

Introduction and the Pinhole Camera

Computer Vision I (CSCI 3240U)

Faisal Z. Qureshi

<http://vclab.science.ontariotechu.ca>



Instructor

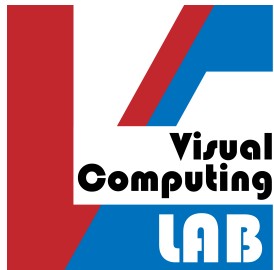
Faisal Qureshi

Professor of Computer Science

Office: SCI 4032

Email: faisal.qureshi@ontariotechu.net

Office hours: Tuesday, 10–11 or by
appointment



<http://www.vclab.ca>

Computer Vision
theory, methods, systems

Given an image, recover something about the world that produced it.





Summer of 66

Playing chess is harder than looking at the world!!!

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

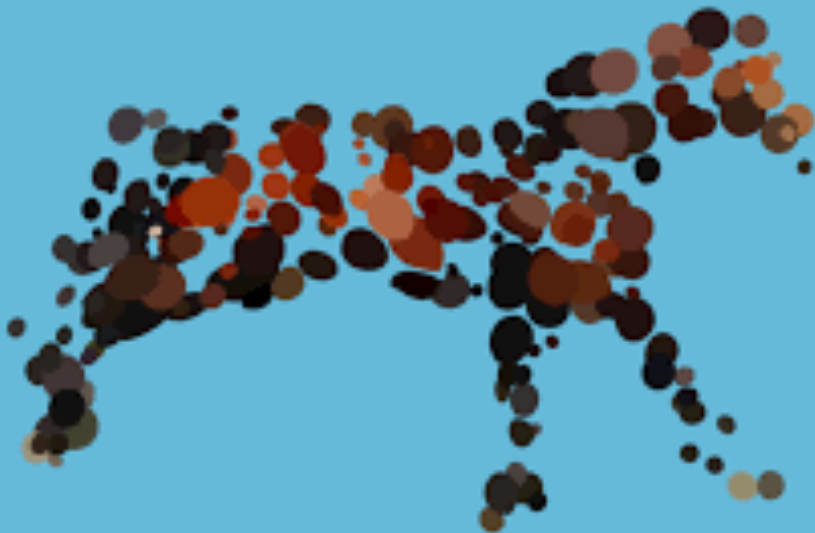
Artificial Intelligence Group
Vision Memo. No. 100.

July 7, 1966

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".







Vision is Hard

- ▶ Visual cortex occupies nearly 50% of Macaque brain
- ▶ Greater fraction of human brain is devoted to vision (processing) than anything else



Credits

These slides draw upon other computer vision courses. I would in particular like to thank S. Seitz, D. Forsyth, K. Derpanis, J. Hoiem, A. Efros, Criminisi and many others for making their material available for teaching and learning purposes.





Varying **SCALES**

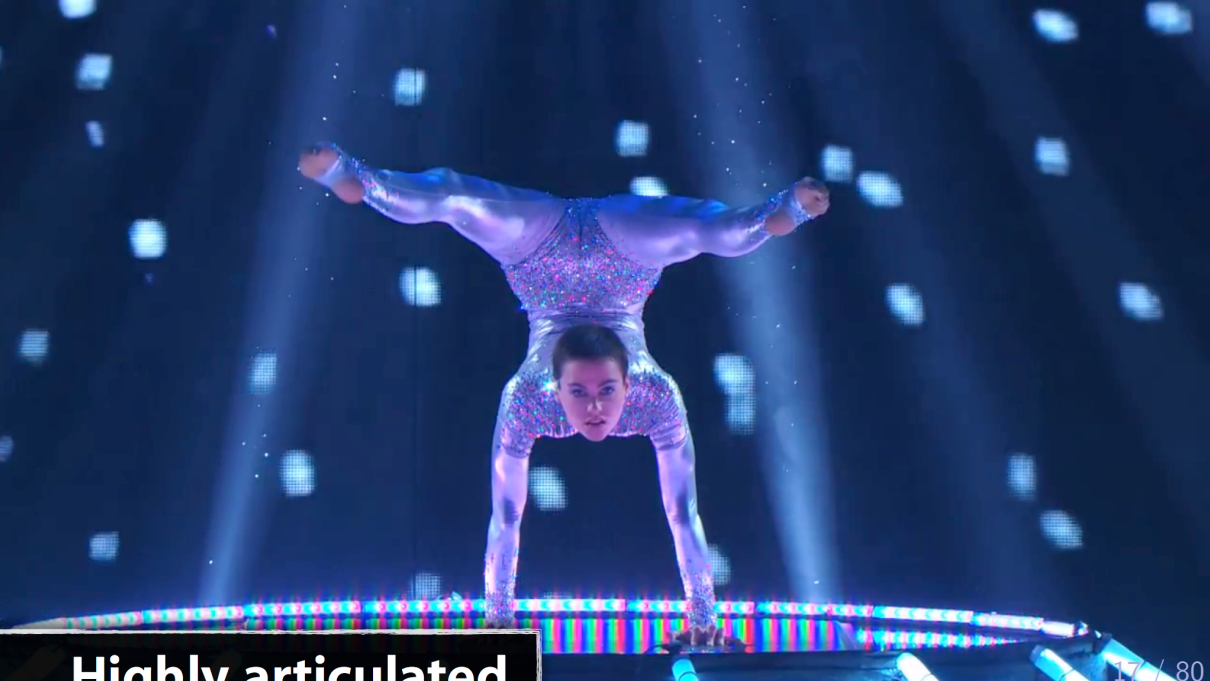




Varying instances



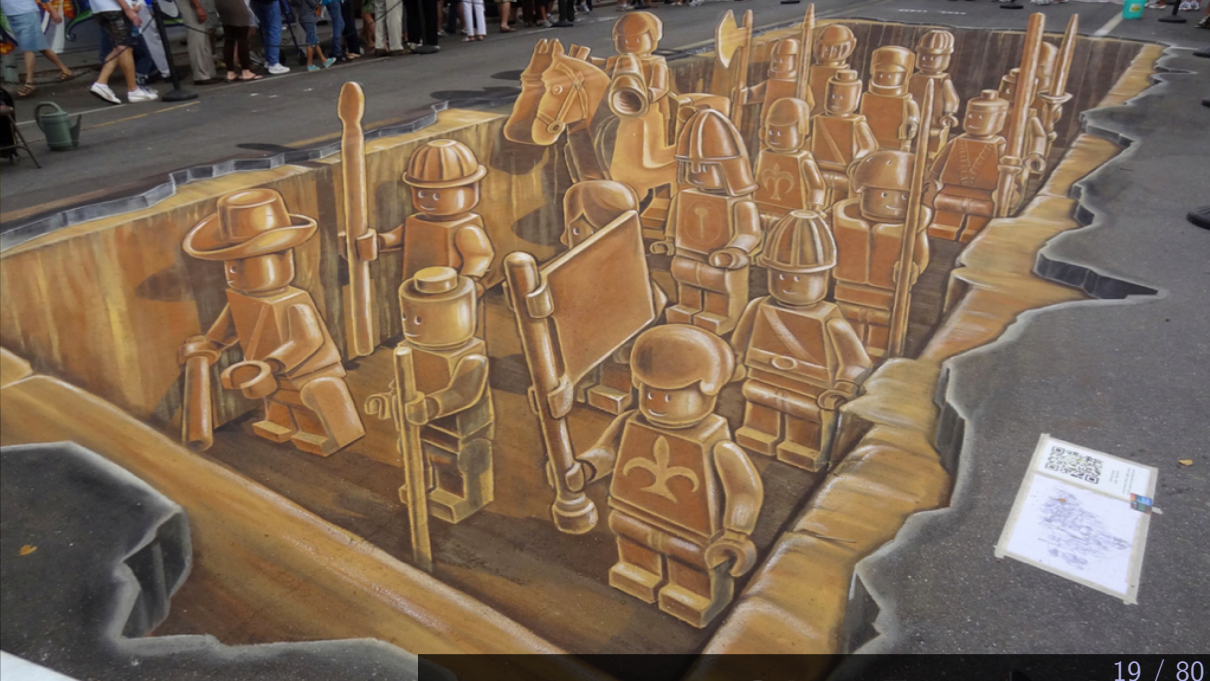
Occlusion



Highly articulated



Viewpoint differences

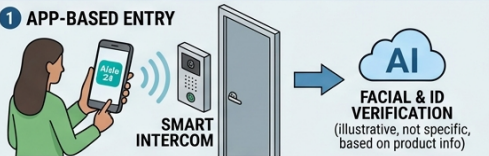




AISLE 24: COMPUTER VISION-ENABLED OPERATIONS

ZONE 1: ACCESS & SHOPPER AUTHENTICATION

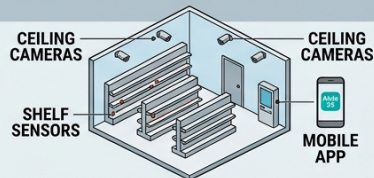
1 APP-BASED ENTRY



Registered member unlocks door via app and connected video intercom.

2 UNMANNED STORE

CV enables 24/7 keyless access.

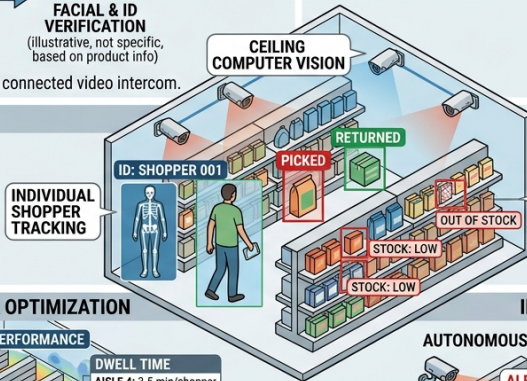


CV enables 24/7 keyless access.

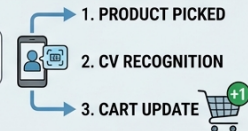
ZONE 2: IN-STORE VISION

ZONE 2: IN-STORE PRODUCT TRACKING & VIRTUAL CART

Real-time object detection & action recognition track items and shoppers.



VIRTUAL CART SYNCHRONIZATION

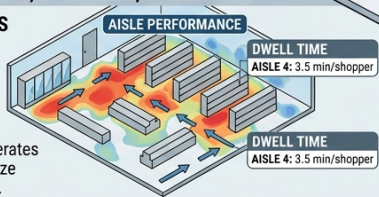


ZONE 3: OPERATIONS, INVENTORY, & OPTIMIZATION

RETAIL ANALYTICS & INSIGHTS

FOOT TRAFFIC HEATMAP
PATHWAYS

Computer vision generates spatial data to optimize layout and marketing.



INVENTORY & STORE MANAGEMENT

AUTONOMOUS RESTOCK ALERTS



SAFETY & SECURITY



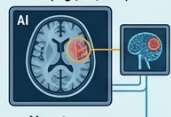
Real-time shelf scanning, inventory forecasting, and safety monitoring

A MULTI-PANEL GUIDE TO COMPUTER VISION APPLICATIONS

EXPLORING 18 KEY EXAMPLES ACROSS INDUSTRIES

PANEL 1: APPLICATIONS IN HEALTHCARE, TRANSPORTATION & SECURITY

MEDICAL IMAGING
(e.g., CT, MRI)



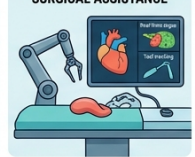
AI system

PATIENT MONITORING



FALL DETECTION

SURGICAL ASSISTANCE



TRAFFIC MANAGEMENT



AUTONOMOUS VEHICLES



LICENSE PLATE RECOGNITION (ALPR)



PLATE: ABC 123

CROWD SURVEILLANCE



INTRUSION DETECTION



DRONE-BASED INSPECTION



A MULTI-PANEL GUIDE TO COMPUTER VISION APPLICATIONS

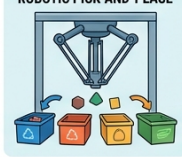
EXPLORING 18 KEY EXAMPLES ACROSS INDUSTRIES

PANEL 2: INDUSTRIAL, COMMERCIAL & EMERGING USES

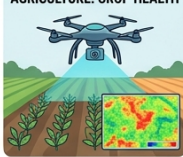
MANUFACTURING QUALITY CONTROL



ROBOTIC PICK-AND-PLACE



AGRICULTURE: CROP HEALTH



LIVESTOCK MONITORING



FACIAL RECOGNITION



AUTHORIZED ACCESS

OBJECT COUNTING & TRACKING



INVENTORY

ACTION RECOGNITION



DOCUMENT DIGITIZATION (OCR)



ANOMALY DETECTION (FIRE & SMOKE)



Week 1 at a glance

► **Lecture 1: What is computer vision? (Sec. 1.1)**

- Why vision is hard
- An image as a function
- What this course covers, and what it does not
- Tools: Python, NumPy, OpenCV

► **Lecture 2: The pinhole camera (Sec. 2.1)**

- Image formation as projection
- Perspective projection from similar triangles
- Aperture: the geometry/diffraction trade-off
- What projection destroys

Reading: Szeliski Sec. 1.1, 2.1. Lab 1 and in-class exercise w01a-pinhole this week.

Lecture 1: What is computer vision?

The problem

Given an image, recover something about the world that produced it.

- ▶ Where is the camera? How far away is that car?
- ▶ What objects are present? Where are their boundaries?
- ▶ How is the scene moving?

The problem

Given an image, recover something about the world that produced it.

- ▶ Where is the camera? How far away is that car?
- ▶ What objects are present? Where are their boundaries?
- ▶ How is the scene moving?

This is *inverse* to computer graphics.

- ▶ Graphics: scene $\rightarrow$ image. Well posed.
- ▶ Vision: image $\rightarrow$ scene. **Ill posed.**

Why it is hard: information is destroyed

A camera maps a 3D world onto a 2D array of numbers. That mapping is **many-to-one**.

Why it is hard: information is destroyed

A camera maps a 3D world onto a 2D array of numbers. That mapping is **many-to-one**.

- ▶ **Depth is lost.** Every point along a ray produces the same pixel.
- ▶ **Illumination and reflectance are confounded.** A dark surface in bright light and a bright surface in dim light give the same measurement.
- ▶ **Occlusion.** Most of the scene is not visible.
- ▶ **Sampling and noise.** We get a finite grid of noisy integers.

Why it is hard: information is destroyed

A camera maps a 3D world onto a 2D array of numbers. That mapping is **many-to-one**.

- ▶ **Depth is lost.** Every point along a ray produces the same pixel.
- ▶ **Illumination and reflectance are confounded.** A dark surface in bright light and a bright surface in dim light give the same measurement.
- ▶ **Occlusion.** Most of the scene is not visible.
- ▶ **Sampling and noise.** We get a finite grid of noisy integers.

Vision is the business of putting back what the camera threw away, using geometry, statistics, and assumptions about the world.

Why it is hard: the same object, different pixels

The pixel values change completely under

- ▶ viewpoint,
- ▶ illumination,
- ▶ scale,
- ▶ deformation,
- ▶ occlusion,
- ▶ clutter.





A human calls all of these "a chair". A naive pixel comparison calls them six unrelated objects. This gap between pixels and meaning is what the whole field is about.

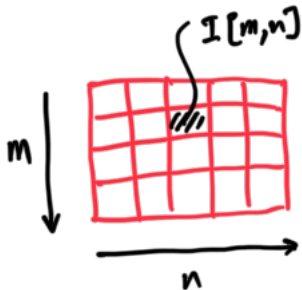
An image is a function

A grayscale image is a function

$$I : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad (x, y) \mapsto \text{intensity}$$

A digital image is that function **sampled** on a grid and **quantized**:

$$I[m, n], \quad m = 0, \dots, H - 1, \quad n = 0, \dots, W - 1$$



An image is a function

A grayscale image is a function

$$I : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad (x, y) \mapsto \text{intensity}$$

A digital image is that function **sampled** on a grid and **quantized**:

$$I[m, n], \quad m = 0, \dots, H - 1, \quad n = 0, \dots, W - 1$$

- ▶ Colour image: $I : \mathbb{R}^2 \rightarrow \mathbb{R}^3$.
- ▶ Video: add time, $I(x, y, t)$.

Thinking of an image as a *function* rather than a *picture* is what lets us apply calculus (derivatives, Week 4) and signal processing (filtering, Week 3).

Picture, or function?

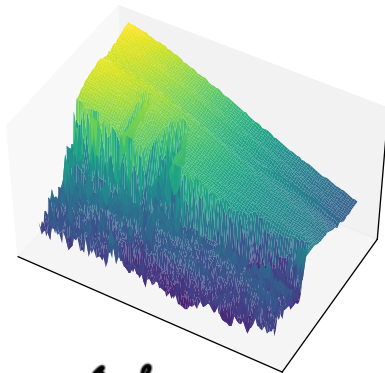
$I[m, n]$ as a picture



$[0, 255]$

of 8-bits

$I(x, y)$ as a function



Surface

Conventions we will use all term

- ▶ **Origin at the top-left.** Row index m increases **downwards**.
- ▶ $I[m, n]$ is row m , column n . **This is (y, x) , not (x, y) .**
- ▶ OpenCV reads colour as **BGR**, not RGB.
- ▶ `img.shape` gives (H, W) for grayscale and (H, W, 3) for colour.

Conventions we will use all term

- ▶ **Origin at the top-left.** Row index m increases **downwards**.
- ▶ $I[m, n]$ is row m , column n . **This is (y, x) , not (x, y) .**
- ▶ OpenCV reads colour as **BGR**, not RGB.
- ▶ `img.shape` gives (H, W) for grayscale and (H, W, 3) for colour.

These four conventions account for most first-month bugs. Write them on the inside cover of your notes.



What this course covers

1. **Geometry**: how cameras map the world to images, and how to invert it (Weeks 1–2, 9–10).
2. **Image processing**: filtering, derivatives, scale (Weeks 3–5).
3. **Estimation**: fitting models to noisy, contaminated data (Weeks 6–7).
4. **Features and motion**: what to measure and how to track it (Week 8).
5. **Learning**: a gentle introduction (Weeks 11–12).

What this course does *not* cover

- ▶ Vision transformers, foundation models, generative image models.
- ▶ Modern learned detection and segmentation.
- ▶ Deep networks trained from scratch.

What this course does *not* cover

- ▶ Vision transformers, foundation models, generative image models.
- ▶ Modern learned detection and segmentation.
- ▶ Deep networks trained from scratch.

Those are **Computer Vision II**, in the winter term.

What this course does *not* cover

- ▶ Vision transformers, foundation models, generative image models.
- ▶ Modern learned detection and segmentation.
- ▶ Deep networks trained from scratch.

Those are **Computer Vision II**, in the winter term.

There is a reason for the order. Classical methods are where the *geometry* and the *estimation* live, and both are assumed knowledge in the learned methods. You will also see, in Week 12, exactly where the classical approach runs out, which is the honest motivation for what replaces it.

Tools

- ▶ **Python 3** with **NumPy**, **SciPy**, **Matplotlib**, **OpenCV**.
- ▶ Jupyter notebooks for labs.
- ▶ PyTorch for Weeks 11–12 only.



Tools

- ▶ **Python 3** with **NumPy**, **SciPy**, **Matplotlib**, **OpenCV**.
- ▶ Jupyter notebooks for labs.
- ▶ PyTorch for Weeks 11–12 only.

Two rules for the labs:

1. When a lab says *implement it yourself*, do so. Calling `cv.GaussianBlur` when asked to write a convolution defeats the purpose, and everything downstream assumes you wrote it.
2. When a lab asks you to *measure* something, report a number, not an impression.

Assessment, briefly

- ▶ Class participation, via in-class exercises and quizzes: 10%
- ▶ Lab and assignment participation and completion: 20%
- ▶ Midterm exams: 50%, split evenly (25% each)
- ▶ Course project: 20%

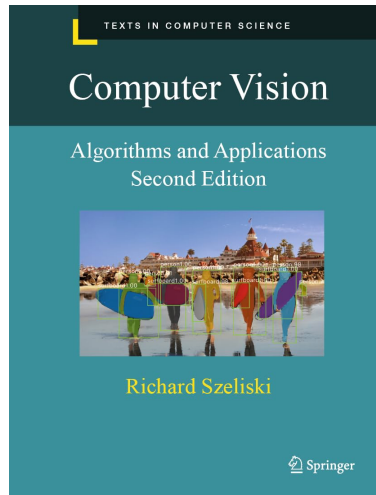
Assessment, briefly

- ▶ Class participation, via in-class exercises and quizzes: 10%
 - ▶ Lab and assignment participation and completion: 20%
 - ▶ Midterm exams: 50%, split evenly (25% each)
 - ▶ Course project: 20%
-
- ▶ Midterm 1 (Oct 8) covers Weeks 1–4.
 - ▶ Midterm 2 (Nov 19) covers Weeks 5–9.
 - ▶ There is **no final exam**.

See the syllabus for the full rules, including the 50% hurdles.

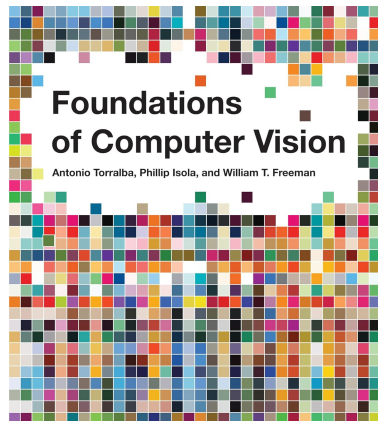
Textbook (required)

Available at <https://szeliski.org/Book/>



Textbook (recommended)

Available at <https://visionbook.mit.edu/>



Lecture 1 recap

- ▶ Vision is the inverse of graphics, and it is ill posed.
- ▶ Projection destroys depth; illumination and reflectance are confounded.
- ▶ An image is a *sampled, quantized function*, which is why calculus and signal processing apply.
- ▶ Top-left origin, (row, col) indexing, BGR in OpenCV.

Next: how exactly does the world get onto the sensor?

Lecture 2: The pinhole camera

The simplest camera that works

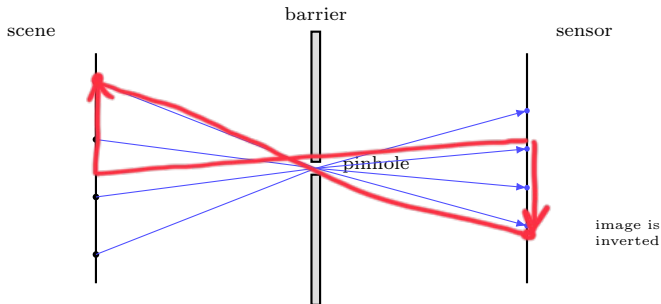
Put a barrier with a small hole between the scene and a sensor.

- ▶ Without the barrier, every scene point illuminates every sensor point. The result is a uniform blur: no image at all.
- ▶ The hole restricts each sensor point to (approximately) **one ray**.

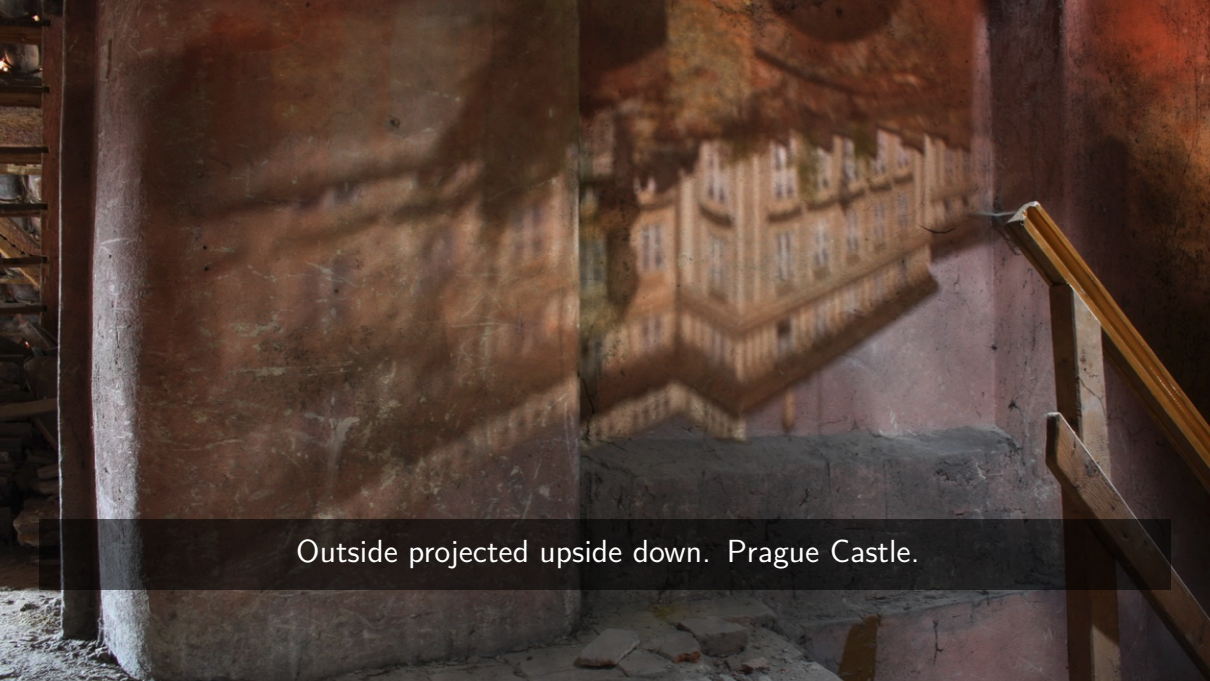
The simplest camera that works

Put a barrier with a small hole between the scene and a sensor.

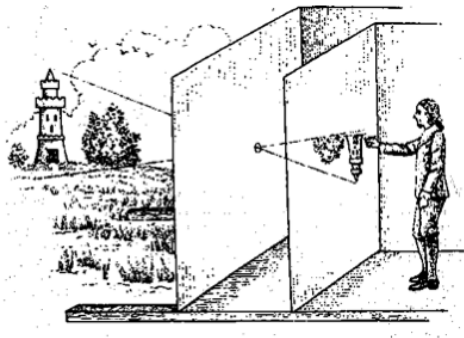
- ▶ Without the barrier, every scene point illuminates every sensor point. The result is a uniform blur: no image at all.
- ▶ The hole restricts each sensor point to (approximately) **one ray**.



That is the whole idea. The pinhole is the geometric model we will use for the rest of the term, and it is what the KITTI cameras in your labs are calibrated against.



Outside projected upside down. Prague Castle.



Freestanding camera obscura at UNC Chapel Hill
(Photo by Seth Ilys)



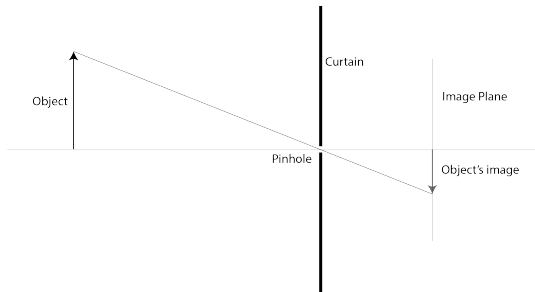
Boulevard du Temple 1838

Terminology

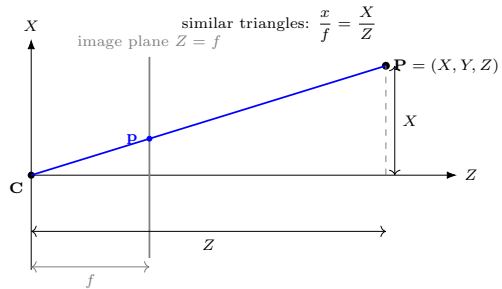
- ▶ **Pinhole / centre of projection**: the point all rays pass through.
- ▶ **Image plane**: where the sensor sits.
- ▶ **Optical axis**: the line through the pinhole, perpendicular to the image plane.
- ▶ **Focal length** f : distance from pinhole to image plane.
- ▶ **Principal point**: where the optical axis meets the image plane.

The real image plane is *behind* the pinhole and the image is inverted. For convenience we place a **virtual image plane** in front, at the same distance, which removes the inversion and changes nothing else.

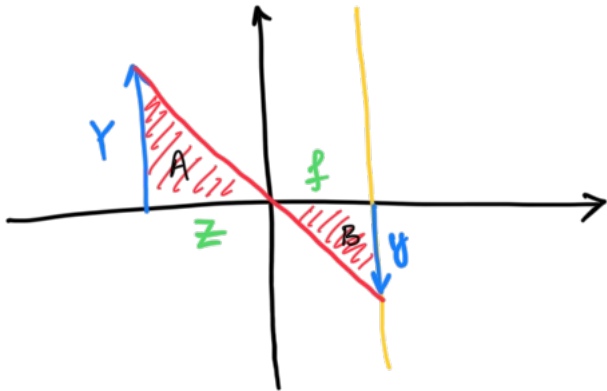
Real image plane vs virtual image plane



Real image plane: **behind** the pinhole.
Image is **inverted**. This is what a physical camera obscura does.



Virtual image plane: **in front** of the pinhole, at the same distance f . Removes the inversion. Same geometry, cleaner algebra. **We use this from now on.**



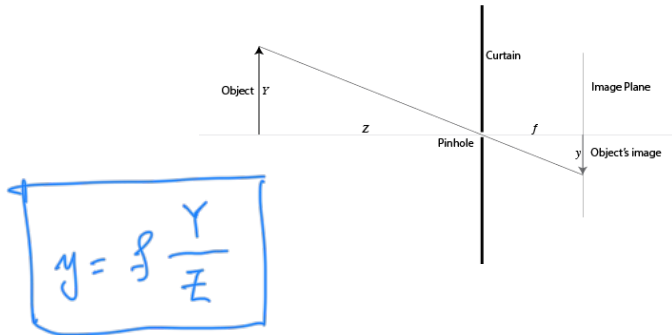
Triangles A and B are similar triangles.

$$\frac{Y}{Z} = \frac{y}{f} \Rightarrow y = f \frac{Y}{Z}$$

↖ size of object
↑
size of image

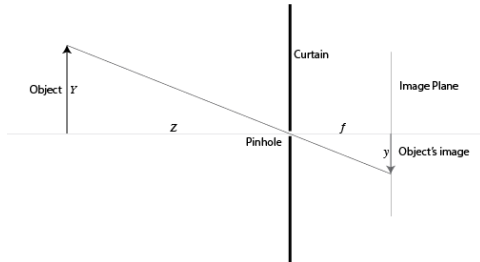
Similar triangles: the 2D view first

Ignore the horizontal axis for a moment. Look at the camera from the side.



Similar triangles: the 2D view first

Ignore the horizontal axis for a moment. Look at the camera from the side.

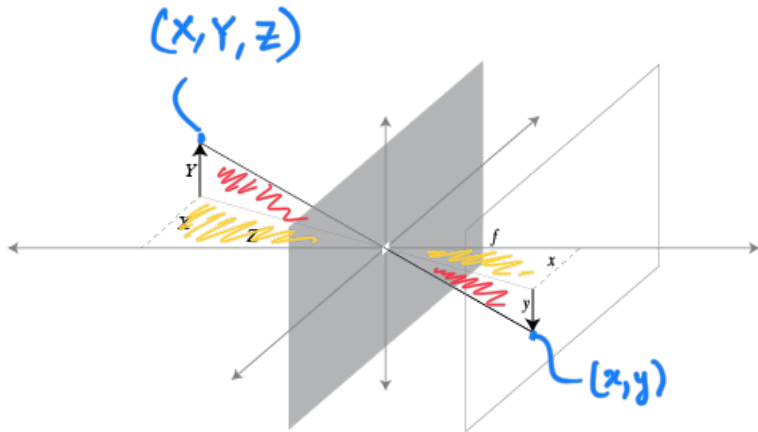


The ray through the pinhole makes two similar triangles, so

$$\frac{y}{f} = \frac{Y}{Z} \quad \implies \quad y = f \frac{Y}{Z}$$

That is the whole of perspective projection, in one dimension.

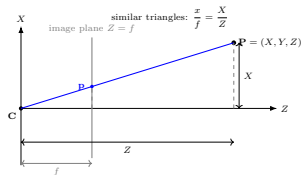
Perspective projection in 3D



$$y = f \frac{Y}{Z}$$
$$x = f \frac{X}{Z}$$

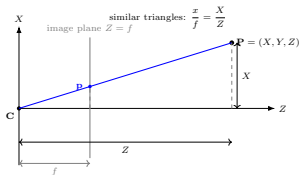
Perspective projection in 3D

Camera at the origin, optical axis along $+Z$, virtual image plane at $Z = f$.



Perspective projection in 3D

Camera at the origin, optical axis along $+Z$, virtual image plane at $Z = f$.



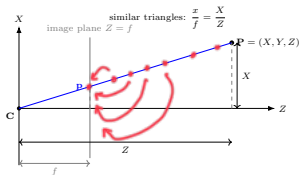
The same similar-triangle argument holds for Y , so

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}$$

↗
Perspective projection Equations.
Divide by Z . Perspective divide.

Perspective projection in 3D

Camera at the origin, optical axis along $+Z$, virtual image plane at $Z = f$.



*matrix product
affine transformations*

The same similar-triangle argument holds for X , so

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}$$

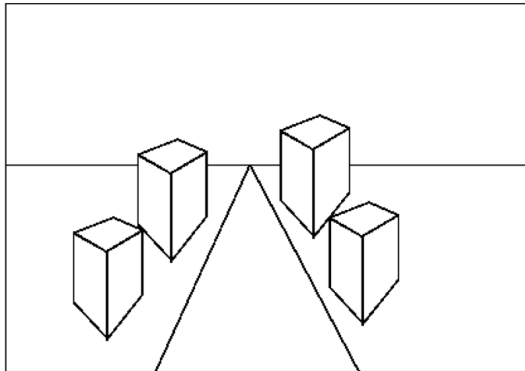
Two consequences, both central to this course:

1. The division by Z makes projection **non-linear**.
2. **Depth is lost**. Every point on the ray through p maps to p .



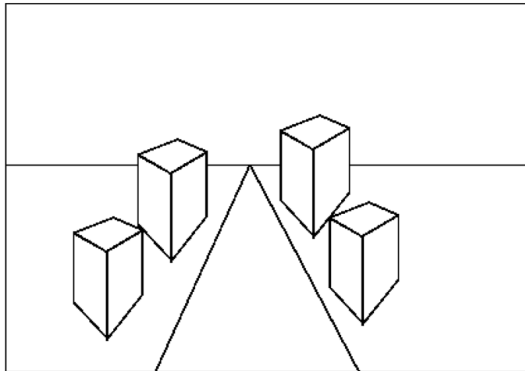
What perspective does

- ▶ **Farther is smaller.** Size in the image goes as $1/Z$.
- ▶ **Parallel lines converge** at a **vanishing point.**
- ▶ **Straight lines stay straight.** Used in Weeks 7 and 9.
- ▶ **Angles are not preserved.** Rectangles become general quadrilaterals.



What perspective does

- ▶ **Farther is smaller.** Size in the image goes as $1/Z$.
- ▶ **Parallel lines converge** at a **vanishing point.**
- ▶ **Straight lines stay straight.** Used in Weeks 7 and 9.
- ▶ **Angles are not preserved.** Rectangles become general quadrilaterals.



Next week: the vanishing point of the optical-axis direction is the **principal point**.
Predicted then measured in Lab 2.

Aperture: a trade-off, not a limit

Shrinking the hole sharpens the image: each sensor point sees a narrower cone.

pinhole.

Aperture: a trade-off, not a limit

Shrinking the hole sharpens the image: each sensor point sees a narrower cone.

But:

- ▶ **Too large:** each scene point spreads over a disc. Geometric blur.
- ▶ **Too small: diffraction** dominates and the spot widens again.
- ▶ **Too small** also means very little light, so exposure time grows and motion blur follows.

Aperture: a trade-off, not a limit

Shrinking the hole sharpens the image: each sensor point sees a narrower cone.

But:

- ▶ **Too large:** each scene point spreads over a disc. Geometric blur.
- ▶ **Too small: diffraction** dominates and the spot widens again.
- ▶ **Too small** also means very little light, so exposure time grows and motion blur follows.

There is an optimum, and it is not “as small as possible”. This is the point of in-class exercise w01a-pinhole.



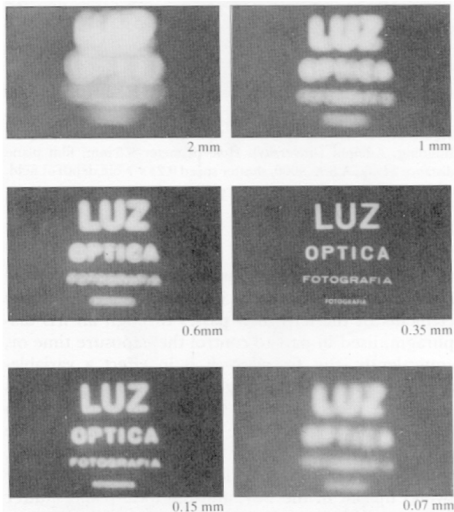
Aperture: the trade-off



Pinhole is too big. Many directions are averaged. Blurring the image.

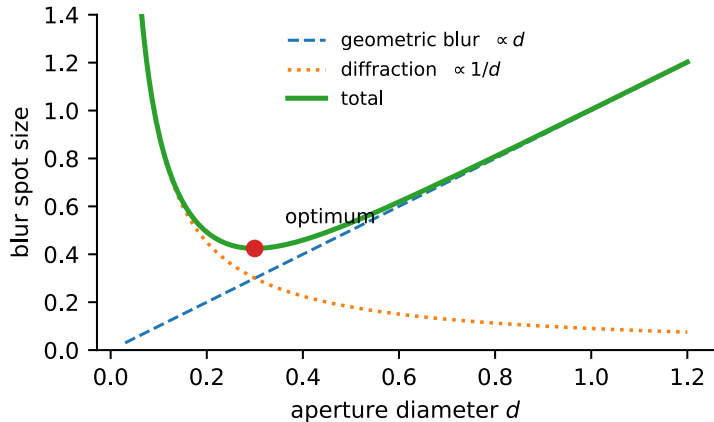
Pinhole is too small. Diffraction effects blur the image.

Images captured by pinhole cameras are generally very dark, since a very small set of rays from any point make it to the image plane



Slide credit: "Computer Vision: A Modern Approach" 2e by D.A. Forsyth

Aperture: the trade-off



Why real cameras use lenses

A lens gathers light over a large aperture and still focuses it to a point (for objects at one particular distance).

- ▶ Much more light, so shorter exposures.
- ▶ At the price of **finite depth of field** and **distortion**.

Why real cameras use lenses

A lens gathers light over a large aperture and still focuses it to a point (for objects at one particular distance).

- ▶ Much more light, so shorter exposures.
- ▶ At the price of **finite depth of field** and **distortion**.

For the geometry in this course, a well-corrected lens behaves like a pinhole. We will model it as one, and add **radial distortion** as a correction in Week 2.

Why real cameras use lenses

A lens gathers light over a large aperture and still focuses it to a point (for objects at one particular distance).

- ▶ Much more light, so shorter exposures.
- ▶ At the price of **finite depth of field** and **distortion**.

For the geometry in this course, a well-corrected lens behaves like a pinhole. We will model it as one, and add **radial distortion** as a correction in Week 2.

Lens design, depth of field and the imaging pipeline belong to a computational photography course. We will not cover them.

In-class exercise: w01a-pinhole

1. A rectangular object $10\text{ m} \times 5\text{ m}$ sits 50 m from a pinhole camera of focal length 5 cm . What is the **area** of its image?
2. An ideal pinhole is made as small as it can be without diffraction, and used to image a distant object. Under what conditions is the image still blurry?

In-class exercise: w01a-pinhole

1. A rectangular object $10\text{ m} \times 5\text{ m}$ sits 50 m from a pinhole camera of focal length 5 cm . What is the **area** of its image?
2. An ideal pinhole is made as small as it can be without diffraction, and used to image a distant object. Under what conditions is the image still blurry?

Hint for 1: magnification is f/Z . Areas scale as the **square** of the magnification. Forgetting the square is a very common slip.

What projection took away

Given a single image and the pinhole model, we know the **ray** each pixel came from, but not **how far** along it the world point sits.

To recover depth we need extra information:

- ▶ a **second view** (stereo, Weeks 9–10),
- ▶ **motion** over time (Week 8),
- ▶ or a **known constraint**, such as the scene lying on a plane.

What projection took away

Given a single image and the pinhole model, we know the **ray** each pixel came from, but not **how far** along it the world point sits.

To recover depth we need extra information:

- ▶ a **second view** (stereo, Weeks 9–10),
- ▶ **motion** over time (Week 8),
- ▶ or a **known constraint**, such as the scene lying on a plane.

In Lab 2 you will use the last of these: knowing the camera's height above a flat road is enough to turn any road pixel back into metres.

Week 1 recap

- ▶ Vision inverts a lossy mapping; that is why it is hard.
- ▶ An image is a sampled function on a top-left-origin grid.
- ▶ The pinhole model: $x = fX/Z$, $y = fY/Z$.
- ▶ Projection is non-linear, destroys depth, preserves straightness, and makes parallel lines converge.
- ▶ Aperture size is a trade-off between geometric blur and diffraction.

Next week: writing this projection as a *matrix*, so we can compose it with camera pose and finally calibrate a real camera.

This week's work

- ▶ **Reading:** Szeliski Sec. 1.1, 2.1.
- ▶ **In-class exercise:** w01a-pinhole.
- ▶ **Lab 1:** Python setup, image and video I/O. Get the environment working *this week*; every later lab builds on it.

Copyright and License

©Faisal Z. Qureshi



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.