middlebury.edu/stereo

Window Based Methods: Results

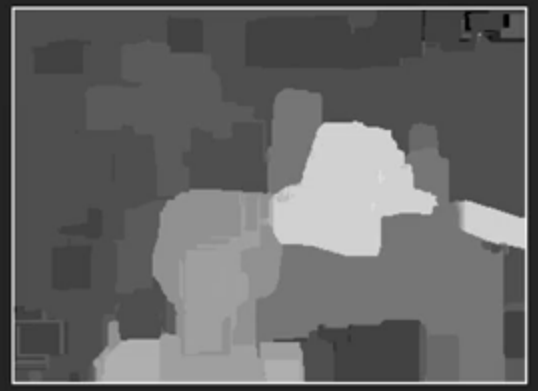


Left Image



Right Image

Ground Truth



SSD (Window size=21)

SSD – Adaptive Window

State of the Art

http://vision.middlebury.edu/stereo