CS195: Computer Vision

January 25, 2022

Md Alimoor Reza
Assistant Professor of Computer Science
Drake University

Road Map

- Brief introduction
- Course logistics
- Topics
  - What is computer vision?
  - What makes vision hard?
  - How does human vision work?

Date	Main Topic	Subtopics
week 1 (Tue: 01/25)	Introduction to Computer Vision Lecture 1 slide	Brief introduction Course logistics What is computer vision? What makes vision hard? How does human vision work?

Introduction

- Md Alimoor Reza
  - Assistant Professor of Computer Science, Dept. of Mathematics and Computer Science, Drake University
  - **Office:** 314 Collier-Scripps
  - **Email:** md.reza@drake.edu
  - **Phone:** 515-271-1972
  - **Office hours:** T/W/R: 12:30-1:30pm CDT
    additionally by appointment

Zoom link (for the first 2 weeks)

http://homes.sice.indiana.edu/mdreza/

Introduction

- Prior teaching experience

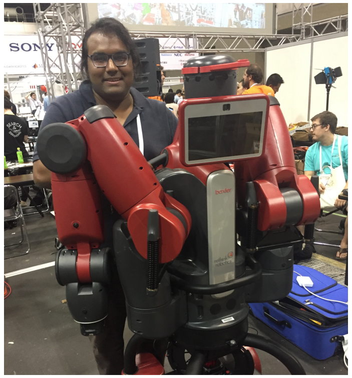
  - Indiana University Bloomington
    - Computer Vision (Spring’21)
    - Artificial Intelligence (Fall’18)
  - George Mason University
    - Introduction to Computing ,
    - Object Oriented Programming,
    - Data Mining
  - Drake University
    - Introduction to Computer Science (Fall’21)

Introduction

- Research interests:
  - Passionate about Artificial Intelligence (A.I.) for robots
  - Specific research interest in the intersection of
    - Computer Vision, Machine Learning, and Robotics
  - Studying images and finding meaningful concepts from them

Object Detection

Semantic Segmentation



Introduction

- Additional help
  - Good news: It is a new course at Drake University
  - Not so great news: CS tutors are not super-familiar with the topics
    - Hence, please feel free to stop by my office hours for help

- Now your turn, briefly introduce yourselves!
  - Your name
  - Major
  - Hobby/interest
  - Why you are interested in this course?

Course mechanics

- Syllabus, labs, assignments, quizzes, announcements, etc. on Drake Blackboard
  - http://blackboard.drake.edu/

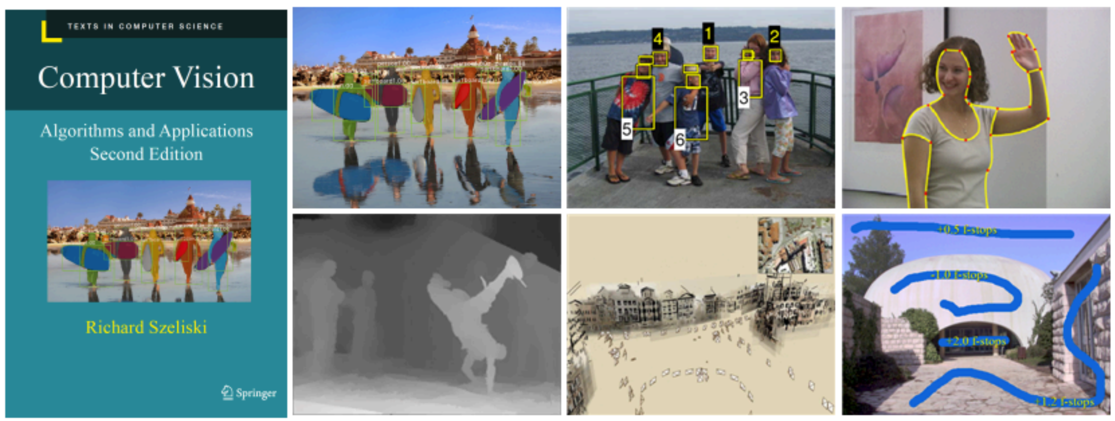
- Readings from papers and online textbooks

Course mechanics

- Recommended 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.

Course overview (tentative)

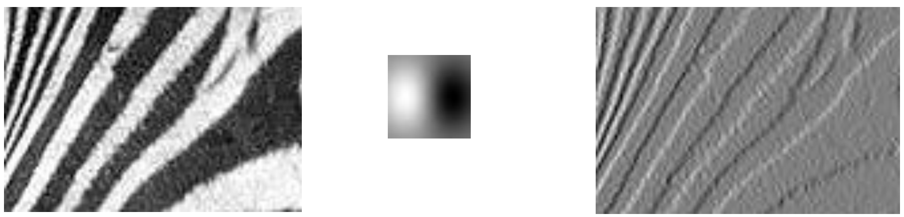
- Bias towards algorithms and mathematical models that are widely applicable (more CS than engineering)
  - Many of these are widely applicable outside of computer vision

- Three broad topics:
  1. Low-level vision
  2. Geometry
  3. Recognition

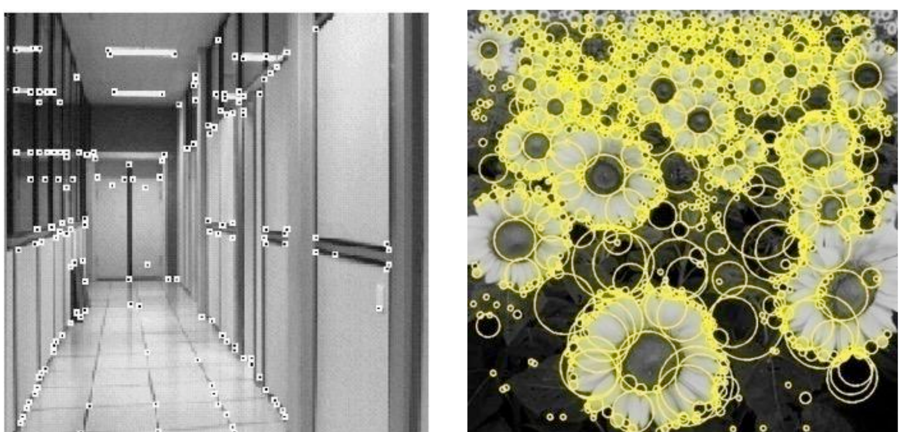
- Three broad technical approaches:
  1. Expert-designed features and reasoning
  2. Statistical graphical models
  3. Machine Learning (especially Deep Learning)

1. Low-level vision

- Basic image processing and image formation



Filtering, edge detection



Feature extraction

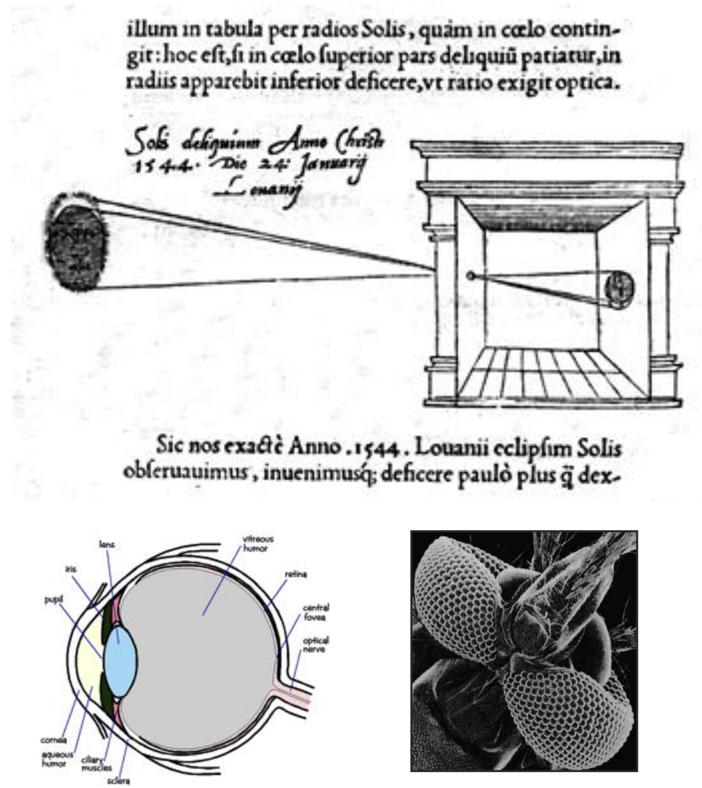
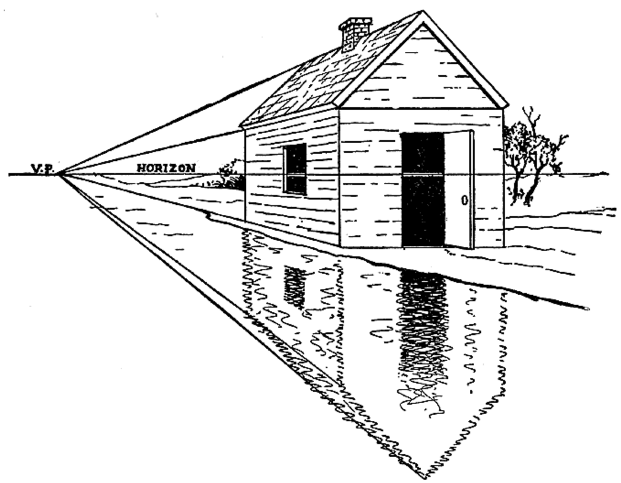
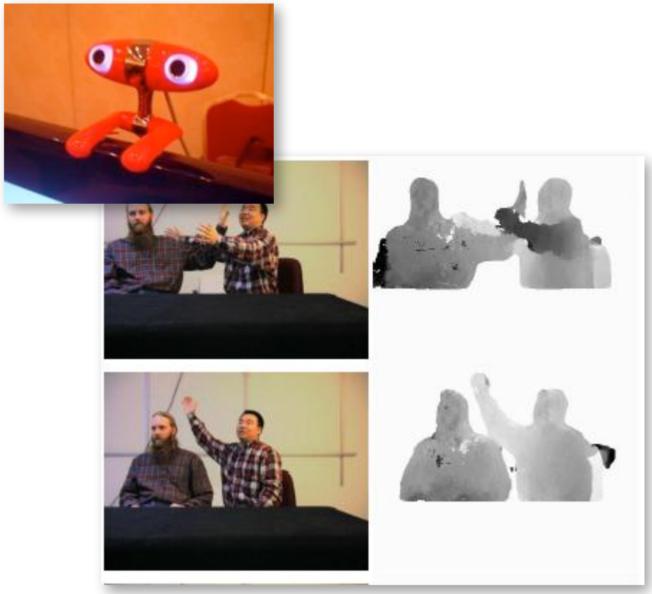


Image formation

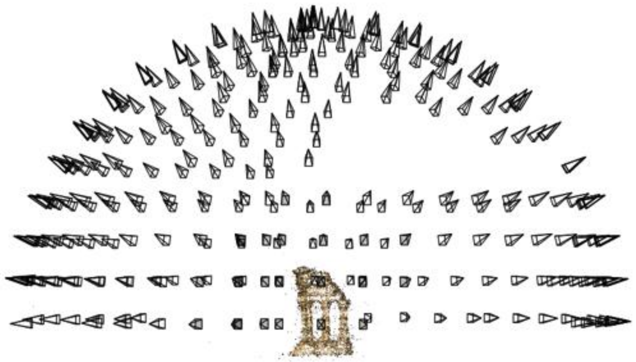
2. Geometry



Projective geometry



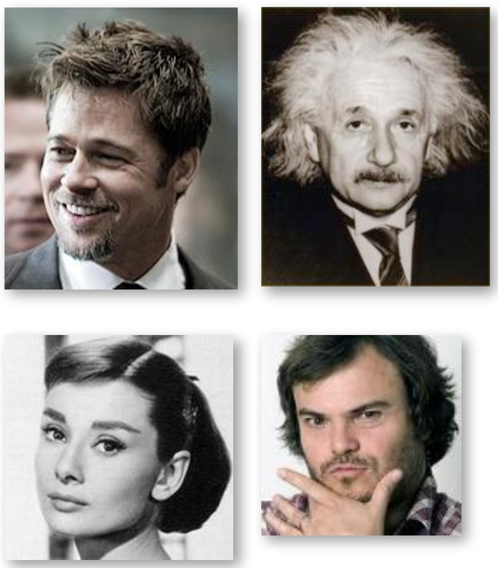
Stereo



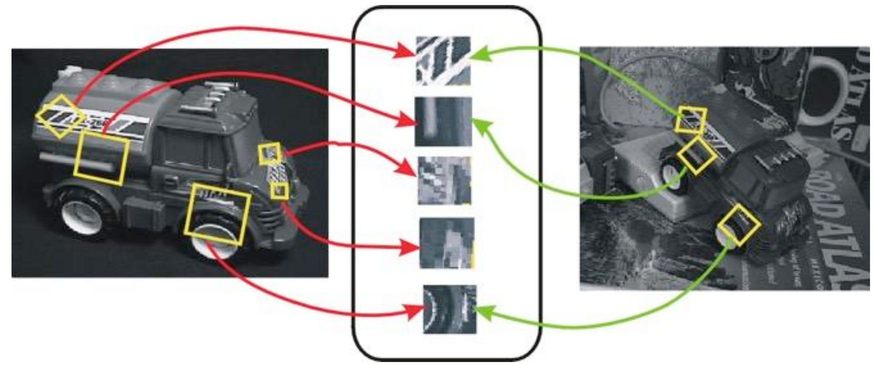
Multi-view stereo

Structure from motion

3. Recognition



Face detection and recognition



Single instance recognition

Category recognition

Course overview (tentative)

- 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:
  - Singular value decomposition, principal component analysis
  - Projective geometry, homogeneous coordinates
  - Statistical graphical models (MRFs) and inference algorithms
  - Statistical learning (e.g. neural networks, deep learning)
  - Dynamic programming, linear programming

Prerequisites

- Practically:
  - Proficiency in a general-purpose programming language, preferably Python or Matlab
  - Some level of mathematical maturity, esp. with linear algebra and statistics
  - Willingness to learn some programming and/or math on your own if necessary

Factoring the projection matrix

- It's often useful to factor the projection matrix into a series of operations

$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$

identity matrix

$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{T}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{translation}}$$

intrinsics projection rotation translation

Energy minimization view

- More generally, we want to minimize:

$$E(Q) = \sum_{v_i \in V} D(O_i, Q_i) + \sum_{(v_i, v_j) \in E} V(Q_i, Q_j)$$

  - Corresponding BP update equations:

$$m^{(t+1)}_{j \to j+1}(q_{j+1}) = \min_{q_j} \Big(D(O_j, q_j) + V(q_j, q_{j+1}) + m^{(t)}_{j-1 \to j}(q_j) \Big)$$

$$m^{(t+1)}_{j \to j-1}(q_{j-1}) = \min_{q_j} \Big(D(O_j, q_j) + V(q_j, q_{j-1}) + m^{(t)}_{j+1 \to j}(q_j) \Big)$$

  - After T iterations, each node computes its MAP state:

$$q^*_j = \arg\min_{q_j} \Big(D(O_j, q_j) + m^{(T)}_{j-1 \to j}(q_j) + m^{(T)}_{j+1 \to j}(q_j) \Big)$$

Why study computer vision?

- Compelling applications
- Difficult computational problems
  - Essentially all non-trivial inference with today's computer vision models is NP-hard
- Draws on many different areas
  - Machine learning, image processing, graph theory, optics, geometry, statistics, linear algebra, algorithms, optimization, computer graphics…
- An exciting time for the field

Grading

- 30% Labs (~6)
  - Implementation activities regarding specific concepts.
  - No late work accepted, but drop the lowest
- 20% Programming assignments (~2)
  - In small groups
  - Any general-purpose programming language will be acceptable as long as you implement image processing and vision routines yourself (Python or Matlab recommended)
- 10% Activities, quizzes, and class participation
  - No late work accepted, but drop the lowest
- 20% Midterm exam (paper based)
- 20% Final project

Grading

- Also discussed in the course syllabus

  - *Online activities and quizzes (10%).*
  - *Labs (30%).* practical implementations of specific concepts taught in class.
  - *Programming assignments (20%).* approximately 2.
  - *Midterm (20%).*
  - *Final project (20%).* see below for details.

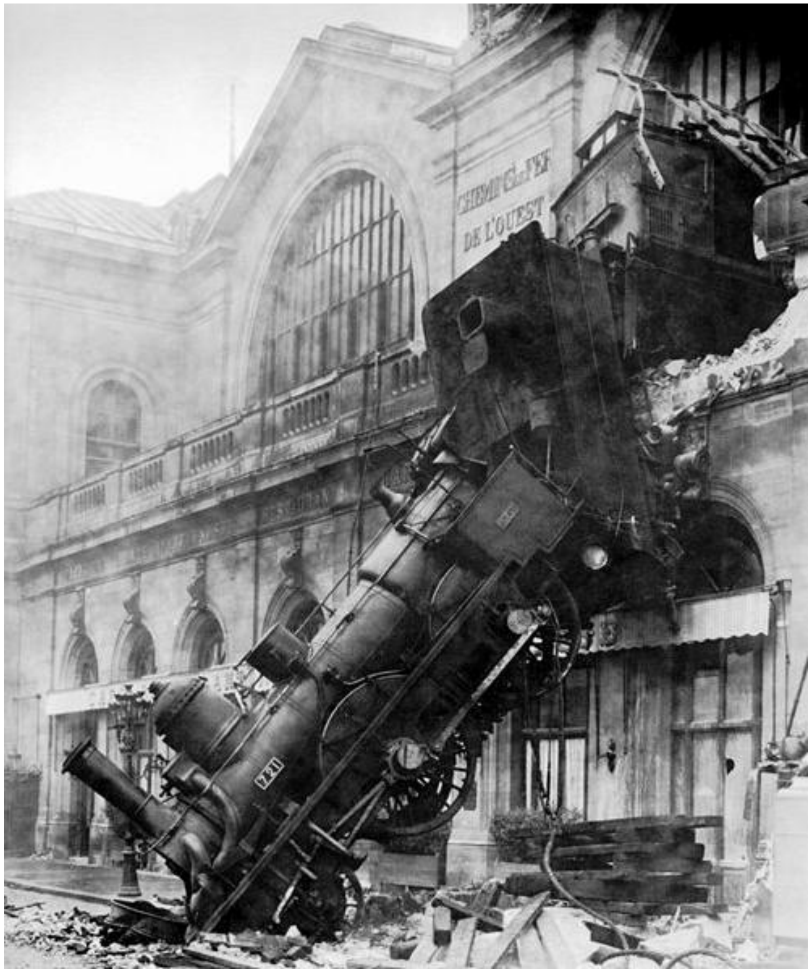
Final project

- On a topic of your choice
- Individually or in small groups (up to 3 students)
- Three deliverables: a brief proposal, a final report (and source code), a video/poster presentation
- Wide range of possible projects, e.g.
  - Develop new technique for problem X
  - Apply existing technique to new application Y
  - Implement technique Z in a significantly faster way
  - Implement and compare techniques W and U
  - Or something else broadly related to vision

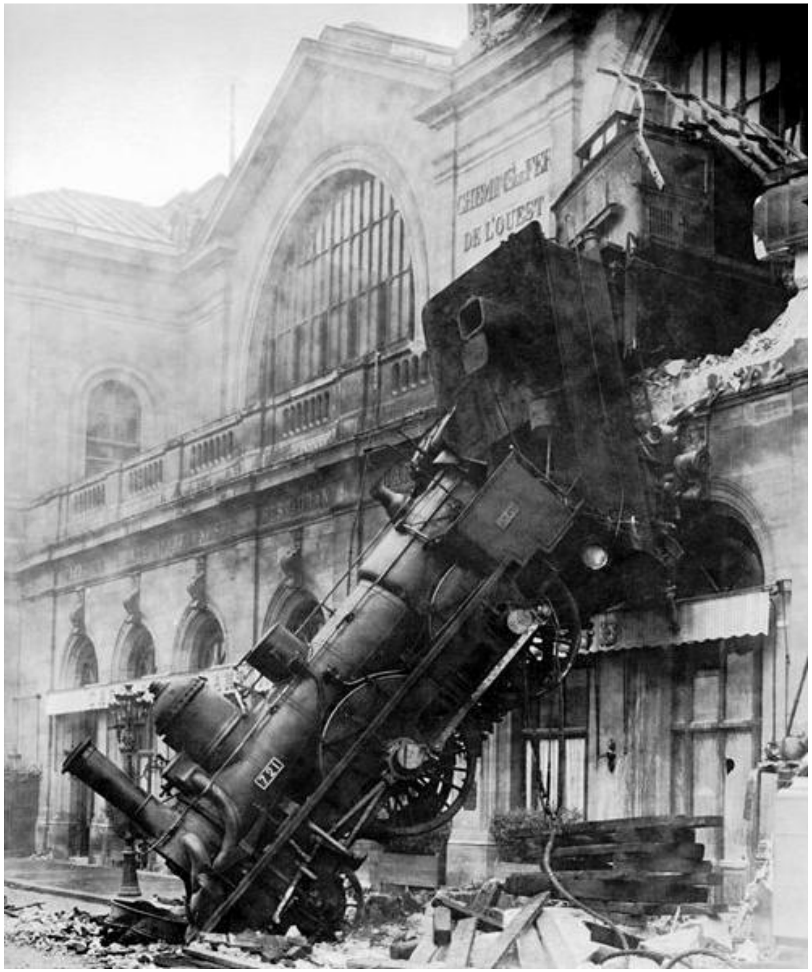
Academic integrity

- Read and understand the academic integrity policy on syllabus
- We will look for academic integrity violations
- Be especially careful with homework assignments
  - You may discuss homework problems at a high level (e.g. general strategies for solving problems), but you may not share code, and you must cite the other student in your submission
  - If you use ideas or code from another source (like a webpage or textbook) you **must** acknowledge the source in your submission

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


DE L'OUEST


721

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

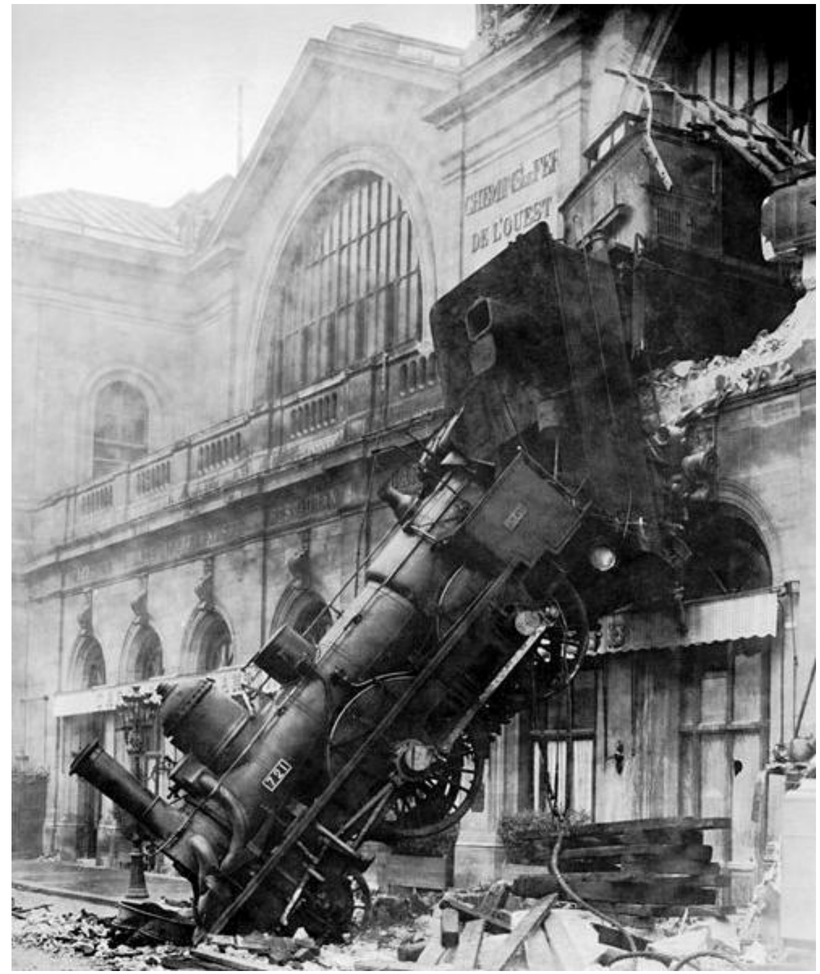


Image processing

- Take an image, produce another image

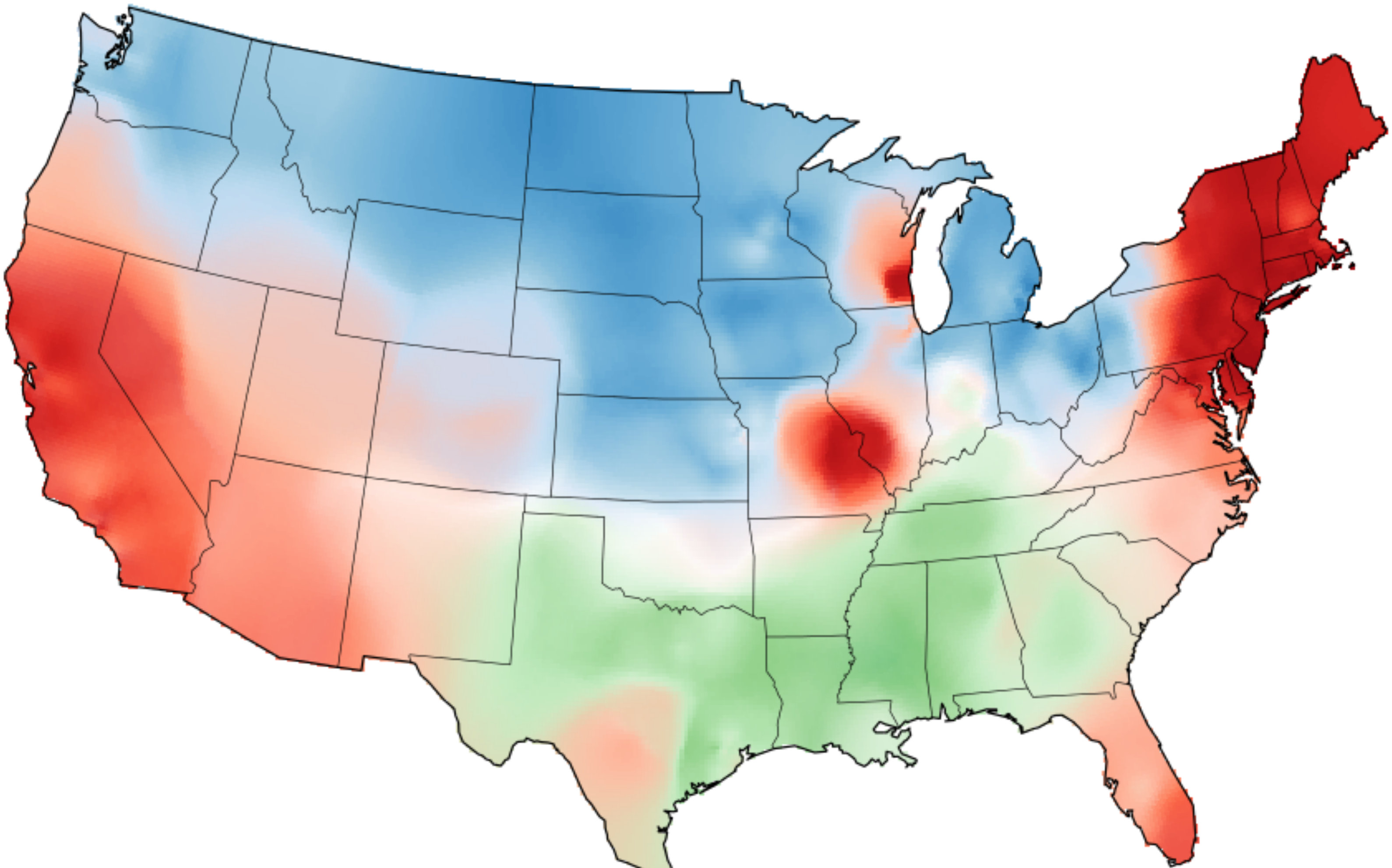
Computer graphics

- Take a description, produce an image

From “Final Fantasy”

Visualization

- Take data, produce an image



Joshua Katz, Department of Statistics, NC State University

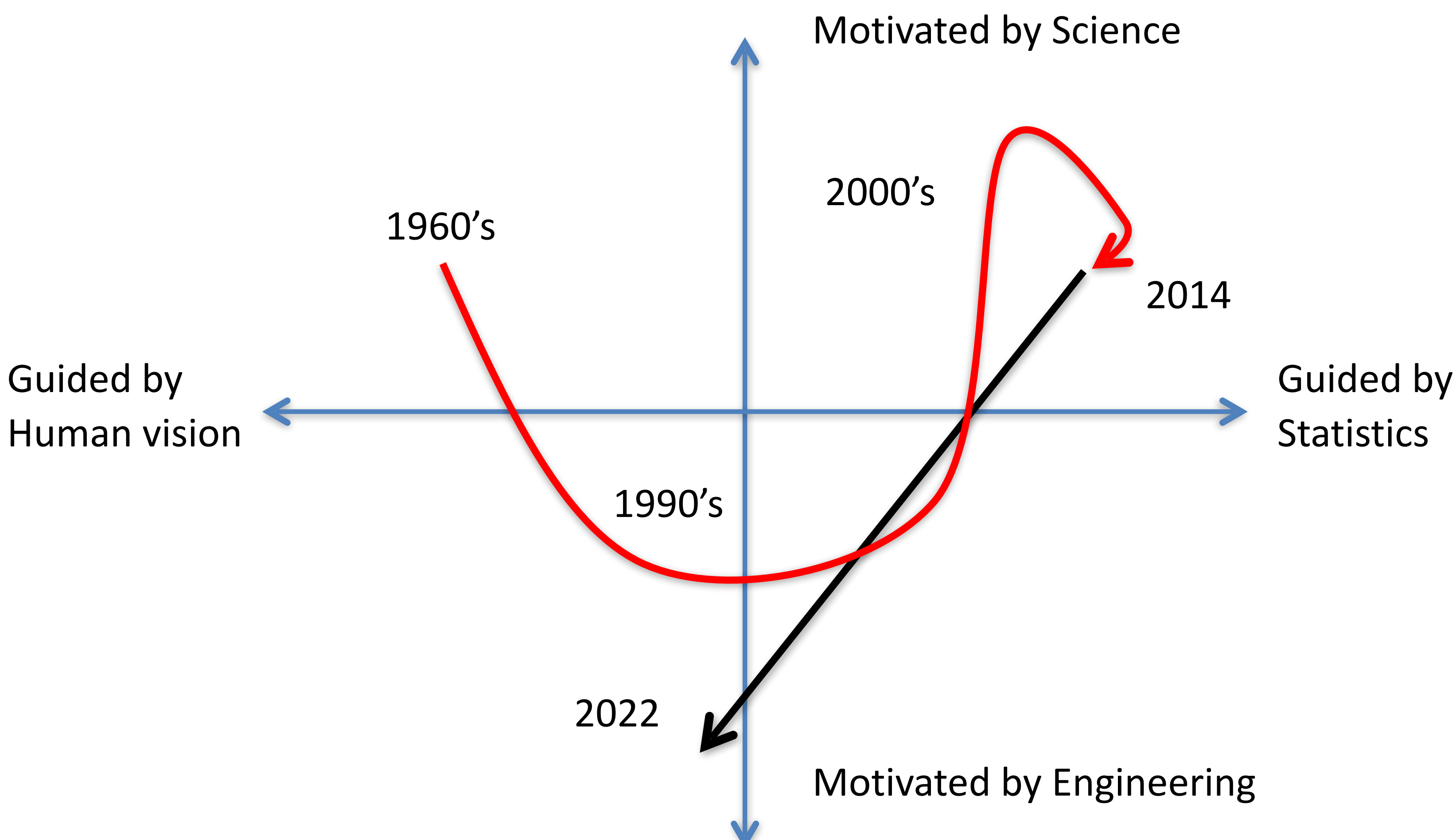
Computer vision

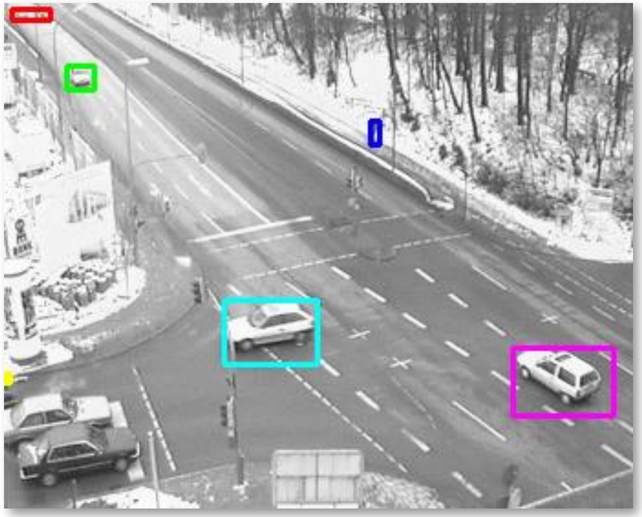
- Takes an image, produces a description

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

Vision: an unusually diverse field

- Computer vision researchers have dramatically different backgrounds and approaches





flickr from YAHOO!

YouTube