Course staff

**Haozhu Wang**  
Graduate student (GSI)

**Anthony Liang**  
Instructional aide (IA)

**Bingqi Sun**  
Instructional aide (IA)
Interacting with us

• Ask questions on Piazza.
• Submit written work to Gradescope
• Office hours on website:

<table>
<thead>
<tr>
<th>Name</th>
<th>Office hour times</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Owens</td>
<td>Fri. 3:00pm - 4:00pm</td>
<td>EECS 4231</td>
</tr>
<tr>
<td>Haozhu Wang</td>
<td>Thu. 7:30pm - 8:30pm</td>
<td>EECS 3312</td>
</tr>
<tr>
<td>Bingqi Sun</td>
<td>Mon. 12:00 - 1:00</td>
<td>EECS 3312</td>
</tr>
<tr>
<td>Anthony Liang</td>
<td>Mon. 1:00 - 2:00pm</td>
<td>EECS 3312</td>
</tr>
</tbody>
</table>
# EECS 504: Foundations of Computer Vision

Instructor: Andrew Owens  
Tues & Thurs, 12:00 - 1:30 in 1610 IOE  
Winter 2020

## Schedule

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Date</th>
<th>Topic</th>
<th>Materials</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lec. 1</td>
<td>Thu, Jan. 9</td>
<td>Introduction / Image filters 1</td>
<td><em>Toniade, Freeman, and Iida: A Simple Visual sys (optional reading)</em></td>
<td>ps1 (filtering) out</td>
</tr>
<tr>
<td>Lec. 2</td>
<td>Tue, Jan. 14</td>
<td>Image filters 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lec. 3</td>
<td>Thu, Jan. 16</td>
<td>Signal processing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lec. 4</td>
<td>Tue, Jan. 23</td>
<td>Temporal filters</td>
<td></td>
<td>ps1 due</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ps2 (signal processing) out</td>
</tr>
<tr>
<td>Lec. 5</td>
<td>Thu, Jan. 23</td>
<td>Multi-scale pyramids</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lec. 6</td>
<td>Tue, Jan. 28</td>
<td>Statistical models of images</td>
<td></td>
<td>ps2 due</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ps3 out (texture)</td>
</tr>
<tr>
<td>Lec. 7</td>
<td>Thu, Jan. 30</td>
<td>Machine learning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lec. 8</td>
<td>Tue, Feb. 4</td>
<td>Neural networks</td>
<td></td>
<td>ps3 due</td>
</tr>
</tbody>
</table>

Course website: [https://web.eecs.umich.edu/~ahowens/eecs504/w20/](https://web.eecs.umich.edu/~ahowens/eecs504/w20/)
Grading

- Assignments (70%)
- Final project (30%)
Assignments

• Weekly homework assignments (≈10 total)
• Due each Tuesday at midnight
• Late submissions penalized 30% per day
  - You have 5 "late days"
• Assignments should be done independently.
  - Encouraged to discuss them
  - Programming and writing should all be yours
Assignments

• Mix of programming and written problems
• Python + numerical computing libraries (numpy, scipy, etc.)
• PyTorch for deep learning
• Linear algebra and multivariable calculus
• Jupyter notebooks and Google Colab for problem sets
Assignments

Table of contents

EECS 504 PS1: Filtering

- Introduction

Starting

Problem 1.2: Pet edge detection (a)

Problem 1.2: Pet edge detection (b)

Problem 1.2: Pet edge detection (c)

Problem 1.2: Pet edge detection (d)

Problem 1.2: Pet edge detection (e)

- Problem 1.2: Pet edge detection (a)

# You can upload images yourself or load them from BING

```python
# Load an image from file
image = cv2.imread('image.png')
```

# Visualize edge maps using matplotlib

```python
plt.figure()
plt.imshow(image, cmap='gray', vmin=0, vmax=1)
plt.axis('off')
```

# Thresholding

```python
edges = cv2.Canny(image, threshold1=100, threshold2=200)
plt.imshow(edges, cmap='gray', vmin=0, vmax=1)
plt.axis('off')
```

- Problem 1.2: Pet edge detection (b)

```python
# Visualize edge maps using matplotlib
plt.figure()
plt.imshow(image, cmap='gray', vmin=0, vmax=1)
plt.axis('off')
```

- Problem 1.2: Pet edge detection (c)

```python
# Visualize edge maps using matplotlib
plt.figure()
plt.imshow(image, cmap='gray', vmin=0, vmax=1)
plt.axis('off')
```

- Problem 1.2: Pet edge detection (d)

```python
# Visualize edge maps using matplotlib
plt.figure()
plt.imshow(image, cmap='gray', vmin=0, vmax=1)
plt.axis('off')
```

- Problem 1.2: Pet edge detection (e)

```python
# Visualize edge maps using matplotlib
plt.figure()
plt.imshow(image, cmap='gray', vmin=0, vmax=1)
plt.axis('off')
```
Project

- Open-ended! Example projects:
  - Implement and extend a recent computer vision paper
  - Use computer vision in your research
  - We’ll also provide a list of project ideas
- Work in small groups (up to 4 people)
- Complete in last month of class.
  - Project proposal (after spring break)
  - Short presentation (finals period)
  - Writeup (finals period)
Readings

http://szeliski.org/Book

https://www.deeplearningbook.org

Manuscript chapters by Torralba, Freeman, and Isola (on course website). Class based on this coursework.

And also occasional paper readings
Class topics
Signal processing

Homework problem:

\[ \text{Apple} + \text{orange} = ? \]

<table>
<thead>
<tr>
<th>Lec. 1</th>
<th>Thu, Jan. 9</th>
<th>Introduction / Image filters 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lec. 2</td>
<td>Tue, Jan. 14</td>
<td>Image filters 2</td>
</tr>
<tr>
<td>Lec. 3</td>
<td>Thu, Jan. 16</td>
<td>Signal processing</td>
</tr>
<tr>
<td>Lec. 4</td>
<td>Tue, Jan. 21</td>
<td>Temporal filters</td>
</tr>
<tr>
<td>Lec. 5</td>
<td>Thu, Jan. 23</td>
<td>Multi-scale pyramids</td>
</tr>
<tr>
<td>Lec. 6</td>
<td>Tue, Jan. 28</td>
<td>Statistical models of images</td>
</tr>
<tr>
<td>Lec. 7</td>
<td>Thu, Jan. 30</td>
<td>Machine learning</td>
</tr>
<tr>
<td>Lec. 8</td>
<td>Tue, Feb. 4</td>
<td>Neural networks</td>
</tr>
<tr>
<td>Lec. 9</td>
<td>Thu, Feb. 6</td>
<td>Optimization</td>
</tr>
<tr>
<td>Lec. 10</td>
<td>Tue, Feb. 11</td>
<td>Convolutional networks</td>
</tr>
<tr>
<td>Tutorial</td>
<td>Tue, Feb. 11</td>
<td>PyTorch tutorial</td>
</tr>
<tr>
<td>Lec.</td>
<td>Date</td>
<td>Topic</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Lec. 1</td>
<td>Thu, Jan. 9</td>
<td>Introduction / Image filters 1</td>
</tr>
<tr>
<td>Lec. 2</td>
<td>Tue, Jan. 14</td>
<td>Image filters 2</td>
</tr>
<tr>
<td>Lec. 3</td>
<td>Thu, Jan. 16</td>
<td>Signal processing</td>
</tr>
<tr>
<td>Lec. 4</td>
<td>Tue, Jan. 21</td>
<td>Temporal filters</td>
</tr>
<tr>
<td>Lec. 5</td>
<td>Thu, Jan. 23</td>
<td>Multi-scale pyramids</td>
</tr>
<tr>
<td>Lec. 6</td>
<td>Tue, Jan. 28</td>
<td>Statistical models of images</td>
</tr>
<tr>
<td>Lec. 7</td>
<td>Thu, Jan. 30</td>
<td>Machine learning</td>
</tr>
<tr>
<td>Lec. 8</td>
<td>Tue, Feb. 4</td>
<td>Neural networks</td>
</tr>
<tr>
<td>Lec. 9</td>
<td>Thu, Feb. 6</td>
<td>Optimization</td>
</tr>
<tr>
<td>Lec. 10</td>
<td>Tue, Feb. 11</td>
<td>Convolutional networks</td>
</tr>
<tr>
<td>Tutorial</td>
<td>Tue, Feb. 11</td>
<td>PyTorch tutorial</td>
</tr>
</tbody>
</table>

Intro to deep learning
<table>
<thead>
<tr>
<th>Tutorial</th>
<th>Tue, Feb. 11</th>
<th>PyTorch tutorial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lec. 11</td>
<td>Thu, Feb. 13</td>
<td>Scene understanding</td>
</tr>
<tr>
<td>Lec. 12</td>
<td>Tue, Feb. 18</td>
<td>Object detection</td>
</tr>
<tr>
<td>Lec. 13</td>
<td>Thu, Feb. 20</td>
<td>Action recognition</td>
</tr>
<tr>
<td>Lec. 14</td>
<td>Tue, Feb. 25</td>
<td>GANs</td>
</tr>
<tr>
<td>Lec. 15</td>
<td>Thu, Feb. 27</td>
<td>Image synthesis</td>
</tr>
</tbody>
</table>

**Spring break**

| Lec. 16  | Tue, Mar. 10 | Representation learning |
| Lec. 17  | Thu, Mar. 12 | Sight, sound, and touch |
| Lec. 18  | Tue, Mar. 17 | Optical flow |
| Lec. 19  | Thu, Mar. 19 | Multi-view geometry |

**Learning for vision**

**INPUT**

```
undo  clear  random
```

**OUTPUT**

```
Beach
```
Homework problem:

Cameras, optics, motion
### Advanced topics and applications

<table>
<thead>
<tr>
<th>Lec.</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>Thu, Mar. 19</td>
<td>Multi-view geometry</td>
</tr>
<tr>
<td>20</td>
<td>Tue, Mar. 24</td>
<td>Structure from motion</td>
</tr>
<tr>
<td>21</td>
<td>Thu, Mar. 26</td>
<td>Depth estimation</td>
</tr>
<tr>
<td>22</td>
<td>Tue, Mar. 31</td>
<td>Graphical models</td>
</tr>
<tr>
<td>23</td>
<td>Thu, Apr. 2</td>
<td>Color</td>
</tr>
<tr>
<td>24</td>
<td>Tue, Apr. 7</td>
<td>Embodied vision</td>
</tr>
<tr>
<td>25</td>
<td>Thu, Apr. 9</td>
<td>Modern CNNs</td>
</tr>
<tr>
<td>26</td>
<td>Tue, Apr. 14</td>
<td>Image forensics</td>
</tr>
<tr>
<td>27</td>
<td>Thu, Apr. 16</td>
<td>Language and vision</td>
</tr>
<tr>
<td>28</td>
<td>Tue, Apr. 21</td>
<td>Datasets and bias</td>
</tr>
</tbody>
</table>
Today

1. A bit of vision history
2. Why vision is hard
3. A simple visual system
Exciting times for computer vision

Robotics
Medical applications
3D modeling
Driving
Mobile devices
Accessibility

Slide credit: Torralba, Freeman, Isola
To see

“What does it mean, to see? The plain man's answer (and Aristotle's, too) would be, to know what is where by looking.”

To discover from images what is present in the world, where things are, what actions are taking place, to predict and anticipate events in the world.
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".
Just a few years ago…

[“HOGgles”, Vondrick et al. , ICCV 2013]
“Mask RCNN”, He et al., ICCV 2017

Slide credit: Torralba, Freeman, Isola
Different signals, same methods

Sound

Touch

WiFi

(Zhao et al. 2019)

(Calandra et al. 2018)
On-screen audio

(Owens and Efros 2018)
What makes vision hard?
To see: perception vs. measurement
To see: perception vs. measurement
Other ambiguities

Figure 1. (a) A line drawing provides information only about the $x$, $y$ coordinates of points lying along the object contours. (b) The human visual system is usually able to reconstruct an object in three dimensions given only a single 2D projection. (c) Any planar line-drawing is geometrically consistent with infinitely many 3D structures.

Sinha & Adelson 93
A simple visual system

• A simple world
• A simple image formation model
• A simple goal
A simple world

MACHINE PERCEPTION OF THREE-DIMENSIONAL SOLIDS

by

LAWRENCE GILMAN ROBERTS

Submitted to the Department of Electrical Engineering on May 10, 1963, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

The problem of machine recognition of pictorial data has long been a challenging goal, but has seldom been attempted with anything more complex than alphabetic characters. Many people have felt that research on character recognition would be a first step, leading the way to a more general pattern recognition system. However, the multitudinous attempts at character recognition, including my own, have not led very far. The reason, I feel, is that the study of abstract, two-dimensional forms leads us away from, not toward, the techniques necessary for the recognition of three-dimensional objects. The per-

Complete Convex Polygons. The polygon selection procedure would select the numbered polygons as complete and convex. The number indicates the probable number of sides. A polygon is incomplete if one of its points is a collinear joint of another polygon.
A simple world
A simple image formation model

Simple world rules:
• Surfaces can be horizontal or vertical.
• Objects will be resting on a white horizontal ground plane

Slide credit: Antonio Torralba
A simple image formation model

World reference system

Camera plane

Slide credit: Antonio Torralba
A simple image formation model

\[ x = X + x_0 \]
\[ y = \cos(\theta) Y - \sin(\theta) Z + y_0 \]

World coordinates

Image and projection of the world coordinate axes into the image plane

World coordinates

Image coordinates

World coordinates

Image coordinates

Slide credit: Antonio Torralba
A simple goal

Recover the 3D structure of the world

We want to recover $X(x,y)$, $Y(x,y)$, $Z(x,y)$ using as input $I(x,y)$

Slide credit: Antonio Torralba
Treating the image as a function
Finding edges in the image

Image gradient:
\[
\nabla I = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)
\]

Approximation image derivative:
\[
\frac{\partial I}{\partial x} \approx I(x, y) - I(x - 1, y)
\]

Edge strength
\[
E(x, y) = |\nabla I(x, y)|
\]

Edge orientation:
\[
\theta(x, y) = \angle \nabla I = \arctan \frac{\partial I/\partial y}{\partial I/\partial x}
\]

Edge normal:
\[
n = \frac{\nabla I}{|\nabla I|}
\]

Slide credit: Antonio Torralba
Finding edges in the image

\[ \nabla I = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \quad n = \frac{\nabla I}{|\nabla I|} \]

\( E(x,y) \) and \( n(x,y) \)

Slide credit: Antonio Torralba
Edge classification

• Figure/ground segmentation
  – Using the fact that objects have color

• Occlusion edges
  – Occlusion edges are owned by the foreground

• Contact edges

Slide credit: Antonio Torralba
From edges to surface constraints

Slide credit: Antonio Torralba
From edges to surface constraints

• Ground
  \[ Y(x,y) = 0 \quad \text{if } (x,y) \text{ belongs to a ground pixel} \]

• Contact edge
  \[ Y(x,y) = 0 \quad \text{if } (x,y) \text{ belongs to foreground and is a contact edge} \]

• What happens inside the objects?
  
  … now things get a bit more complicated.

Slide credit: Antonio Torralba
From edges to surface constraints

How can we relate the information in the pixels with 3D surfaces in the world?

Vertical edges

Given the image, what can we say about $X$, $Y$ and $Z$ in the pixels that belong to a vertical edge?

$Z = \text{constant along the edge}$

$$\frac{\partial Y}{\partial y} = \frac{1}{\cos(\theta)}$$

Slide credit: Antonio Torralba
From edges to surface constraints

- **Horizontal edges**

Given the image, what can we say about X, Y and Z in the pixels that belong to an horizontal 3D edge?

\[
\begin{align*}
Y &= \text{constant along the edge} \\
\frac{\partial Y}{\partial t} &= 0 \\
\text{Where } &t \text{ is the vector parallel to the edge} \\
& t = (-n_y, n_x) \\
\frac{\partial Y}{\partial t} &= -n_y\frac{\partial Y}{\partial x} + n_x\frac{\partial Y}{\partial y}
\end{align*}
\]
From edges to surface constraints

- What happens where there are no edges?

Assumption of planar faces:

\[
\begin{align*}
\frac{\partial^2 Y}{\partial x^2} &= 0 \\
\frac{\partial^2 Y}{\partial y^2} &= 0 \\
\frac{\partial^2 Y}{\partial y \partial x} &= 0 \\
\end{align*}
\]

Information has to be propagated from the edges

Slide credit: Antonio Torralba
A simple inference scheme

All the constraints are linear!

\[ Y(x,y) = 0 \]

if \((x,y)\) belongs to a ground pixel

\[ \frac{\partial Y}{\partial y} = \frac{1}{\cos(\theta)} \]

if \((x,y)\) belongs to a vertical edge

\[ \frac{\partial Y}{\partial t} = 0 \]

if \((x,y)\) belongs to an horizontal edge

\[ \frac{\partial^2 Y}{\partial x^2} = 0 \]

\[ \frac{\partial^2 Y}{\partial y^2} = 0 \]

\[ \frac{\partial^2 Y}{\partial y \partial x} = 0 \]

if \((x,y)\) is not on an edge

A similar set of constraints could be derived for \(Z\)

Slide credit: Antonio Torralba
Discrete approximation

We can transform every differential constraint into a linear constraint on $Y(x,y)$

$$\frac{dY}{dx} \approx Y(x,y) - Y(x-1,y)$$
Discrete approximation

Transform the “image” $Y(x,y)$ into a column vector:

$Y(x,y) = \begin{bmatrix}
0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}$

$x=2, y=2$

$\frac{dY}{dx} \approx Y(x,y) - Y(x-1,y) = Y(2,2) - Y(1,2) = \begin{bmatrix}
0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}$

Slide credit: Antonio Torralba
A simple inference scheme: solve for $Y$

$$A Y = b$$

Constraint weights

$$\frac{\partial Y}{\partial y} = \frac{1}{\cos(\theta)}$$

Slide credit: Antonio Torralba
Results

Edge normals

Edge strength

3D orientation

Contact edges

Depth discontinuities

Slide credit: Antonio Torralba
Changing view point

New view points:

Slide credit: Antonio Torralba
Failure cases... even in a simple world!
Failure cases... even in a simple world!
Failure cases... even in a simple world!

More on this next week!
Questions about the course?