Lecture 2: Image Formation

(unless otherwise stated slides taken or adapted from Bob Woodham, Jim Little, Fred Tung and Leon Sigal)
Menu for Today (January 13, 2021)

Topics:
— Image Formation
— Cameras and Lenses
— Projection
— Human eye (as camera)

Readings:
— **Today’s** Lecture: Forsyth & Ponce (2nd ed.) 1.1.1 — 1.1.3
— **Next** Lecture: Forsyth & Ponce (2nd ed.) 4.1, 4.5

Reminders:
— Complete **Assignment 0** (ungraded) by Monday, **January 18**
— Google Colab Tutorials next week
— **TA and Office** hours are posted and will start on Monday, **January 18**
Today’s “fun” Example

Photo credit: reddit user Liammm
Today’s “fun” Example: Eye Sink Illusion

Photo credit: reddit user Liammm
Today’s “fun” Example: Eye Sink Illusion

“Tried taking a picture of a sink draining, wound up with a picture of an eye instead”

Photo credit: reddit user Liammm
Surprising faces!

Photo credit: camaro5.com
Lecture 1: Re-cap

Types of computer vision problems:
— Computing properties of the 3D world from visual data (measurement)
— Recognition of objects and scenes (perception and interpretation)
— Search and interact with visual data (search and organization)
— Manipulation or creation of image or video content (visual imagination)

Computer vision challenges:
— Fundamentally ill-posed
— Enormous computation and scale
— Lack of fundamental understanding of how human perception works
Computer vision technologies have moved from research labs into commercial products and services. Examples cited include:

— broadcast television sports
— electronic games (Microsoft Kinect)
— biometrics
— image search
— visual special effects
— medical imaging
— robotics

... many others
Related Disciplines

Artificial Intelligence (AI)

Computer Vision

Scope of CPSC 425

Image Processing
Geometric Reasoning
Recognition

Deep Learning

Machine Learning

Robotics

Graphics

Human Computer Interaction

Computational Photography

Medical Imaging

Optics

Neuroscience

Slide Credit: James Hays (GA Tech)
Related Disciplines: Vision and Graphics

Inverse problems: analysis and synthesis

(it is sometimes useful to think about computer vision as inverse graphics)

Slide Credit: Kristen Grauman (UT Austin)
Why Study Computer Vision?

It is one of the **most exciting areas of research** in computer science.

Among the **fastest growing technologies** in the industry today.
WHO’S SHAPING THE DIGITAL WORLD?
63. Yann LeCun

*Director of AI research, Facebook, Menlo Park*

LeCun is a leading expert in deep learning and heads up what, for Facebook, could be a hugely significant source of revenue: understanding its user’s intentions.

62. Richard Branson

*Founder, Virgin Group, London*

Branson saw his personal fortune grow £550 million when Alaska Air bought Virgin America for $2.6 billion in April. He is pressing on with civilian space travel with *Virgin Galactic*.

61. Taylor Swift

*Entertainer, Los Angeles*
CVPR Attendance
Lecture 2: Goal

To understand how images are formed
What is **Computer Vision**?

Compute vision, broadly speaking, is a research field aimed to enable computers to **process and interpret visual data**, as sighted humans can.
The **image formation process** that produces a particular image depends on:

- **Lighting** conditions
- **Scene geometry**
- **Surface** properties
- **Camera optics**
- **Sensor properties**

Sensor (or eye) **captures amount of light** reflected from the object.
(small) **Graphics** Review
Graphics Review

Surface reflection depends on both the viewing \((\theta_v, \phi_v)\) and illumination \((\theta_i, \phi_i)\) direction, with Bidirectional Reflection Distribution Function: \(\text{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v)\).

**Lambertian surface:**

\[
\text{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho d}{\pi}
\]

constant, called **albedo**

---

Slide adopted from: Ioannis (Yannis) Gkioulekas (CMU)
Surface reflection depends on both the **viewing** $(\theta_v, \phi_v)$ and **illumination** $(\theta_i, \phi_i)$ direction, with Bidirectional Reflection Distribution Function: \[ \text{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) \]

**Lambertian surface:**
\[ \text{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho d}{\pi} \]
\[ L = \frac{\rho d}{\pi} I(\vec{i} \cdot \vec{n}) \]

A Lambertian surface appears the same (brightness) from all directions.

*Slide adopted from:* Ioannis (Yannis) Gkioulekas (CMU)
Lambertian sphere

“A surface that looks the same from any direction, i.e., surface points are the same brightness
**Graphics Review**

Surface reflection depends on both the **viewing** \((\theta_v, \phi_v)\) and **illumination** \((\theta_i, \phi_i)\) direction, with Bidirectional Reflection Distribution Function: \(\text{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v)\)

**Lambertian** surface:

\[
\text{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho_d}{\pi}
\]

**Mirror** surface: all incident light reflected in one direction \((\theta_v, \phi_v) = (\theta_r, \phi_r)\)

*Slide adopted from: Ioannis (Yannis) Gkioulekas (CMU)*
Let's design a camera

— **Idea #1:** Lets put a film (or digital sensor) in front of an object
— Do we get a reasonable image?

*Slide Credit: Stephen Seitz (University of Washington)*
Pinhole Camera

The opening (hole) in the barrier is called an aperture

Let's design a camera

— **Idea #2**: Add a barrier to block off the majority of the rays

— This reduces blurring

*Slide Credit: Stephen Seitz (University of Washington)*
Cameras

Old school **film** camera

Digital **CCD/CMOS** camera
Cameras

Old school film camera

Digital CCD/CMOS camera
Let’s say we have a sensor …

Digital CCD/CMOS camera

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
... and the **object** we would like to photograph

What would an image taken like this look like?

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Bare-sensor imaging

real-world object

digital sensor (CCD or CMOS)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Bare-sensor imaging

real-world object

digital sensor (CCD or CMOS)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Bare-sensor imaging

real-world object

digital sensor (CCD or CMOS)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Bare-sensor imaging

All scene points contribute to all sensor pixels

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Bare-sensor imaging

All scene points contribute to all sensor pixels

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
What would an image taken like this look like?

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Pinhole Camera

real-world object

digital sensor (CCD or CMOS)

most rays are blocked

one makes it through

Slide Credit: Ioannis (Yannis) Gkioulkas (CMU)
Pinhole Camera

Each scene point contributes to only one sensor pixel

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Camera Obscura (Latin for “dark chamber”) principles behind the pinhole camera or camera obscura were first mentioned by Chinese philosopher Mozi (Mo-Ti) (470 to 390 BCE).

Reinerus Gemma-Frisius observed an eclipse of the sun at Louvain on January 24, 1544. He used this illustration in his book, “De Radio Astronomica et Geometrica,” 1545. It is thought to be the first published illustration of a camera obscura.

Credit: John H., Hammond, “Th Camera Obscure, A Chronicle”
First **Photograph** on Record

*La table servie*

Credit: Nicéphore Niepce, 1822
A pinhole camera is a box with a small hall (aperture) in it.
A pinhole camera is a box with a small hall (aperture) in it.
Image Formation

Forsyth & Ponce (2nd ed.) Figure 1.1

Credit: US Navy, Basic Optics and Optical Instruments. Dover, 1969
Elsa Dorfman and the large format camera

Credit: TuftsNow
Accidental Pinhole Camera

The pinhole camera is not just an abstraction; these are images of "accidental" room-sized pinhole cameras that resulted from having blinds with small holes in them.

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)
What’s this?
**Pinhole Camera (Simplified)**

$f'$ is the **focal length** of the camera.
Pinhole Camera (Simplified)

\[ f' \] is the **focal length** of the camera

**Note:** In a pinhole camera we can adjust the focal length, all this will do is change the **size** of the resulting image.
Pinhole Camera (Simplified)

It is convenient to think of the image plane which is in from of the pinhole

What happens if object moves towards the camera? Away from the camera?
Perspective Effects

Far objects appear smaller than close ones

Forsyth & Ponce (2nd ed.) Figure 1.3a
Perspective Effects

Far objects appear smaller than close ones

Forsyth & Ponce (2nd ed.) Figure 1.3a
Perspective Effects

Far objects appear **smaller** than close ones

Size is **inversely** proportional to distance
**Perspective Effects**

Parallel lines meet at a point (**vanishing point**)

Forsyth & Ponce (1st ed.) Figure 1.3b
Vanishing Points

Each set of parallel lines meets at a different point — the point is called the vanishing point.
Vanishing Points

Each set of parallel lines meet at a different point — the point is called **vanishing point**

Sets of parallel lines one the same plane lead to **collinear** vanishing points — the line is called a **horizon** for that plane
Vanishing Points

- Draw a horizon line.
Vanishing Points

1. Draw a horizon line.
2. Make a vanishing point.
3. Draw a square or rectangle.
4. Draw orthogonals from shape corners to vanishing point.
5. Draw a vertical line to make the form’s side.
6. Erase the orthogonals.
7. Draw another form!
8. Add windows and doors.

Slide Credit: David Jacobs
Vanishing Points

Each set of parallel lines meet at a different point
— the point is called **vanishing point**

Sets of parallel lines on the same plane lead to **collinear** vanishing points
— the line is called a **horizon** for that plane

Good way to **spot fake images**
— scale and perspective do not work
— vanishing points behave badly
Vanishing Points

One point perspective

Two point perspective

Slide Credit: Efros (Berkeley), photo from Criminisi
Perspective Aside

One point perspective

Two point perspective

Three point perspective

Image credit: http://www.martinacecilia.com/place-vanishing-points/
Perspective Aside

Image credit: http://www.martinacecilia.com/place-vanishing-points/60
Perspective Aside

Image credit: http://www.martinacecilia.com/place-vanishing-points/
Properties of Projection

— Points project to points
— Lines project to lines
— Planes project to the whole or half image
— Angles are not preserved
— Incidences (intersections) are preserved
Properties of Projection

— Points project to points
— Lines project to lines
— Planes project to the whole or half image
— Angles are not preserved

Degenerate cases
— Line through focal point (pinhole or aperture) projects to a point
— Plane through focal point projects to a line
Projection Illusion
Optional subtitle
With one-point and two-point perspective, all the height lines on the building are perpendicular from the ground up.

With three-point perspective, the height lines converge at one point at the top.

This technique is used for buildings viewed from below or above and brings out the power of the structure.
Perspective Projection

3D object point $P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$ where

\[
\begin{align*}
x' &= f' \frac{x}{z} \\
y' &= f' \frac{y}{z}
\end{align*}
\]

Note: this assumes world coordinate frame at the optical center (pinhole) and aligned with the image plane, image coordinate frame aligned with the camera coordinate frame.
Perspective Projection: Proof

3D object point $P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$ where

$$x' = f' \frac{x}{z}$$
$$y' = f' \frac{y}{z}$$

Note: this assumes world coordinate frame at the optical center (pinhole) and aligned with the image plane, image coordinate frame aligned with the camera coordinate frame.
Aside: Camera Matrix

Forsyth & Ponce (1st ed.) Figure 1.4

3D object point

\[
P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
\]

projects to 2D image point

\[
P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}
\]

where

\[
P' = CP
\]

Camera Matrix

\[
C = \begin{bmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}
\]

Note: this assumes world coordinate frame at the optical center (pinhole) and aligned with the image plane, image coordinate frame aligned with the camera coordinate frame
**Perspective Projection: Proof**

Forsyth & Ponce (1st ed.) Figure 1.4

3D object point

\[ P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \]

projects to 2D image point

\[ P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \]

where

\[ P' = CP \]

**Note:** this assumes world coordinate frame at the optical center (pinhole) and aligned with the image plane, image coordinate frame aligned with the camera coordinate frame
Aside: Camera Matrix

**Camera Matrix**

\[
C = \begin{bmatrix}
f' & 0 & 0 & 0 \\
0 & f' & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

\[
x' = f' \frac{x}{z} \\
y' = f' \frac{y}{z}
\]

\[
P = \begin{bmatrix}
x \\
y \\
z \\
1 \\
\end{bmatrix}
\]

projects to 2D image point

\[
P' = \begin{bmatrix}
x' \\
y' \\
1 \\
\end{bmatrix}
\]

where

\[
P' = CP
\]
Aside: Camera Matrix

**Camera Matrix**

\[
C = \begin{bmatrix}
f' & 0 & 0 & 0 \\
0 & f' & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
f' & 0 & 0 & 0 \\
0 & f' & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= \begin{bmatrix}
f'x \\
f'y \\
f'z \\
z
\end{bmatrix}
= \begin{bmatrix}
\frac{x}{z} \\
\frac{y}{z}
\end{bmatrix}
\]

\[x' = f' \frac{x}{z} \]
\[y' = f' \frac{y}{z}\]

\[P = \begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}\] projects to 2D image point

\[P' = \begin{bmatrix}
x' \\
y'
\end{bmatrix}\] where \[P' = CP\]
Aside: Camera Matrix

Camera Matrix

\[ C = \begin{bmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \]

Pixels are squared / lens is perfectly symmetric
Sensor and pinhole perfectly aligned
Coordinate system centered at the pinhole

\[ P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \] projects to 2D image point
\[ P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \]

where \[ P' = CP \]
Aside: Camera Matrix

Camera Matrix

\[
C = \begin{bmatrix}
  f_x' & 0 & 0 & 0 \\
  0 & f_y' & 0 & 0 \\
  0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

Pixels are squared / lens is perfectly symmetric
Sensor and pinhole perfectly aligned
Coordinate system centered at the pinhole

\[
P = \begin{bmatrix}
  x \\
  y \\
  z \\
  1 \\
\end{bmatrix}
\]
projects to 2D image point

\[
P' = \begin{bmatrix}
  x' \\
  y' \\
  1 \\
\end{bmatrix}
\]
where

\[
P' = CP
\]
Aside: Camera Matrix

**Camera Matrix**

\[
C = \begin{bmatrix}
  f'_x & 0 & 0 & c_x \\
  0 & f'_y & 0 & c_y \\
  0 & 0 & 1 & 0
\end{bmatrix}
\]

- Pixels are squared / lens is perfectly symmetric
- Sensor and pinhole perfectly aligned
- Coordinate system centered at the pinhole

\[
P = \begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
projects to 2D image point

\[
P' = \begin{bmatrix}
x' \\
y'
\end{bmatrix}
\]
where

\[
P' = CP
\]
Aside: Camera Matrix

**Camera Matrix**

\[ C = \begin{bmatrix} f_x' & 0 & 0 & c_x \\ 0 & f_y' & 0 & c_y \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbb{R}^{4 \times 4} \]

- Pixels are squared / lens is perfectly symmetric
- Sensor and pinhole perfectly aligned
- Coordinate system centered at the pinhole

\[ P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \] projects to 2D image point \[ P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \] where \[ P' = CP \]
Aside: Camera Matrix

Camera Matrix

\[
C = \begin{bmatrix}
  f' & 0 & 0 & c_x \\ 0 & f' & 0 & c_y \\ 0 & 0 & 1 & 0
\end{bmatrix}_{4 \times 4}
\]

Camera calibration is the process of estimating parameters of the camera matrix based on set of 3D-2D correspondences (usually requires a pattern whose structure and size is known)

\[
P = \begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]

projects to 2D image point

\[
P' = \begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix}
\]

where

\[
P' = CP
\]
Perspective Projection

3D object point

\[ P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \]

projects to 2D image point

\[ P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \]

where

\[ x' = f' \frac{x}{z} \]
\[ y' = f' \frac{y}{z} \]

Note: this assumes world coordinate frame at the optical center (pinhole) and aligned with the image plane, image coordinate...
Weak Perspective

Forsyth & Ponce (1st ed.) Figure 1.5

3D object point $P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ in $\Pi_0$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$

where $x' = mx$ and $y' = my$

and $m = \frac{f'}{z_0}$
Orthographic Projection

3D object point \( P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \) projects to 2D image point \( P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \)

where \( x' = x \) and \( y' = y \)

Forsyth & Ponce (1st ed.) Figure 1.6
Summary of **Projection Equations**

3D object point \( P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \) projects to 2D image point \( P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \) where

- **Perspective**
  \[
  \begin{align*}
  x' &= f' \frac{x}{z} \\
  y' &= f' \frac{y}{z}
  \end{align*}
  \]

- **Weak Perspective**
  \[
  \begin{align*}
  x' &= mx \\
  y' &= my \\
  m &= \frac{f'}{z_0}
  \end{align*}
  \]

- **Orthographic**
  \[
  \begin{align*}
  x' &= x \\
  y' &= y
  \end{align*}
  \]
Projection Models: Pros and Cons

Weak perspective (including orthographic) has simpler mathematics
— accurate when object is small and/or distant
— useful for recognition

Perspective is more accurate for real scenes

When maximum accuracy is required, it is necessary to model additional details of a particular camera
— use perspective projection with additional parameters (e.g., lens distortion)
Readings:

— **Today’s** Lecture: Forsyth & Ponce (2nd ed.) 1.1.1 — 1.1.3
— **Next** Lecture: Forsyth & Ponce (2nd ed.) 4.1, 4.5

Reminders:

— Complete **Assignment 0** (ungraded) by Monday, **January 18**
— **Piazza**: [piazza.com/ubc.ca/winterterm12019/cpsc425](piazza.com/ubc.ca/winterterm12019/cpsc425)