

CS181: Computer Vision

Introduction to Computer Vision

August 25, 2026

Md Alimoor Reza
Associate Professor of Computer Science
Drake University

https://analytics.drake.edu/~reza/teaching/cs181_fall26/cs181_schedule.html



Introduction



<https://analytics.drake.edu/~reza>

Alimoor Reza

- Associate Professor of Computer Science, Dept. of Mathematics and Computer Science, Drake University
- Office: 323 Collier-Scripps
- Email: md.reza@drake.edu
- Phone: 515-271-1972
- Office hours: Tue/Thur: 12:00-2:30pm CDT
additionally by appointment
[Zoom link](#)

What do you see in this image?



What is the object inside the cropped portion?



What is the object inside that cropped portion?

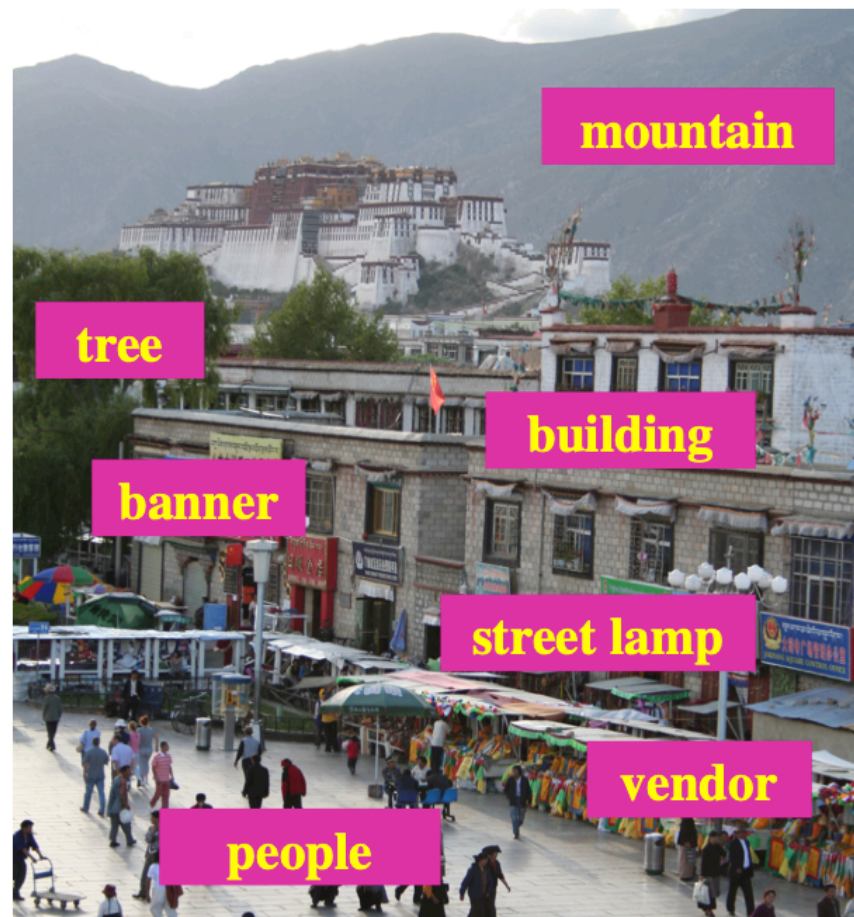


A street lamp

What are the different objects out there?



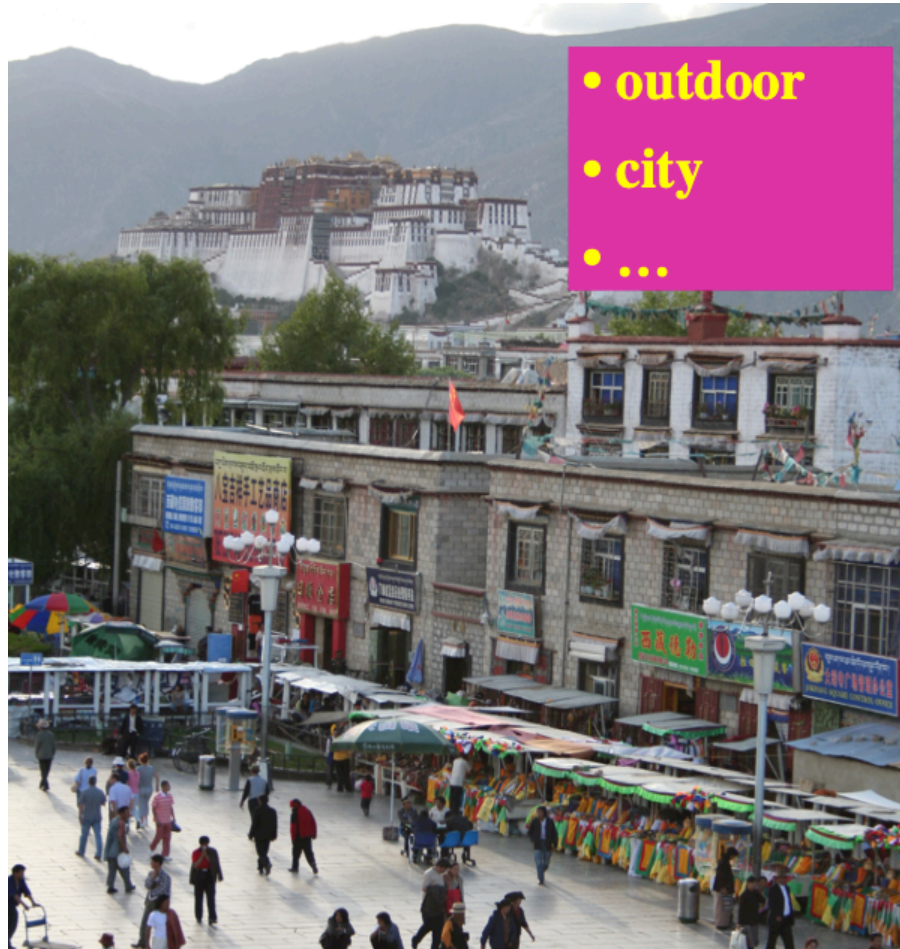
What objects are out there?



What type of scene is this?



What type of scene is this?



Where are the people? Can we locate them?



Where are the people? Can we locate them?



What are the “things” and “stuff” in this image?

Can you precisely demarcate all of them with tight boundaries?



What are the “things” and “stuff” in this image?

Can you precisely demarcate all of them with tight boundaries?



[Reference: Efficient Panoptic Segmentation](#)

What an state-of-the-art AI Model sees?



I am attaching a photo. Explain what do you see in this image?

The image shows a **busy street and market area in Lhasa, Tibet**, with the famous **Potala Palace** dominating the hillside in the background.

Here are the main elements:

- **Potala Palace:** The large white-and-red structure on the mountain is the Potala Palace, historically the winter residence of the Dalai Lamas and one of Tibet's most iconic landmarks.
- **Mountainous landscape:** The palace sits against a dramatic, high-altitude mountain backdrop, characteristic of the Tibetan Plateau.
- **Traditional Tibetan architecture:** The foreground buildings have flat roofs, whitewashed walls, decorative window frames, and colorful Tibetan-style ornamentation.
- **Market stalls:** Numerous stalls are selling brightly colored textiles, clothing, scarves, and other goods. The colorful umbrellas and awnings make the lower part of the image particularly lively.
- **Religious/cultural elements:** You can see **Tibetan prayer flags** and other decorative religious elements around the buildings.
- **Chinese signage:** Several storefronts have Chinese-language signs, reflecting the location's contemporary urban environment.
- **People and street life:** The plaza is crowded with pedestrians and vendors, giving the scene the feel of a busy tourist and commercial district.

Overall, it appears to be a **street-level view of Lhasa with the Potala Palace in the background**, combining Tibetan religious heritage, traditional architecture, tourism, and everyday urban life.



[Reference: What ChatGPT sees.](#)

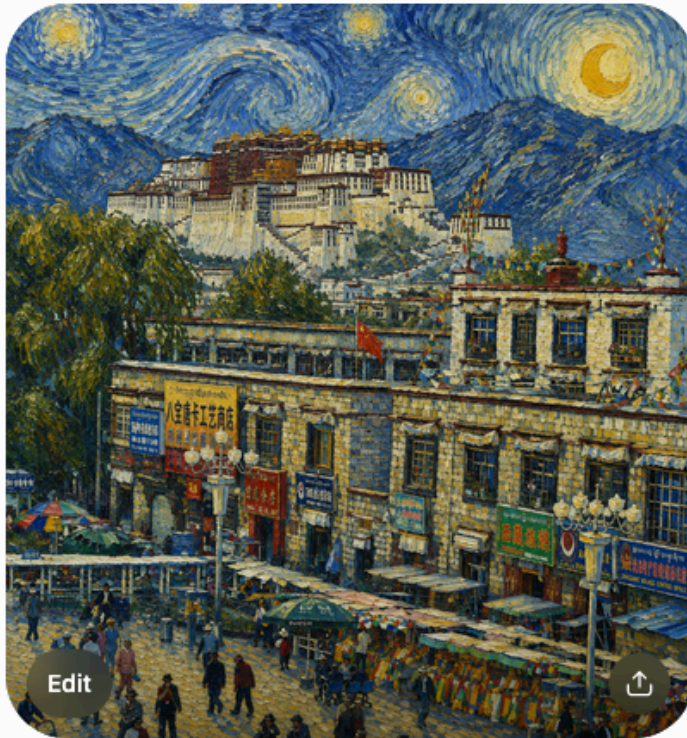
Generate a new image using a generative AI model



[Reference: Picture with an artistic sytel using a Generative-AI Model](#)

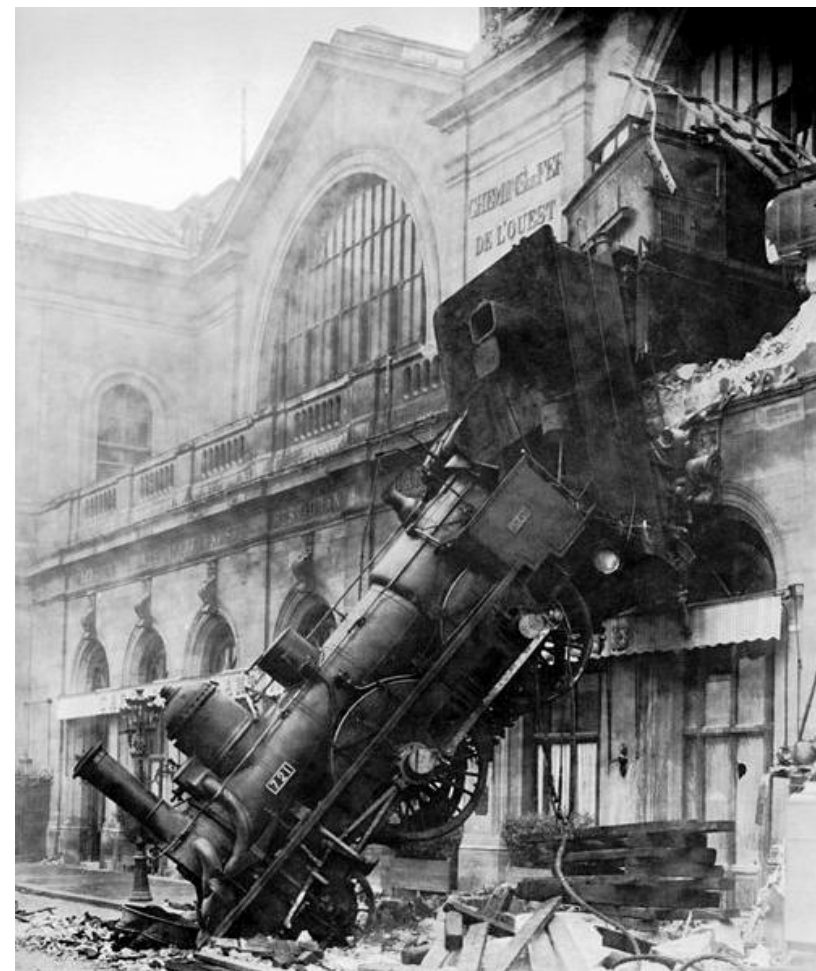
Generate a new image using a generative AI model

Since you are a generative AI model, can you generate a painting of the same scene in the artistic style of Van Gogh?

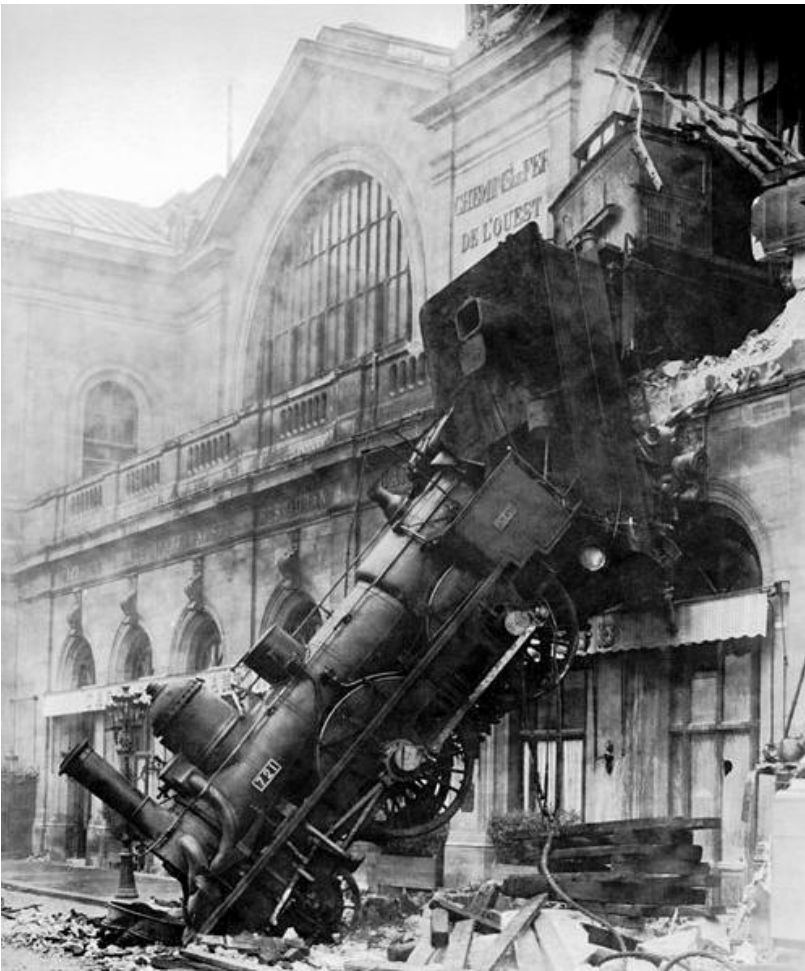


[Reference: Picture with an artistic sytel using a Generative-AI Model](#)

What is computer vision?



0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0



Goal: from images to meaning

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

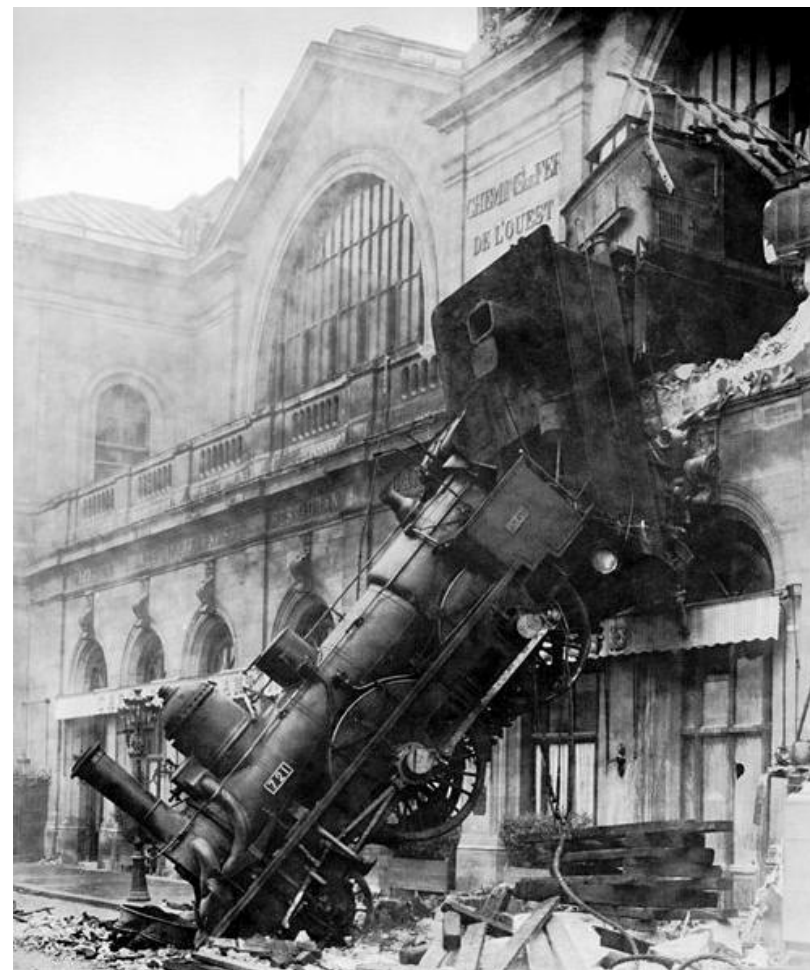


Image processing

- Take an image, produce another image



Computer graphics

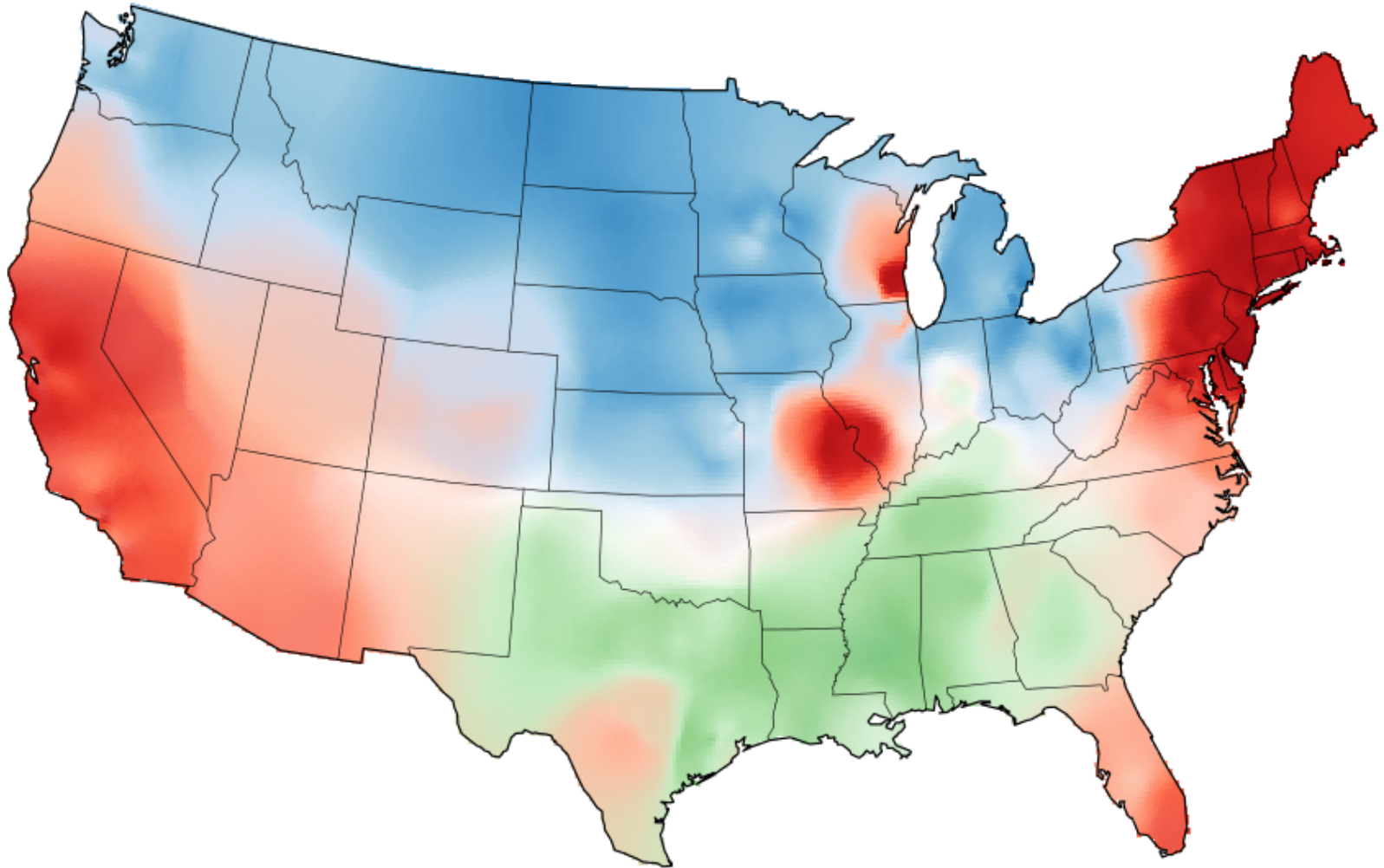
- Take a description, produce an image



From “Final Fantasy”

Visualization

- Take data, produce an image



Computer vision

- Takes an image, understands it (eg, produces a description)



Indoors, office, people, scale, laughing, humor, Obama, mirrors, ...

Can machines see as well as humans?

- Yes and no, but mostly it progressing rapidly
- Current vision technology is useful in select applications, with:
 - Specific, constrained environments, and/or
 - High tolerance for errors

Why is computer vision difficult?

Why is computer vision difficult?



Viewpoint variation



Illumination changes



Scale changes

Why is computer vision difficult?



Intra-class variation



Motion (Source: S. Lazebnik)

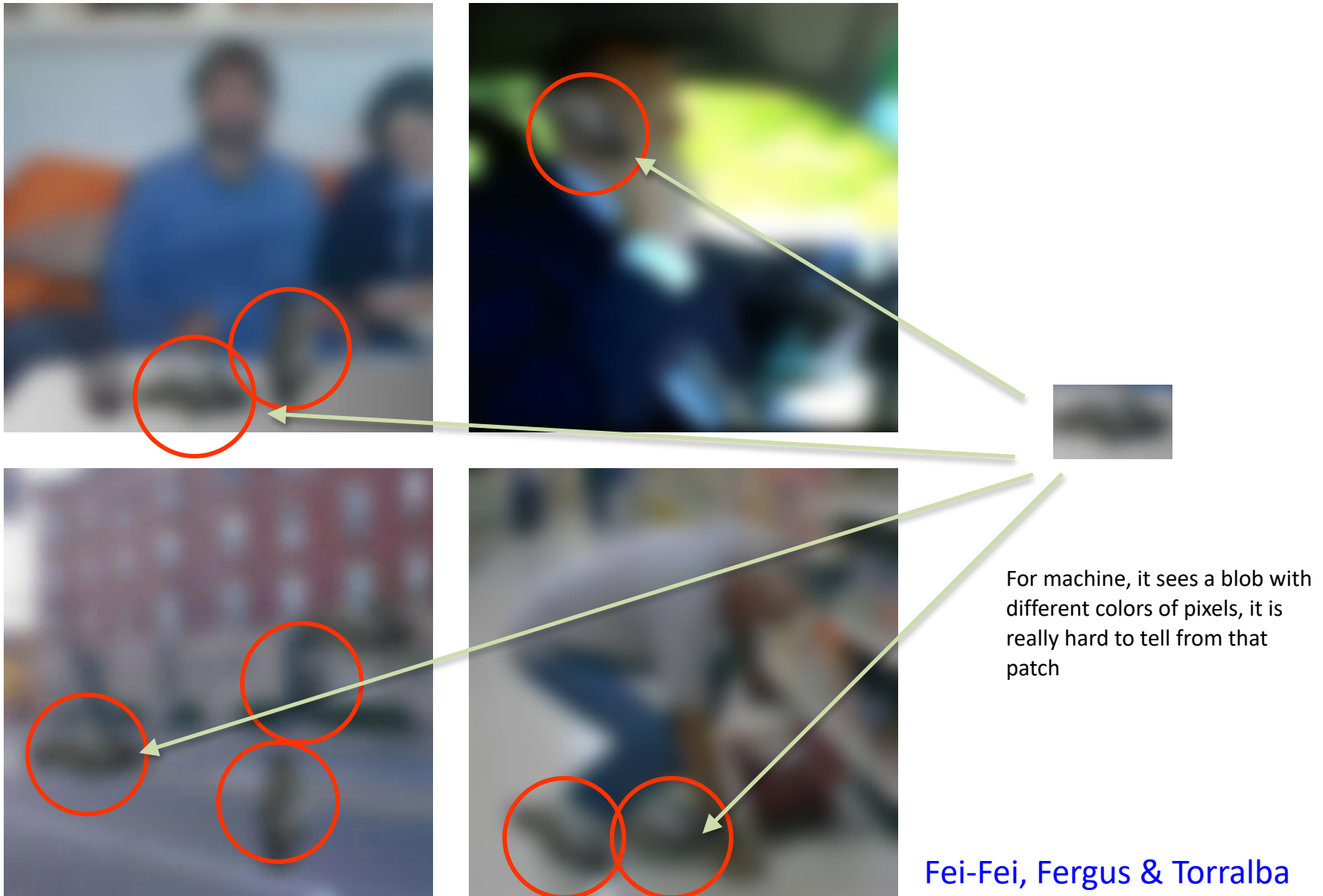


Background clutter



Occlusion

Role of high-level reasoning (using context)



Perception is inherently ambiguous

Many scenes could have been created a given 2D image

- People figure out the "most likely" one based on experience, intuition, convention, ... ?



Artwork from
Julian Beever

Perception is inherently ambiguous

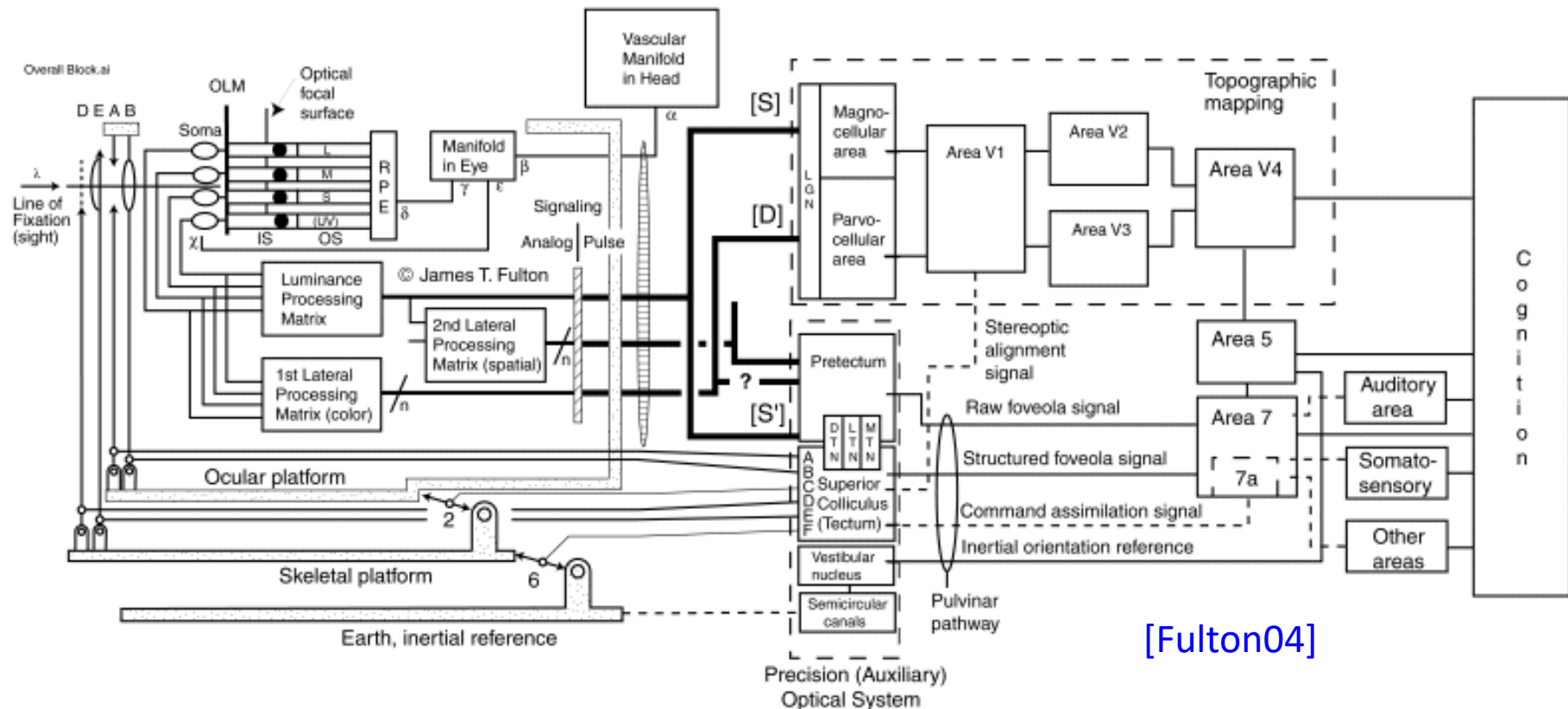


Julian Beever
Artwork

How does human vision work?

How do people (and animals) see?

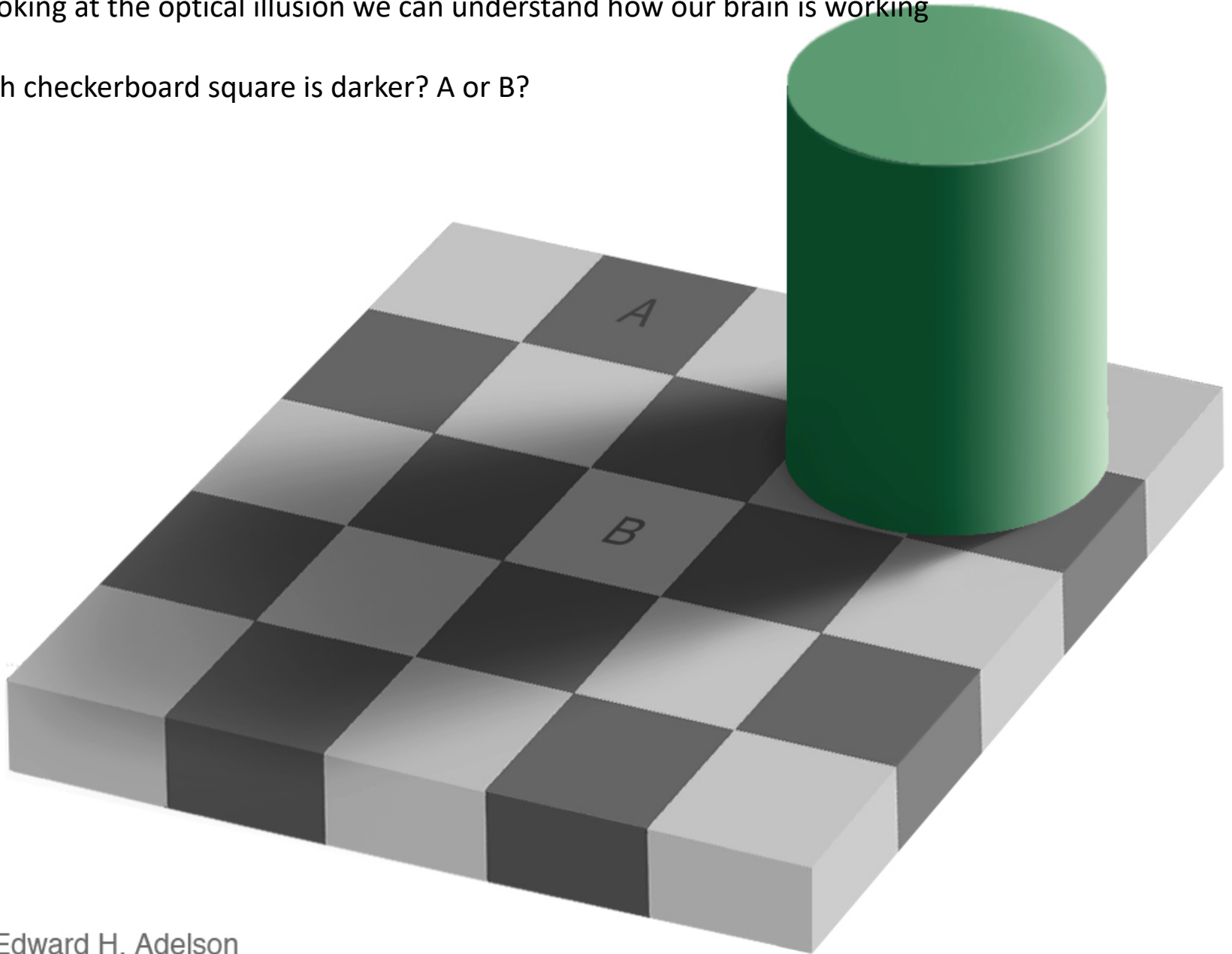
- We don't really know how it works
 - But we use 75% of our brain for visual processing

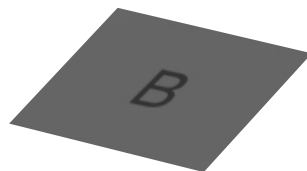
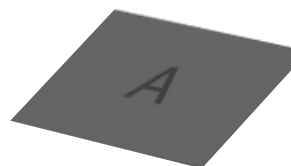


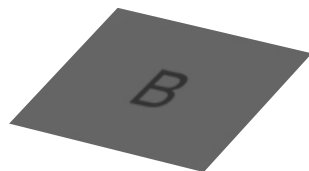
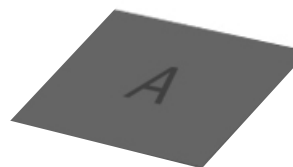
How hard our visual system is working?

By looking at the optical illusion we can understand how our brain is working

Which checkerboard square is darker? A or B?







How does human vision work?

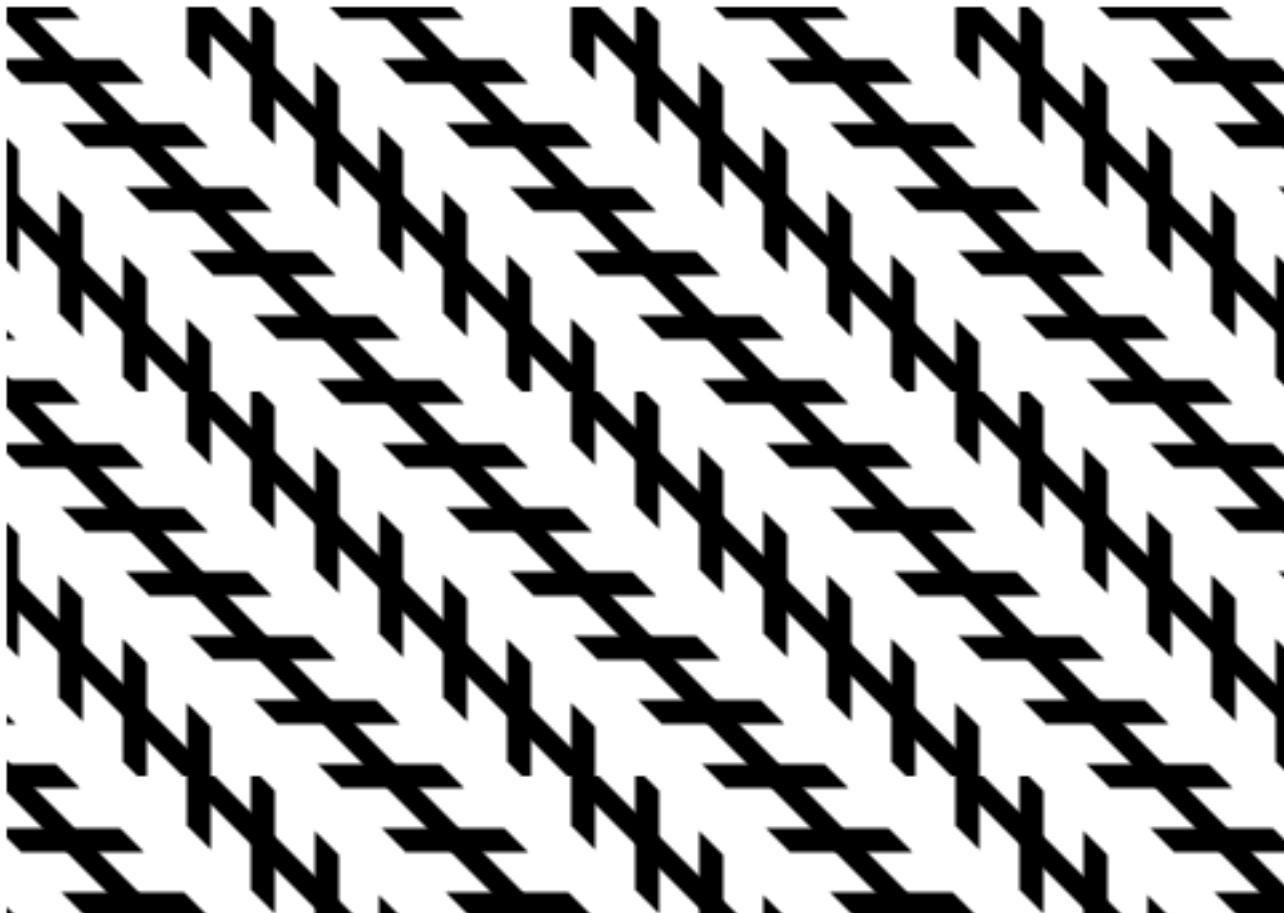


How does human vision work?



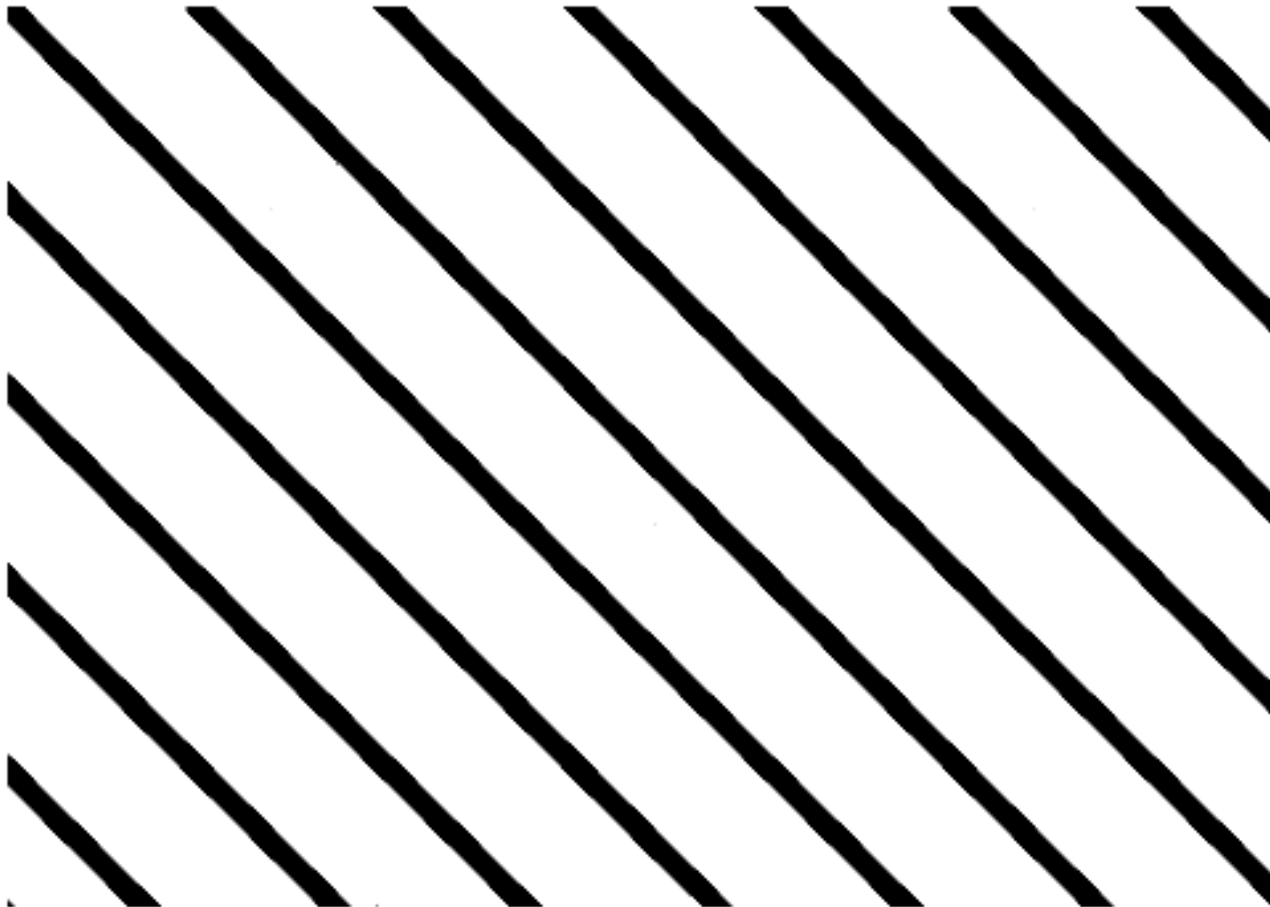
Zollner illusion

- Are these lines parallel?



Zollner illusion

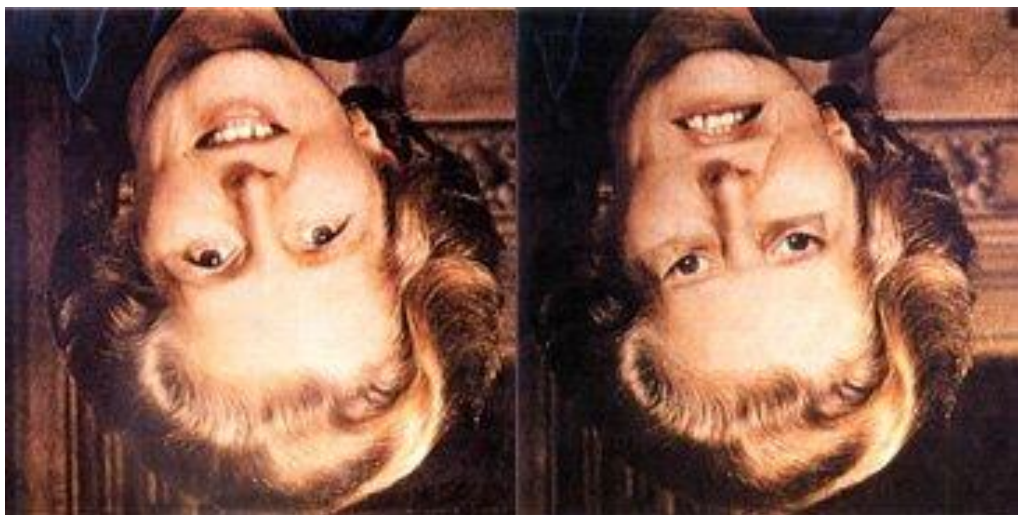
- After removing the hatches on these lines, they look parallel



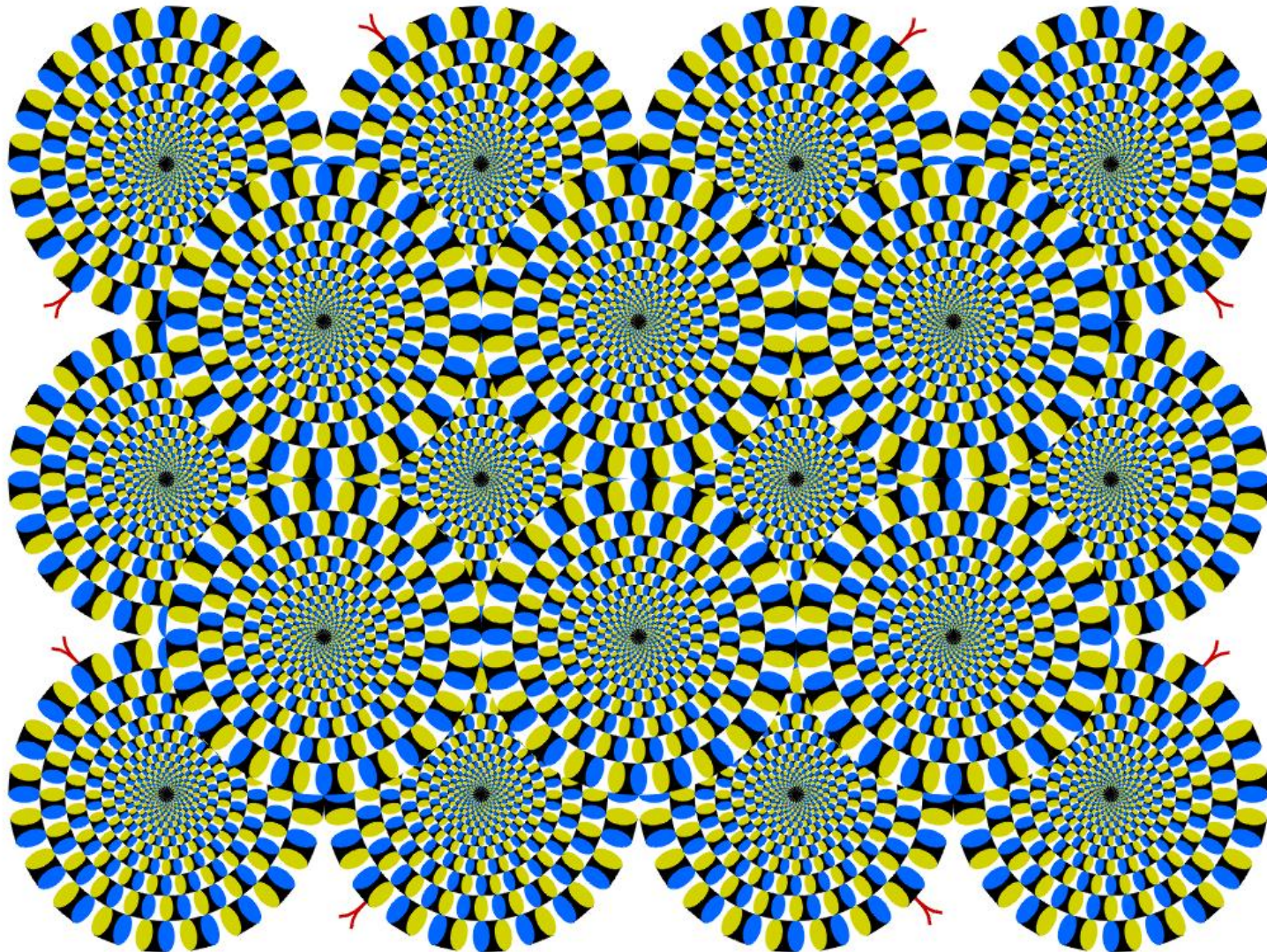
The Thatcher effect



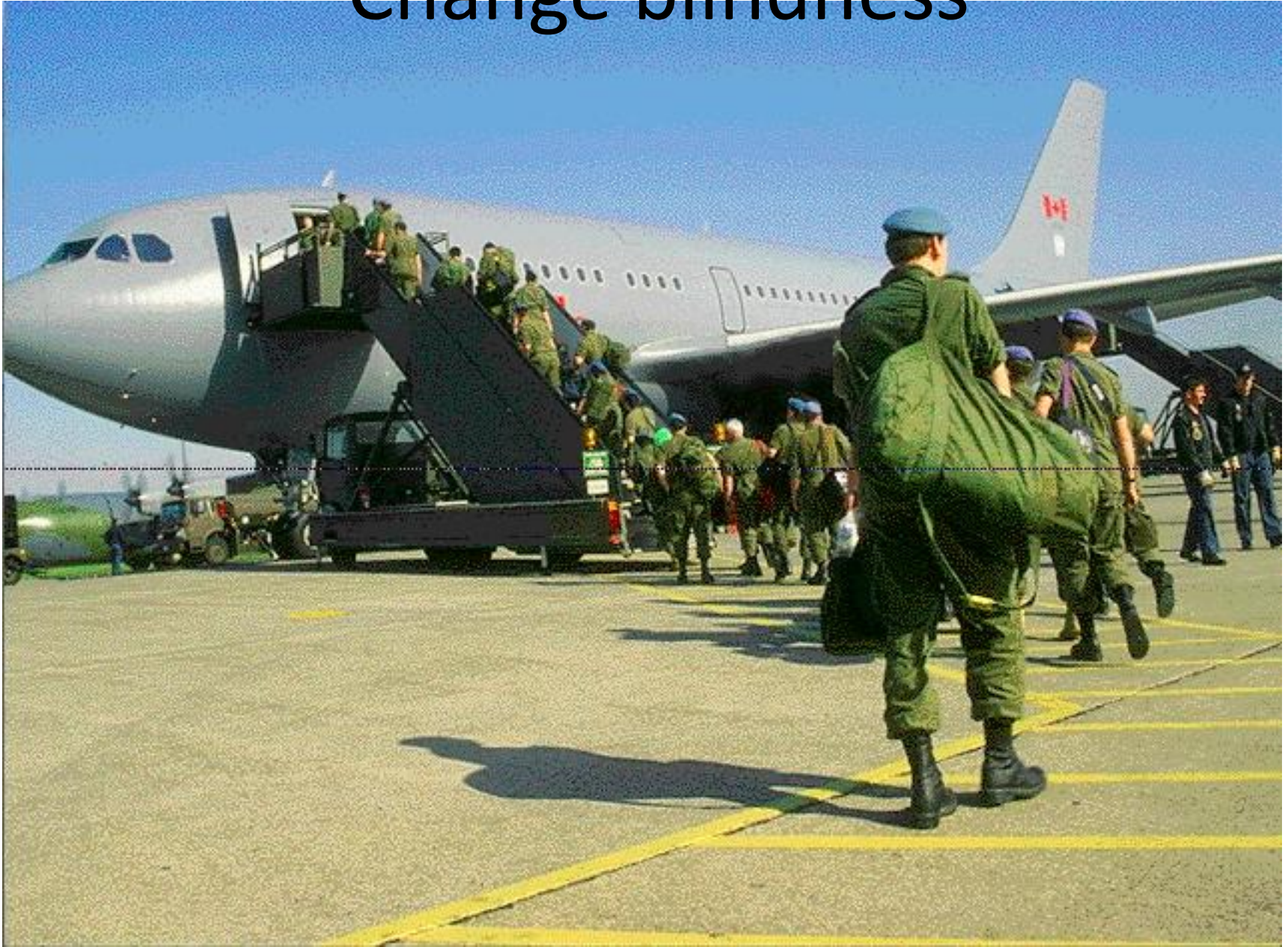
[Thompson 1980]



Rotation illusion



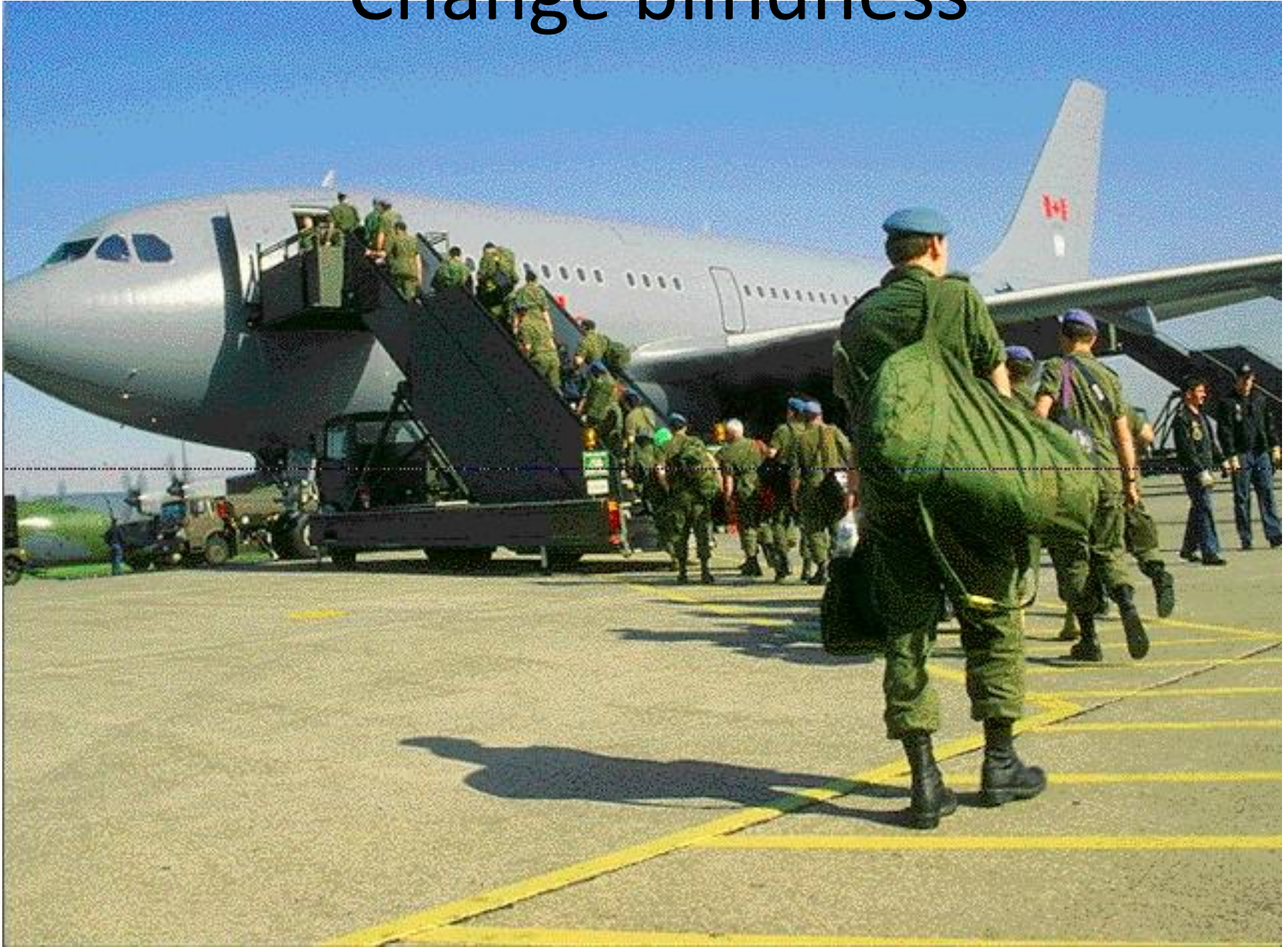
Change blindness

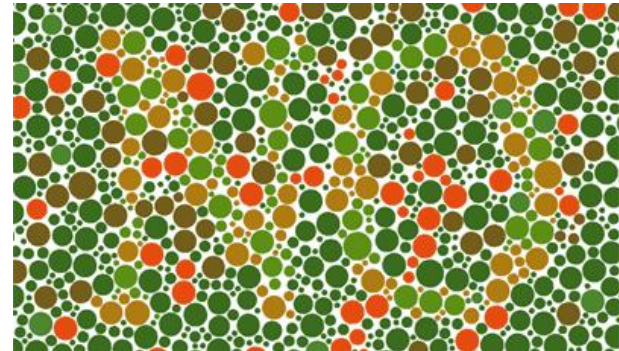
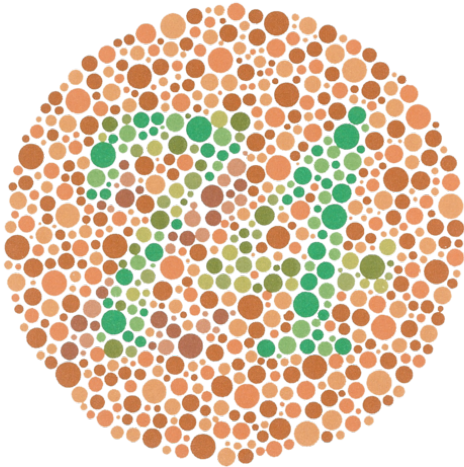
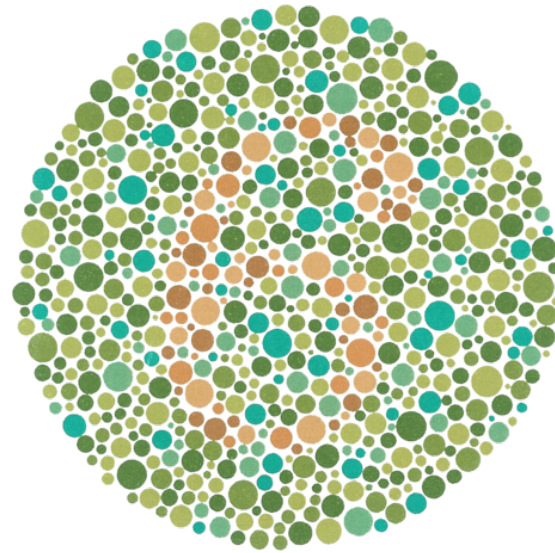
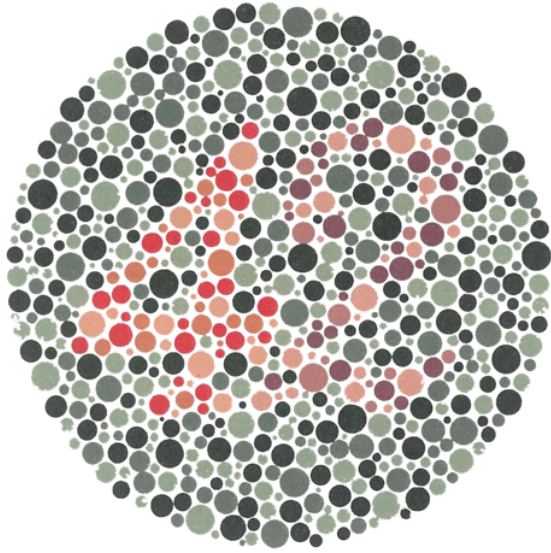


Change blindness



Change blindness





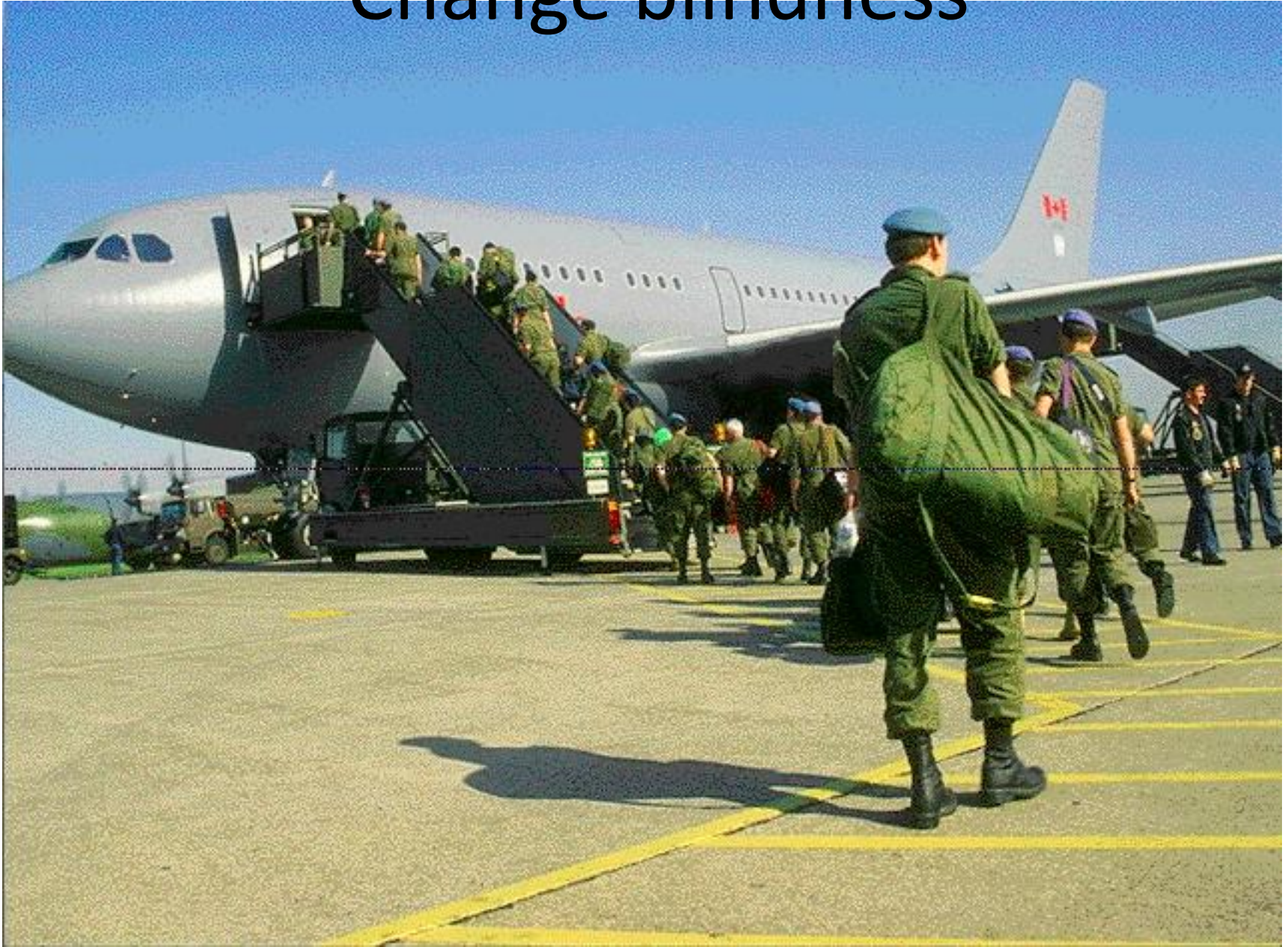
We see the world differently. Even if figure out how to replicate the human vision system the next question would be — Whose vision system do we replicate?

This is yet another example where we experience the world differently

Change blindness



Change blindness



Course overview

- Bias towards algorithms and mathematical models that are widely applicable (more CS than engineering)
 - Many of these are widely applicable outside of computer vision
- Several broad topics:
 1. Recognition techniques
 - multi-label image classification, object detection, and semantic segmentation using different datasets and state-of-the-art models
 2. Reorganization techniques
 - clustering based, graph- based, and vision foundation model-based segmentation
 3. Reproduction techniques
 - content generation with generative modeling such as Generative Adversarial Network (GAN), Variational Autoencoder (VAE), and Diffusion Models
 4. Reconstruction techniques
 - 3D model reconstruction from images

1. Recognition

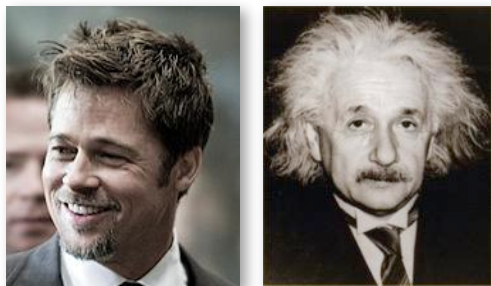
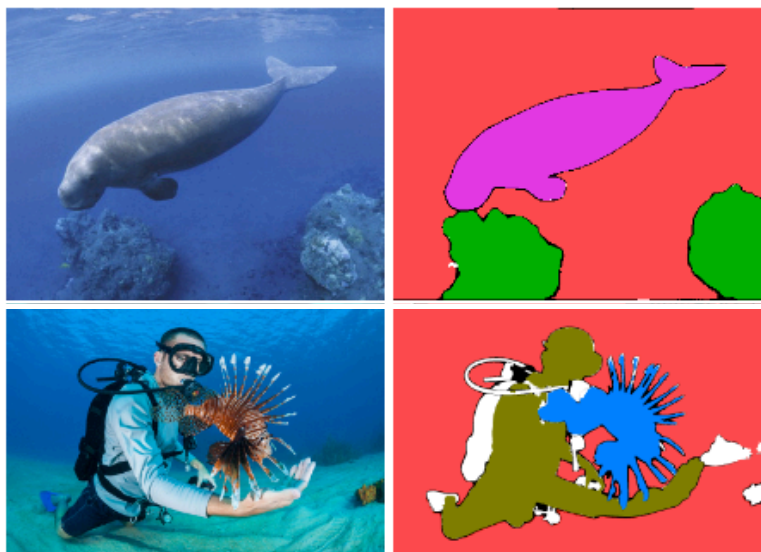


Image classification

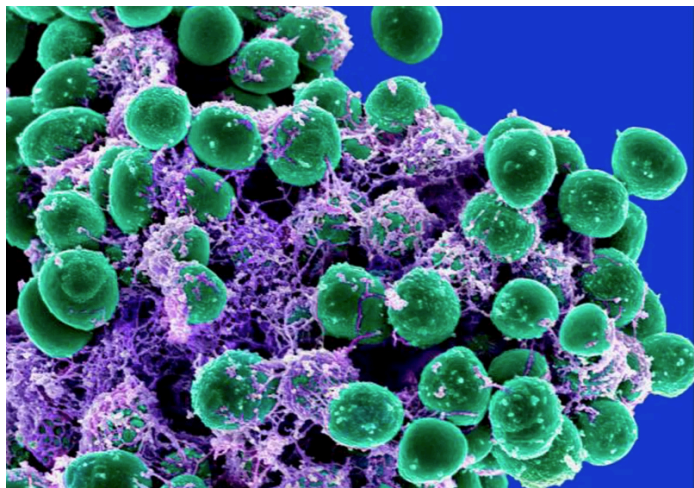
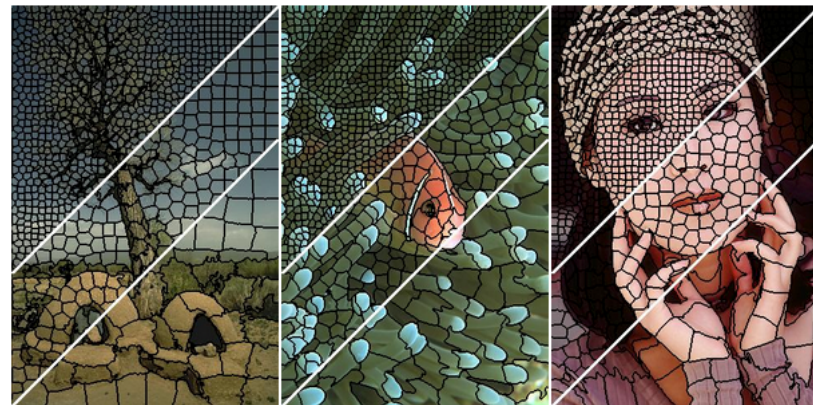
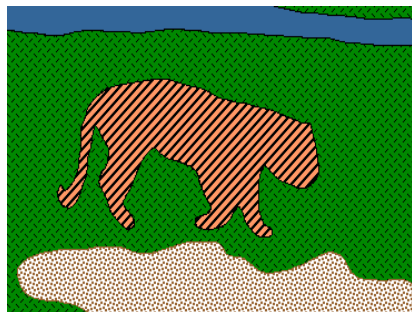


Object Detection

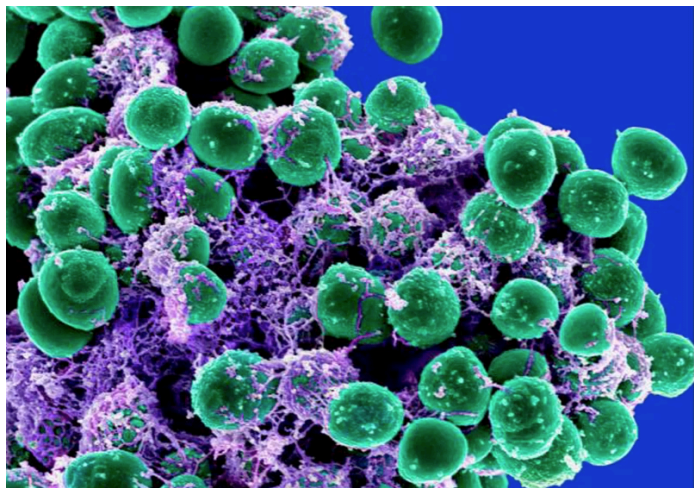
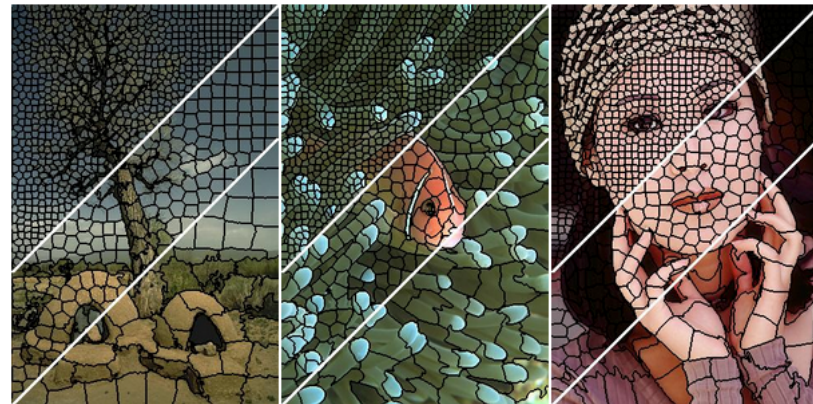
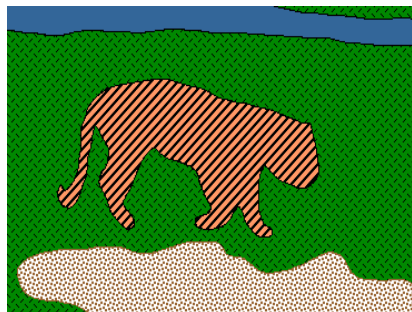


Semantic segmentation

2. Reorganization



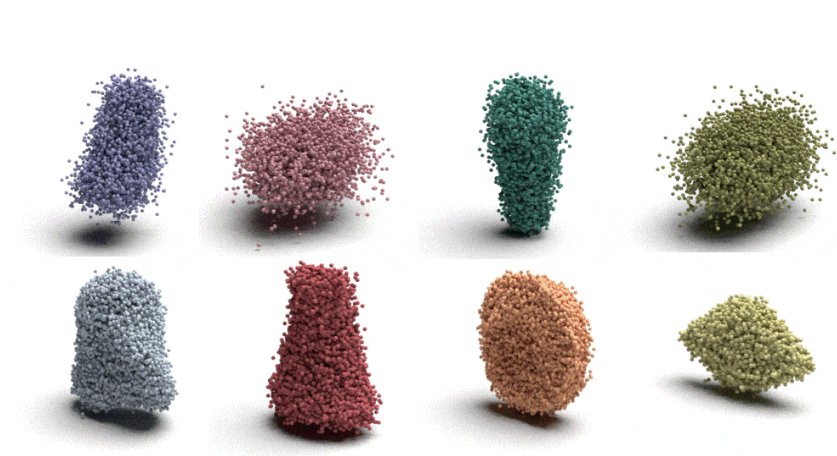
2. Reorganization



3. Reproduction (new content generation)

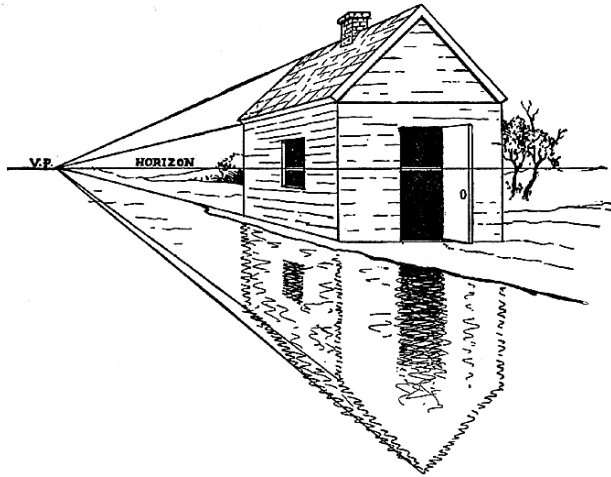


[Content generation \(Neural Style Transfer\)](#)

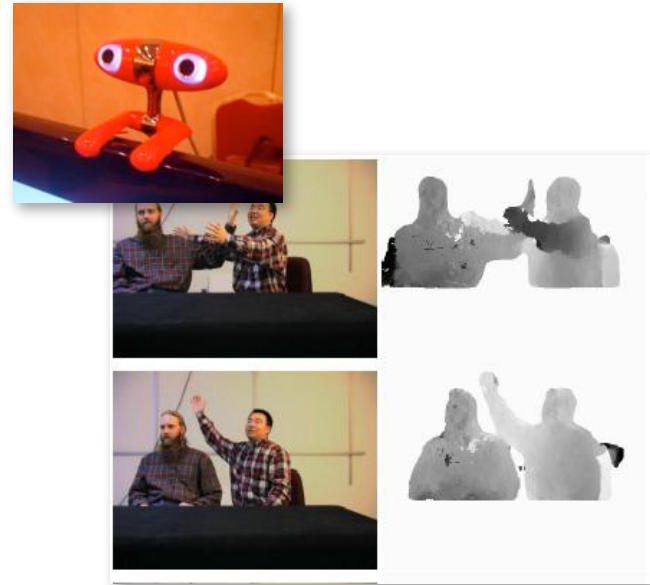


[Learning Gradient Fields for Shape Generation - ECCV'20](#)

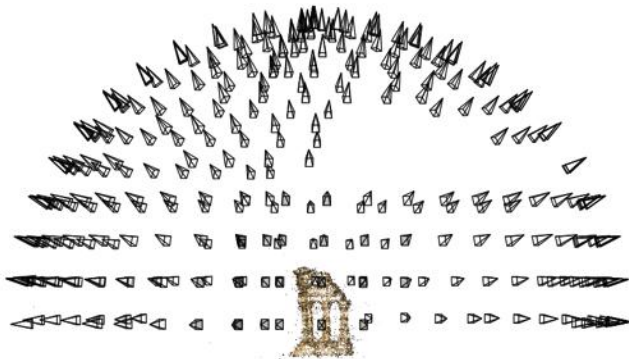
4. Reconstruction (3D modeling)



Projective geometry



Stereo



Multi-view stereo



Structure from motion

Course overview

- Bias towards algorithms and mathematical models that are widely applicable (more CS than engineering)
 - Many of these are widely applicable outside of computer vision
- Some of the techniques we'll encounter:
 - Deep learning e.g., neural networks (CNN, Vision Transformer)
 - Deep neural network features (CLIP, Dino-v2)
 - Probabilistic learning and inference e.g., maximum likelihood (ML) estimate
 - Probabilistic models e.g., Variational Autoencoder (VAE), diffusion model
 - Projective geometry, homogeneous coordinates

Prerequisites

- Practically:
 - CS167 (ML) or CS143 (AI) or equivalent is recommended but not necessary
 - Proficiency in a programming language, preferably Python
 - Willingness to learn some programming and/or math on your own if necessary
 - Some level of mathematical maturity, esp. with linear algebra and statistics

Conclusion: why is computer vision so difficult?

Bad news:

- Computers lack higher-level prior knowledge
- Perception is inherently ambiguous
- We don't know how the human brain works
- Haven't found mathematical models that represent human vision well
- The models we do have require intensive (usually intractable) computation

Good news:

- So much progress is being made! Especially in applications where perfect performance isn't needed.

Course mechanics

- Syllabus, schedule, assignments, announcements, etc. on Drake Blackboard
 - <https://drake.blackboard.com>
 - [Syllabus link](#)
- Class meeting times and locations:

Meeting time: Tue/Thur (3:30 pm - 4:45 pm), Class room: Collier-Scripps # 301

Course mechanics

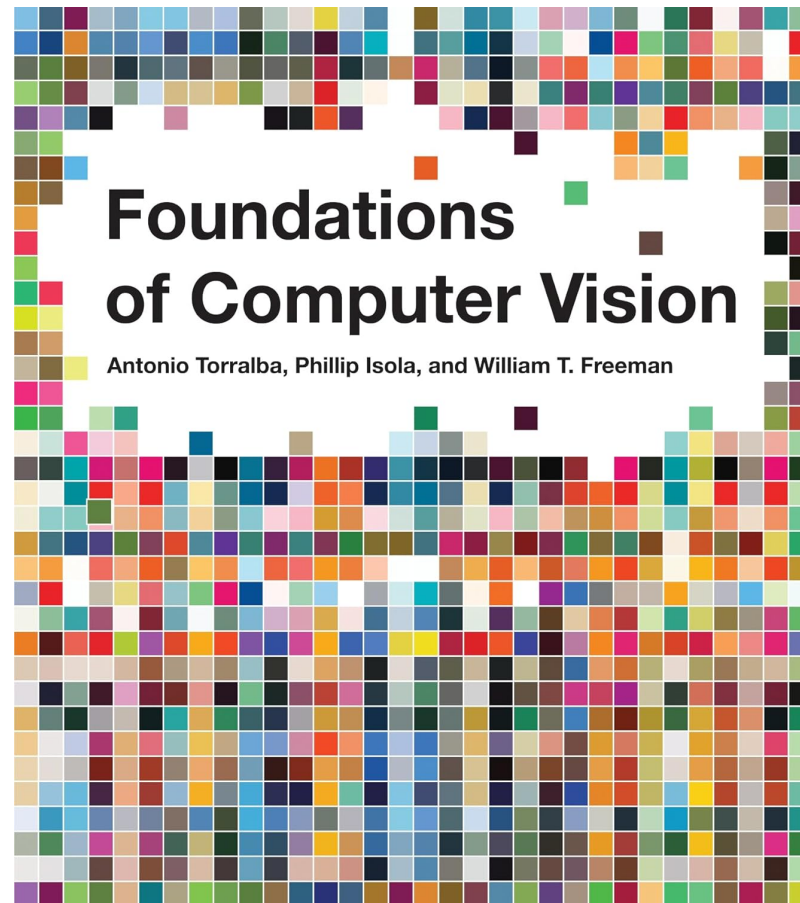
- Office hour times and location:

Office hours (Collier-Scripps#323): Tue/Thu (12:00pm-2:30pm) or by appointment

Additionally by appointment

[Zoom link](#)

Course mechanics



Recommended textbook has a free PDF version

<https://visionbook.mit.edu>

Course mechanics

- Another good reference textbook has a free PDF version

Computer Vision: Algorithms and Applications, 2nd ed.

© 2022 [Richard Szeliski](#), The University of Washington



Welcome to the website (<https://szeliski.org/Book>) for the second edition of my computer vision textbook, which is now available for purchase at [Amazon](#), [Springer](#), and other booksellers.

To [download](#) an electronic version of the book, please fill in your information on [this page](#). You are welcome to download the PDF website for personal use, but **not** to repost it on any other website; please post a link to [this URL](#) instead.

Note that while the content of this electronic version and the hardcopy versions are the same, the page layout is different, since the electronic version is optimized for online reading. The PDF should be enabled for commenting in your viewer. Also, hyper-links to sections, equations, and references are enabled. To get back to where you were, use the Previous View (Alt-Left-Arrow) command in Acrobat.

The current download count is [190](#) (since 1/23/2022).

This book is largely based on the computer vision courses that I have co-taught at the University of Washington ([2020](#), [2008](#), [2005](#), [2001](#)) with [Steve Seitz](#) and Harpreet Sawhney and at Stanford (2003) with [David Fleet](#).

Grading and Requirement

- Notebook assignments (30%): *take home assignments (submit on Blackboard)*.

Notebooks Assignments (30%) We will explore various computer vision techniques using Python programming language. You'll regularly submit notebook files (.ipynb) to showcase your skills. Since I won't typically run your code due to long computation times, please ensure your results are saved in the notebook. You can expect to submit 4-5 assignments throughout the course.

Grading and Requirement

- **In-class activities (20%):** *10-12 simple coding activities or paper-based tasks (submit on Blackboard).*

In-class Activities (20%) You will also submit notebook files with simple coding or paper-based activities. These are easier, as they follow the lecture and are usually done during or right after class. Since I won't run your code, be sure to save the results in the notebook. You can expect to submit about **10-12 in-class activities** throughout the course.

Grading and Requirement

- **Quizzes (30%):** *3 quizzes, 10% each. Mostly coming from lecture slides to assess conceptual understanding.*

Quizzes (30%) 3 timed quizzes, administered via Blackboard, will be given during the semester. Each will be conducted within a several-hour window, during which students are expected to complete the quiz individually. Students may consult external resources, such as class notes and internet materials, as these quizzes are open-book and open-notes; course materials may therefore be used throughout. **The use of AI, however, is strictly prohibited.** Any sources consulted during completion of the quizzes, apart from class notes, must be properly cited. Content will be drawn primarily from the lecture slides, with an emphasis on assessing conceptual understanding.

Grading and Requirement

- Final Project (15%): *preferably a group project*

Final project (15%): 1 final project will consist of research or implementation on a topic of your choice (in consultation with the instructor). Projects may be done individually or in groups of up to 2 students. There will be three deliverables:

- *Project proposal:* a brief (1-2 pages) proposal, due approximately 11th week into the semester.
- *Project report:* a summary report along with an electronic copy of the source code developed during the project.
- *Project presentation:* a class presentation. students are encouraged but not required to submit their work to a regional conference (eg, CCSC) or workshop.

Final project

- On a topic of your choice
- Individually or in small groups (up to 2 students)
- Three deliverables: a brief proposal, a final presentation, and a final report (and source code)
- Wide range of possible projects, e.g.
 - Apply existing technique to new application X
 - Implement and compare techniques U and V
 - Something else broadly related to vision
 - Or develop new technique for problem Y (not necessary but you are welcome)

Grading and Requirement

- **Attendance/Participation (05%):** *Participation in polls/papers, not based on correctness*

Attendance/Participation (05%): This class is highly interactive, meaning that active participation is both expected and the norm. You will receive credit for your participation, and it will be counted towards your final grade. I will keep track of your involvement using a signature sheet. Throughout the course, I will pose questions using polling software and conduct in-class Q&A sessions to better understand how the class is grasping the content. These responses will not be evaluated for correctness but rather for completion.

Summary: Grading and Requirement

- Notebook assignments (30%): *take home assignments (submit on CodePost)*
- In-class activities (20%): *simple coding activities or paper-based tasks (submit on Blackboard)*
- Quizzes (30%): *3 quizzes, 10% each*
- Final Project (15%): *preferably group project*
- Attendance/Participation (05%): *Participation in polls, not based on correctness, physical attendance during lecture time*

Grading Scale

- The tentative grading scale for this course would be as follows:

A (93%-100%)	A- (90%-92.9%)	B+ (87%-89.9%)
B (84%-86.9%)	B- (80%-83.9%)	C+ (77%-79.9%)
C (74%-76.9%)	C- (70%-73.9%)	D (60%-69.9%)
F (0%-59.9%)		