



Robot Perception

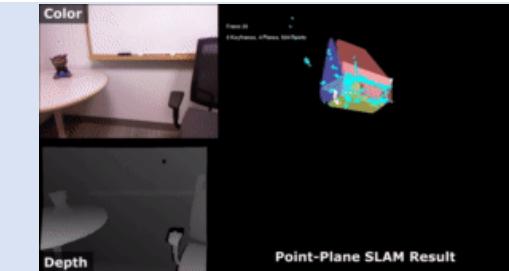
Introduction

Dr. Chen Feng

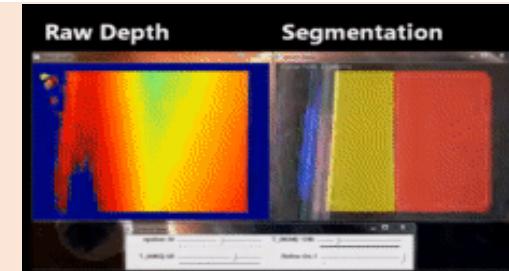
cfeng@nyu.edu

ROB-GY 6203, Fall 2024

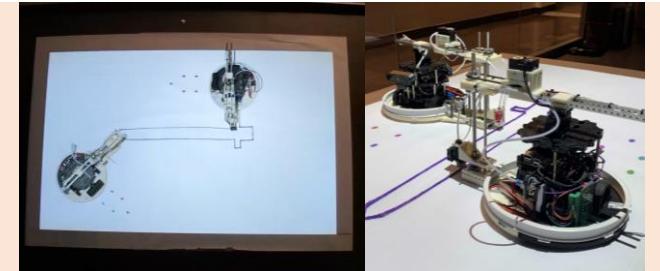
AI4CE Lab's Research in Spatial AI (Vision, Learning, & Robotics)



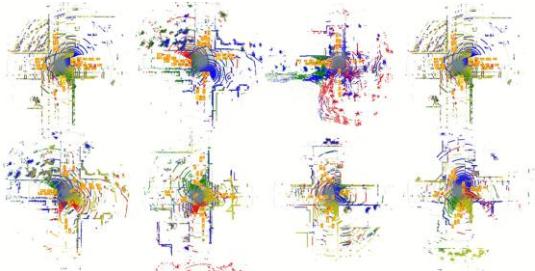
PP-SLAM, ICRA'13



PEAC, ICRA'14



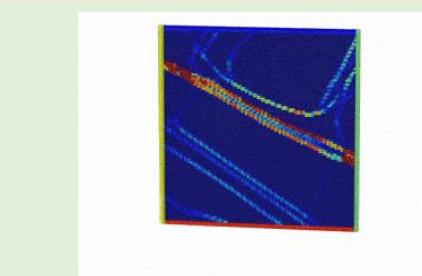
M3DP, ICRA'21



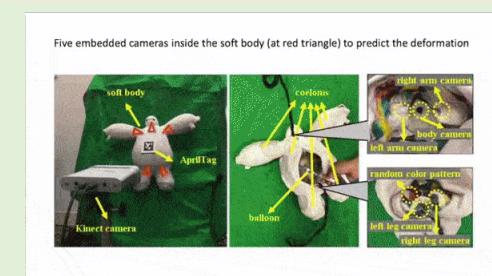
CoPerception, NeurIPS'21



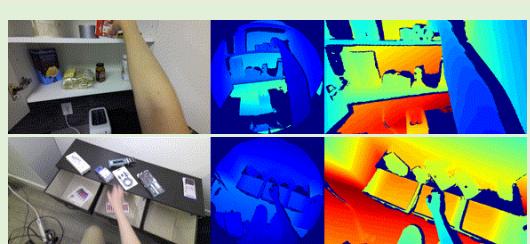
DeepExplorer, RSS'23



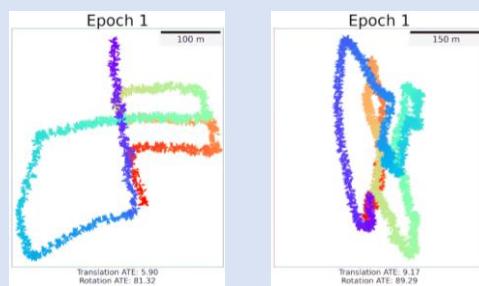
FoldingNet, CVPR'18



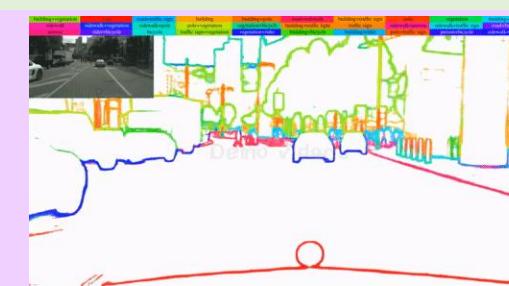
DeepSoRo, RAL'20



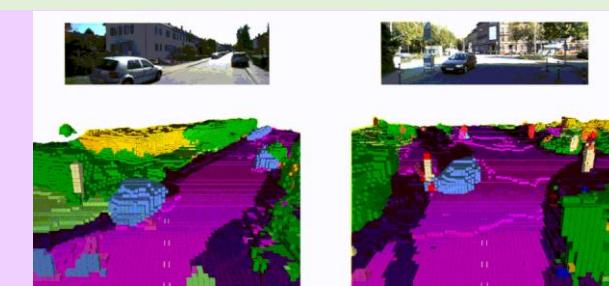
EgoPAT3D, CVPR'22



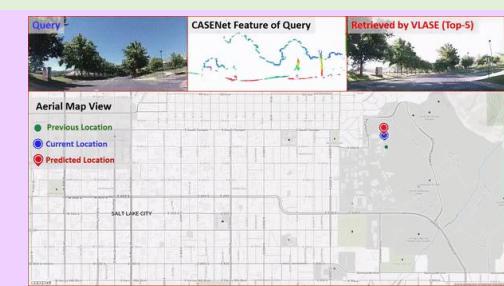
DeepMapping2, CVPR'23



CASENet, CVPR'17



VoxFormer, CVPR'23



VLASE, IROS'18



Course Information

- [Syllabus](#)
- [Slack](#)
- Students:
 - 10 PhD students, 45 Master students, 1 Junior student
 - 24 Robotics, 9 EE, 6 CE, 7 MAE, 5 CS, 1 CUE/CUSP, 1 BME, 1 NYUSH



Prerequisites

- Need-to-have
 - Linear Algebra
 - Probability
 - Vector Calculus
- Should-have
 - Assignments: Python (unless you are a fast learner)
 - Project: Python/MATLAB/C/C++/Java/...
- Nice-to-have
 - Image processing
 - Machine learning

Course Objectives

- Describe robot perception models/algorithms and their applicable problem types.
- Compare assumptions in each model/algorithm and their strengths and weaknesses.
- Determine appropriate robot perception models/algorithms for a given problem.
- Formulate a robot perception problem formally.
- Solve the formulated problem with programming tools.



My Teaching Philosophy

- Focus on delivering the knowledge by explaining intuitions in solutions
 - I hope to teach you how to **formulate** problems before discussing the solutions
 - So you can apply the algorithms for new applications in your future career
 - **Expectation for you:**
 - read the given textbook (pick at least one) chapters/sections **before coming to the class**
- Hands-on robot perception experiences via project and homeworks
 - Project-based Learning
 - But this is **not** a programming course
 - So our assignments are not programming heavy, but focus on the knowledge
 - **Expectation for you:**
 - Write codes and play with them **after the class** (use APIs wisely)
- To introduce state-of-the-art **research** in computer vision with a robotics and engineering context
 - **Expectation for you:**
 - **Take this course proactively**, not passively: you can **self-study ahead** of our weekly schedule!



Evaluation Methods



- All deliverables are due before the start of the class.
- 30%: 3 in-class quizzes, close-book (no more than 10min each)
- 30%: 2 homework (15% each)
- 40%: course project
 - Embodied AI Challenge (details to be announced in class)
 - Given onboard camera views, your team should navigate a mobile robot to find goal locations as soon as possible
 - It does not require any experience in real-world robotics
 - Any methods (fully manual, semi-automatic, fully autonomous) are welcome, but that would affect the final points you receive on this course project
 - 1 - 4 students per team
 - Detailed evaluation metrics are to be announced in class
 - See last year's highlight video:
<https://www.linkedin.com/feed/update/urn:li:activity:7143437082135334913/?originTrackingId=11Nlpsp4RXG5tWMkEs6UGQ%3D%3D>

Useful Resources

- Wikipedia
 - Good way to learn new concepts
 - But do not trust it wholeheartedly
- CVonline: <https://sites.google.com/site/cvonlinewiki>
 - A dedicated list of wiki links to computer vision
 - And a list of vision-related online books
 - Some maybe outdated
- Paper with code: <https://paperswithcode.com/>
- Google/Bing/Google-Scholar should be your best friend!
- **We are ChatGPT friendly!**
 - Homework? Sure, but don't trust it on its own, cross validate!
 - Project? Sure, feel free to take advantage of it!
 - **Quiz? No**, and you won't have time anyway.

Related Conferences

- CVPR: Computer Vision and Pattern Recognition
- ICCV/ECCV
- ICRA: IEEE International Conference on Robotics and Automation
- IROS
- 3DV
- ISARC: International Symposium on Automation and Robotics in Construction

Open-Source Libraries & Tools

- OpenCV
 - PCL: Point Cloud Library
 - Open3D
 - NumPy/Eigen
 - Google Ceres, G2o
 - Scikit-learn
 - PyTorch, TensorFlow/Keras
-
- MeshLab/CloudCompare
 - COLMAP



Q & A / Office hours

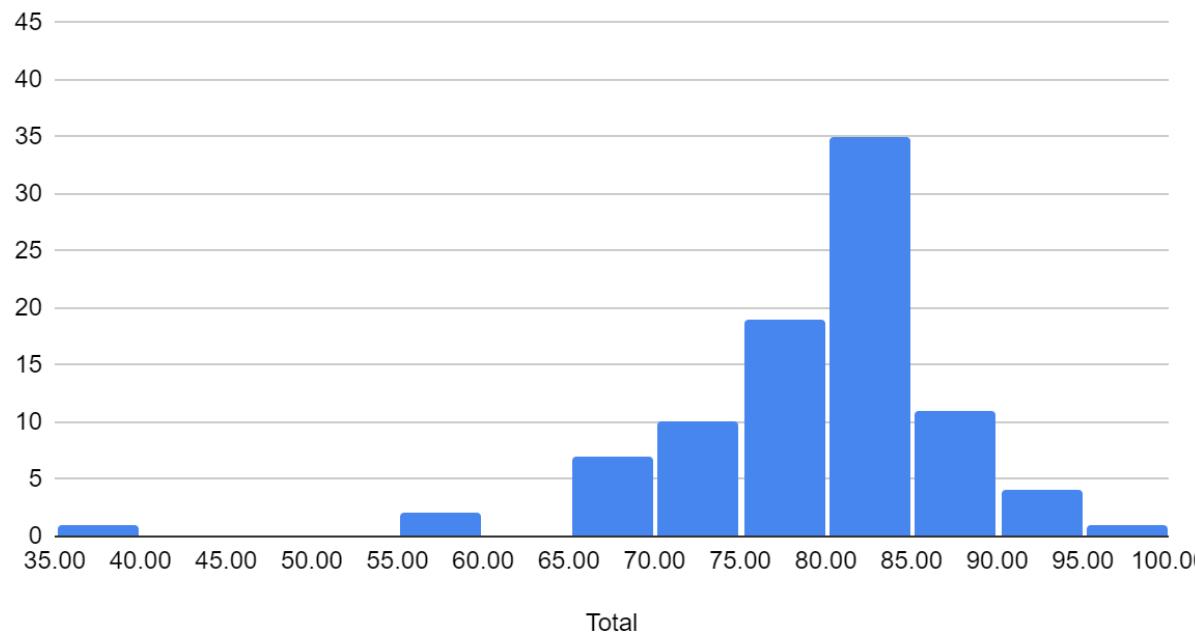
- Always use Slack first
 - DM the TAs if the question only applies to yourself
 - Use #qna if the question (and answers) might help more students
- Office hours will be offered by our TAs (Time & Location/Zoom TBD)
 - We have a large cohort, so first come, first served
 - Use these hours wisely:
 - Get TAs' help to deep dive into concepts you could not understand after reading on your own
 - Ask TAs' experience when they learned the courses in previous years
 - Get TAs' feedback on your course project solutions and potential bugs in your codes (TAs will offer suggestions but they may not have enough time to debug for you)
- Don't
 - Ask for HW/Project solutions **directly** --- TAs are asked NOT to respond to such questions
 - Ask for debugging **unless** you tried your best (including asking ChatGPT) and still have no clue
 - Ask for debugging **without** enough contexts (screenshots, detailed explanations, etc.) for people to help you



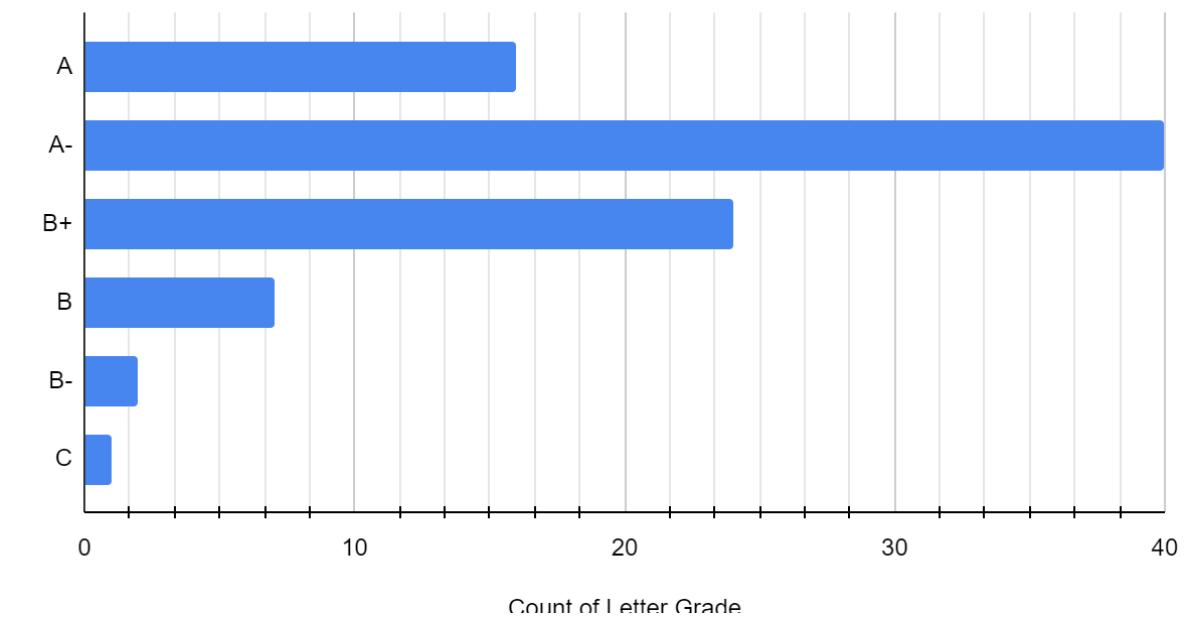
Statistics from Fall'23

- Overall Course Score/Grade Distribution

Histogram of Total



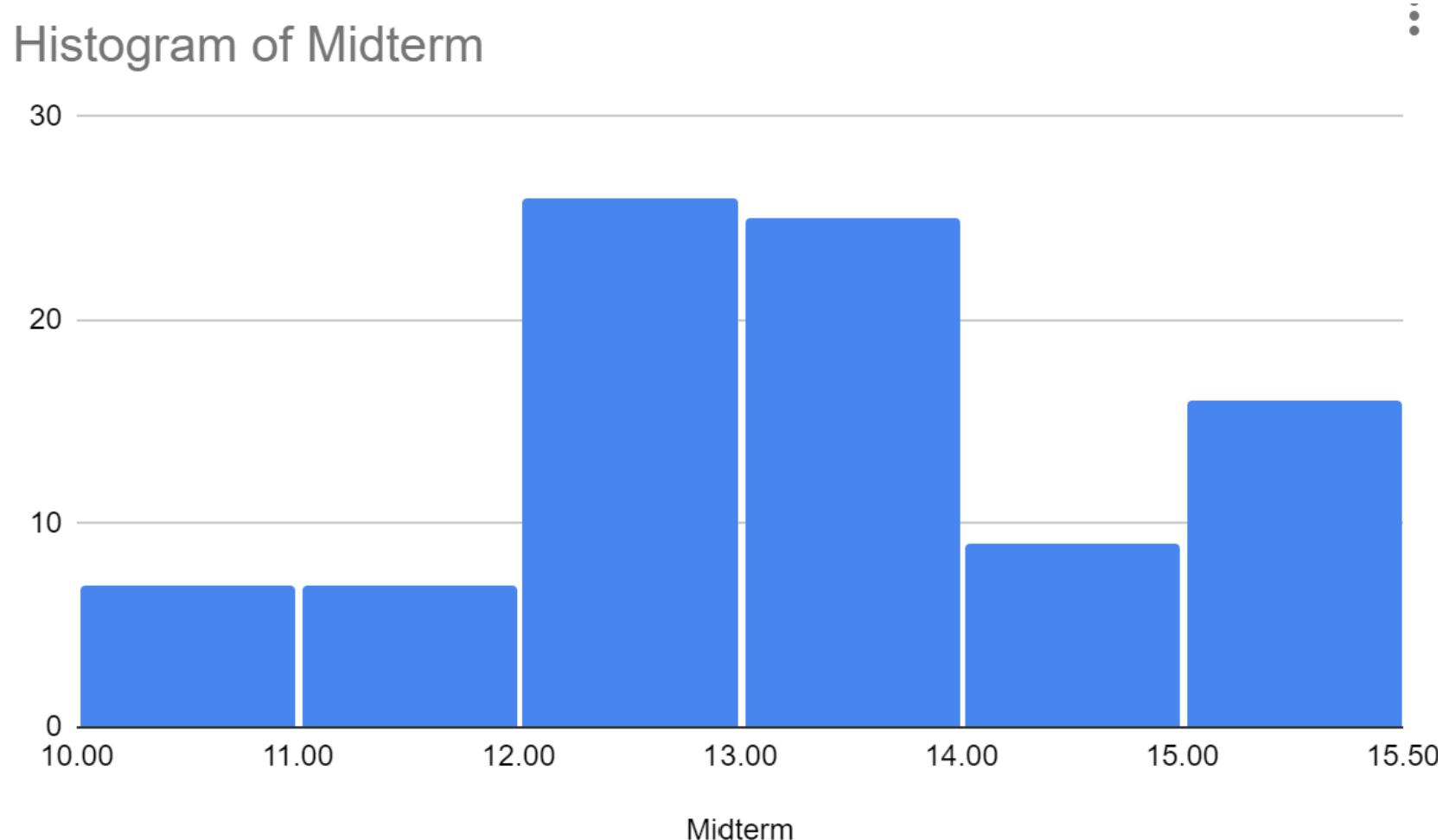
Count of Letter Grade





Statistics from Fall'23

- Course Project Midterm Grade Distribution

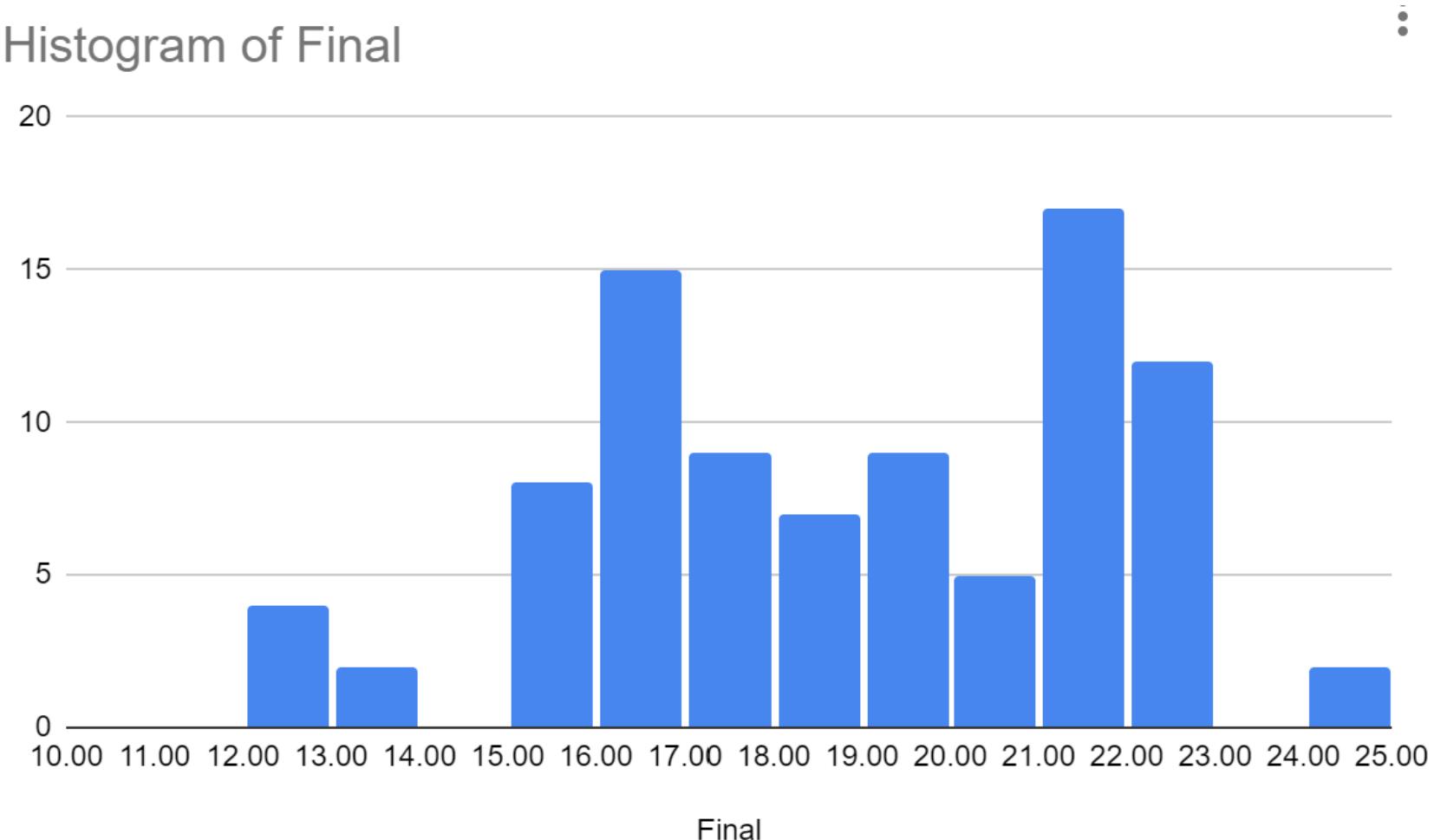




Statistics from Fall'23

- Course Project Final Grade Distribution

Histogram of Final

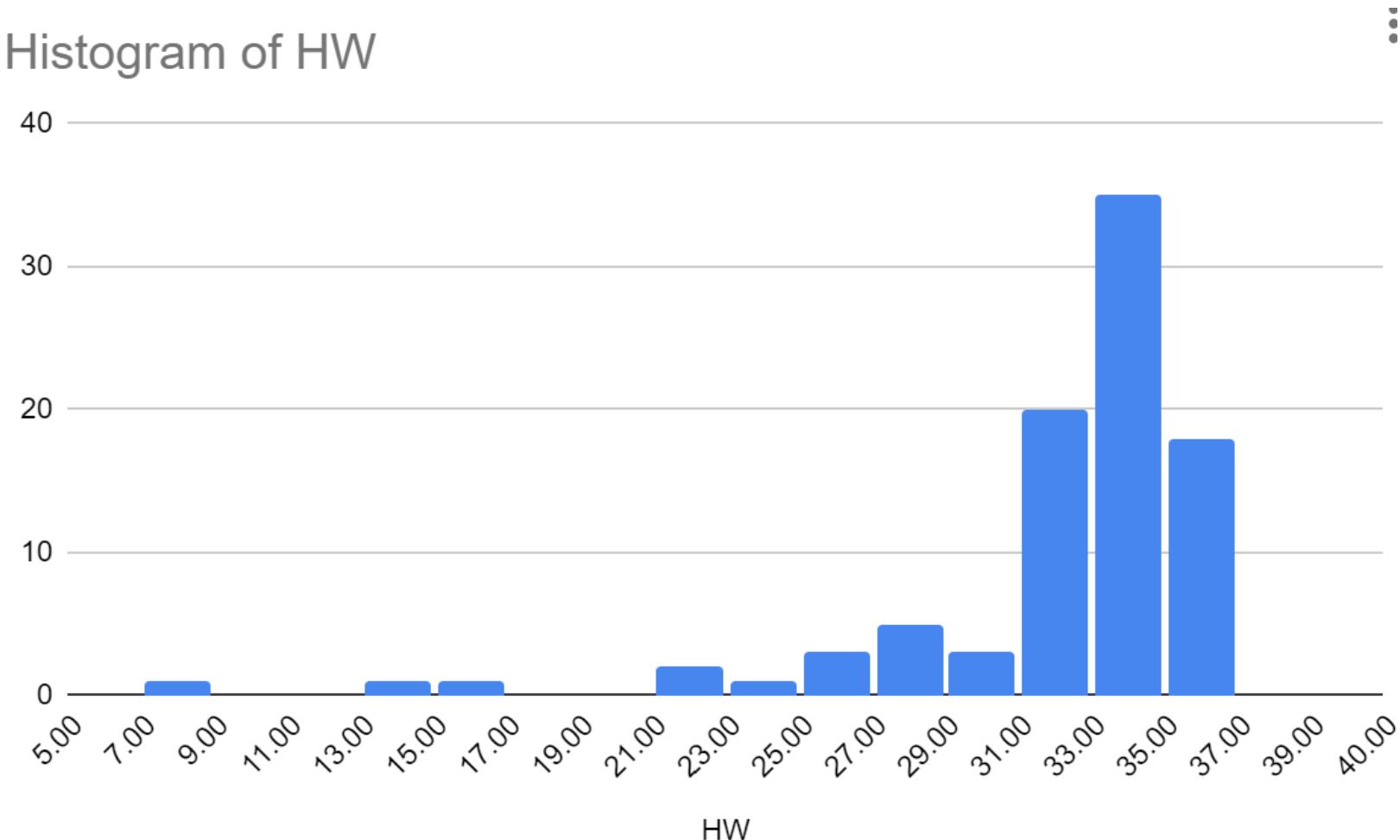




Statistics from Fall'23

- HW Grade Distribution

Histogram of HW

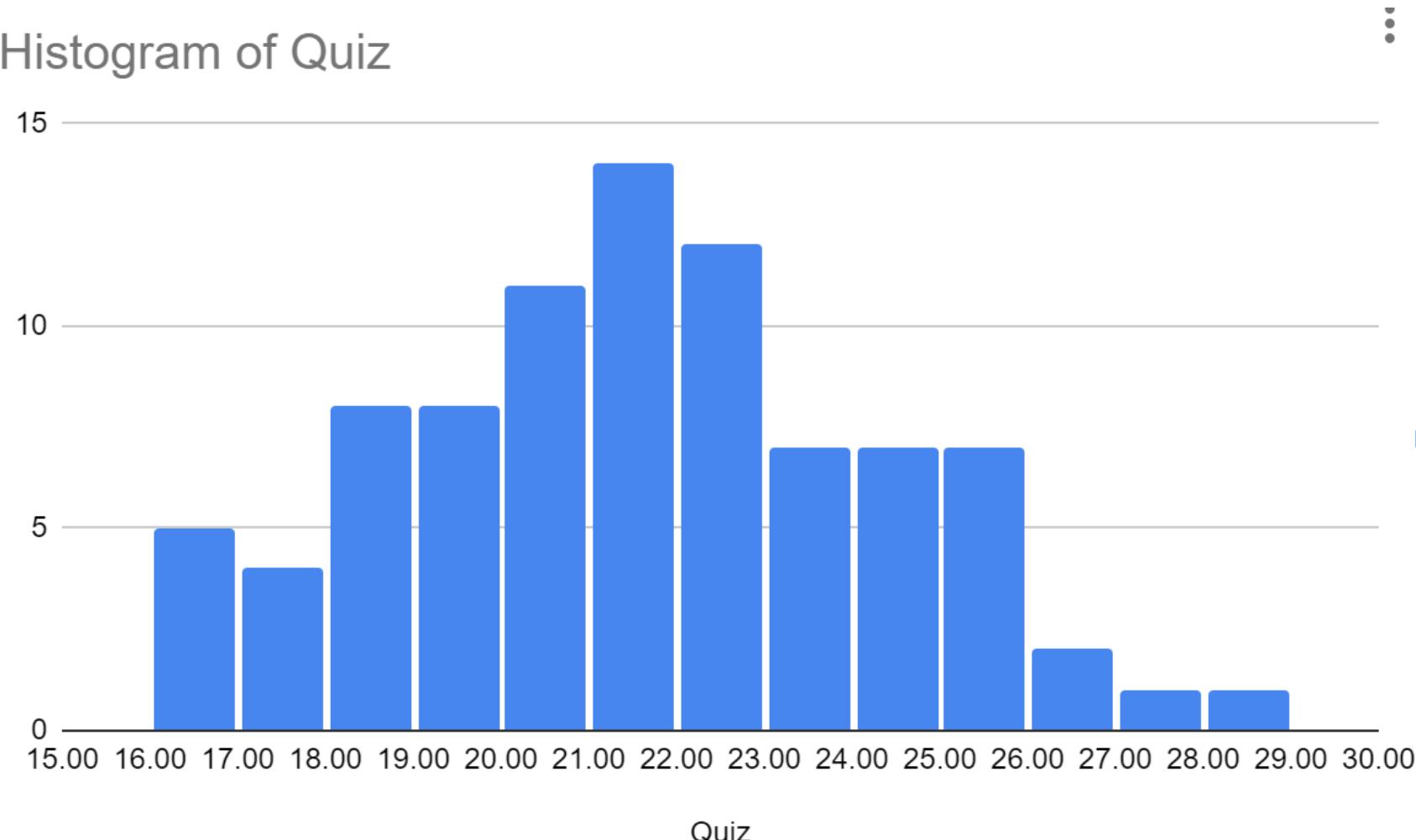




Statistics from Fall'23

- Quiz Grade Distribution

Histogram of Quiz





Definition of Robot Perception in This Course

- Robot
 - A machine capable of carrying out a complex series of actions automatically
 - Regardless of how it looks
 - With a bit of a stretch in scale, an automated system also counts
- Perception
 - The *awareness* of the elements of environment through physical *sensation*
 - The process of becoming *aware* of something through the *senses*



Definition of Robot Perception in This Course

- Robot Perception
 - The hardware and software
 - to sense and process raw signals
 - to become aware of the states of a system itself and/or its surroundings
 - which enables the system to take proper actions to fulfil its design goals
- Shortened version
 - **Observing data to understand situations** for achieving system objectives



Types of Robot Perception

- Visual
 - Hardware: cameras, LIDAR (line-of-sight sensors)
 - Software: **computer vision, machine learning**
 - The focus of this course
- Auditory
- Tactile



Related Disciplines – Can you link different research areas?



Artificial
Intelligence

Neuroscience
Cognitive Science

Image Processing

Robot/Computer Vision

NLP/Speech

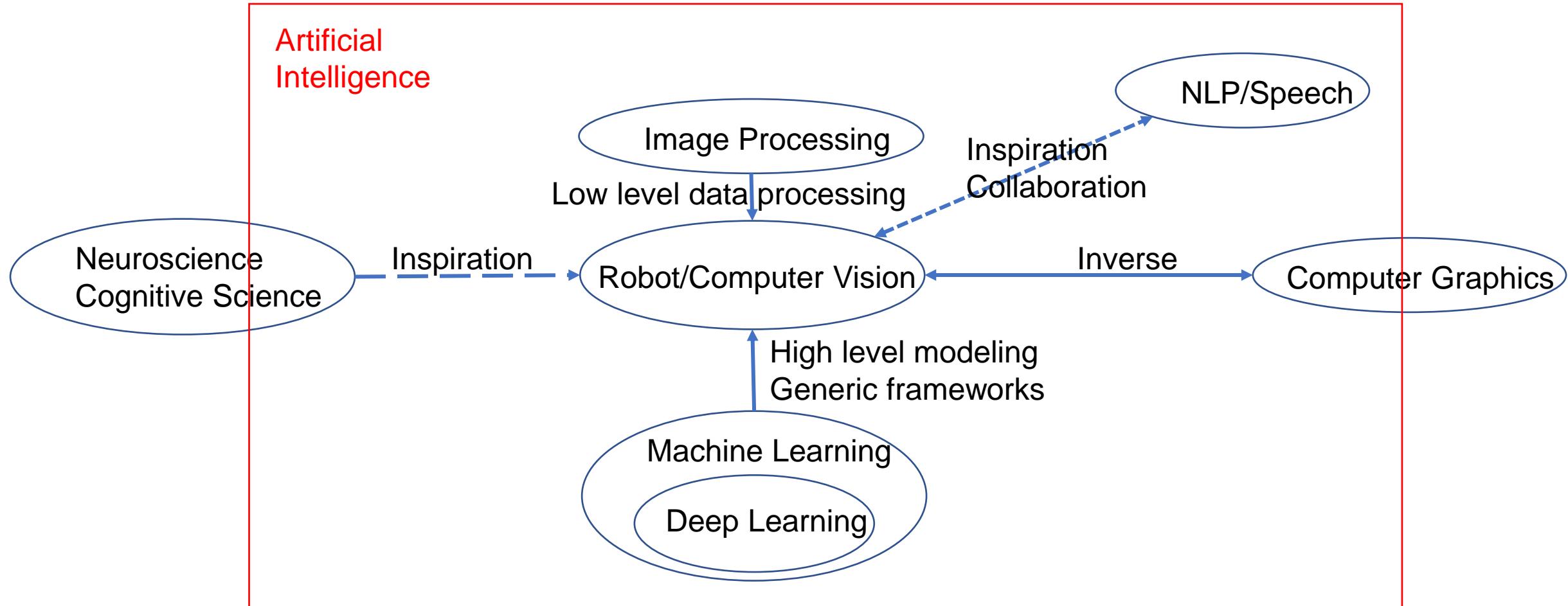
Machine Learning

Deep Learning

Computer Graphics

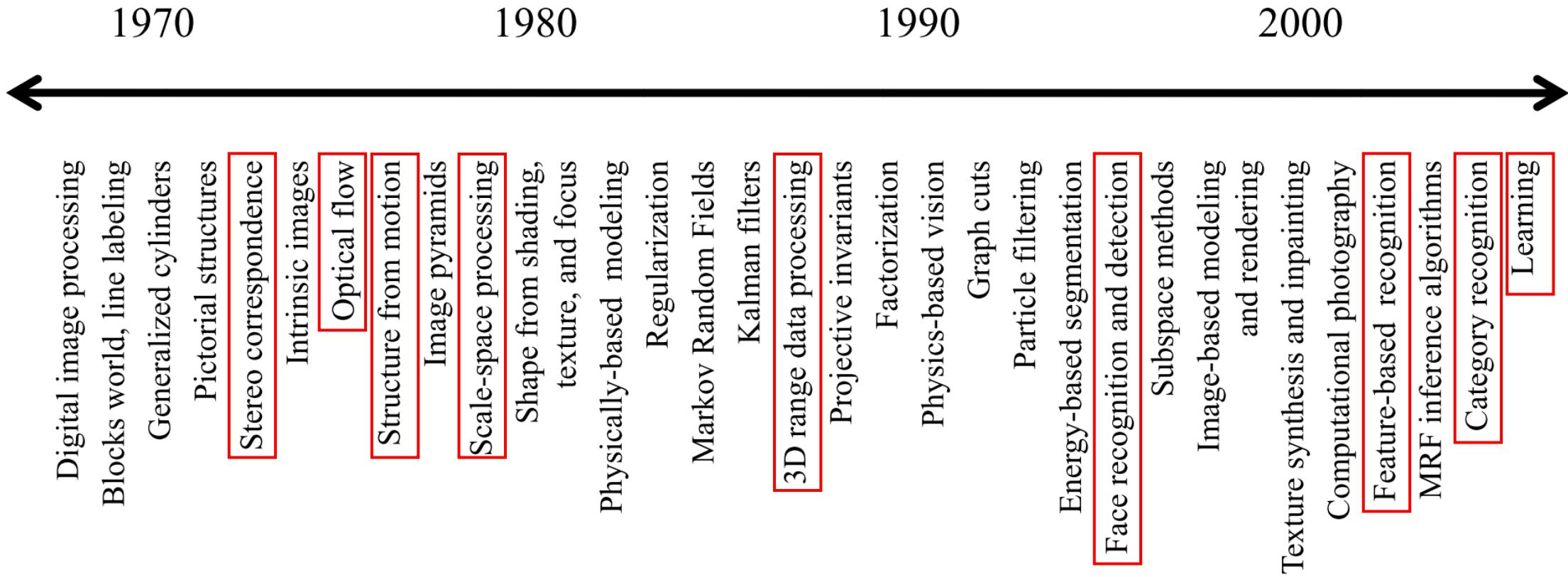


Related Disciplines





A Brief History of Computer Vision



Main Topics in Robot Vision

- Shape/Structure
- Pose
- Segmentation
- Detection
- Tracking
- Recognition

Motivating Examples

- Humanoid/Legged Robots
- Manufacturing
- Transportation
- Civil/Construction Engineering
- Space Exploration
- Architecture
- Agriculture
- Medical/Health
- Entertainment



DARPA Robotics Challenge



Video from: <https://youtu.be/wX0KagJ1du8>

Boston Dynamics



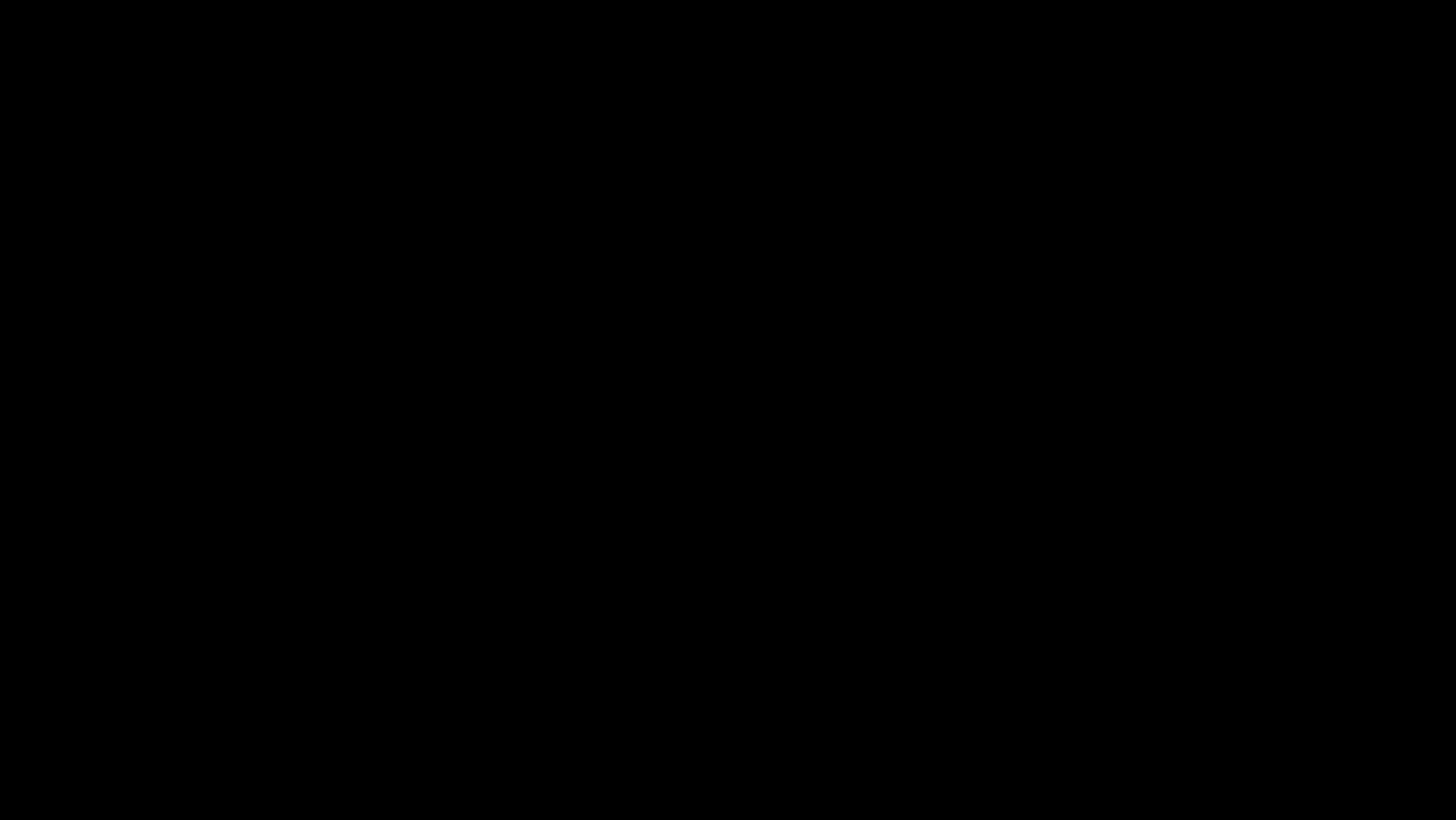
Video from: https://youtu.be/Ve9kWX_KXus

Drones in the lab



Video from: <https://youtu.be/w2itwFJCgFQ>

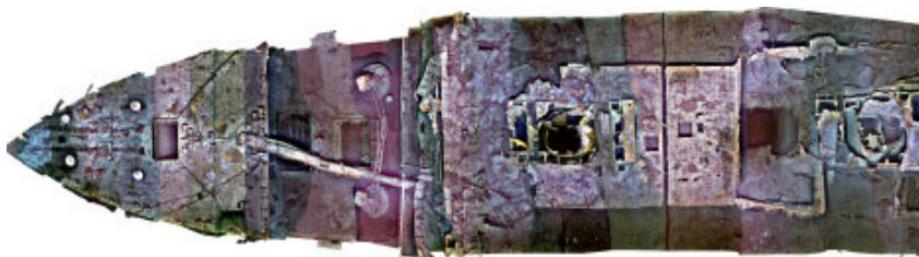
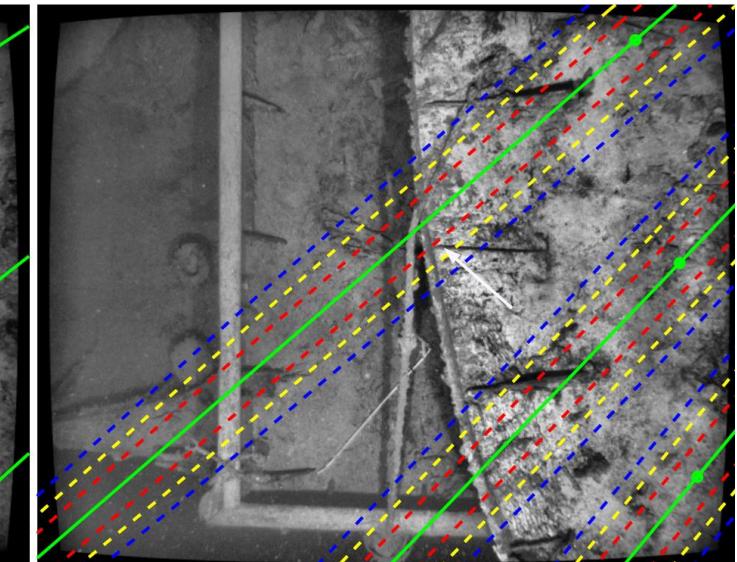
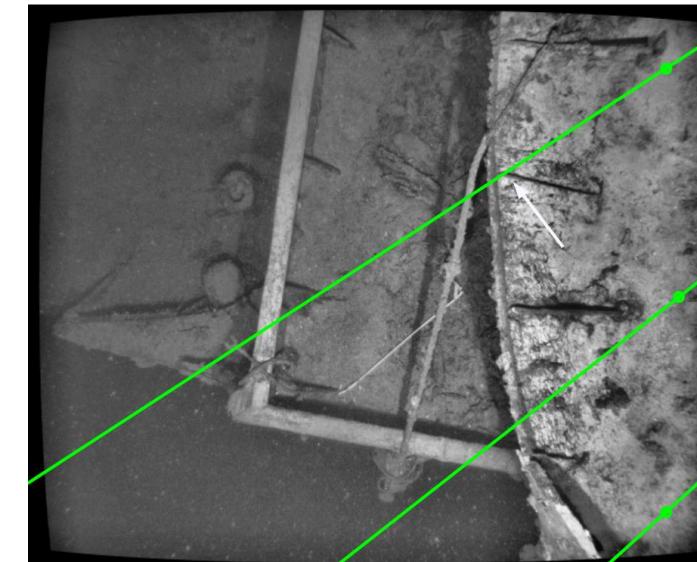
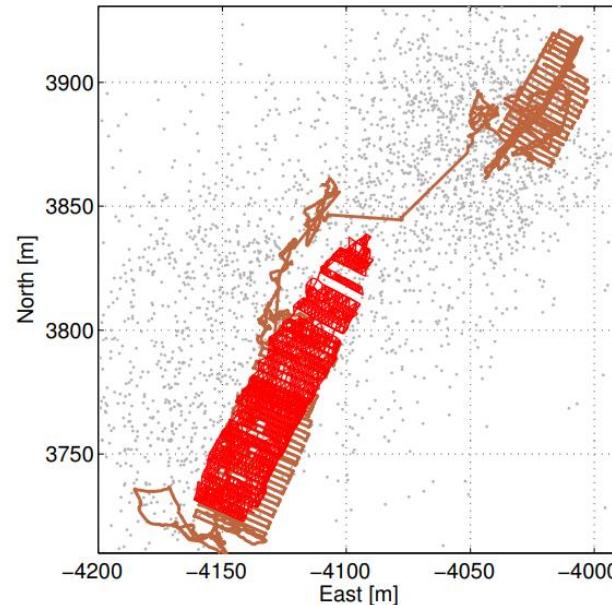
Drones in the Wild



Video from: https://youtu.be/p1d_ptE6yrc



Underwater Robots



Eustice, R., Singh, H., Leonard, J. J., Walter, M. R., & Ballard, R. (2005). *Visually Navigating the RMS Titanic with SLAM Information Filters*. In *Robotics: Science and Systems* (Vol. 2005, pp. 57-64).



Roomba



Video from: <https://youtu.be/tZ0bq-jlg-o>



Video from: <https://youtu.be/oj3Vawn-kRE>

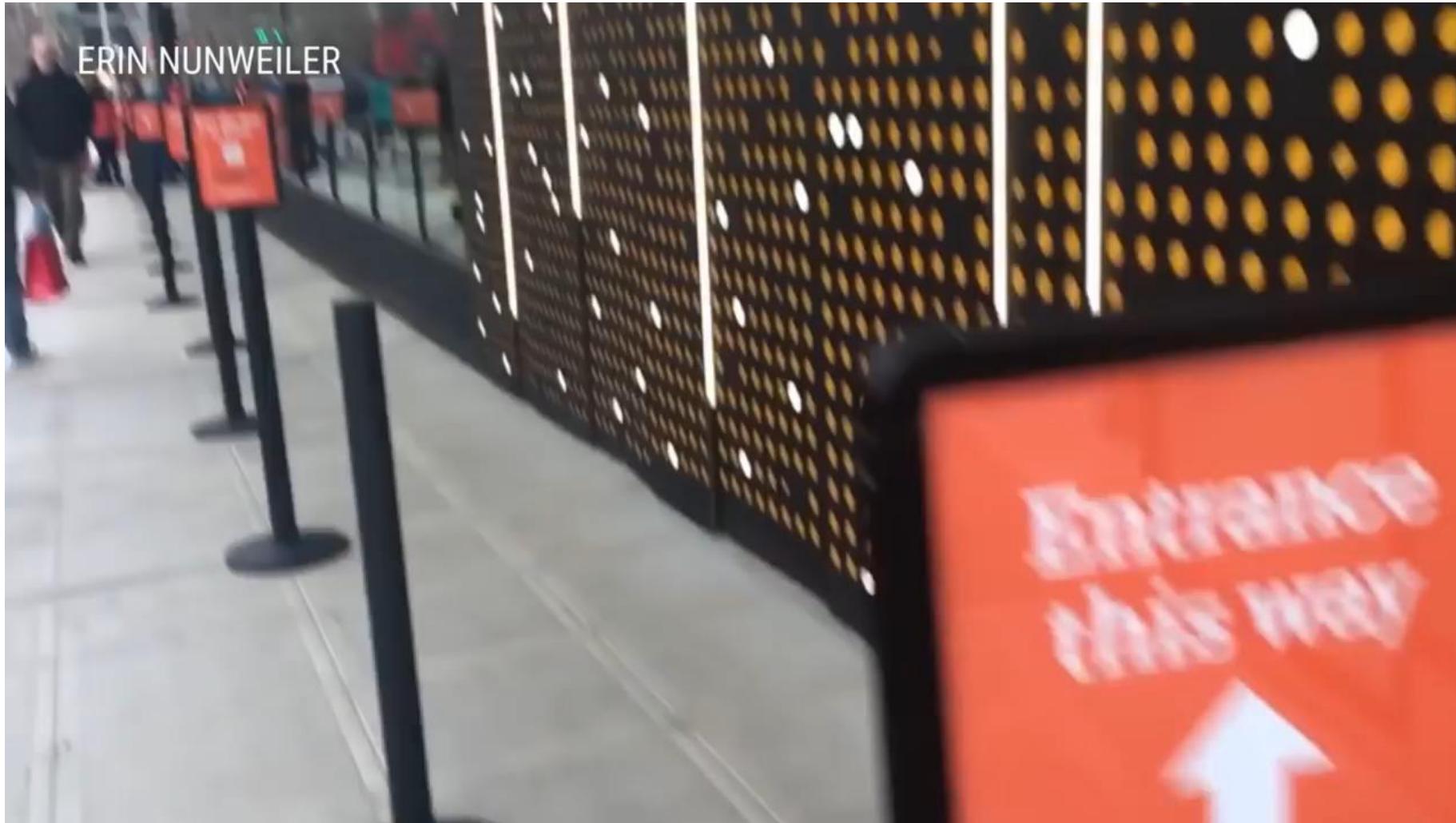
Amazon Robotics Challenge

Cartman: The low-cost Cartesian Manipulator that won the Amazon Robotics Challenge

D. Morrison, A.W. Tow, M. McTaggart, R. Smith, N. Kelly-Boxall, S. Wade-McCue, J. Erskine, R. Grinover, A. Gurman, T. Hunn, D. Lee, A. Milan, T. Pham, G. Rallos, A. Razjigaev, T. Rowntree, K. Vijay, Z. Zhuang, C. Lehnert, I. Reid, P. Corke, and J. Leitner

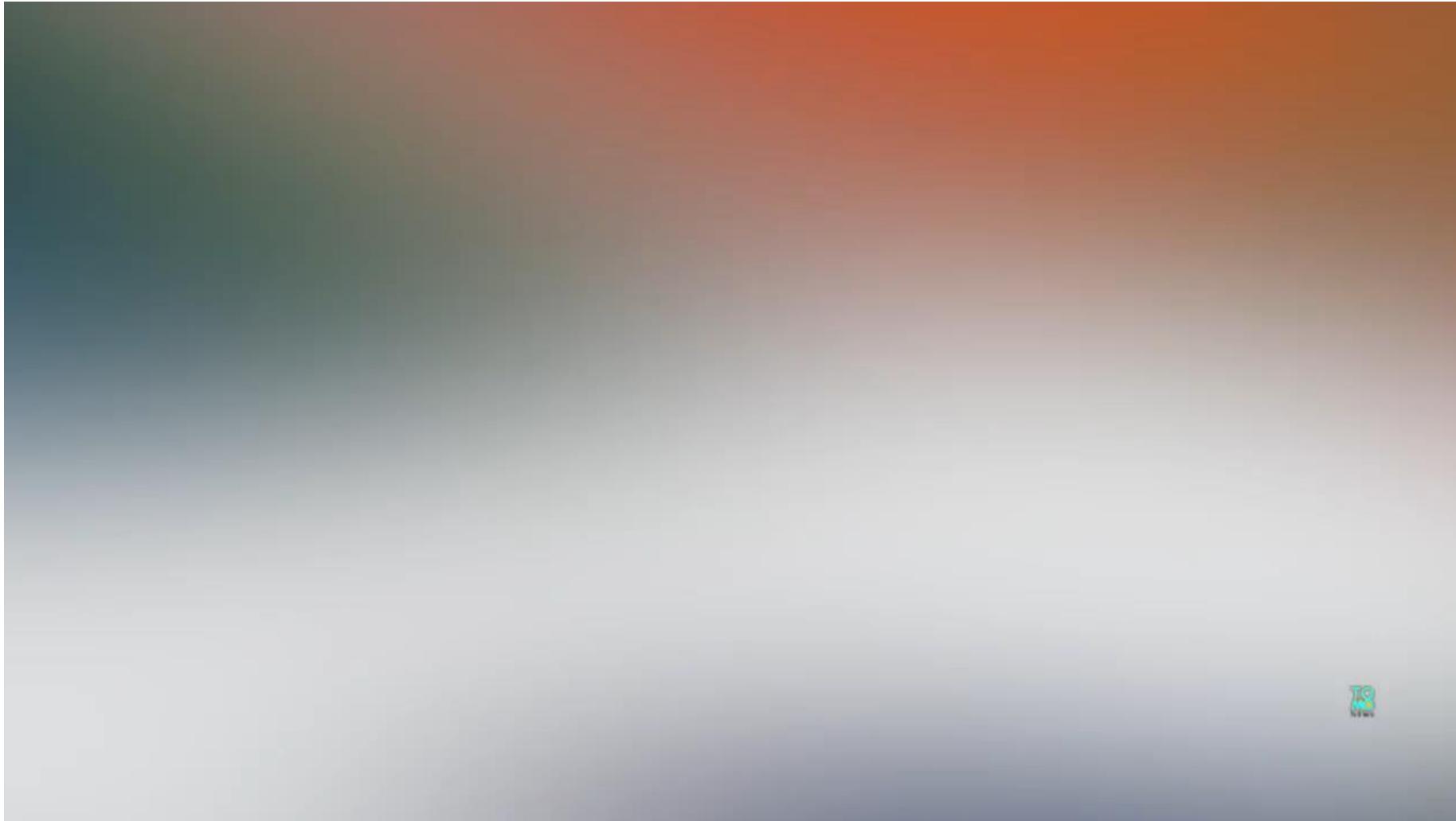


Amazon Go



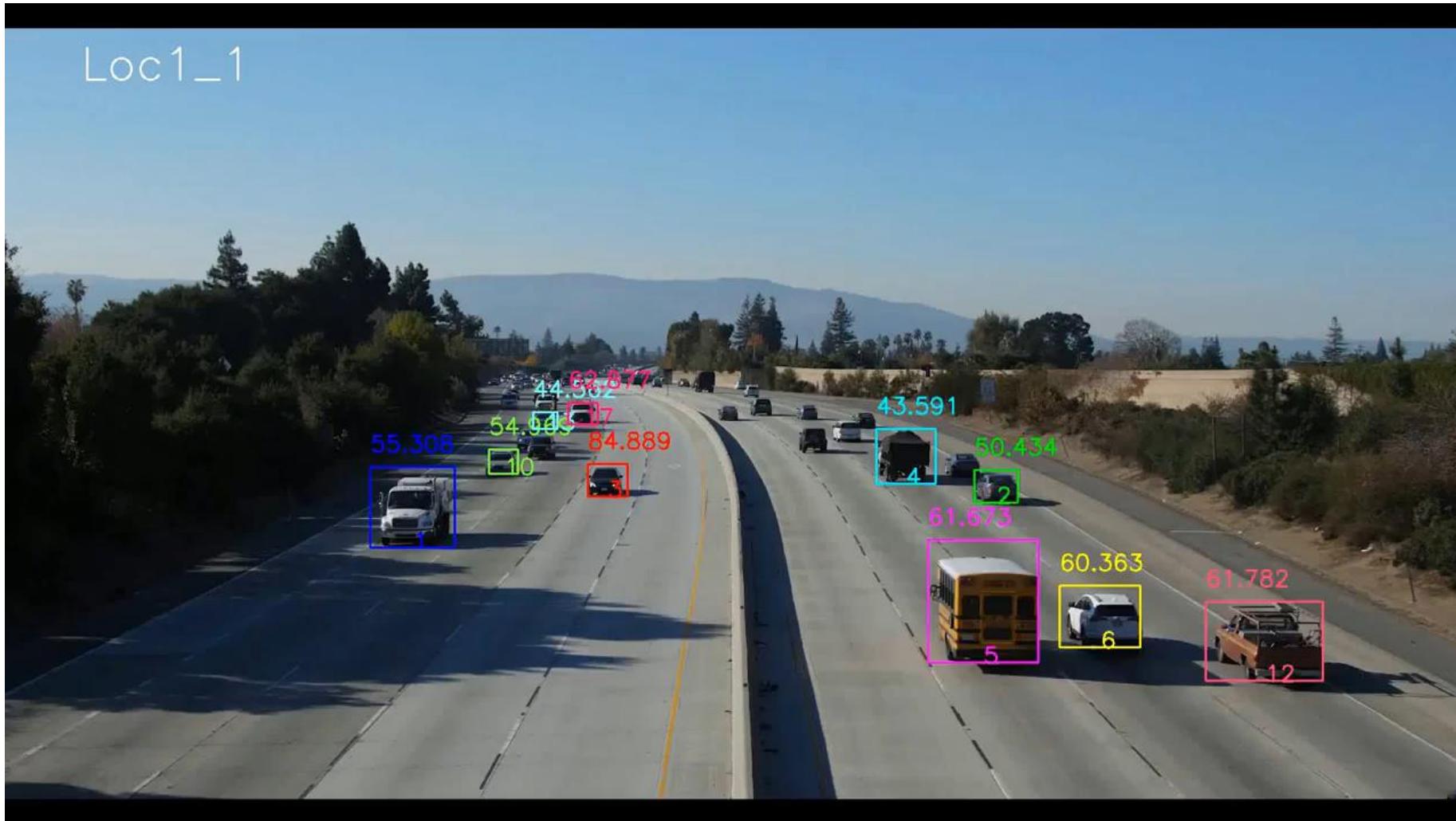
Video from: <https://youtu.be/zdbumR6Bhd8>

Autonomous Driving



Video from: <https://youtu.be/7oCe0aLye-U>

Transportation Surveillance



Video from: https://youtu.be/_i4numqiv7Y

Space Exploration



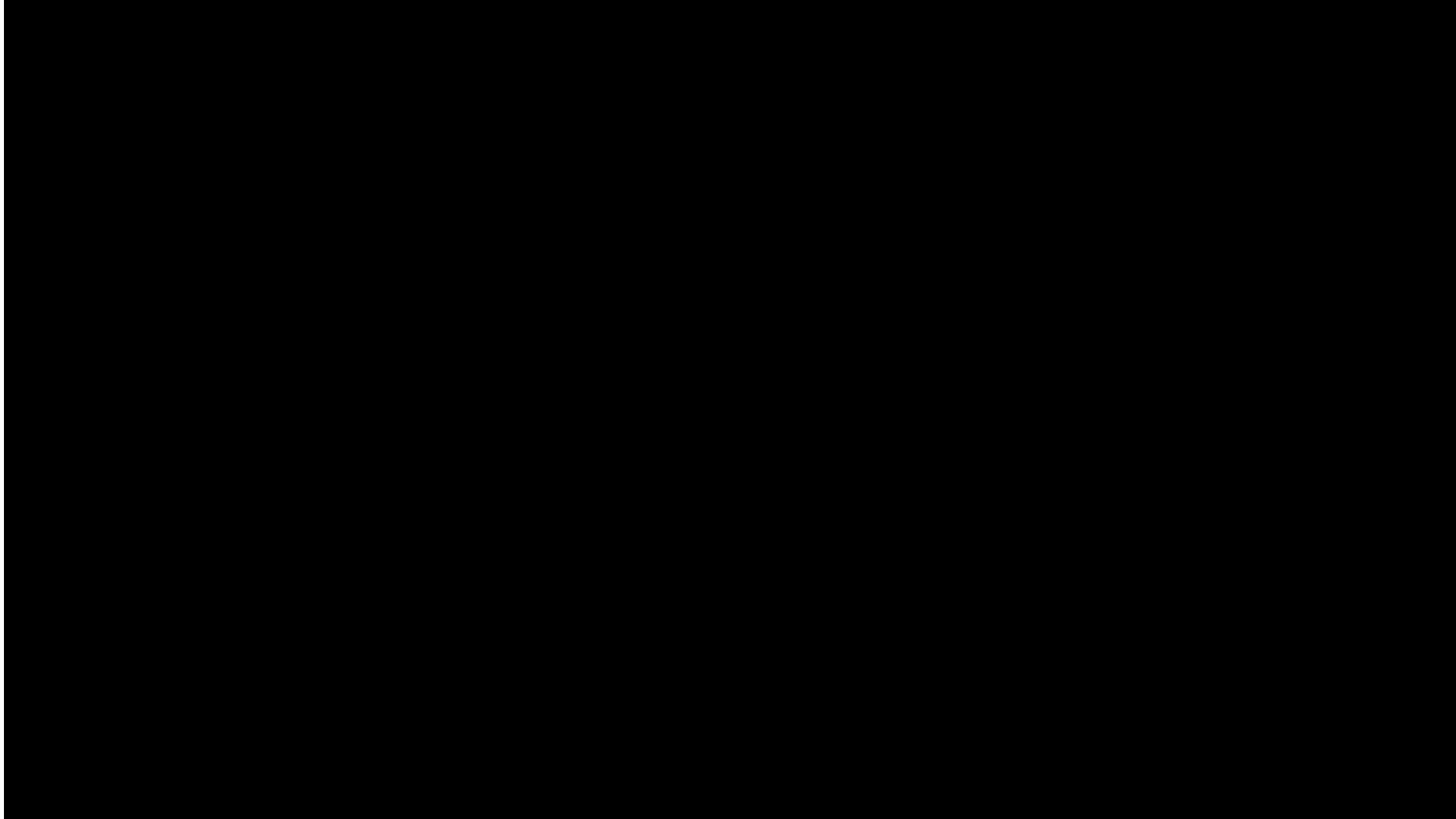
<https://mars.jpl.nasa.gov/mer/gallery/artwork/rover1browse.html>



Video from: <https://youtu.be/kr58r0b5LKM>



Smart Jobsite Surveillance



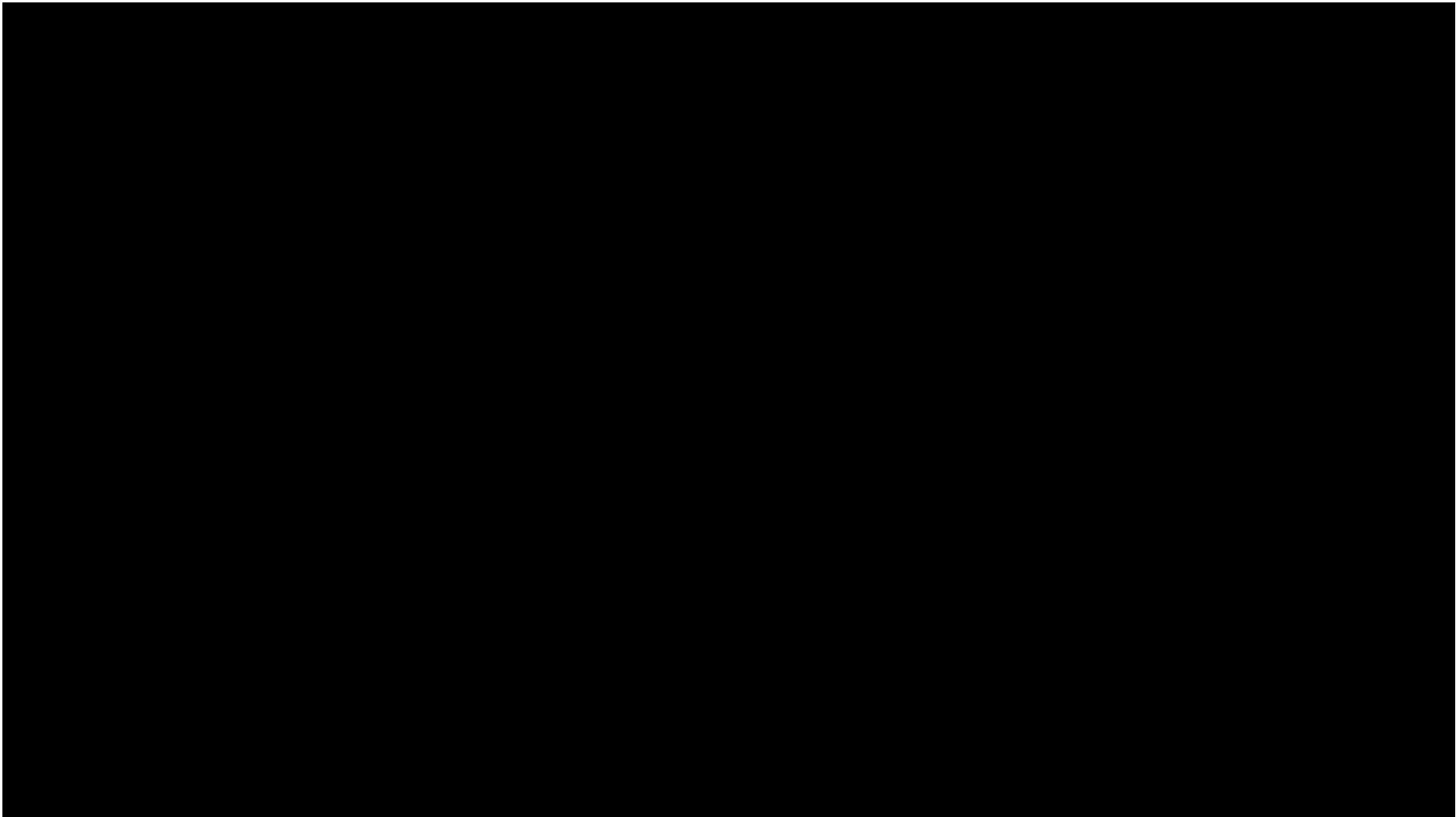
Video from: <https://youtu.be/pL-c00M2CnI>

Robotic Jobsite Scanner



Video from: <https://www.doxel.ai>

Grit Blasting Robot



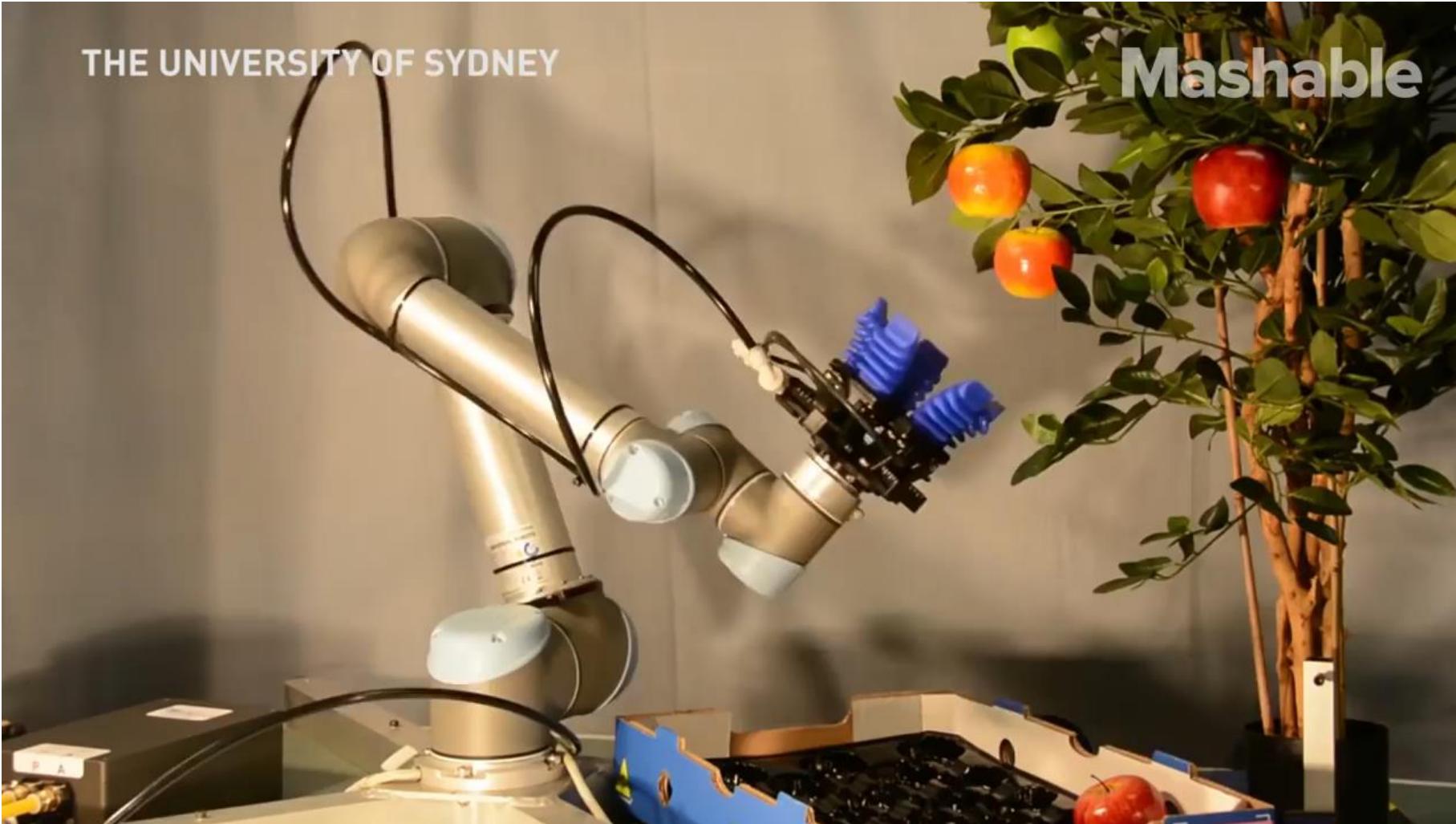
Video from: https://youtu.be/zREgx4NsA_Q

Architecture



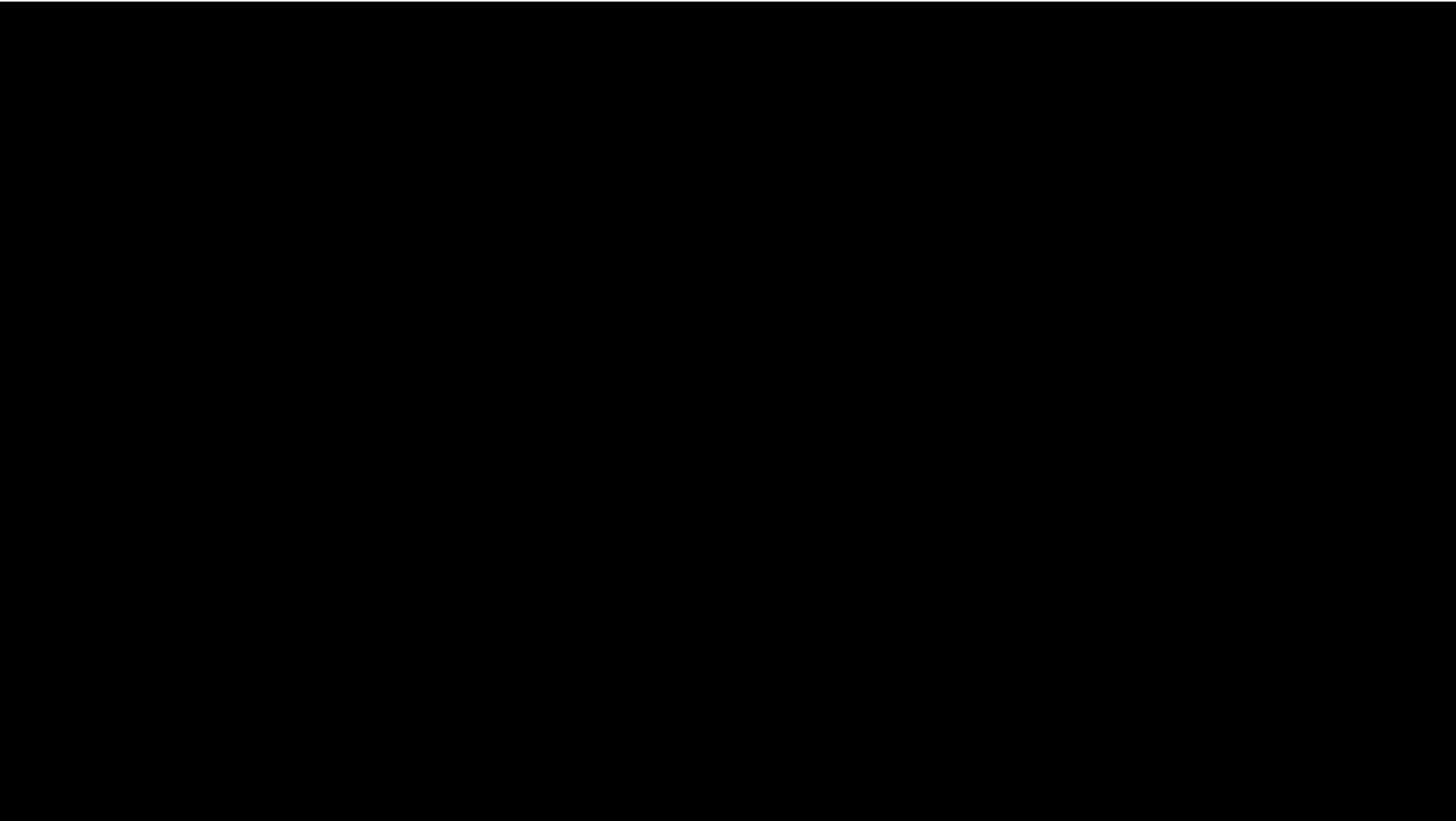
Video from: <https://youtu.be/TCJOQkOE69s>

Agriculture



Video from: <https://youtu.be/NO8PmqEI0cc>

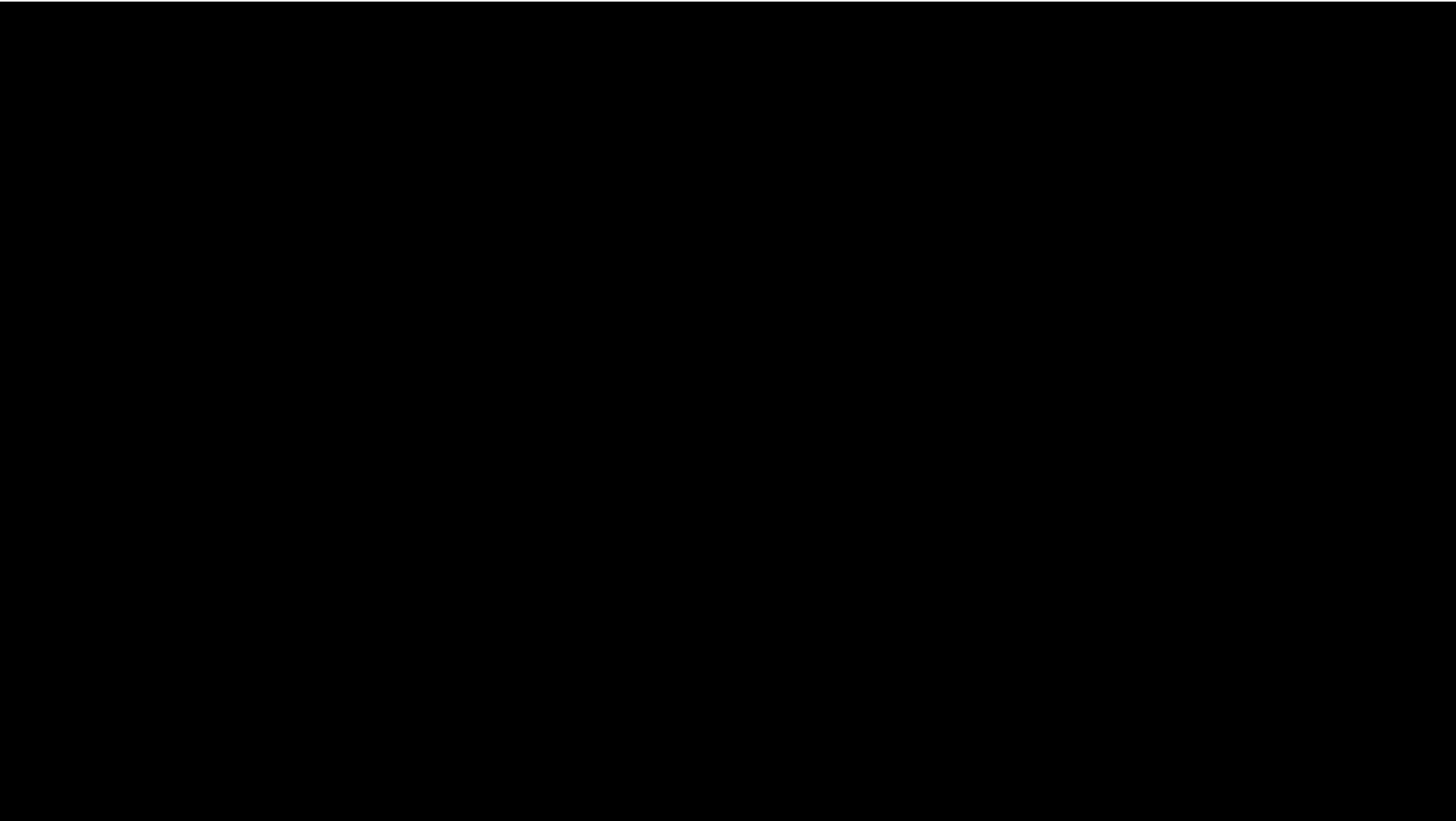
Surgical Robot



Video from: <https://youtu.be/R17lsilRjbM>



Hollywood AR Car



Video from: <https://youtu.be/OnBC5bwV5y0>

Vision for Tactile Sensing

GelSight

Retrographic sensing for touch, texture and shape

Micah K. Johnson, Edward H. Adelson and Alvin Raj

MIT Department of Brain and Cognitive Sciences
MIT Computer Science and Artificial Intelligence Lab



Massachusetts
Institute of
Technology

Presented at SIGGRAPH Emerging Technologies, 2009

LinkedIn job posts



Jet Propulsion Laboratory
California Institute of Technology

Robotics Engineer - New Grad

NASA Jet Propulsion Laboratory · Pasadena, CA, US

Posted 1 day ago · 82 views



2 company alumni work here

Robotics engineers use multi-disciplinary skills to develop robotic systems for space or terrestrial applications. These systems are remotely-controlled or autonomous machines that sense, understand, and take useful actions in unknown and unstructured environments. Typical sub-disciplines include computer vision and perception, multi-sensor estimation and control,

FANUC

Senior Engineer Machine Vision Products

FANUC America Corporation · 3900 W Hamlin Rd, Rochester Hills, MI 48309, US

Posted 1 week ago · 37 views



Robotics Perception Validation Engineer

Volt Workforce Solutions · Pittsburgh, Pennsylvania

Posted 2 weeks ago · 68 views



Be an early applicant



Robotics Research Hardware Engineer (NCG)

NVIDIA · Redmond, WA, US

Posted 5 days ago · 64 views

Save

Apply

experiments. Experience with core robotics areas such as manipulation, vision, control, etc. is a significant plus. Candidates should have deep and gauges, tactile sensing, cameras (depth, binocular, high-speed, high-resolution, calibration, communication, etc.), real-time



Systems QA Robotics Automation Engineer

Apple · Cupertino, CA, US

Posted 2 days ago · 30 views

- Developing and executing testing using computer vision.



Advanced Robotics R&D Engineer Intern Summer 2019

Amazon Robotics · North Reading, MA, US

Posted 1 week ago · 578 views

The Intern Will Be Responsible For

- Identifying creative solutions for challenging research problems in robotics and computer vision



Robot Perception (cfeng@nyu.edu)

BIG CLOUD

Robotics Engineer - SLAM
Big Cloud · Sunnyvale, California
Posted 6 days ago · 28 views

[Save](#) [in Easy Apply](#)

Chris Bradbury can refer you
Get referred to [increase your chances](#) of landing an interview.

[Ask for a referral](#)

Job description
An exciting autonomous driving start-up is expanding its team. They need world-class machine learning researchers to help them solve complex problems in machine learning, computer vision, data engineering and robotics. They just closed an impressive series A and have an incredible technical leadership team.

They're looking for candidates with real-world experience of robot

How you match
Criteria provided by job poster

Skills

- Computer Vision
- Machine Learning
- Robotics
- SLAM

Senior Robotics Engineer
Quest Groups LLC · San Francisco Bay Area
New · Posted 20 hours ago · 75 views

[Save](#) [in Easy Apply](#)

Job description
We are building the future of construction.
Our mission is to take the latest sensors from self-driving cars, retrofit them into proven equipment from the job site, and develop a suite of autonomous software designed specifically for the requirements of construction and earthmoving. And over the last two years, with a team of a dozen dedicated engineers, roboticists, and construction experts, that's what we've done. It hasn't been easy—in fact, no one has ever done what we're doing—but with

Job Description

- As a robotics engineer on our team you'll be designing, developing, testing and improving our autonomous track loader (ATL). You'll be a generalist, doing everything from optimizing controllers to designing high-level planners for new to construction tasks to improving the accuracy of our object recognition systems. You'll have a significant impact on the product road map, and you'll deploy code every day.

Qualifications

- BS, MS or PhD in computer science or related field
- 2+ years of software development experience
- Proficiency in Python
- Experience with distributed systems
- Familiarity with ROS is a plus
- Knowledge of computer **vision** is a plus
- Mechanical engineering experience is a plus, but not required
- Controls experience is a plus, but not required

Compensation & Benefits

- Competitive equity and salary (we believe great engineers are worth it)
- Free catered lunch and snacks in the office
- In-office gym

TMG The Mackenzie Group

Robotics Engineer
The Mackenzie Group, LLC · Detroit, Michigan, United States
Posted 1 week ago · 135 views

[Save](#) [in Easy Apply](#)

Job description
The Mackenzie Group has been retained by a Detroit area automation company to fill a ROBOTICS ENGINEERING position. If you are interested in this exciting opportunity, please respond.

Skills Required/Details:

- Strong and recent background in two of the following robots: Fanuc, ABB, Kuka, or Comau.

How you match
Criteria provided by job poster

Skills

- Computer Vision
- Programming
- Robotics
- Electrical Troubleshooting

Robotics Software Engineer
Fox Robotics · Austin, Texas
Posted 2 days ago · 1,368 views

[Save](#) [in Easy Apply](#)

Job description
Develop navigation and perception capabilities on a mobile robot. You will need to rapidly integrate and evaluate third party libraries, keep abreast of current research, and develop core autonomy capabilities.

Requirements

- strong C++ programming skills
- experience with localization and mapping algorithms
- experience with 3D obstacle avoidance and navigation algorithms

Bonus points for

- proven ability to ship working robotic applications

How you match
Criteria provided by job poster

Skills

- Computer Vision
- C++
- Mobile Robotics
- Machine Learning
- Robotics
- Algorithms
- Localization
- Computer Hardware

Computer Vision Engineer - Robotics
Harnham · Greater Pittsburgh Area
Posted 2 weeks ago · 271 views

[Save](#) [in Easy Apply](#)

Sam Brown can refer you
Get referred to [increase your chances](#) of landing an interview.

Job description
Computer Vision Engineer - Robotics
Pittsburgh, PA
\$150,000 - 180,000 + Benefits + Equity

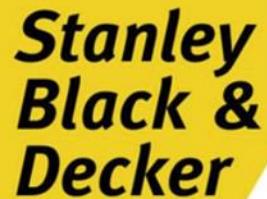
Harnham has been retained by a rapidly growing and well funded start up with over 40 employees globally and 10 million in initial funding. Focusing on localization and mapping as well as image processing, their mission is to develop robotic solutions, allowing robots to better perceive, understand and navigate in both indoor and outdoor environments.

Very much focused on C++ programming, you'll be an integral part of their

How you match
Criteria provided by job poster

Skills

- Research
- Computer Vision
- C++
- Robotics
- Algorithms
- SLAM
- Computer Hardware



Robotics Engineer Intern

Stanley Black & Decker, Inc. · Atlanta, GA, US

Posted 5 days ago · 165 views

Reporting to the Technical Manager of Robotic Systems, the Robotics Engineer engages in design and development of one or more areas related to advanced sensors and sensing technologies, algorithms development for perception, AI, robotic **vision**, control and data analysis, and integration of modern methods of navigation and path planning (specially with application to obstacle avoidance and autonomous operations.)

In this role, the ideal candidate is expected to have the following technical qualifications:

- BSc, MSc or PhD in Mechanical/Electrical/Robotic Engineering or Computer Science preferred, Bachelor's degree with appropriate additional years of experience also acceptable.
- Interest and/or experience developing robots and/or autonomous mobile systems. Hands on design/build experience of robot/UAV systems highly desired.
- Ability to do research in Robotics **vision** and or computational



Amazon Robotics Software Engineer - Autonomous Mobility

Amazon Robotics · North Reading, MA, US

Posted 1 week ago · 136 views

Job description

As an Amazon Robotics software engineer on the Autonomous Mobility team, you will own key components of navigation, **vision**, control, and coordination for single and multi-agent systems. You will collaborate cross-functionally to define clear requirements, deliverables, and test cases in an exciting, highly innovative environment.



Robotics Engineer (LiDAR / Computer Vision / C++)

MoTek Technologies · San Francisco Bay Area

Posted 3 weeks ago · 99 views

[Save](#)

[in Easy Apply](#)



Robotics Computer Vision Engineer

Bastian Solutions · Boise, ID, US

Posted 2 weeks ago · 193 views



Robotics Engineer - Relo to Austin

CyberCoders · New York City, NY, US

Posted 3 days ago · 12 views

[Be an early applicant](#)

Seniority Level
Entry level

Industry
Hospital & Health Care

Employment Type
Full-time

Job Functions
Information Technology

Job description

Robotics Engineer

Location: Austin, TX (relocation assistance)

Salary: \$80,000 - \$120,000 + equity

Top Reasons to Work with Us

- Be part of the founding team.
- Influence the direction of the company and product.
- Experience with robotics
- Chance to live in amazing Austin, Texas!

What You Will Be Doing

- Writing our core software stack
- Have an interest in:
- Web Based Applications
- Navigation for dynamic door environments
- Machine Learning and data visualization
- Manipulation in semi-structured environments
- Computer vision on real systems

What You Need for this Position

- Experience with deploying algorithms on real robots
- ROS



Software Engineer, Researcher / Robotics
MoTek Technologies · Palo Alto, California
Posted 3 weeks ago · 160 views

Save **in Easy Apply**

Jerald Baker can refer you
Get referred to **increase your chances** of landing an interview.

Ask for a

Job description

Software Engineer, Researcher / Robotics - Level 5 Autonomous Driving

Our software allows our vehicles to perceive the world around them, make the right decision for every situation, and deliver people safely to their destinations. We think deeply and solve complex technical challenges in areas like robotics, perception, decision-making and deep learning, while collaborating with hardware and systems engineers. If you're a software engineer or researcher who's curious and passionate about Level 5 autonomous driving, we'd like to meet you.

How you match
Criteria provided by job poster

Skills

- Research
- Computer Vision
- C++
- Machine Learning
- Robotics
- Algorithms

Sensor Fusion
Object Tracking
Computer Hardware
Automotive

Level of education

- Doctor of Philosophy

In this researcher/robotics role you'll:

- Work with world-class experts in the field of autonomous vehicles and advance the state of the art in areas such as computer vision, sensor fusion, machine learning, object tracking, and motion planning.

We'd like you to have:

- MS/PHD in Robotics, Computer Science, or equivalent practical experience.
- Experience in hands-on robotics research and expertise in one or more of the following: computer vision, LIDAR, object tracking, sensor fusion, perception, machine learning, motion planning, and control.
- Experience in data structures and advanced algorithms.
- Experience programming in C++ and with robust, safety-critical, efficient code.
- Experience with field robotics and systems design.

Contact the job poster

 **Jerald Baker** 1st
Technical Recruiter

Message

Job Details



Software Engineer, Researcher/Robotics
Waymo · Ann Arbor, MI, US
Posted 1 week ago · 1,965 views

In This Researcher/Robotics Role You'll

- Work with world-class experts in the field of autonomous vehicles and advance the state of the art in areas such as computer vision, sensor fusion, machine learning, object tracking, and motion planning

Job description

Waymo is the self-driving technology that makes it easy for people to get from point A to point B. Our software allows our vehicles to perceive the world around them, make the right decision for every situation, and deliver people safely to their destinations. We think deeply and solve complex technical challenges in areas like robotics, perception, decision-making and deep learning, while collaborating with hardware and systems engineers. If you're a software engineer or researcher who's curious and passionate about Level 5 autonomous driving, we'd like to meet you.

We'd Like You To Have

- MS/PHD in Robotics, Computer Science, or equivalent practical experience
- Experience in hands-on robotics research and expertise in one or more of the following: computer vision, LIDAR, object tracking, sensor fusion, perception, machine learning, motion planning, and control
- Experience in data structures and advanced algorithms
- Experience programming in C++ and with robust, safety-critical, efficient code
- Experience with field robotics and systems design



Robotics Research Engineer
CyberCoders · Pittsburgh, PA, US
Posted 1 day ago · 8 views

Be an early applicant

NICE

- Experience with distributed computing, inter-processor communication, or low-level networking
- Experience with voting architectures, safety-critical systems, reliable computing, or redundant processing
- Experience in developing spacecraft electronics systems or flight software
- Experience with applications such as GPS-denied navigation, localization, SLAM, computer vision, and deep learning
- Experience with basic systems engineering principles, risks, and requirements development

SP1419: Simulation of Multimodal Sensors

MERL is seeking a motivated intern to assist in generating simulated multimodal data for machine learning applications. The project involves integrating several existing software components to generate optical and radar data in a variety of sensing scenarios, and executing the simulations under a variety of conditions. The ideal candidate should have experience with C++, Python, and scripting methods. Some knowledge or experience with Blender, computer graphics, and computer vision would be preferred, but is not required. Project duration is flexible in the range of 1-2 months. Intern has the choice of part-time or full-time occupation and may start immediately.

Research Areas: **Artificial Intelligence, Computer Vision, Signal Processing**

Host: **Petros Boufounos**

[APPLY NOW](#)

CV1462: Graph neural networks for computer vision

MERL is seeking a motivated intern to conduct research on graph neural networks and computer vision. The ideal candidate should have experience with Pytorch/Tensorflow, and scripting methods. Some knowledge or experience with graph neural networks would be preferred, but is not required. Project duration is flexible in the range of 2-4 months. Intern has the choice of part-time or full-time occupation and may start immediately. Candidates in their junior or senior years of a Ph.D. program are encouraged to apply.

Research Areas: **Computer Vision, Signal Processing**

Host: **Siheng Chen**

[APPLY NOW](#)



Lyft logo

PhD Research Intern, Computer Vision/LiDAR (Fall 2019)

Lyft • Palo Alto, CA, US

Responsibilities

- You and your team will work on core perception algorithms such as sensor calibration, objection detection, tracking, segmentation, and state space estimation.
- You will build indoor and outdoor calibration algorithms for camera, LiDAR, IMU and radar.
- Develop segmentation and classification algorithms on LiDAR point cloud data.
- Implement state-of-the-art CV models based on latest publications in computer vision, perception and machine learning.
- Develop sensor fusion algorithms for radar, LiDAR, and vision modalities.
- Participate in team advancement of tools and infrastructure to evaluate the performance of perception stack and track it over time.

Software Engineering Intern – Computer Vision & Deep Learning



NVIDIA
Santa Clara, CA

[Apply on HiredInn](#)

Over 1 month ago • Internship

NVIDIA's invention of the GPU in 1999 sparked the growth of the PC gaming market, redefined modern computer graphics, and revolutionized parallel computing. More recently, GPU deep learning ignited modern AI—the next era of computing—with the GPU acting as the brain of computers, robots, and self-driving cars that can perceive and understand the world. Today, we are increasingly known as "the AI computing company." We're looking to grow our company, and build our teams with the smartest people in the world. Join us at the forefront of technological advancement! We are currently seeking a software engineer intern with strong Computer Vision and Deep Learning fundamentals and robust C skills to contribute to the development of RTX Broadcast engine - a comprehensive suite of SDKs and libraries that enable AI-driven broadcast features. What you'll be doing: You will work alongside brilliant engineers on core technologies, and implement and optimize software to solve challenging...

[READ MORE](#)

[Report this listing](#)

Image Formation

- + Digital Image & Camera Selection
 - * Pinhole Camera Model
 - ++ Projective Geometry Primitives
 - ++ Transformations
 - + Lens Distortion

*: know how to code
++: know how to derive
+: know the concept

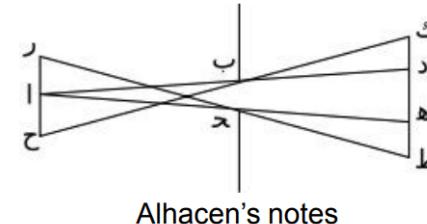
References

- FP2011:
 - Chapter 1, Section 3.1, 3.4, 3.5
- Sz2011:
 - Section 2.1, 2.3, 2.3.2, 6.3.5
- Co2011:
 - Chapter 2, Section 11.1.4
- HZ2003:
 - Section 2.2, 2.3, 2.4, 3.1, 3.4, 6.1
- Linear algebra:
 - Sz2011: section A.1.1, A.1.2, A.1.3, A.2, A.2.1
 - HZ2003: A5.1, A5.2, A5.3

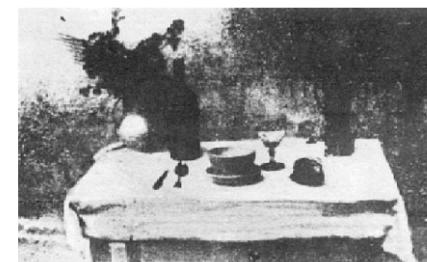


Historic Milestones of Camera

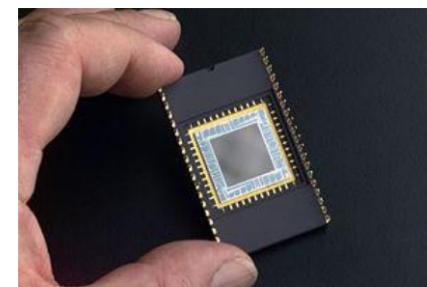
- **Pinhole model:** Mozi (470-390 BCE), Aristotle (384-322 BCE)
- **Principles of optics (including lenses):** Alhacen (965-1039 CE)
- **Camera obscura:** Leonardo da Vinci (1452-1519), Johann Zahn (1631-1707)
- **First photo:** Joseph Nicephore Niepce (1822)
- **Daguerreotypes** (1839)
- **Photographic film** (Eastman, 1889)
- **Cinema** (Lumière Brothers, 1895)
- **Color Photography** (Lumière Brothers, 1908)
- **Television** (Baird, Farnsworth, Zworykin, 1920s)
- **First consumer camera with CCD:** Sony Mavica (1981)
- **First fully digital camera:** Kodak DCS100 (1990)



Alhacen's notes



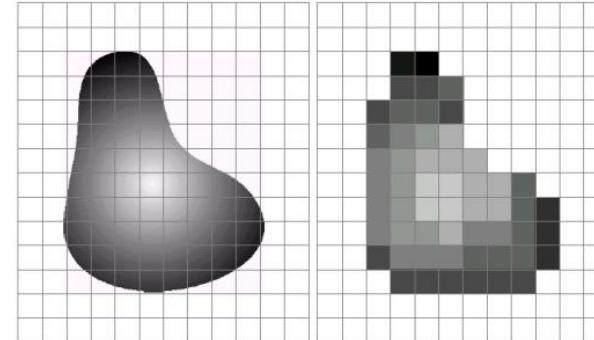
Niepce, "La Table Servie," 1822



CCD chip



Digital Camera



a b

FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

- A digital camera replaces film with a sensor array
 - Each cell in the array is light-sensitive diode that converts photons to electrons
 - Two common types
 - Charge Coupled Device (CCD)
 - Complementary metal oxide semiconductor (CMOS)



Digital Image



45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

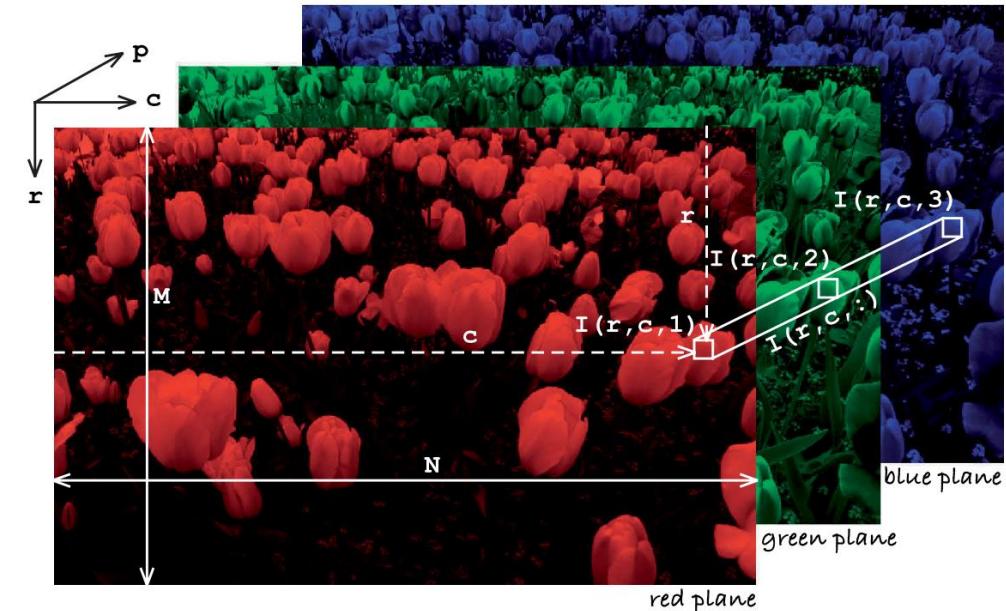
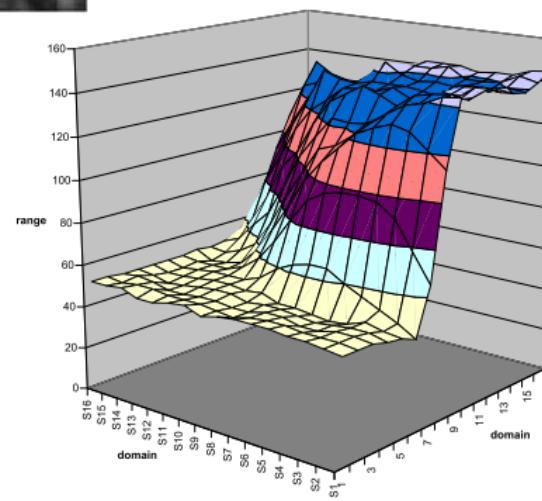


Image from Sz2011, Co2011; A good read: <https://www.scratchapixel.com/lessons/digital-imaging/digital-images/image-file.html>

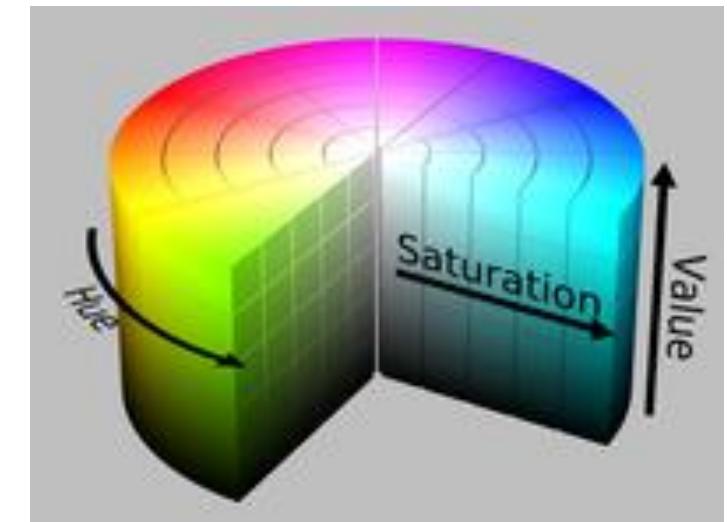
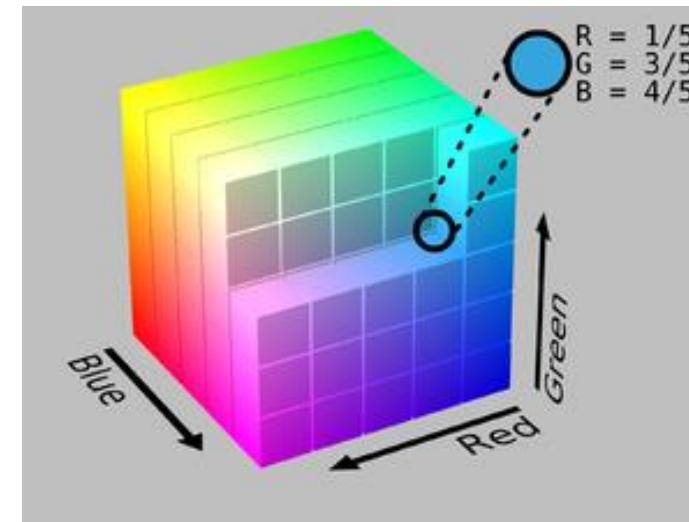
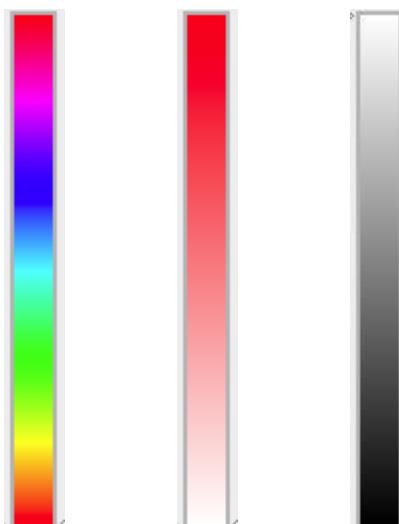


Color Space Transformations

- RGB2Gray

$$\text{RGB}[A] \text{ to Gray: } Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$
$$0.2126R + 0.7152G + 0.0722B$$

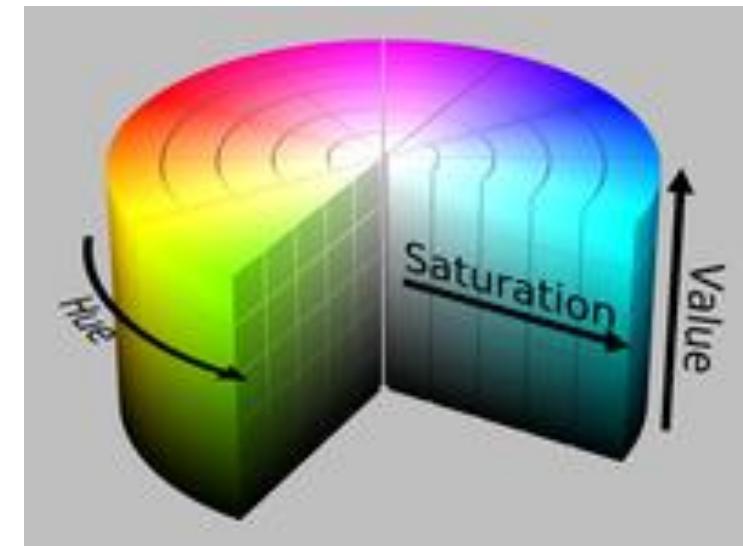
- RGB2HSV





Color Space Transformations

- Higher values are shown as white pixels
- Lower values are shown as darker pixels
- Which one of b & c represents the Hue channel?



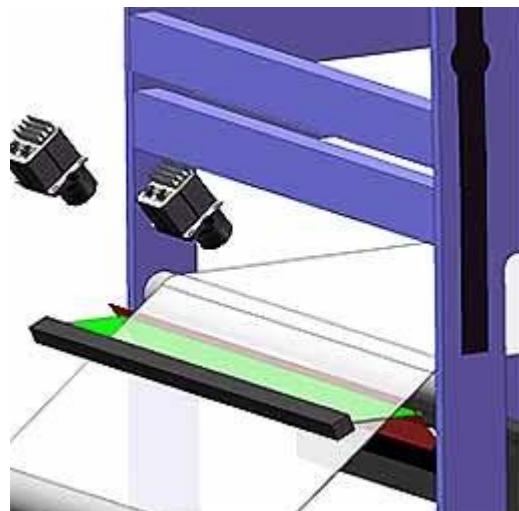
Hue

Saturation

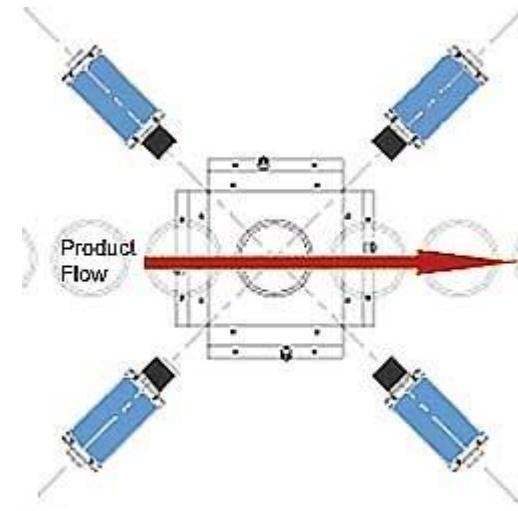
Choosing Machine Vision Cameras

- Area scan or Line scan?
- Color or Monochrome?
- CMOS or CCD?
- Global or Rolling Shutter?
- Frame Rate?
- Resolution?
- Connection Interface?
- Lens Focal length

Area Scan or Line Scan?

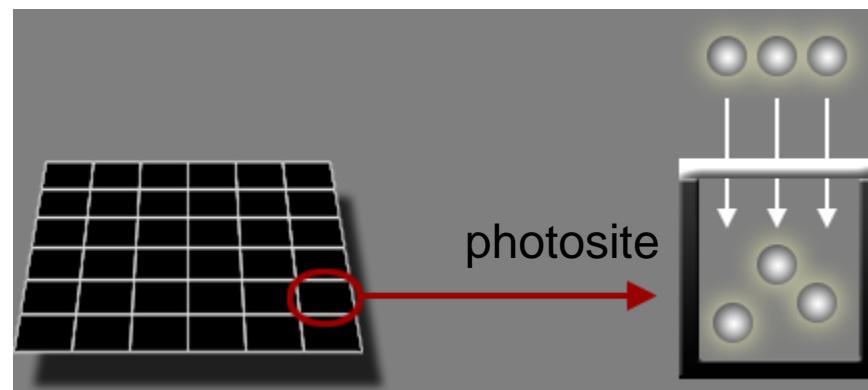


Line scan

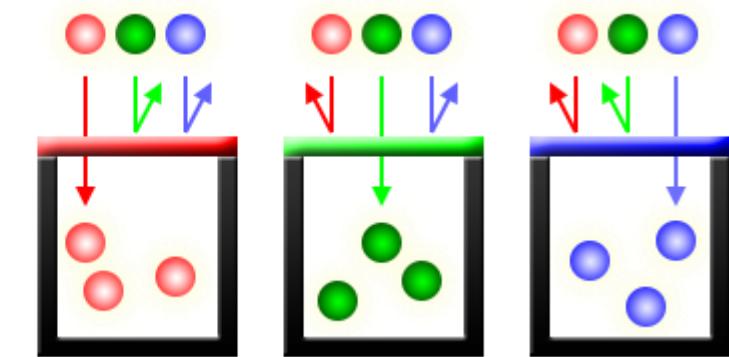
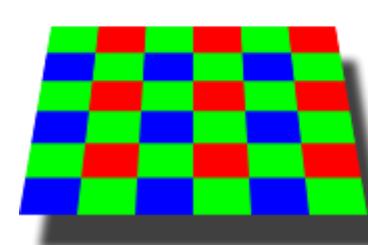
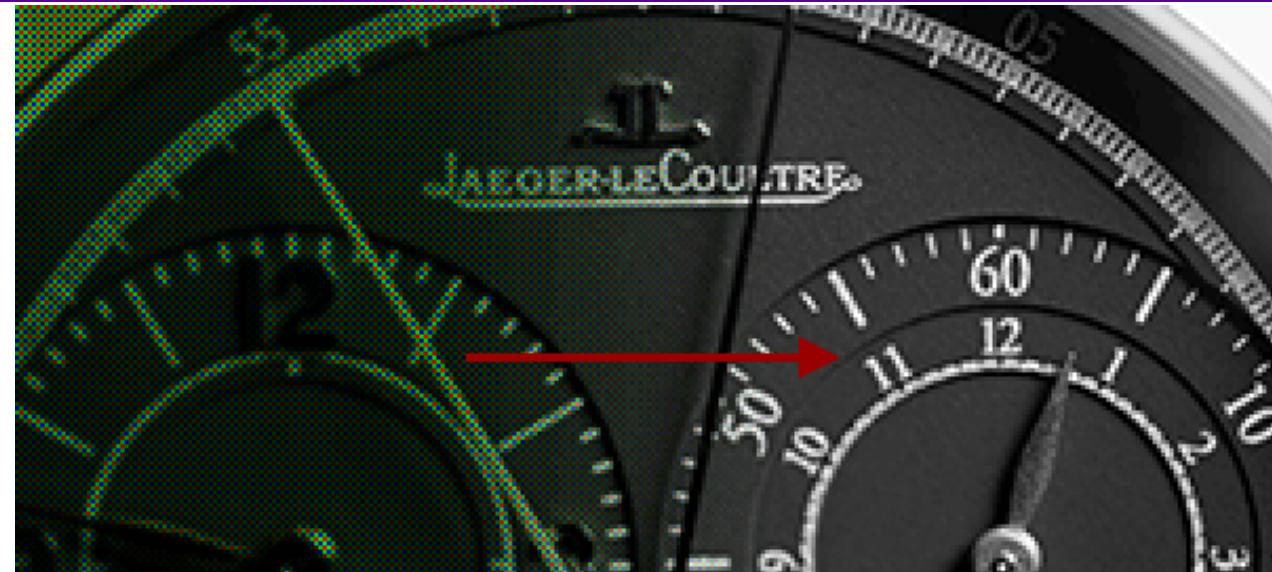


Area scan

Color or Monochrome?



Monochrome sensor



Color sensor (with color filter array)

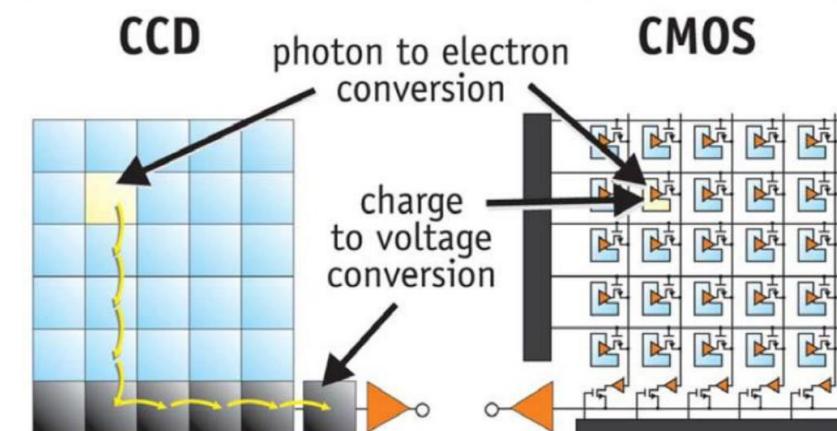
Images from <http://www.red.com/learn/red-101/color-monochrome-camera-sensors>



CMOS or CCD?

- CCD: charge coupled device
- CMOS: complementary metal oxide semiconductor
- Pros/Cons
 - Noise: CMOS > **CCD**
 - Light sensitivity: CMOS < **CCD**
 - Pixel quality: CMOS < **CCD**
 - Power consumption: **CMOS** < CCD
 - Speed/frame-rate: **CMOS** > CCD
 - Price: **CMOS** < CCD
- Current winner: CMOS

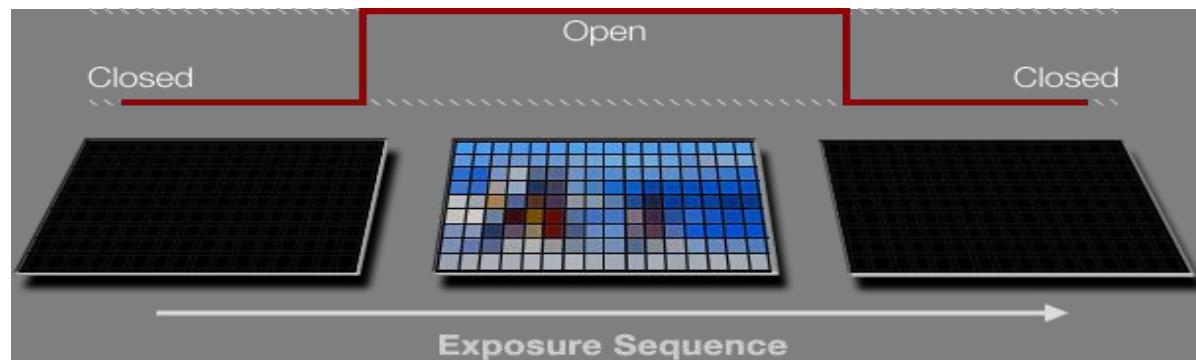
<http://electronics.howstuffworks.com/digital-camera.htm>



CCDs move photogenerated charge from pixel to pixel and convert it to voltage at an output node. CMOS imagers convert charge to voltage inside each pixel.

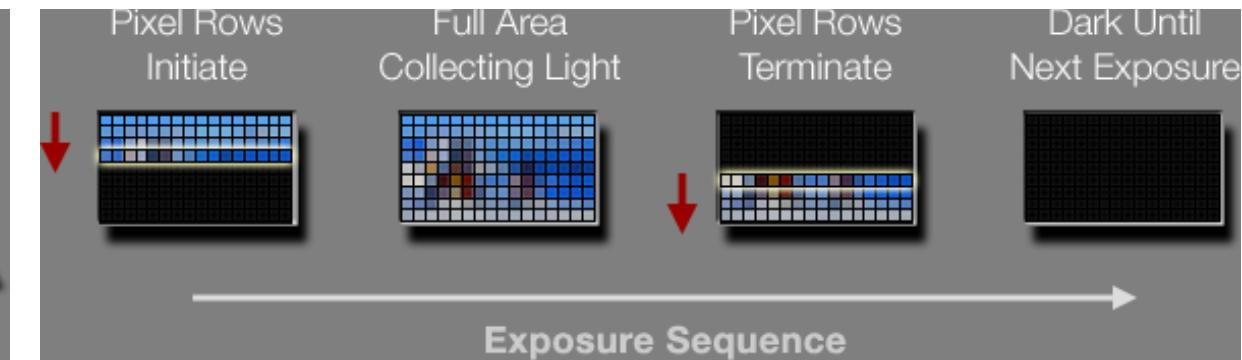
http://www.dalsa.com/shared/content/pdfs/CCD_vs_CMOS_Litwiller_2005.pdf

Global or Rolling Shutter



Global Shutter

Image from <http://www.red.com/learn/red-101/global-rolling-shutter>



Rolling Shutter



Effect of Rolling Shutter with Fast Motion

- Is the photographer moving towards left, or right?
 - If rolling shutter moves from the top to bottom of the image



Image from <https://www.bhphotovideo.com/explora/video/tips-and-solutions/rolling-shutter-versus-global-shutter>

Other Factors

- Resolution
 - QVGA (320x240), VGA (640x480)
 - Choose based on your application!
- Frame rate
 - 15Hz, 30Hz, 60Hz
- Connection interface
 - USB3.0: **350MB/s**, up to 8m cable (power + data), plug-and-play
 - GigE: 100MB/s, up to **100m** cable, good for multiple cameras
- Focal length
 - Fixed: good for geometric vision (3D reconstruction/pose estimation)
 - Auto-focus



Interface	Cable Lengths	Bandwidth max. in MB/s.	Multi-Camera	Cable Costs	“Real-time”	“Plug & Play”
	100 m	100				
	8 m	350				
	10 m	850				

Pinhole Camera Model



Image from <http://thedelightsofseeing.blogspot.com/2010/10/pinhole-photography-and-camera-obscura.html>



Evolution of Pinhole Camera Model

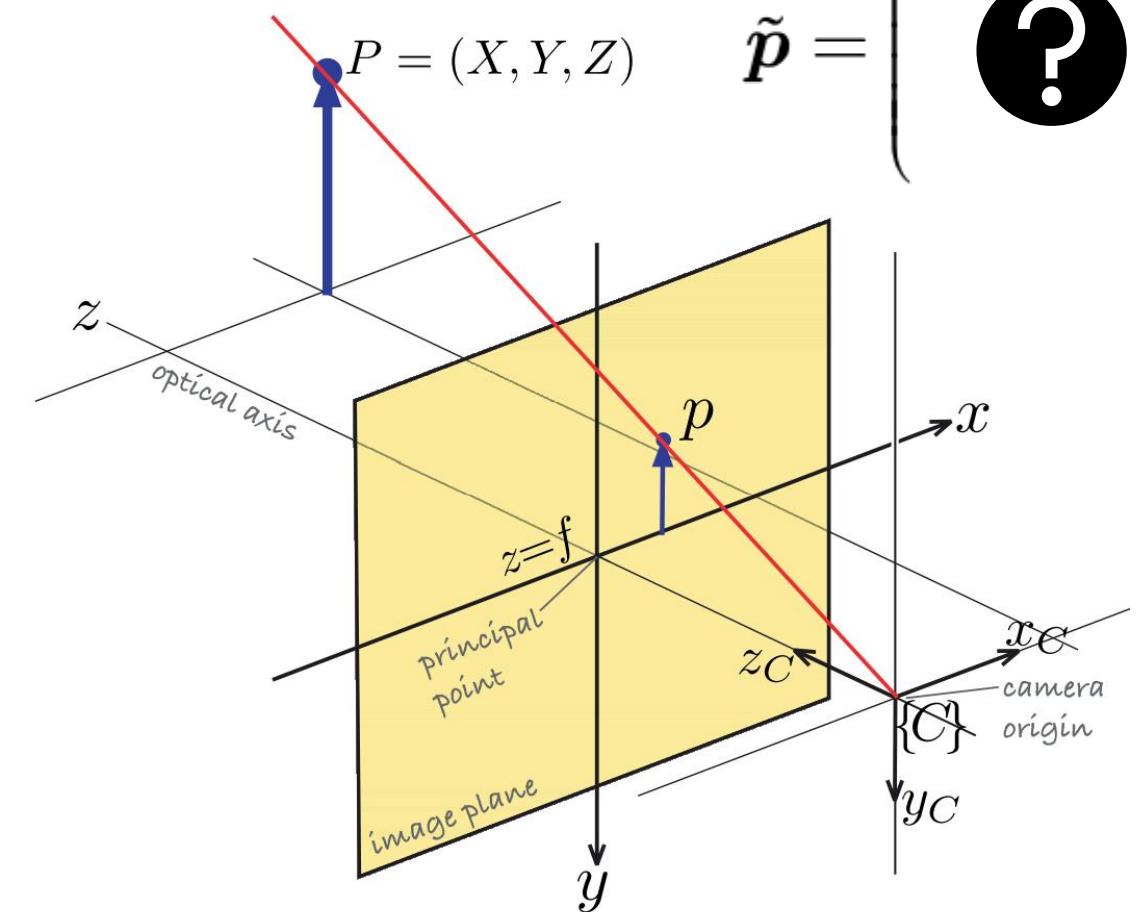
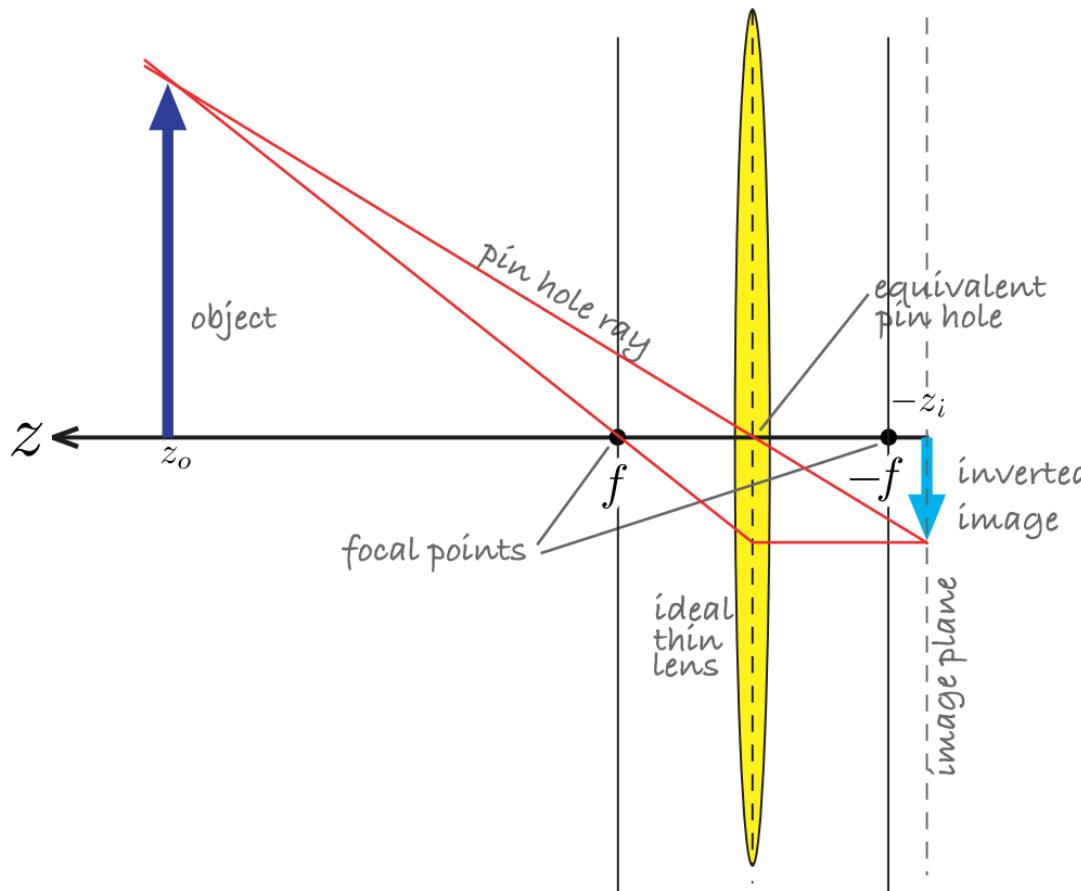
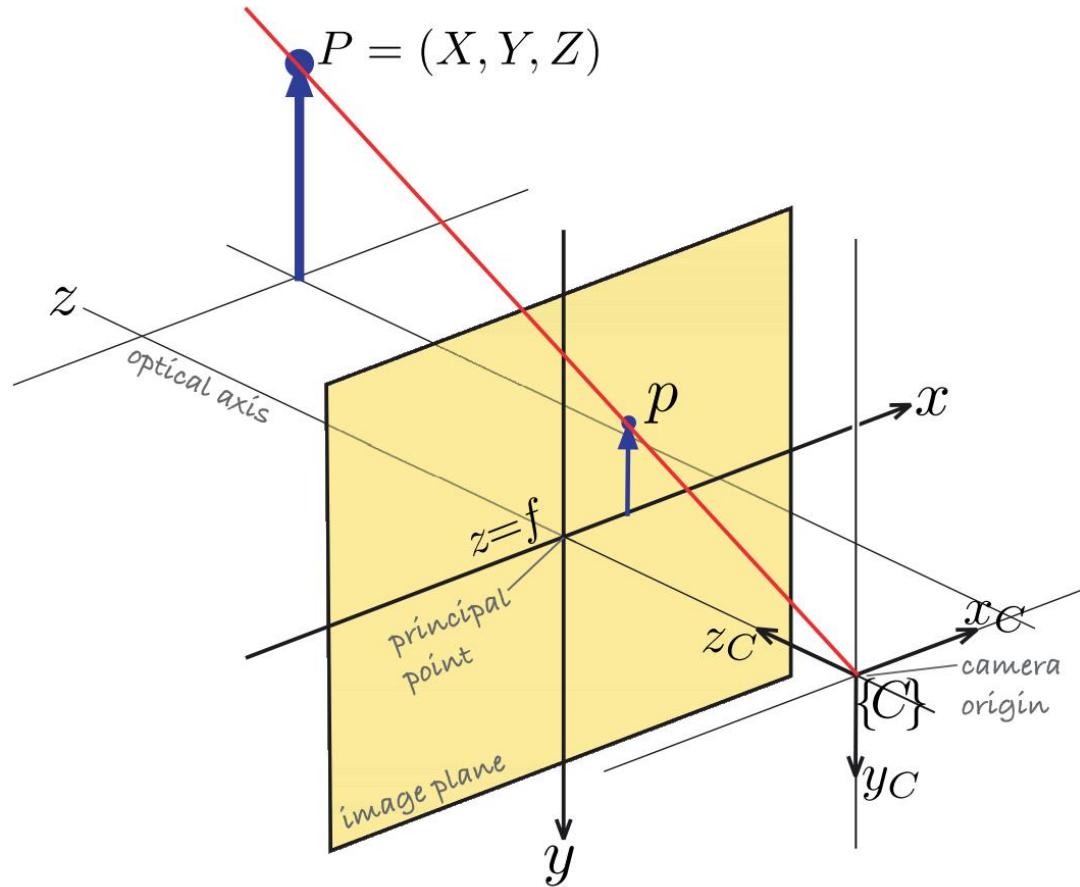


Image from Corke 2011



Evolution of Pinhole Camera Model



$$\tilde{\mathbf{p}} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\tilde{\mathbf{p}} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} {}^C \tilde{\mathbf{P}}$$



Evolution of Pinhole Camera Model

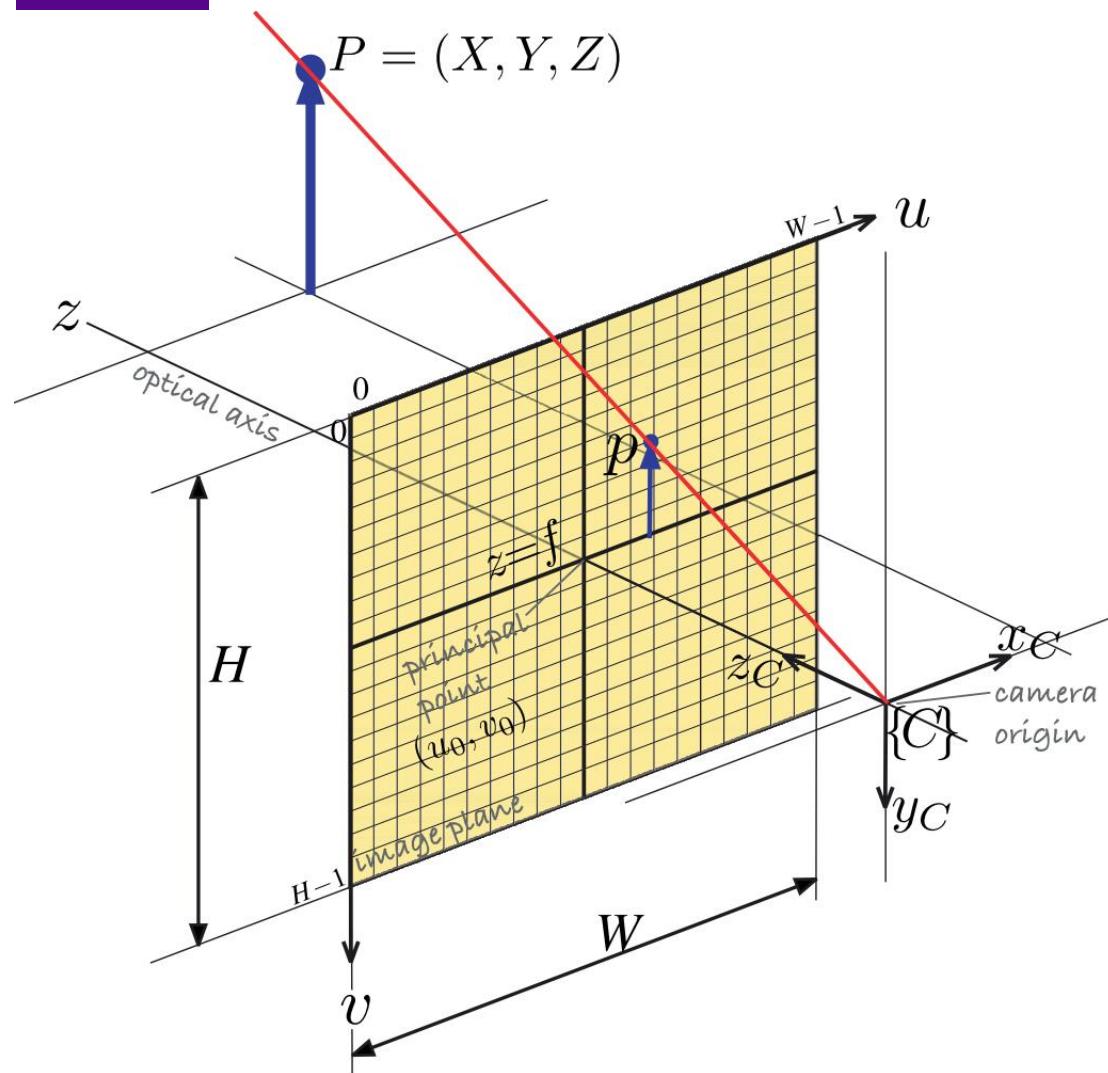


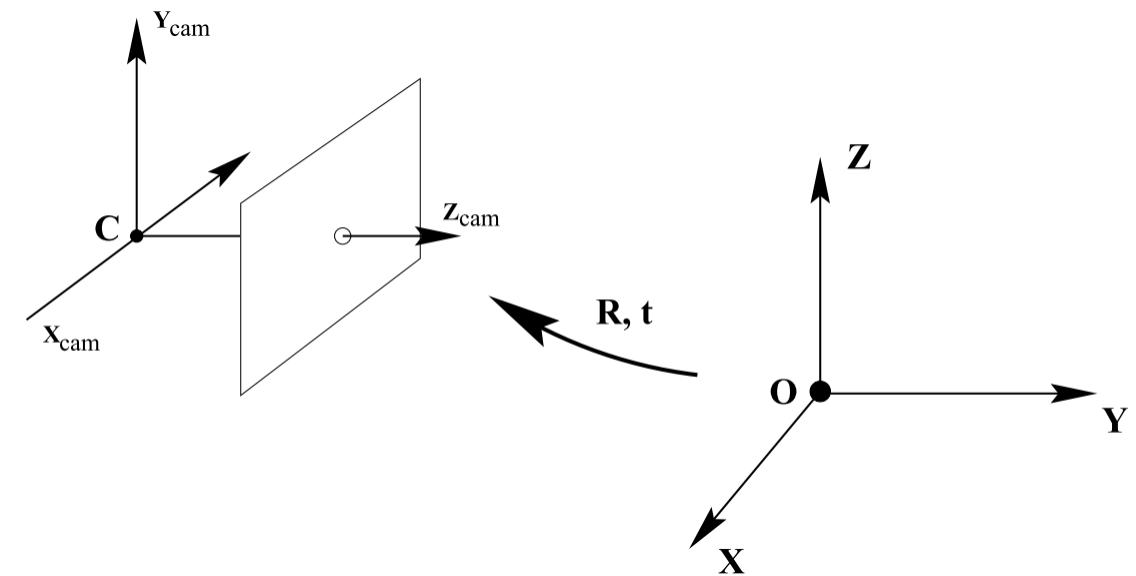
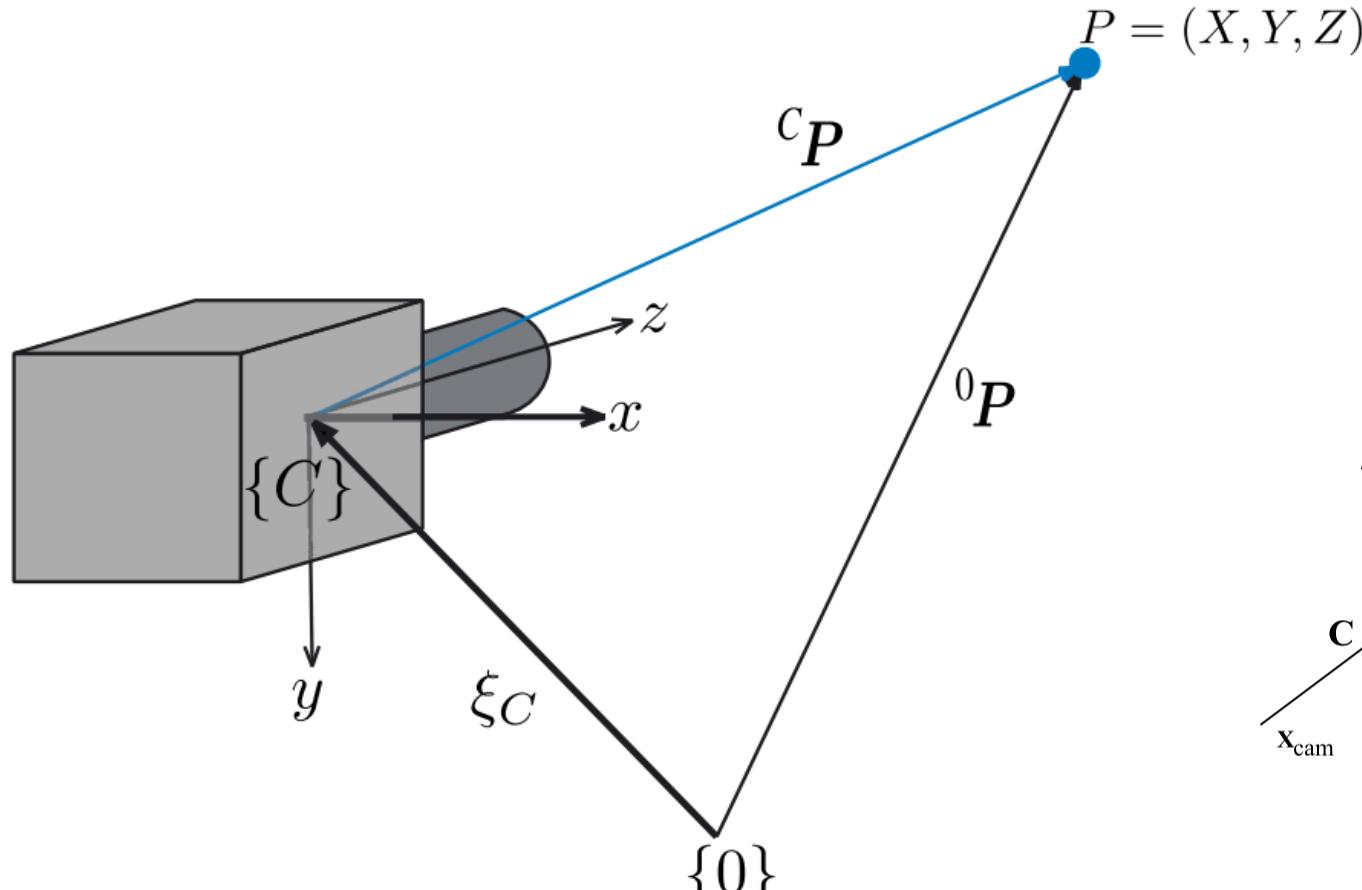
Image from Corke 2011

$$\tilde{\mathbf{p}} = \underbrace{\begin{pmatrix} 1/\rho_w & 0 & u_0 \\ 0 & 1/\rho_h & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_K \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}^C \tilde{\mathbf{P}}$$

ρ_w and ρ_h are the width and height of each pixel



Camera Coordinate Frames



Camera Matrix

- Putting everything together

$$\text{OpenCV} \quad K \begin{bmatrix} R \mid t \end{bmatrix}$$

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{aligned}
 \tilde{\mathbf{P}} &= \underbrace{\begin{pmatrix} f/\rho_w & 0 & u_0 \\ 0 & f/\rho_h & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{intrinsic}} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{extrinsic}} (\mathbf{T}_C^0)^{-1} \tilde{\mathbf{P}} \\
 &= K \mathbf{P}_0^0 \mathbf{T}_C^{-1} \tilde{\mathbf{P}} \\
 &= \mathcal{C} \tilde{\mathbf{P}}
 \end{aligned}$$

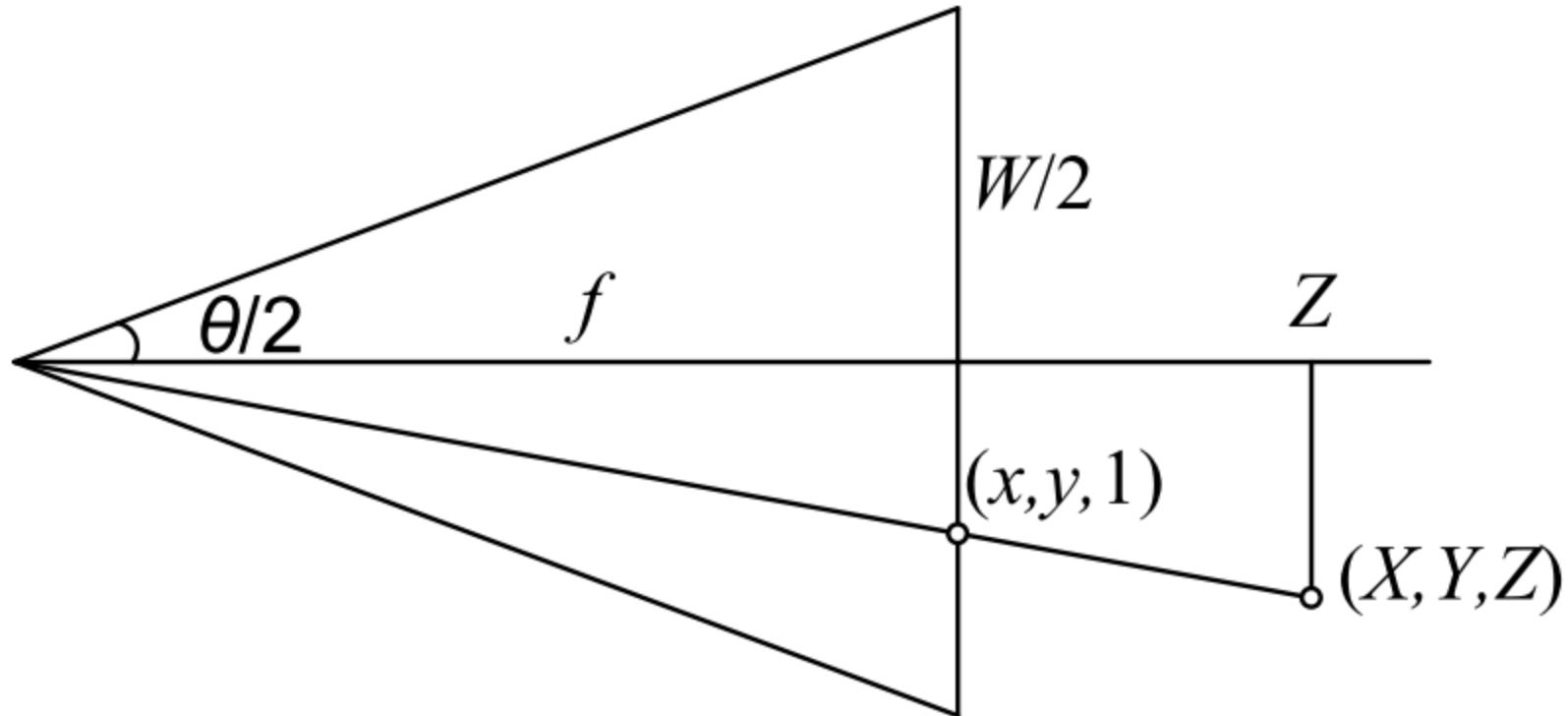
Simplifying Calibration Matrix K

Skew of un-orthogonal u- and v- axis

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{K} = \begin{bmatrix} f & s & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

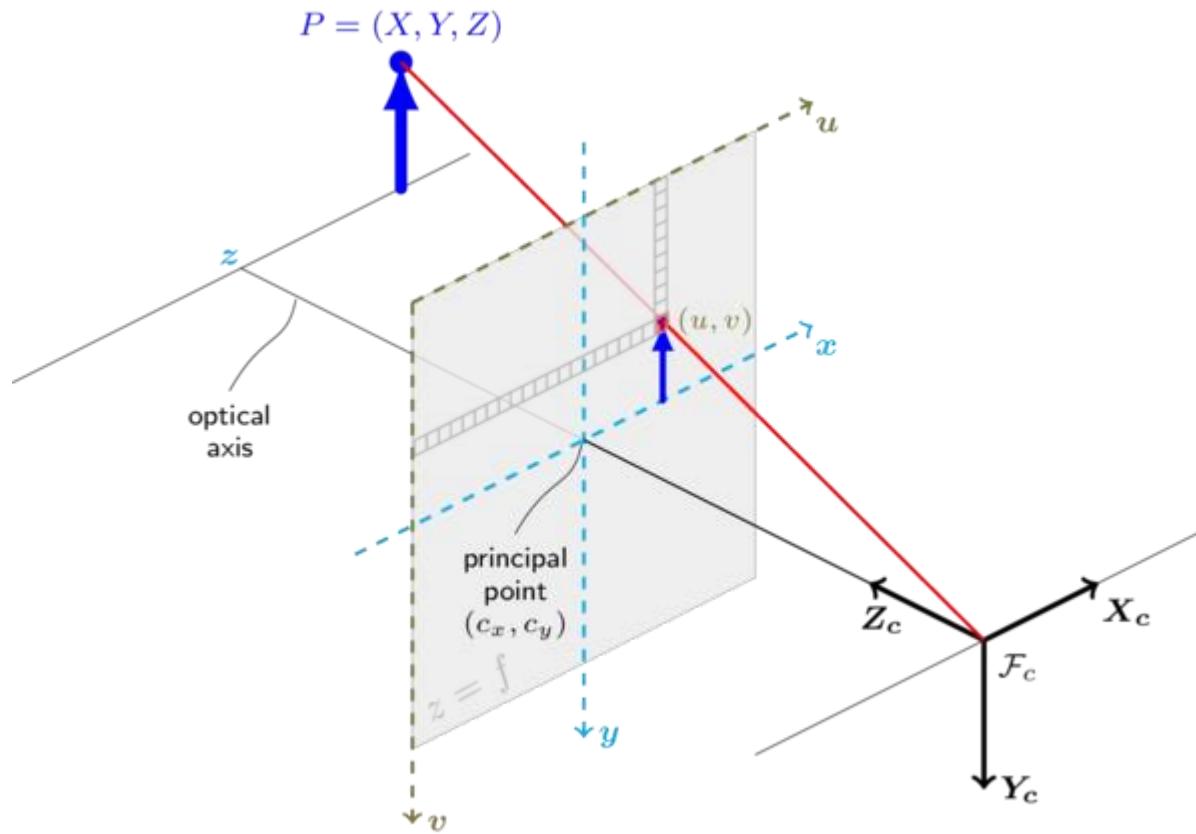


The Unit of Focal Length



$$\tan \frac{\theta}{2} = \frac{W}{2f} \quad \text{or} \quad f = \frac{W}{2} \left[\tan \frac{\theta}{2} \right]^{-1}$$

Putting Everything Together in OpenCV



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$
$$x' = x/z$$
$$y' = y/z$$
$$u = f_x * x' + c_x$$
$$v = f_y * y' + c_y$$



Representing Geometric Primitives in Homogenous Coordinates

- Non-homogeneous/Inhomogeneous point coordinates

$$\mathbf{x} = (x, y) \in \mathcal{R}^2$$

- Homogeneous point coordinates

$$\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) \in \mathcal{P}^2 \quad \text{2D projective space } \mathcal{P}^2 = \mathcal{R}^3 - (0, 0, 0)$$
$$\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = \tilde{w}\bar{\mathbf{x}}$$

- Points at infinity (or ideal points): $\{(\tilde{x}, \tilde{y}, \tilde{w}) | \tilde{w} = 0\}$
- Algebraically treats all points in the projective plane (both Euclidean and ideal) equally
- The camera project equation is *non-linear* when expressed in *non-homogeneous coordinates*, but is *linear* in *homogeneous* coordinates.



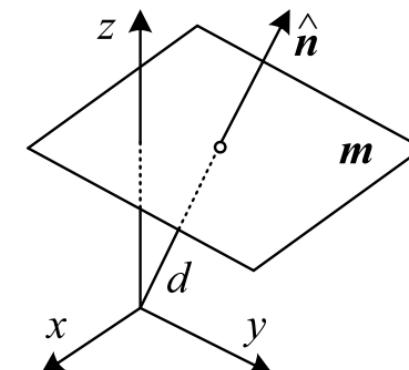
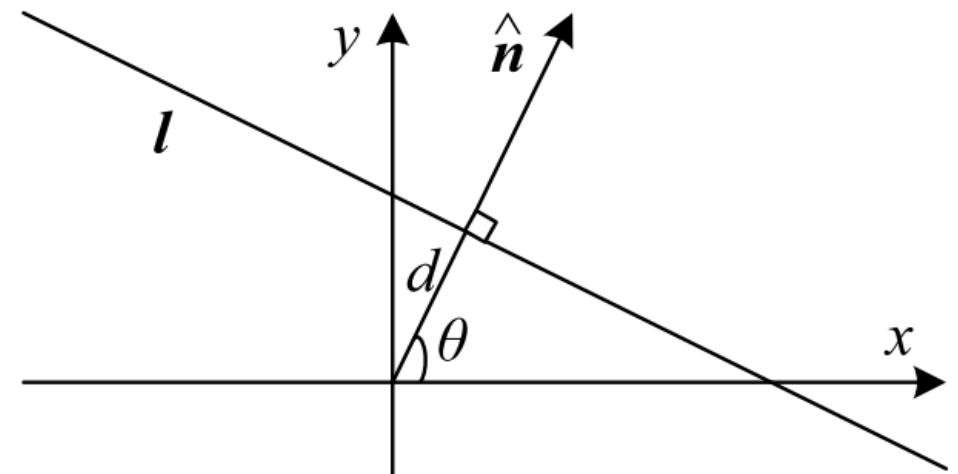
Representing Geometric Primitives in Homogenous Coordinates

- Homogeneous 2D line

$$\bar{x} \cdot \tilde{l} = ax + by + c = 0$$

$$\tilde{l} = (a, b, c)$$

- Homogeneous 3D plane



Advantages of Homogenous Coordinates

- Joints and intersections
 - Intersection point of two lines

$$\tilde{x} = \tilde{l}_1 \times \tilde{l}_2$$

- Line jointing two points

$$\tilde{l} = \tilde{x}_1 \times \tilde{x}_2$$

- Duality between 2D lines and 2D points

$$\begin{array}{l|ccc} +\mathbf{i}\mathbf{u}_2\mathbf{v}_3 & \mathbf{i} & \mathbf{j} & \mathbf{k} \\ +\mathbf{u}_1\mathbf{v}_2\mathbf{k} & u_1 & u_2 & u_3 \\ +\mathbf{v}_1\mathbf{j}\mathbf{u}_3 & v_1 & v_2 & v_3 \\ -\mathbf{v}_1\mathbf{u}_2\mathbf{k} & \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -\mathbf{i}\mathbf{v}_2\mathbf{u}_3 & u_1 & u_2 & u_3 \\ -\mathbf{u}_1\mathbf{j}\mathbf{v}_3 & \mathbf{i} & \mathbf{j} & \mathbf{k} \end{array}$$



Advantages of Homogenous Coordinates

- Intersection of parallel lines $ax+by+c = 0$ and $ax+by+c' = 0$?
 - Point at infinity $(b, -a, 0)^T$
 - Line at infinity (or ideal line)?

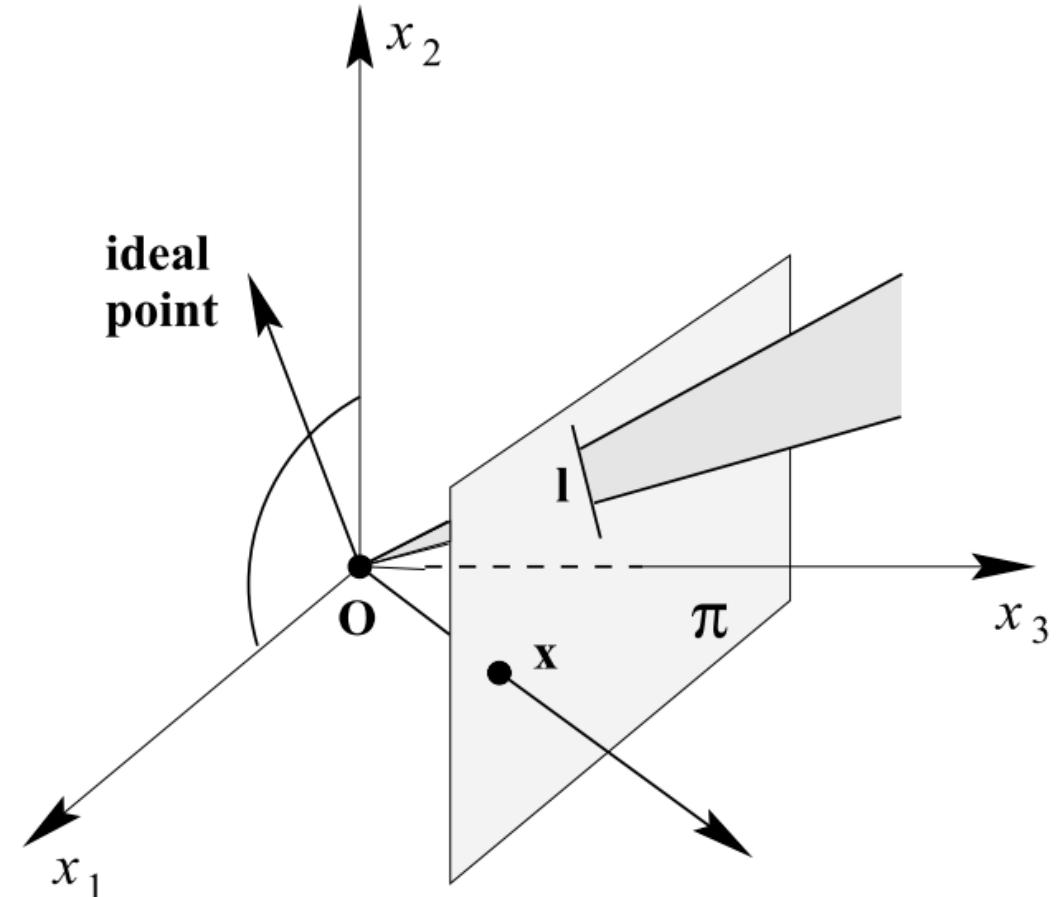
$$\mathbf{l}_\infty = (0, 0, 1)^T$$

- The set of all ideal points
- The set of directions of lines



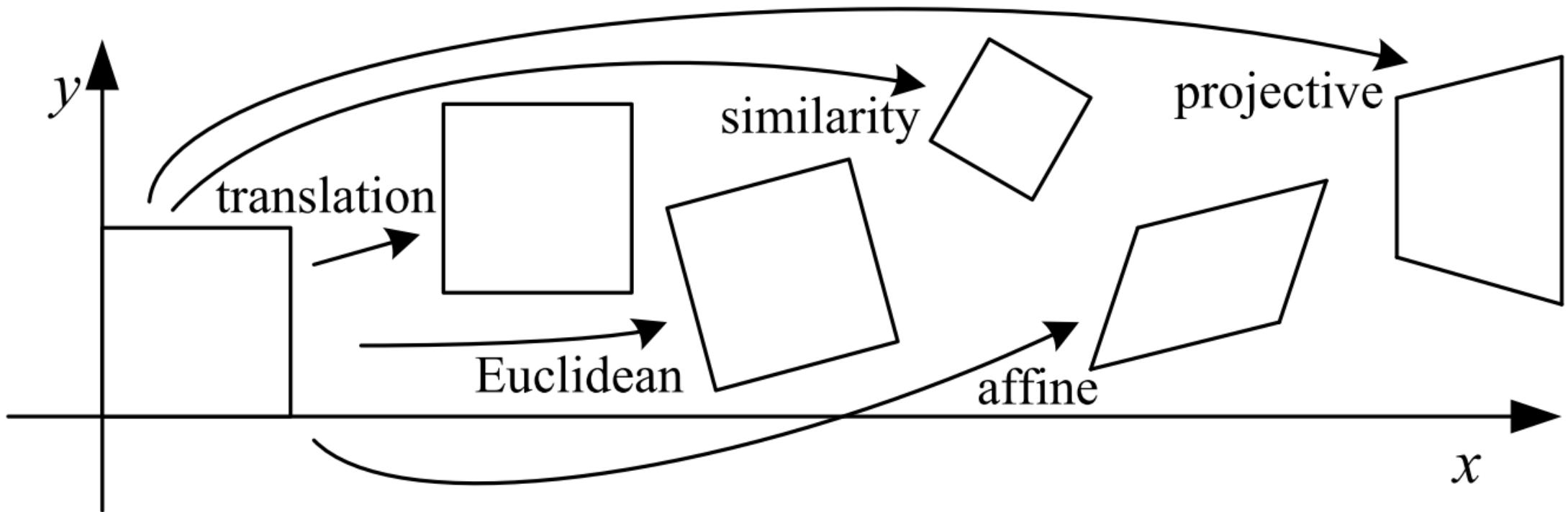
Projective Space

- Projective geometry
 - The study of the geometry of projective space
 - **A point** in 2D projective space
 - Can be seen as **a ray** passing through the origin in 3D Euclidean space
 - No need to distinguish between points/lines at infinity and ordinary points/lines
 - One of the most important theoretic foundation of geometric computer vision





2D-to-2D Transformations





Hierarchy of 2D Transformations

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	4	angles	
affine	$\begin{bmatrix} \mathbf{A} \end{bmatrix}_{2 \times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{3 \times 3}$	8	straight lines	



Hierarchy of 3D Transformations

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\left[\begin{array}{c c} \mathbf{I} & \mathbf{t} \end{array} \right]_{3 \times 4}$		orientation	
rigid (Euclidean)	$\left[\begin{array}{c c} \mathbf{R} & \mathbf{t} \end{array} \right]_{3 \times 4}$		lengths	
similarity	$\left[\begin{array}{c c} s\mathbf{R} & \mathbf{t} \end{array} \right]_{3 \times 4}$		angles	
affine	$\left[\begin{array}{c} \mathbf{A} \end{array} \right]_{3 \times 4}$		parallelism	
projective	$\left[\begin{array}{c} \tilde{\mathbf{H}} \end{array} \right]_{4 \times 4}$		straight lines	

3D Rotation Representation

- Orthonormal matrix
 - Some simple examples (rotation around coordinate axis)

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3D Rotation Representation

- Euler angles
 - 3 successive rotations
 - Eulerian: with repetition in axis
 - XYX, XZX, YXY, YZY, ZXZ, ZYZ
 - Cardanian: without repetition
 - XYZ, XZY, YZX, YXZ, ZXY, ZYX
- 12 types
 - And you should always check handedness, intrinsic or extrinsic rotation

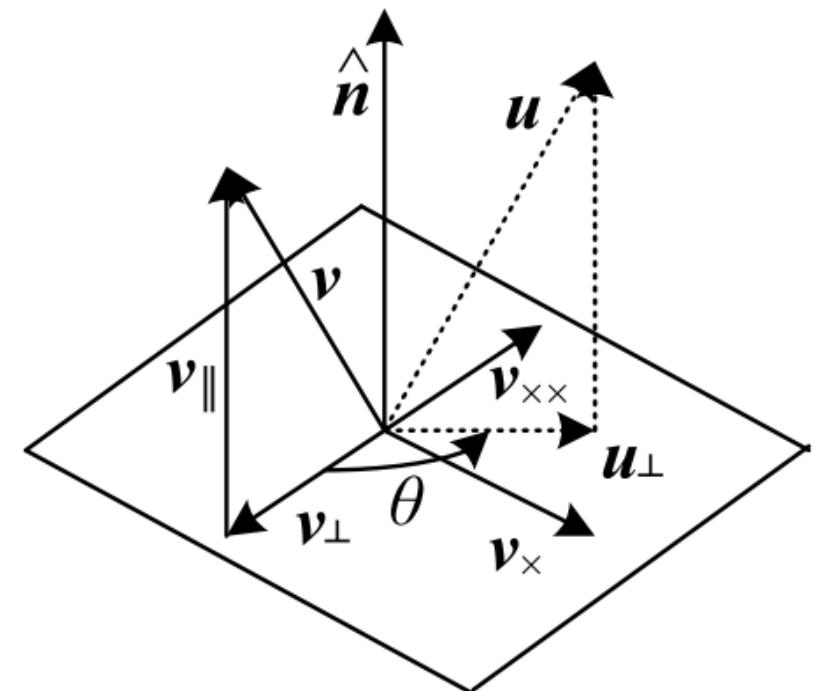
$$R = R_z(\phi)R_y(\theta)R_z(\psi)$$

3D Rotation Representation

- Axis-angle

$$\boldsymbol{R}(\hat{\boldsymbol{n}}, \theta) = \boldsymbol{I} + \sin \theta [\hat{\boldsymbol{n}}]_{\times} + (1 - \cos \theta) [\hat{\boldsymbol{n}}]_{\times}^2$$

$$[\hat{\boldsymbol{n}}]_{\times} = \begin{bmatrix} 0 & -\hat{n}_z & \hat{n}_y \\ \hat{n}_z & 0 & -\hat{n}_x \\ -\hat{n}_y & \hat{n}_x & 0 \end{bmatrix}$$



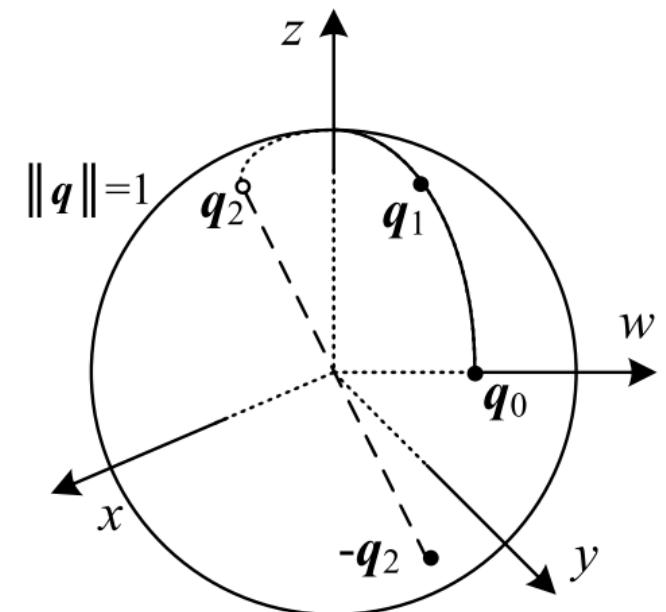
3D Rotation Representation

- Unit quaternion

$$\mathbf{q} = (\mathbf{v}, w) = \left(\sin \frac{\theta}{2} \hat{\mathbf{n}}, \cos \frac{\theta}{2} \right)$$

$$\mathbf{q} = (x, y, z, w)$$

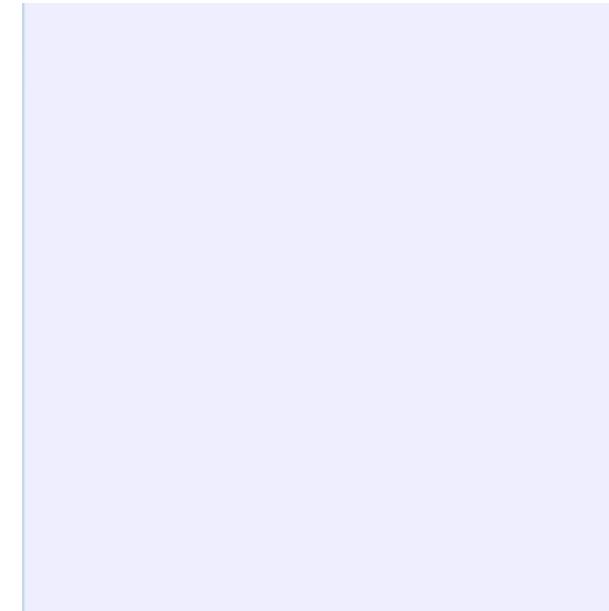
$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - zw) & 2(xz + yw) \\ 2(xy + zw) & 1 - 2(x^2 + z^2) & 2(yz - xw) \\ 2(xz - yw) & 2(yz + xw) & 1 - 2(x^2 + y^2) \end{bmatrix}$$





3D Rotation Representation/Parameterization

- Euler
 - Minimal
 - Many ambiguities
 - Gimbal Lock
- Axis-angle
 - Minimal
 - Not unique
- Unit quaternion
 - Continuous, almost unique
 - Easy inverse/multiplication
 - Interpolation using *slerp*
- *Two more new rotation representation with Deep Learning*
 - RSS'20: <https://github.com/utiasSTARS/bingham-rotation-learning>
 - CVPR'19: https://zhouyisjtu.github.io/project_rotation/rotation.html

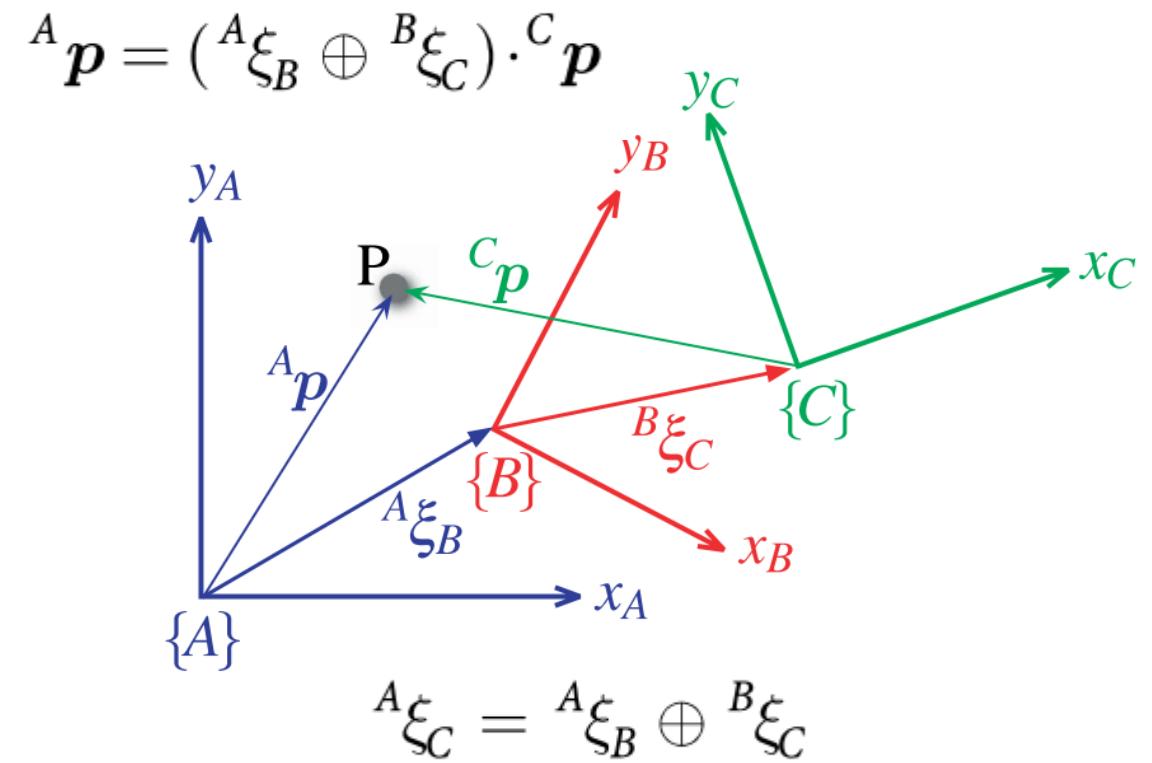
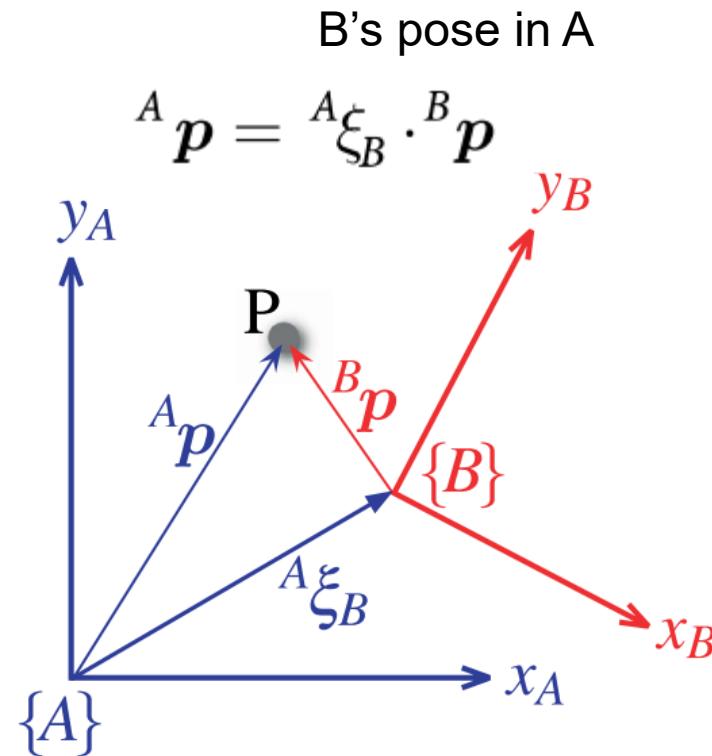


procedure *slerp*($\mathbf{q}_0, \mathbf{q}_1, \alpha$):

1. $\mathbf{q}_r = \mathbf{q}_1/\mathbf{q}_0 = (\mathbf{v}_r, w_r)$
2. if $w_r < 0$ then $\mathbf{q}_r \leftarrow -\mathbf{q}_r$
3. $\theta_r = 2 \tan^{-1}(\|\mathbf{v}_r\|/w_r)$
4. $\hat{\mathbf{n}}_r = \mathcal{N}(\mathbf{v}_r) = \mathbf{v}_r/\|\mathbf{v}_r\|$
5. $\theta_\alpha = \alpha \theta_r$
6. $\mathbf{q}_\alpha = (\sin \frac{\theta_\alpha}{2} \hat{\mathbf{n}}_r, \cos \frac{\theta_\alpha}{2})$
7. **return** $\mathbf{q}_2 = \mathbf{q}_\alpha \mathbf{q}_0$

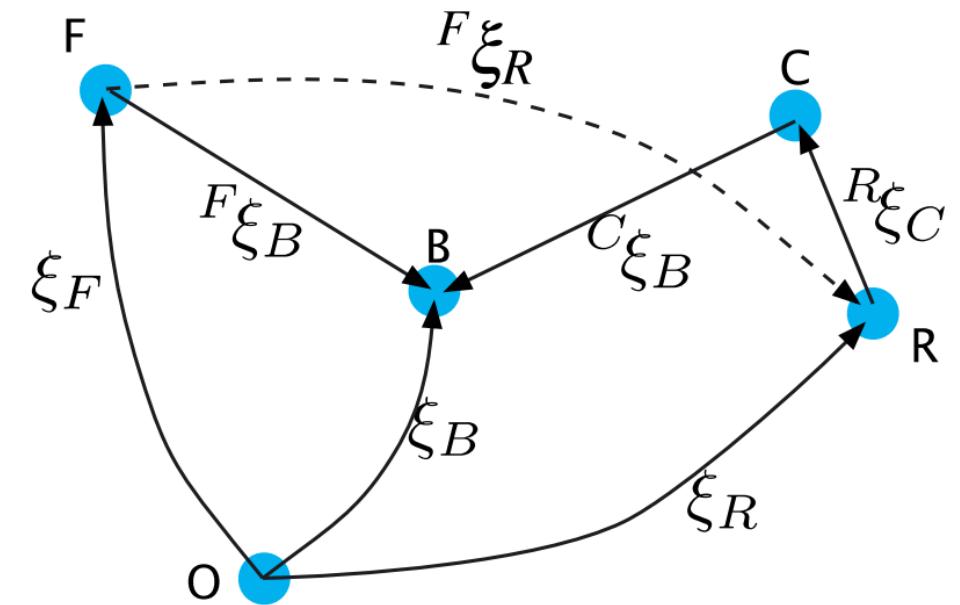
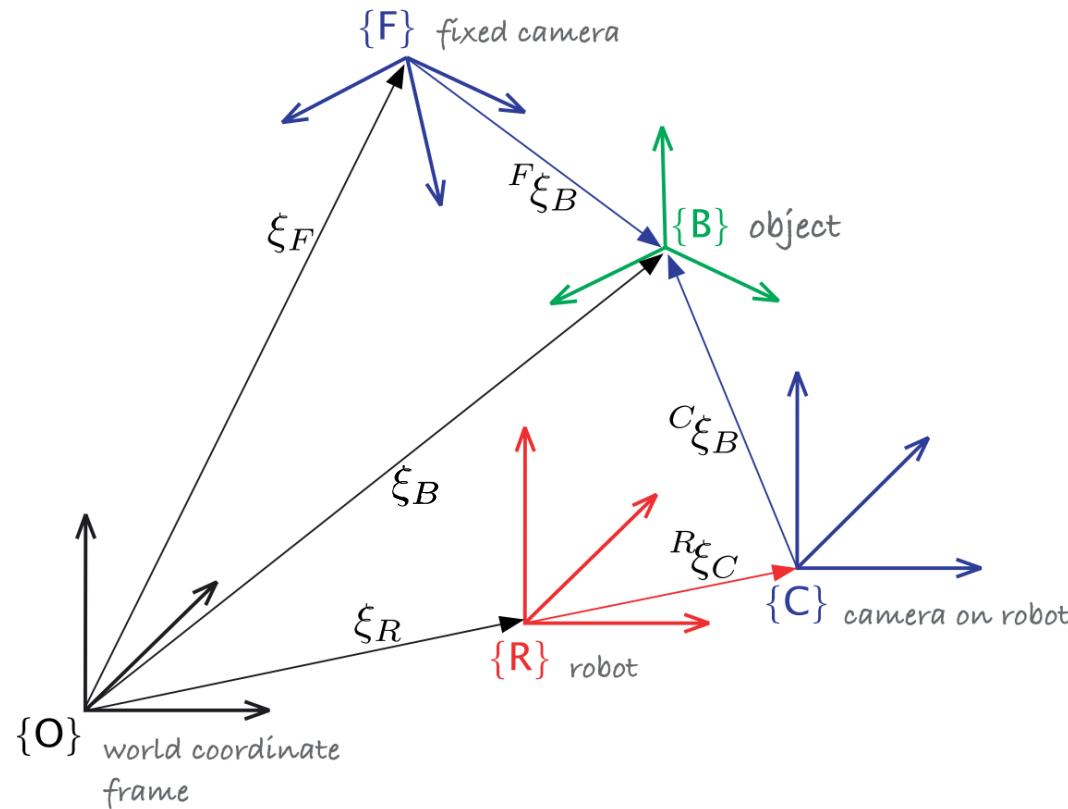


Coordinate Frames and Their Compositions



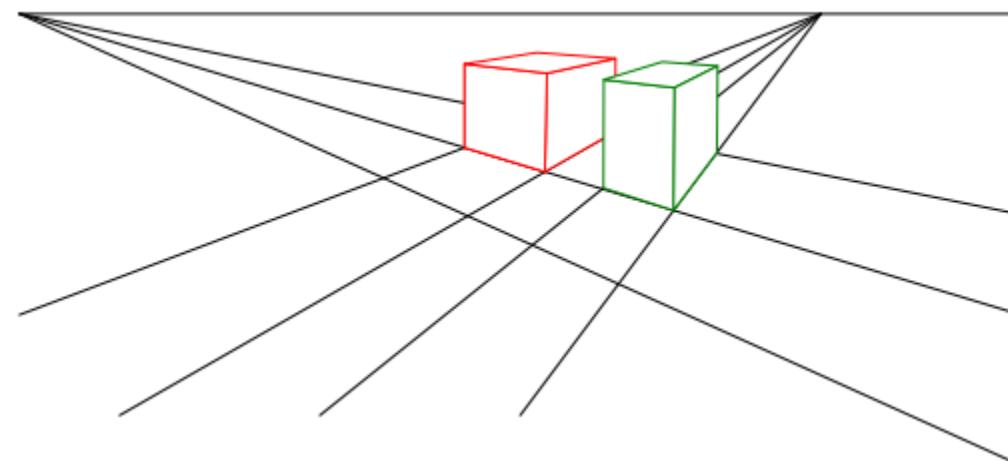


Coordinate Frames and Their Compositions





3D to 2D Projection

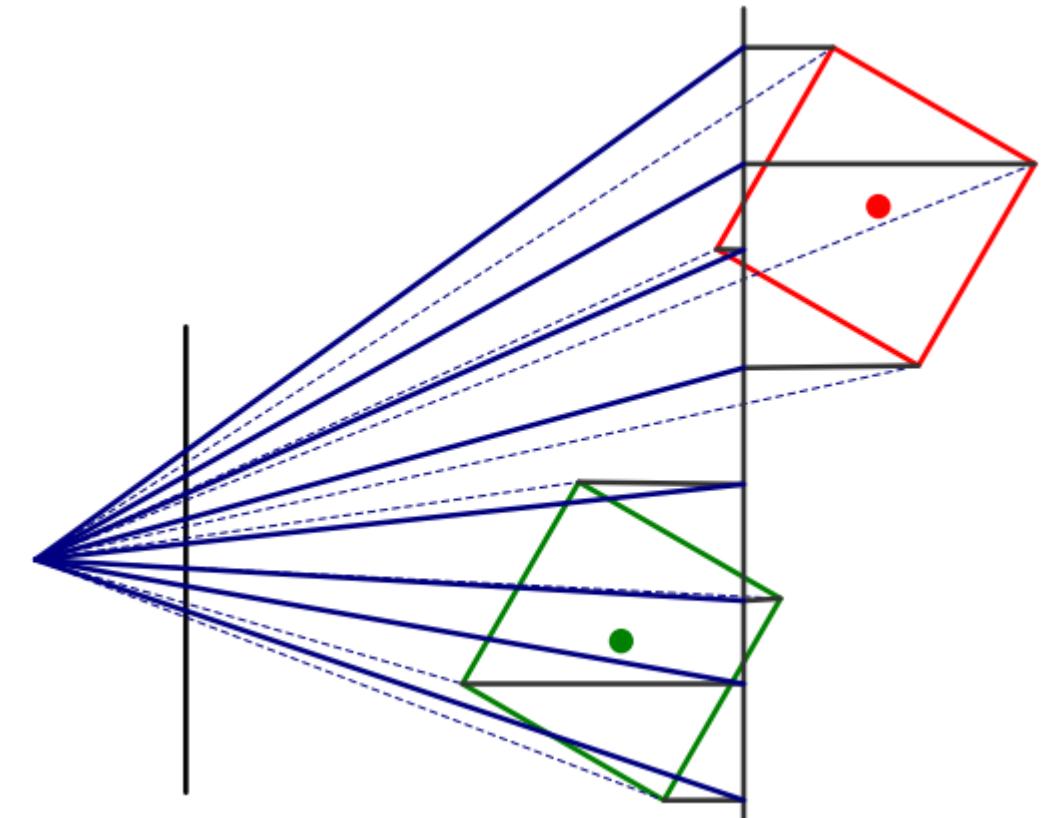




3D to 2D Projection

- Orthographic projection
- Approximation of telephoto

$$\tilde{x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{p}$$

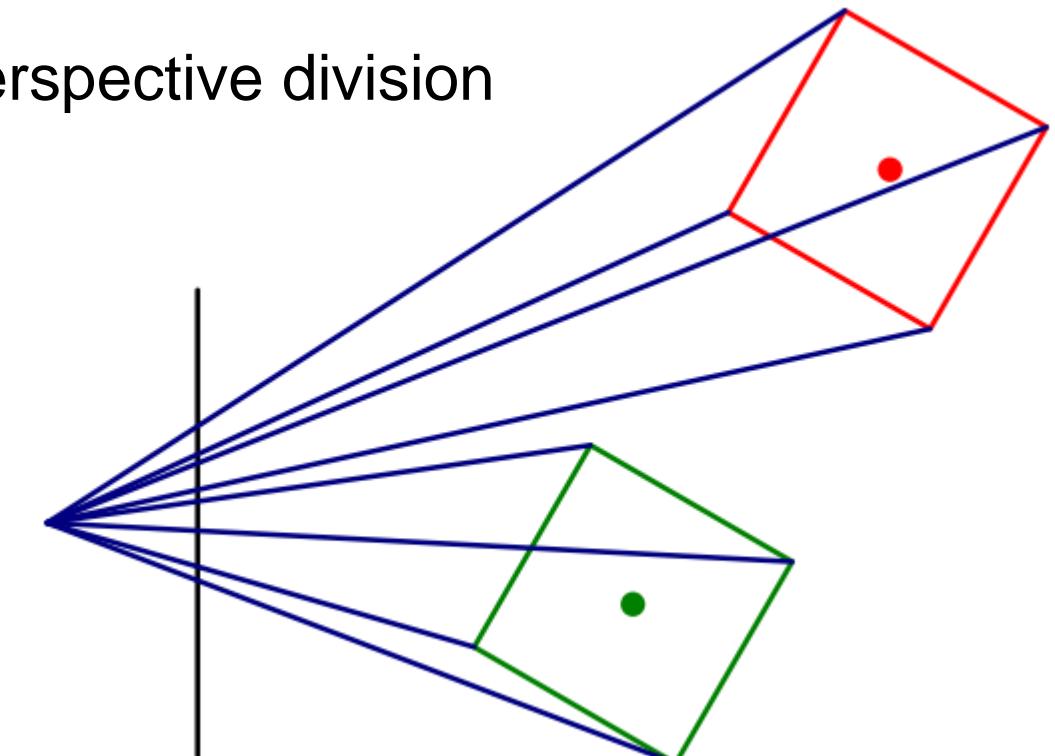




3D to 2D Projection

- Perspective projection

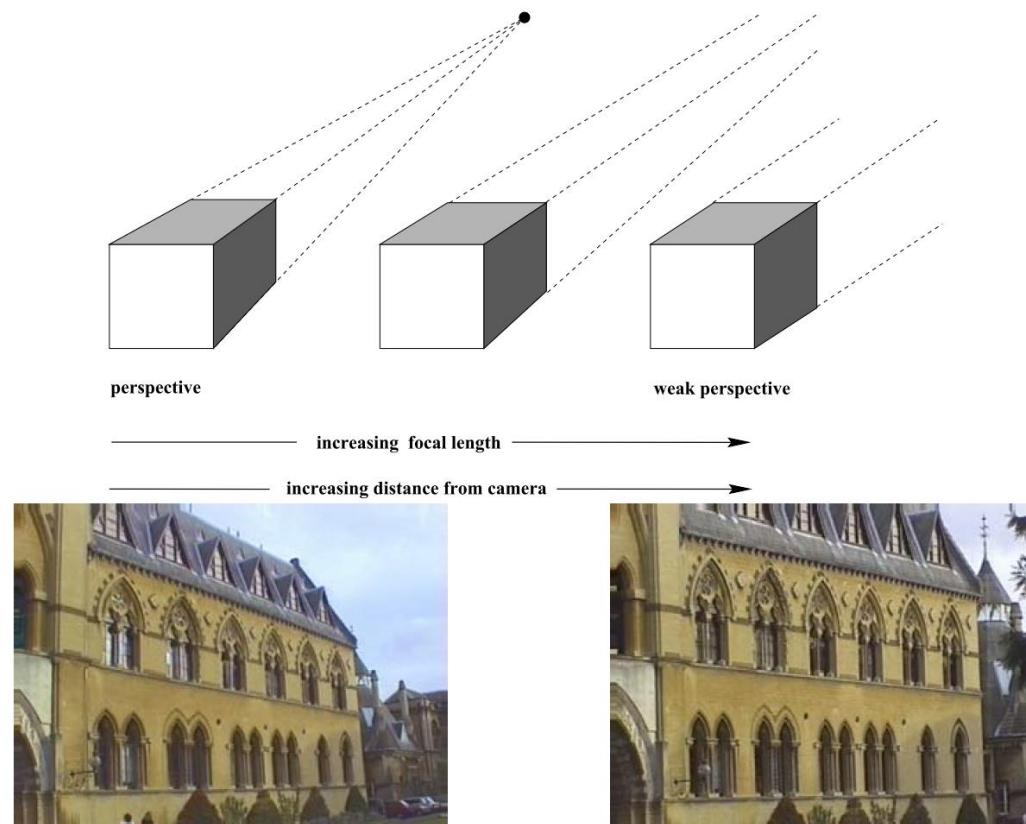
$$\bar{x} = \mathcal{P}_z(\mathbf{p}) = \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix}$$
$$\tilde{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{p}}$$





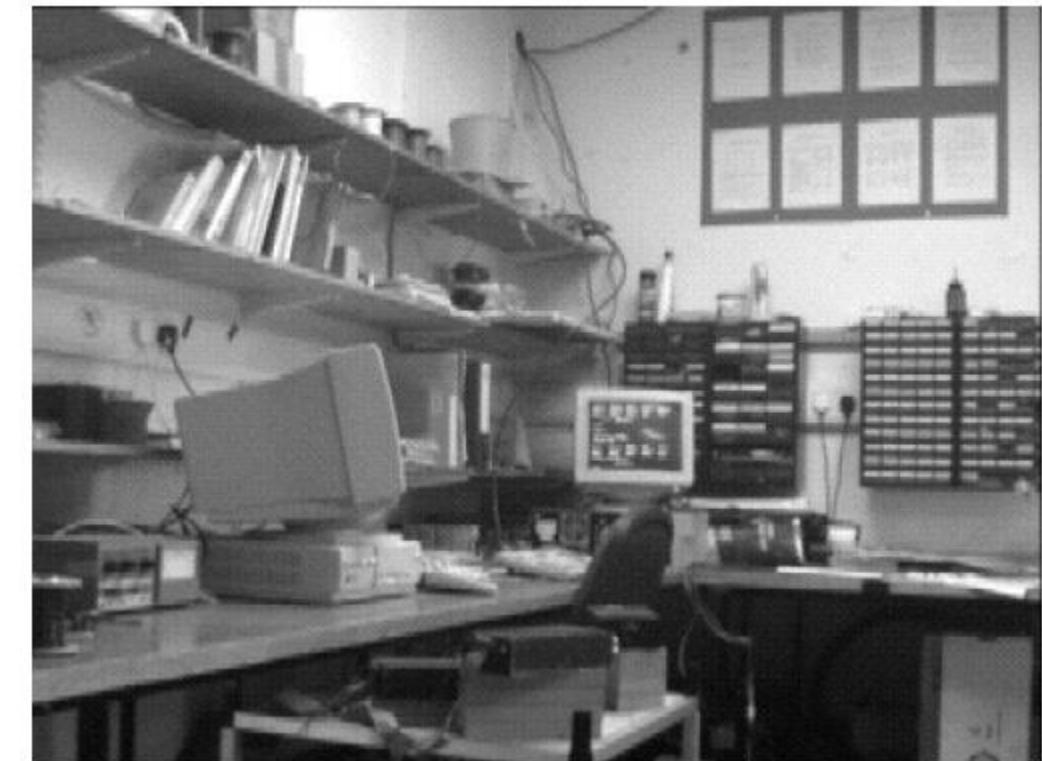
3D to 2D Projection

- Weak perspective projection



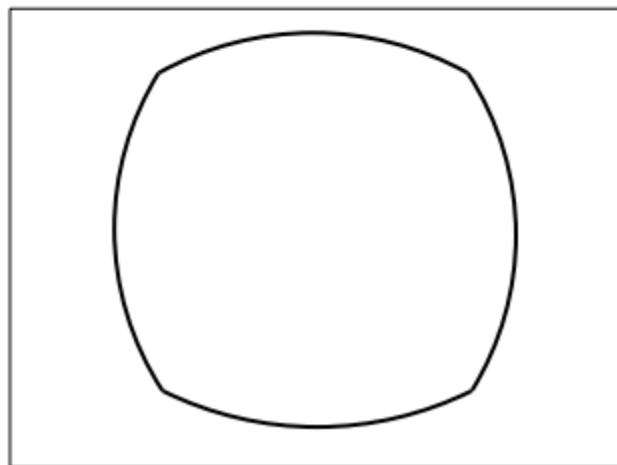
Lens Distortion

- Radial distortion



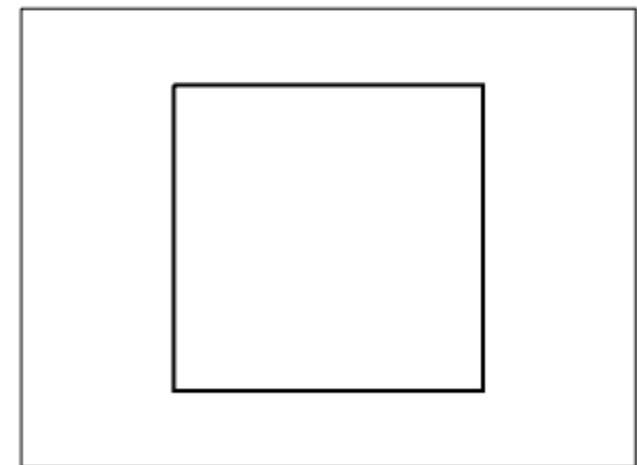
Lens Distortion

radial distortion



correction

linear image



$$x_{\text{corrected}} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

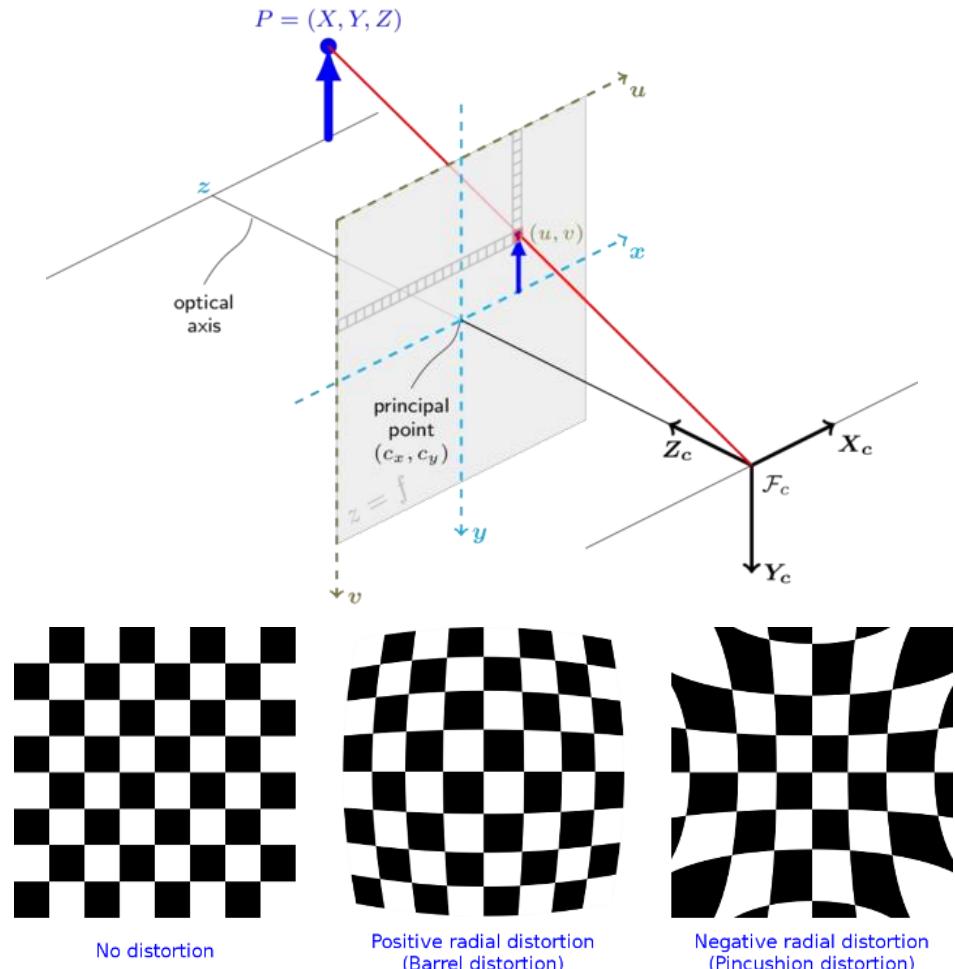
$$y_{\text{corrected}} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

<https://hal-enpc.archives-ouvertes.fr/hal-01556898/document>

https://jo.dreggn.org/home/2016_optics.pdf



Everything Together in OpenCV (Full Model)



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$x'' = x' \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} + 2p_1x'y' + p_2(r^2 + 2x'^2)$$

$$y'' = y' \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} + p_1(r^2 + 2y'^2) + 2p_2x'y'$$

$$\text{where } r^2 = x'^2 + y'^2$$

$$u = f_x * x'' + c_x$$

$$v = f_y * y'' + c_y$$

Next Week

+ Overview of Deep Learning

++ Computational Graph

++ Back Propagation

+ DNN Architectures

RNN, CNN, ResNet, VAE

+ Common Practices

*: know how to code

++: know how to derive

+: know the concept

The Week After the Next Week

+ AprilTags

* Homography Estimation

++ $Ax=b$, $Ax=0$

* Camera Calibration, Zhang's method

+ DLT

+ Vanishing Points & Lines

*: know how to code

++: know how to derive

+: know the concept



References for the Week After the Next Week

- HZ2003:
 - Section 2.3, 4.1, 4.4, 7.1, 7.2, 7.4, 8.6
- FP2011:
 - Section 1.2, 1.3, 12.1
- Sz2022:
 - Section 11.1, 11.4.5
- Co2011:
 - Section 11.2, 11.1
- Linear algebra:
 - Sz2011: section A.1.1, A.1.2, A.1.3, A.2, A.2.1
 - HZ2003: A5.1, A5.2, A5.3