# Computer Science Research

CS 197 | Stanford University | Arpita Singhal
cs197.stanford.edu

# What is computer science research?

# *Seeking a Better Way to Find Web Images*

By JOHN MARKOFF NOV. 19, 2012

STANFORD, Calif. — You may think you can find almost anything on the Internet.

But even as images and video rapidly come to dominate the Web, search engines can ordinarily find a given image only if the text entered by a searcher matches the text with which it was labeled. And the labels can be unreliable, unhelpful ("fuzzy" instead of "rabbit") or simply nonexistent.

To eliminate those limits, scientists will need to create a new generation of visual search technologies — or else, as the Stanford computer scientist Fei-Fei Li recently put it, the Web will be in danger of "going dark."

Now, along with computer scientists from Princeton, Dr. Li, 36, has bu
the world's largest visual database in an effort to mimic the human vis
system. With more than 14 million labeled objects, from obsidian to
orangutans to ocelots, the database has because a vital resource for
computer vision researchers.

CLOUD COMPUTING

## Making Cloud-Computing Systems More Efficient

BY QUENTIN HARDY MARCH 6, 2014 7:00 AM 3

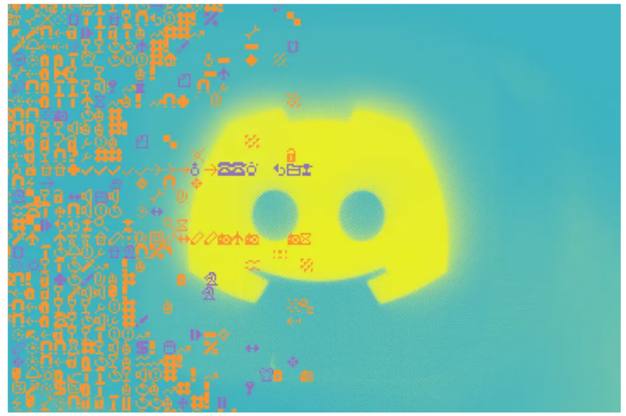
Email
Share
Tweet
Save
More

Christos Kozyrakis, professor of electrical engineering and computer science at Stanford University, headed the creation of management software called Quasar.

Computer scientists at Stanford have developed software that works like the recommendation engines for Amazon shoppers and Netflix movie subscribers, only for the distribution of workloads across large computing environments. The results, they say, can triple the efficiency of cloud-computing systems.

The New York Times

Plus: How A.I. is bringing us closer to "Westworld."



Kirill Kudryavtsev/Agence France-Presse — Getty Images

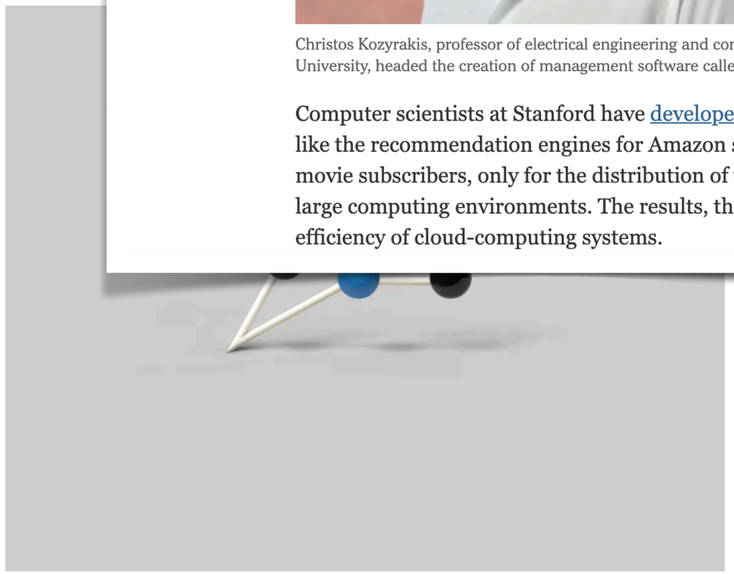
## Stanford Researcher Finds Lots of Leaky Web Sites

BY SOMINI SENGUPTA OCTOBER 11, 2011 6:32 PM 6

The Web is porous. Remarkable information trickles in from everywhere. It also sometimes spills out without its users knowin
exactly where or how.

Take for instance these findings, released on Tuesday by comput
scientists at Stanford University. If you type a wrong password in
the Web site of The Wall Street Journal, it turns out that your e-
mail address quietly slips out to seven unrelated Web sites. Sign
to NBC and, likewise, seven other companies can capture your e-
mail address. Click on an ad on HomeDepot.com and your first
name and user ID are instantly revealed to 13 other companies.

These findings, released by the Center for Internet and Society at
Stanford Law School, are among the leaks found on 185 top Web



Doug Chayka

## *A.I. Is Learning What It Means to Be Alive*

Given troves of data about genes and cells, A.I. models have made some surprising discoveries. What could they

The New York Times

SCIENCETAKE

## *OceanOne, a Mer-Bot Dive Buddy With a 'Friendly Face'*

# What will this course achieve?

# Your experience in CS 197

Work on bleeding-edge topics now, rather than in two years

Fashion a project that you can publish as a work-in-progress or workshop paper

Find an onramp to research in the department, and to research and advanced development in industry

# Today

What is research, vs. industry?

How does this course work?

Research mindset

# Arpita Singhal

**2016-2020:** U.C. Berkeley, Computer Science and Bioengineering, worked with Nir Yosef

**2020-2021:** Software Engineer, Salesforce

**2021-Present:** PhD in Computer Science - AI/ Computational Biology, Anshul Kundaje's lab

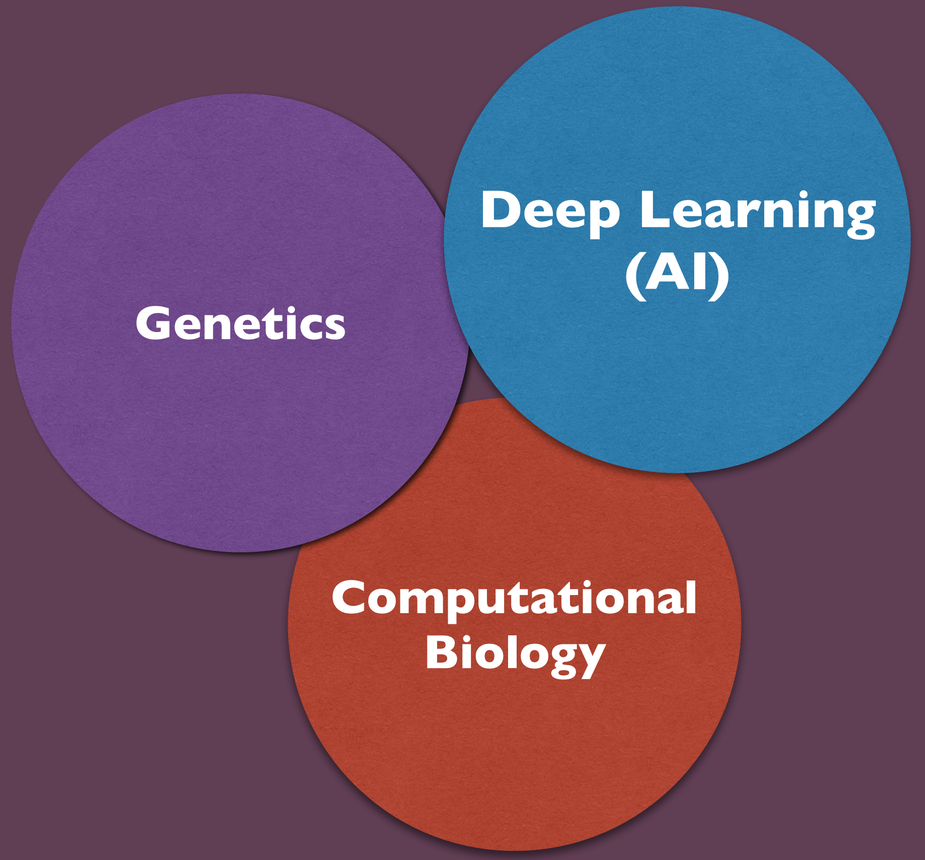
# My research history

**High School** — Science Fairs (intro to research)

**Undergrad** — Computer Science and Bioengineering, Computational Biology Research

**Salesforce** — Software - Hardware to Software Security

**PhD** — Deep Learning models for Computational Biology



# Section

Section is led by a PhD student who (1) is doing research in that area, and (2) has been selected for mentorship skills.

Sections are Tuesdays 4:30pm-5:50pm.

# Section application

We will align each of you into projects based on your preferences

Section application is due tonight at 11:59pm. Fill it out if you haven't already! Linked from cs197.stanford.edu.

- Decisions announced later this week
- Sections start next week

# TA: Yanay Rosen

**2017-2021** BA UC Berkeley,
Major: CS, Minor: Data Science, Nir Yosef's lab

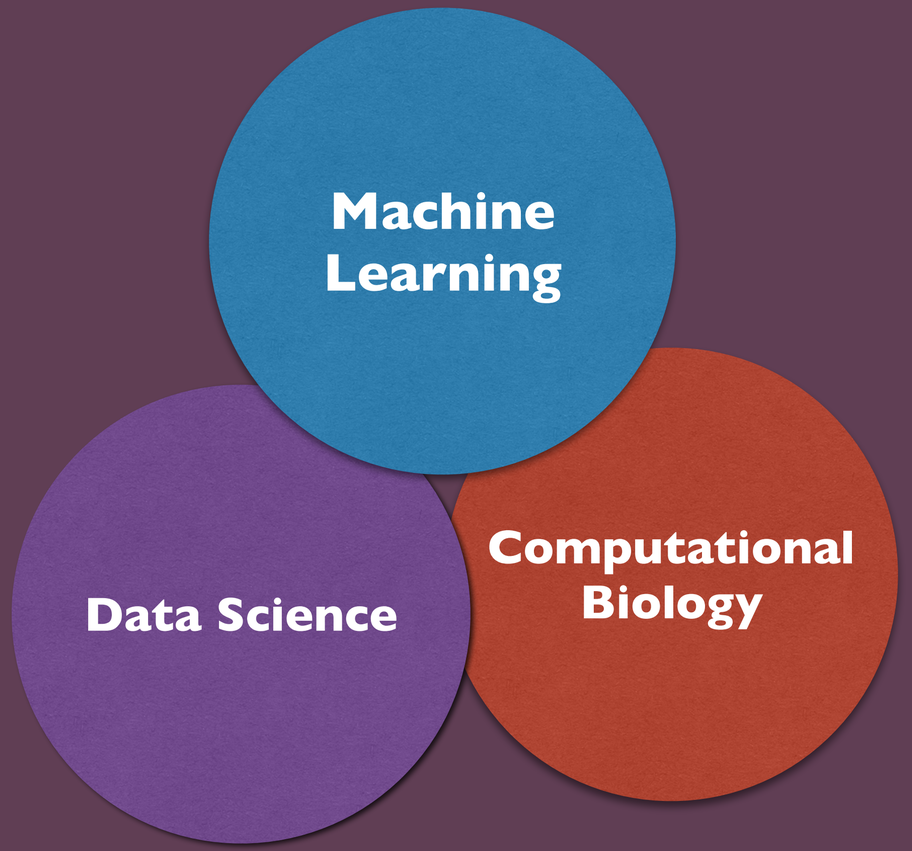
**2021-Present:** PhD, Stanford CS Department,
Jure Leskovec's lab, "Biocomputation" Track

# Yanay's research history

**High School** — Software Engineering, Data Portals, Cloud Computing

**Undergrad** — Computational Biology, Data Science, single-cell RNA

**PhD** — Machine Learning, Foundation models / Language Models for Biology, "AI Virtual Cell"



# Computer science research

What is the goal of research?
Why has it driven major innovations in computing?
What separates research from advanced development?

# A Tale of Three Turing Awards

# Hennessy and Patterson: RISC

Computer architecture was increasing in complexity, in order to enable more and more advanced computation.

Everyone thought that increasingly powerful processors needed increasingly complicated instruction sets to take advantage of them.

The New York Times

GIVE THE TIMES

## *Computer Chip Visionaries Win Turing Award*



Dave Patterson, right, and John Hennessy in the early 1990s. The men won the Turing Award for their pioneering work on a computer chip design that is now used by most of the tech industry. Shane Harvey

**By Cade Metz**

March 21, 2018

SAN FRANCISCO — In 1980, Dave Patterson, a computer science professor, looked at the future of the

# Hennessy and Patterson: RISC

**"No, let's do it this way instead:"** have a very simple instruction set. That way you can compare performance, optimize, and prevent errors.

This became known as Reduced Instruction Set Computer (RISC). It led to a sea change in architectures, and the founding of multiple major silicon valley companies.

The New York Times

GIVE THE TIMES

*Computer Chip Visionaries Win Turing Award*

Dave Patterson, right, and John Hennessy in the early 1990s. The men won the Turing Award for their pioneering work on a computer chip design that is now used by most of the tech industry. Shane Harvey

By Cade Metz

March 21, 2018

SAN FRANCISCO — In 1980, Dave Patterson, a computer science professor, looked at the future of the

# Engelbart: interactive computing

When computers originated, they were used for, well, computing: calculating mathematical functions.

This meant that computers were seen as most appropriate for slow, batch interaction, shared by entire teams.

The New York Times

GIVE THE TIMES

DOUGLAS C. ENGELBART, 1925-2013

## *Computer Visionary Who Invented the Mouse*

By John Markoff

July 3, 2013

Douglas C. Engelbart was 25, just engaged to be married and thinking about his future when he had an epiphany in 1950 that would change the world.

He had a good job working at a government aerospace laboratory in California, but he wanted to do something more with his life, something of value that might last, even outlive him. Then it came to him. In a single stroke he had what might be safely called a complete vision of the information age.

The epiphany spoke to him of technology's potential to expand human intelligence, and from it he spun out a career that indeed had lasting impact. It led to a host of inventions that became the basis for the Internet and the modern personal computer.

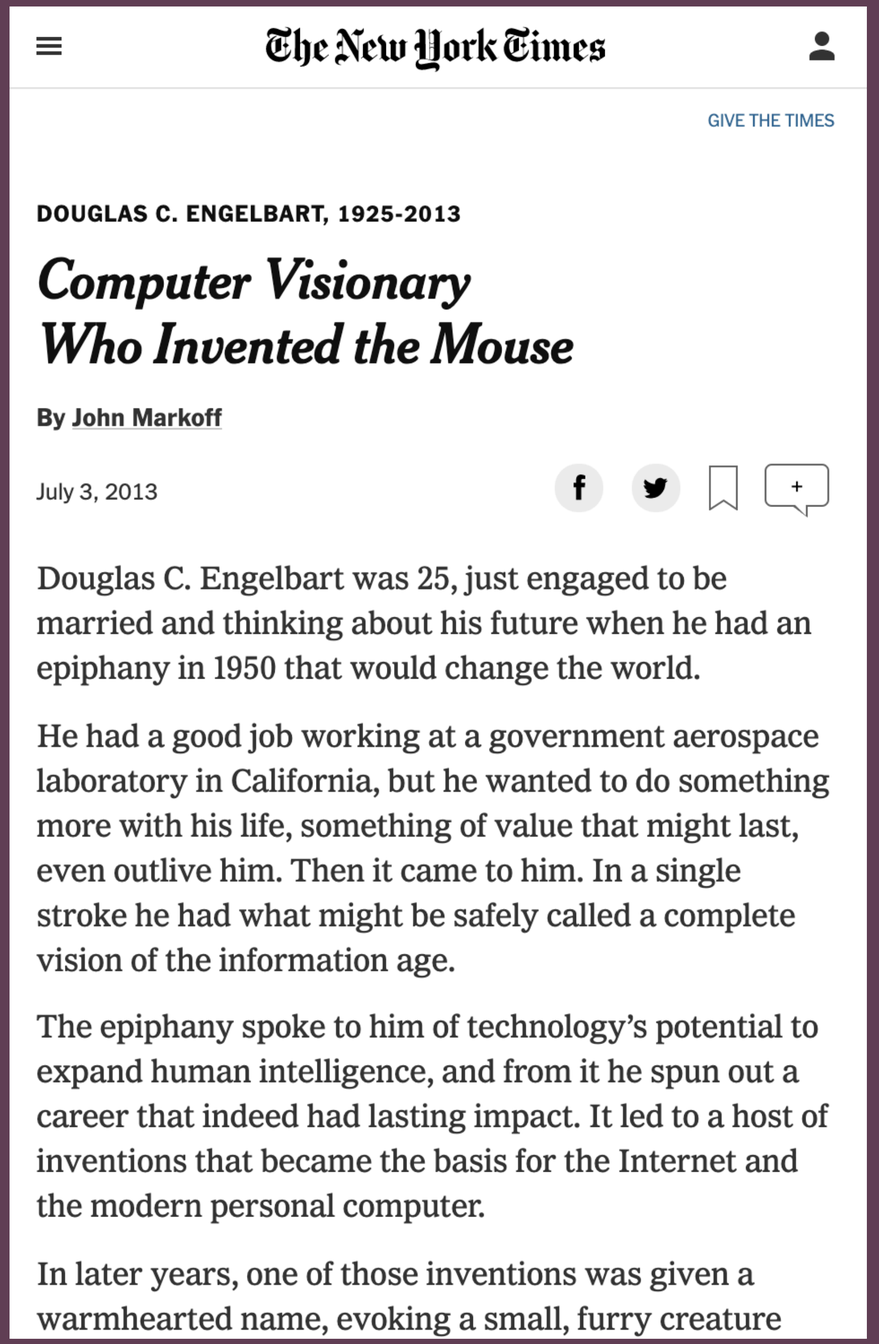
In later years, one of those inventions was given a warmhearted name, evoking a small, furry creature

# Engelbart: interactive computing

**"No, let's do it this way instead:"**
computing should be used as a tool for thought. We must move from batch-style computing to interactive computing.

His result was the "Mother of All Demos": mouse, hypertext, bitmapped screens, collaborative software, and more.

This led to Xerox Star. Steve Jobs saw it, was wow'ed, and infused the ideas into the Mac.



The New York Times

GIVE THE TIMES

DOUGLAS C. ENGELBART, 1925-2013

## *Computer Visionary Who Invented the Mouse*

By John Markoff

July 3, 2013

Douglas C. Engelbart was 25, just engaged to be married and thinking about his future when he had an epiphany in 1950 that would change the world.

He had a good job working at a government aerospace laboratory in California, but he wanted to do something more with his life, something of value that might last, even outlive him. Then it came to him. In a single stroke he had what might be safely called a complete vision of the information age.

The epiphany spoke to him of technology's potential to expand human intelligence, and from it he spun out a career that indeed had lasting impact. It led to a host of inventions that became the basis for the Internet and the modern personal computer.

In later years, one of those inventions was given a warmhearted name, evoking a small, furry creature

# LeCun, Hinton, Bengio: deep learning

The idea of neural networks had been around for fifty years, but unsuccessful. Major AI figures had trashed it, even proving that early versions had very limited expressiveness.

Instead, machine learning was based on other models, for example the support vector machine and graphical models. Neural networks did not perform well.

The New York Times

GIVE THE TIMES

*Turing Award Won by 3 Pioneers in Artificial Intelligence*

From left, Yann LeCun, Geoffrey Hinton and Yoshua Bengio. The researchers worked on key developments for neural networks, which are reshaping how computer systems are built.

From left, Facebook, via Associated Press; Aaron Vincent Elkaim for The New York Times; Chad Buchanan/Getty Images

By Cade Metz

March 27, 2019

# LeCun, Hinton, Bengio: deep learning

**"No, let's do it this way instead:"** these networks learn extremely complex functions, so they need much more data than existing machine learning approaches, GPUs to train, and algorithms to enable them to learn more effectively.

Around 2010, these models began smashing records in speech and image recognition. They are now foundational to ML.

The New York Times

GIVE THE TIMES

*Turing Award Won by 3 Pioneers in Artificial Intelligence*

From left, Yann LeCun, Geoffrey Hinton and Yoshua Bengio. The researchers worked on key developments for neural networks, which are reshaping how computer systems are built.
From left, Facebook, via Associated Press; Aaron Vincent Elkaim for The New York Times; Chad Buchanan/Getty Images

By Cade Metz

March 27, 2019

# Not all research wins Turing Awards. But…

It all follows this same formula —

**An implicit assumption:** Industry and other researchers all thought one way about a problem

**"No, let's do it this way instead:"** The researcher offers a new perspective that nobody had ever considered or made feasible before. They proved out their idea as the better approach.

# And now, a definition.

Research introduces a fundamental **new idea** into the world.

Examples:

Simple instruction sets for complex computer architecture
Computing that is interactive, not batch
Algorithms needed to make deep learning effective

These ideas did not exist in any mature or well-articulated way before their creators developed them.

If the idea is already in the world, for example published by someone else, it is not considered **novel**, and thus not research.

Before: small computer vision datasets

After: huge computer vision dataset, and algorithms to utilize it

Before: programmers manually reserve resources for cloud computing

After: programmers provide needs, software allocates resources

Before: models of people are sparse

After: generative AI agents capture broad behavior

Before: we think web tracking is isolated to the intended site

After: it's much leakier than we realized

Before: biologists had to discover cell functions

After: AI can discover cell functions

Before: underwater robots should look and feel like boats

After: humanoid underwater robotics

# Research creates industry


PageRank algorithm


Stanford University Network workstation (SUNet)


Computer graphics architectures


Online education


Computer virtualization

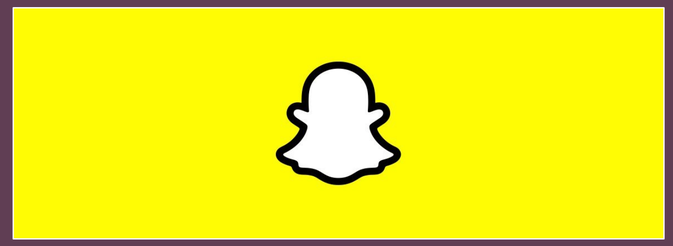
# Industry and research

# Industry vs. research

What makes other start-ups and industry different than research?

If the core idea already exists, but needs to be refined in order to see success…it might be important, but it's not research.





# Industry vs. research

Companies can and do engage in development that is research…

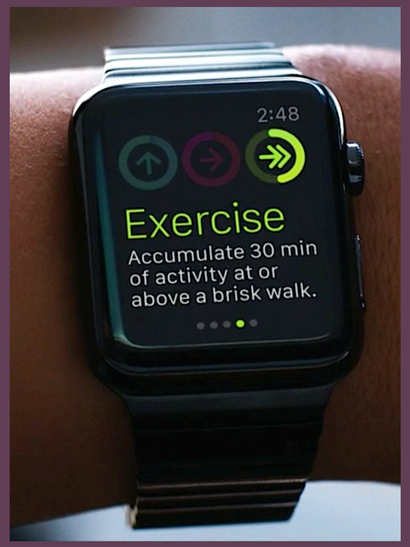
MapReduce and Spanner at Google

Kinect at Microsoft

…but typically companies are working to scale out ideas that exist.

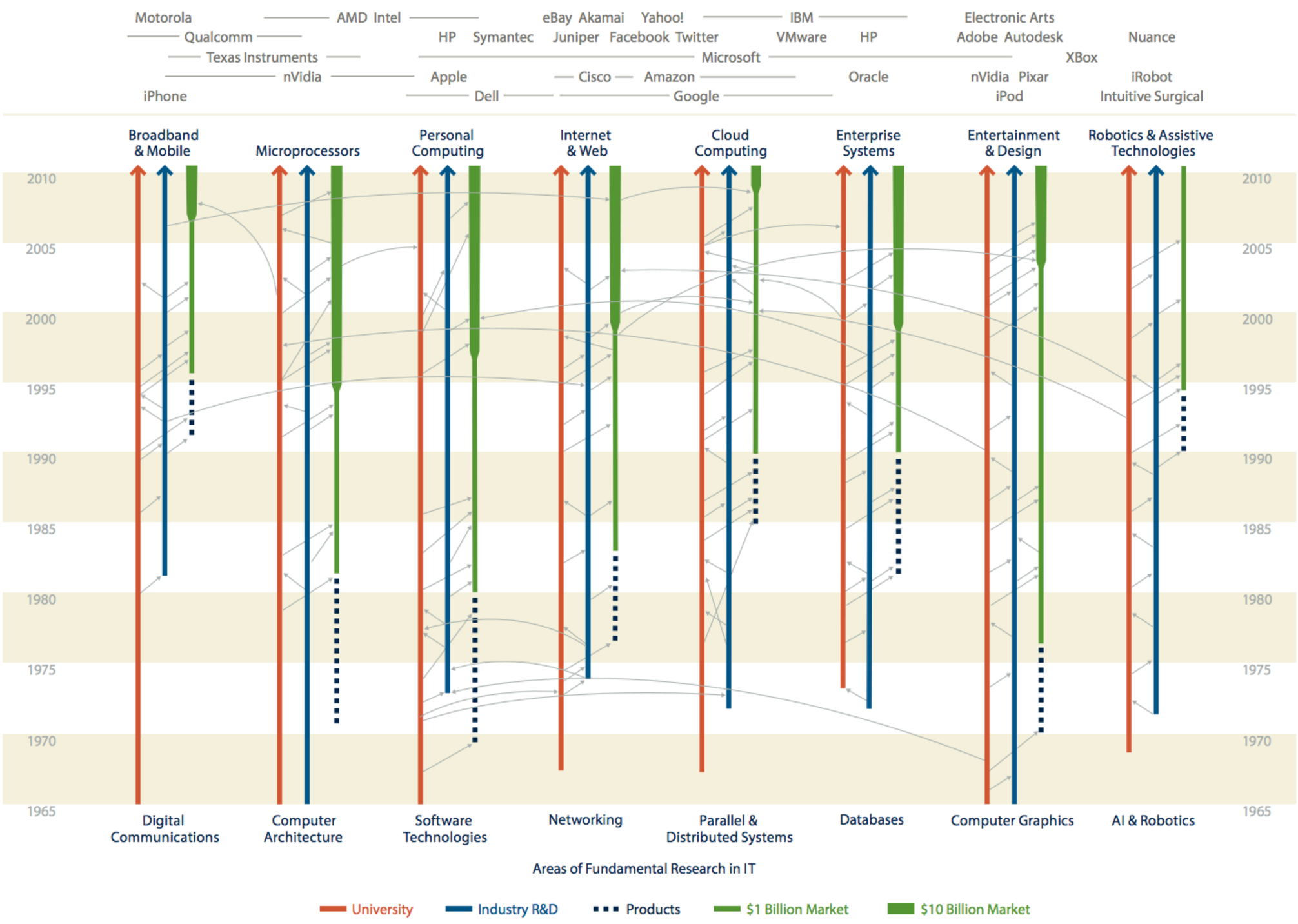

Landay, 2000s:
activity sensing

Credit because he developed the concept and popularized it




Apple, 2010s:
Apple Watch

Credit because they engineered it to work and launched it




# CS tire tracks diagram

Implication: by doing research, you are living about 15 years in the future.

# (An incomplete list of) research areas in computer science

# Flavors of CS research

Computer science is field held together by a shared phenomenon of interest: computing.

This sets it apart from some other fields, which are drawn together by a shared theory or shared methodology. While this is a simplification, it is a helpful first cut:

Psychology: methodology of randomized experiment

Sociology: shared theories — Weber, Marx, Durkheim, DuBois, Quetelet [https://mastodon.social/@kjhealy/109381636840967937]

Math: methodology of formal proof

Anthropology: methodology of participant observation

Architecture

Artificial intelligence

Computational biology

Computer graphics

Computer security

Computer systems

Computer vision

Data science

Education

Human-computer interaction

Machine learning

Natural language processing

Networking

Operating/distributed systems

Programming systems/verification

Robotics

Theory

# Topic: artificial intelligence

Architecture

**Artificial intelligence**

**Computational biology**

Computer graphics

Computer security

Computer systems

**Computer vision**

Data science

Education

Human-computer interaction

**Machine learning**

**Natural language processing**

Networking

Operating/distributed systems

Programming systems/verification

**Robotics**

Theory

# Topic: computer systems

**Architecture**

Artificial intelligence

Computational biology

Computer graphics

**Computer security**

**Computer systems**

Computer vision

Data science

Education

Human-computer interaction

Machine learning

Natural language processing

**Networking**

**Operating/distributed systems**

**Programming systems/ verification**

Robotics

Theory

# Topic: theory

Architecture

Artificial intelligence

Computational biology

Computer graphics

**Computer security**

Computer systems

Computer vision

Data science

Education

Human-computer interaction

**Machine learning**

Natural language processing

Networking

Operating/distributed systems

**Programming systems/ verification**

Robotics

**Theory**

# Method: engineering

**Architecture**

Artificial intelligence

Computational biology

**Computer graphics**

**Computer security**

**Computer systems**

Computer vision

**Data science**

Education

**Human-computer interaction**

Machine learning

Natural language processing

**Networking**

**Operating/distributed systems**

**Programming systems/ verification**

**Robotics**

Theory

# Method: probability and modeling

Architecture

**Artificial intelligence**

**Computational biology**

Computer graphics

Computer security

Computer systems

**Computer vision**

Data science

Education

Human-computer interaction

**Machine learning**

**Natural language processing**

Networking

Operating/distributed systems

Programming systems/verification

**Robotics**

**Theory**

## Method: formal reasoning and proof

Architecture

**Artificial intelligence**

Computational biology

Computer graphics

**Computer security**

**Computer systems**

Computer vision

Data science

Education

Human-computer interaction

**Machine learning**

Natural language processing

Networking

**Operating/distributed systems**

**Programming systems/ verification**

Robotics

**Theory**

# Method: design

Architecture

Artificial intelligence

Computational biology

Computer graphics

Computer security

Computer systems

Computer vision

Data science

Education

**Human-computer interaction**

Machine learning

Natural language processing

Networking

Operating/distributed systems

Programming systems/verification

Robotics

Theory

**Method: empirical measurement and hypothesis testing**

Architecture

Artificial intelligence

**Computational biology**

Computer graphics

Computer security

Computer systems

Computer vision

**Data science**

**Education**

**Human-computer interaction**

Machine learning

Natural language processing

Networking

Operating/distributed systems

Programming systems/verification

Robotics

Theory

# Research mindset

# Research is different than your usual coursework.

Coursework tends to be very clearly defined. Research tends to be exploratory and iterative.

You probably won't know the right answer: if we knew whether it was going to work, it wouldn't be research.

**If you are stuck, do not wait until the next time you meet your TA. Proactively reach out!**

## “I like” from research:

“The free-form structure of our project”

“The freedom to choose the questions and methods I find interesting”

“The independence I got in establishing a research direction”

“That I have had the opportunity to do a lot of self guided research and reading. I feel very free to shape parts of my learning and research experience.”

## “I wish” from research:

“That there was more structure or well-defined expectations.”

“I had a clearer idea or more deliverables and felt the barrier of being unfamiliar with certain parts of the project or coming on late less.”

"I had been able to narrow my scope a little earlier"

**Research is a new and different skill. Embrace and navigate through the uncertainty.**

# How this course works

# Learning goals

Execute a first research project at the scale that can be submitted to a workshop or work-in-progress at a top-tier conference.

Understand the major research topics currently active in your area. Be able to read a research paper and perform a literature review in that area.

Apply vectoring and velocity skills for navigating the open-ended nature of research.

Design and execute an appropriate evaluation of your method.

Write a paper and engage in the peer review process.

# Is this course right for me?

CS 197 is the best fit if you're…

- Just starting your research journey
- Done with CS 106B

It's not the best fit if you're…

- Looking for an advisor for your own research idea
- Just looking for research units (connect with a lab and take CS 195 / CS 399)

# Research project

This class is structured around a quarter-long research project. The project is completed in groups of two–three within a section.

These projects are designed to be accessible to you, of interest to the research community, and achievable within the timeline of the course.

> "I have my own idea!": mention it to your TA. We are unlikely to bend given those goals of accessibility, broader research interest, and achievability, but want to hear your ideas — it's possible!

# Groups and projects

Form project teams and align with topics in section during Week 2.

You can pick your groups, and your group can pick a project from a prepared list of options.

> You will have some freedom to evolve the shape of that project…
>
> …but we chose it to scope your project to something we know we can advise well, and that we think you can finish by the end of the quarter.

# Assignments

Assignments offer waypoints in support of the project.

- Assignment 1 (individual): learning about the project area, and learning how to read a paper
- Assignment 2: literature review
- Assignment 3: introduction
- &etc.

Progress Reports

All details are at cs197.stanford.edu

# Grading

Grades will be returned as ✓-, ✓, ✓+

- ✓- means "needs improvement"
- ✓ means "solid work"
- ✓+ means "wow, nice job!"

Rough benchmark: if you're getting ✓ on every assignment, you're on track for an A in the class

# Responding to TA feedback

Your assignments are building toward a full research paper

Your TA will give feedback on each submission. To encourage mastery learning, we expect that you will address that feedback in your final version (final talk, final paper)

- Look for feedback to be tagged as **required**, **recommended**, or **FYI**
- Final submissions that do not address feedback will be penalized in the final talk and final paper grades

# What after CS 197?

Our goal is for CS 197 to be an onramp for you to research in Computer Science. We will:

- Perform outreach to faculty in CS or at Stanford to help introduce you so you can work on research projects after demonstrating excellence here
- Support you in submitting your work to flagship conferences, and connect you with funding opportunities to travel to present the work

# Questions?

# Before 11:59pm: sign up!

Priority placement to those who fill it out today. Link at cs197.stanford.edu and on Canvas.

# Computer Science Research