Principles of Robot Autonomy I

Course overview, intro to robotic systems and ROS

Stanford University

Autonomous Systems Lab

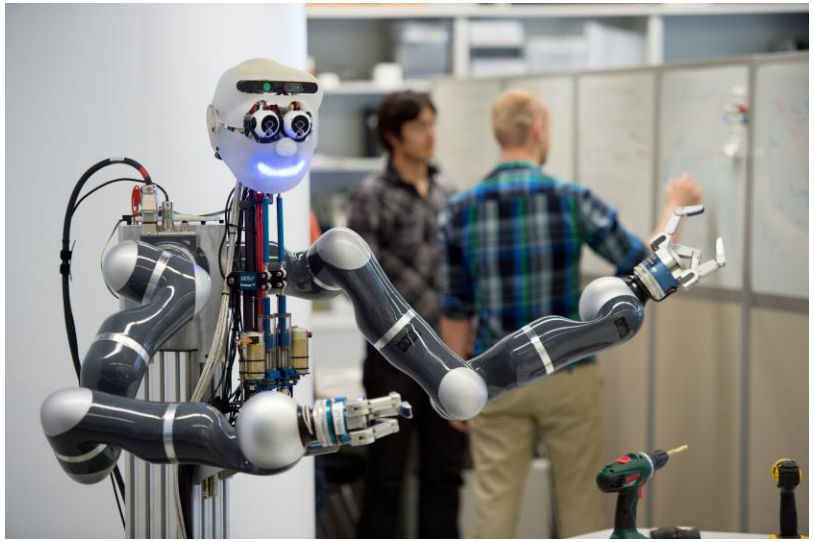
From automation…

…to autonomy

Waymo Self-Driving Car

Intuitive DaVinci Surgical Robot

Apollo Robot at MPI



Astrobee - NASA

Silicon Valley

CBRE

Automotive R&D

April 2016

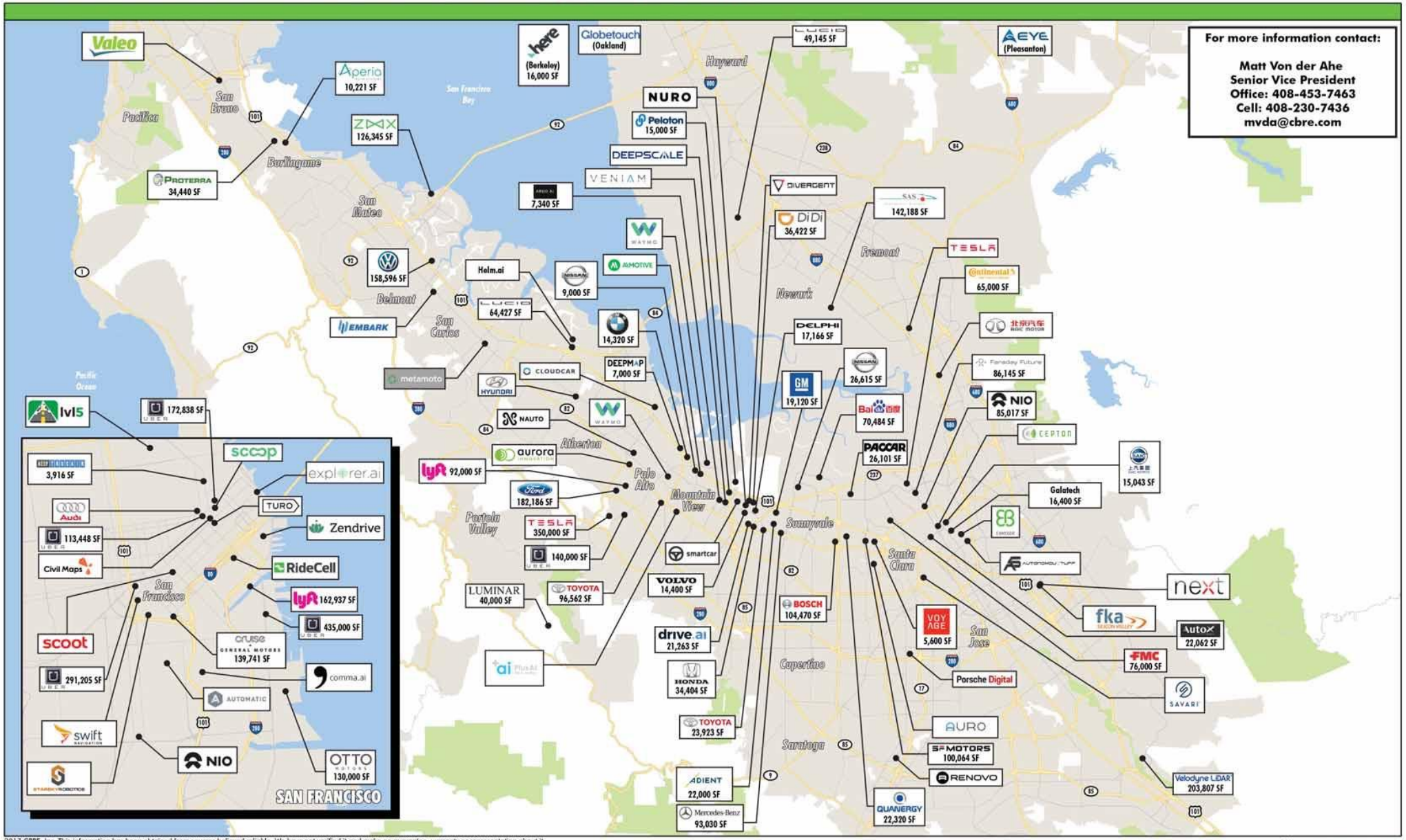
Silicon Valley

CBRE

AUTO LAB MAP APRIL 2017 SILICON VALLEY

CBRE

AUTO LAB MAP SEPTEMBER 2017 SILICON VALLEY

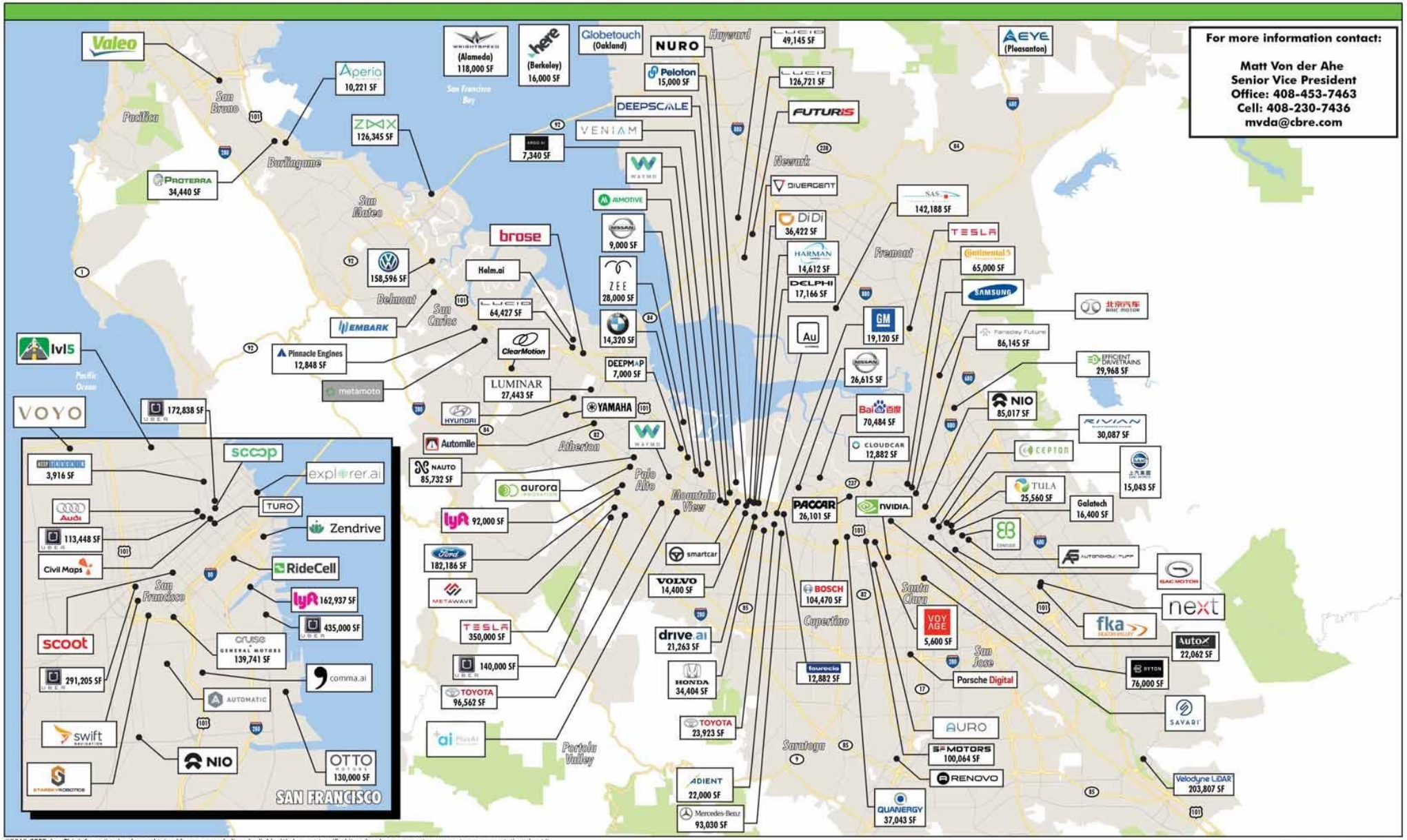


0 1.5 3 6 Miles

N

CBRE

AUTO LAB MAP DECEMBER 2017 SILICON VALLEY

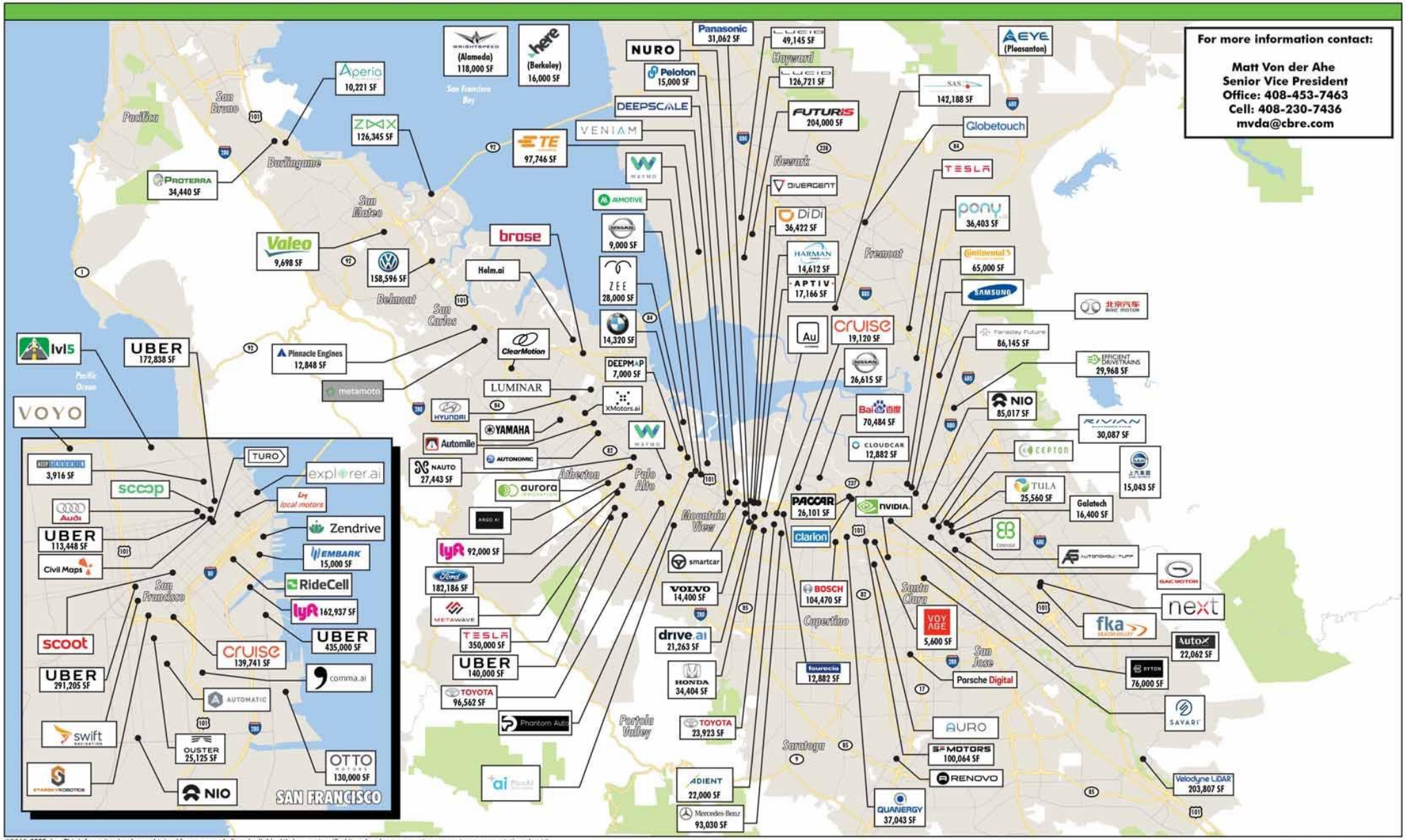


0 1.5 3 6 Miles

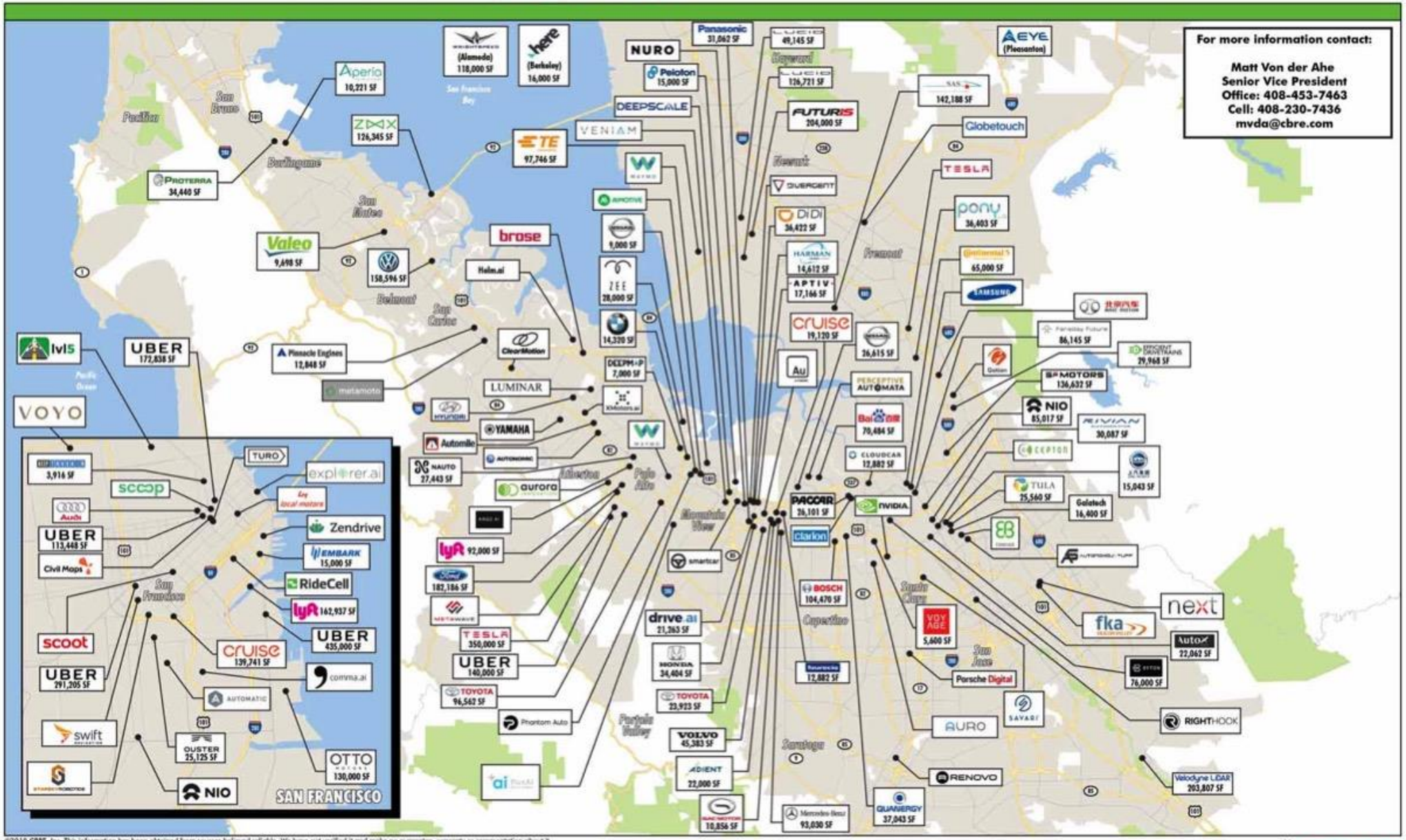
N

CBRE

AUTO LAB MAP FEBRUARY 2018 SILICON VALLEY



AUTO LAB MAP MARCH 2018 SILICON VALLEY

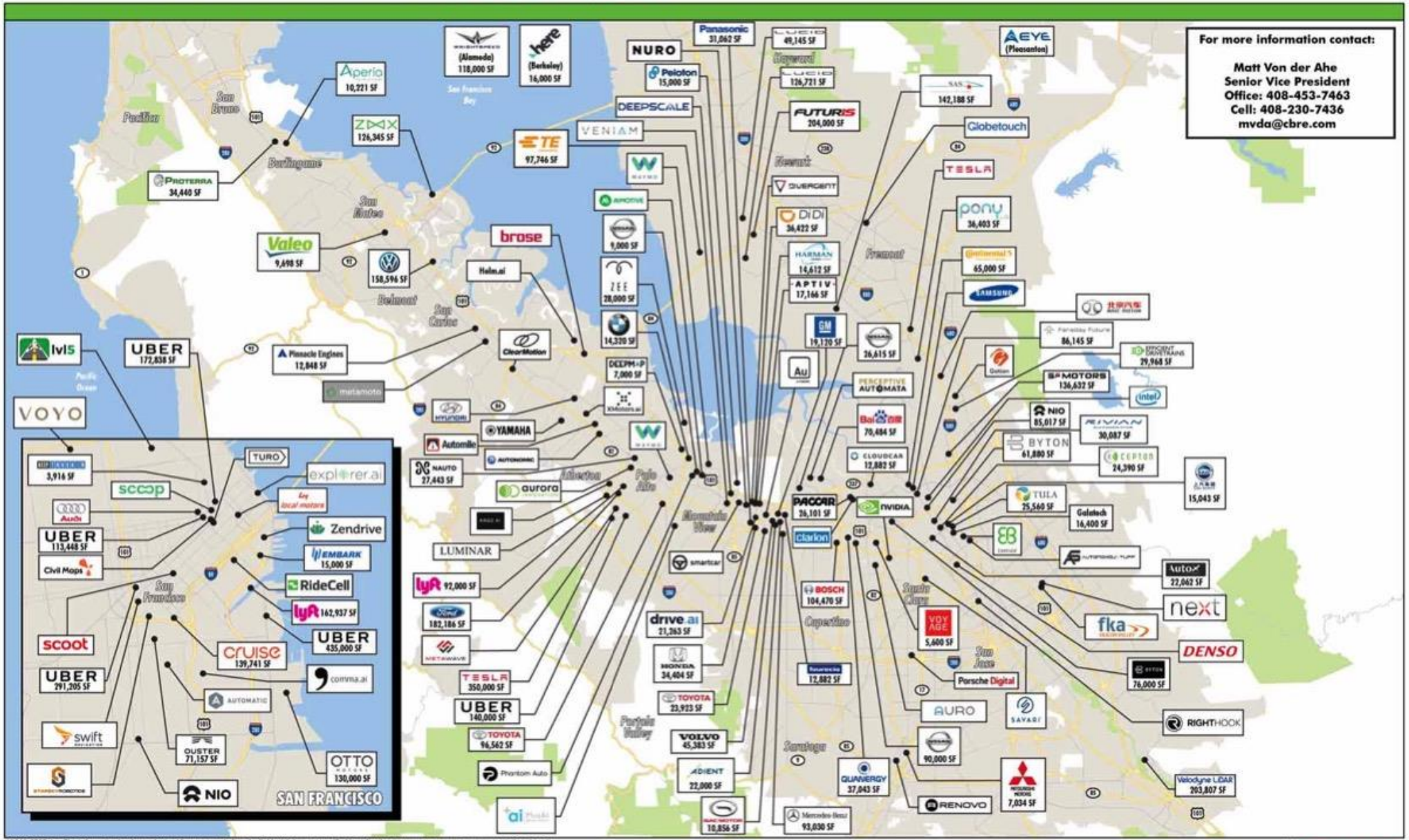


0 1.5 3 6 Miles

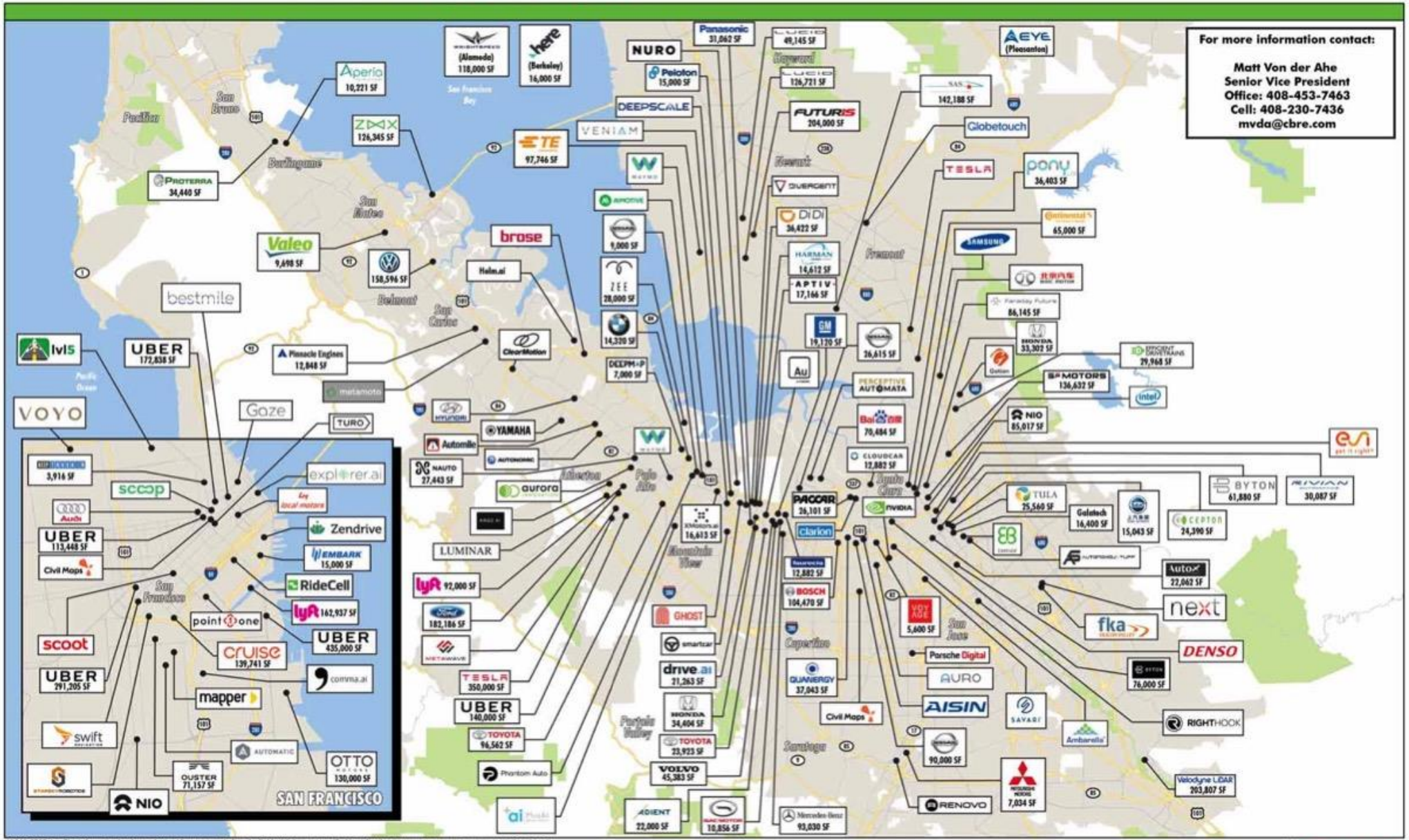
N

CBRE

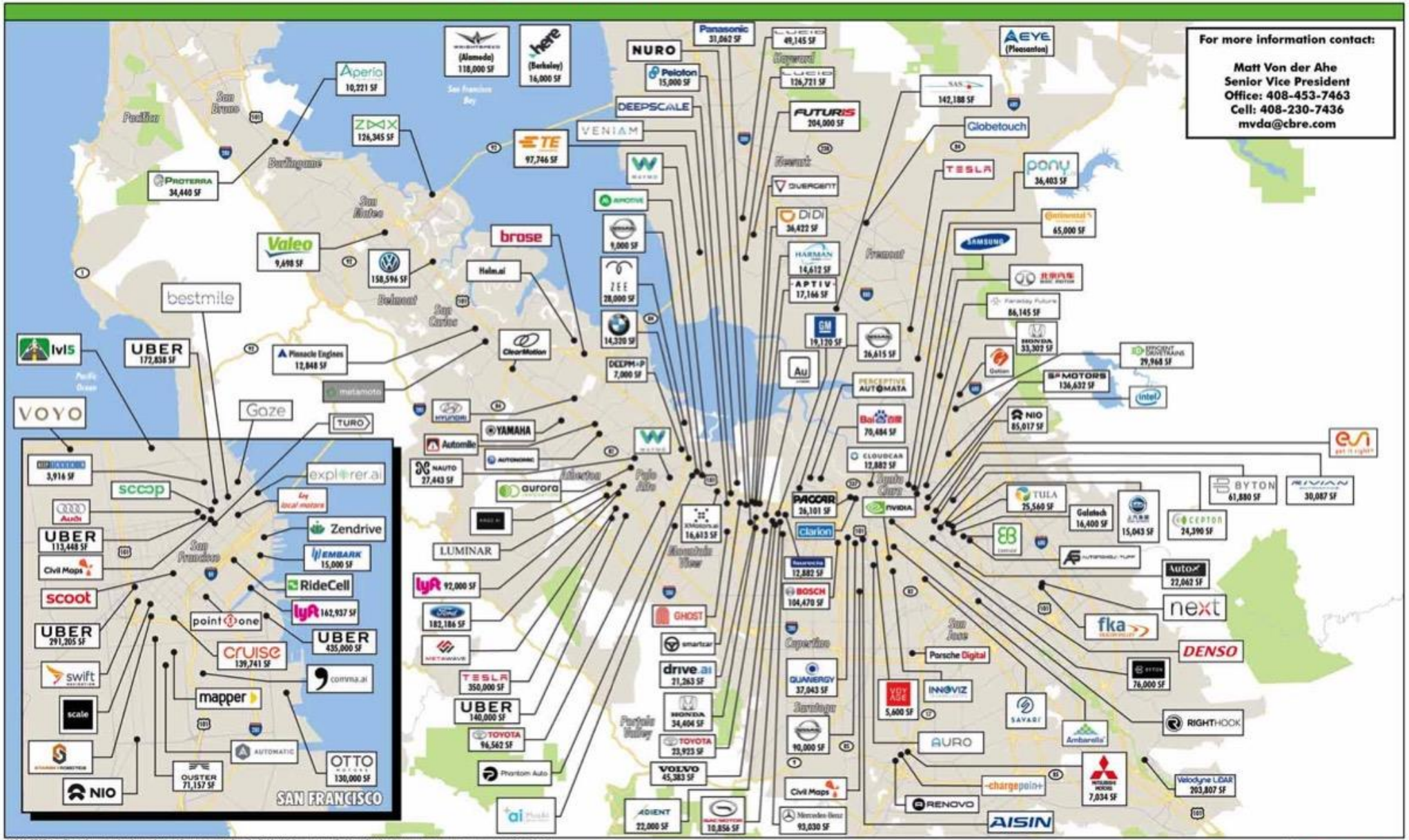
AUTO LAB MAP APRIL 2018 SILICON VALLEY



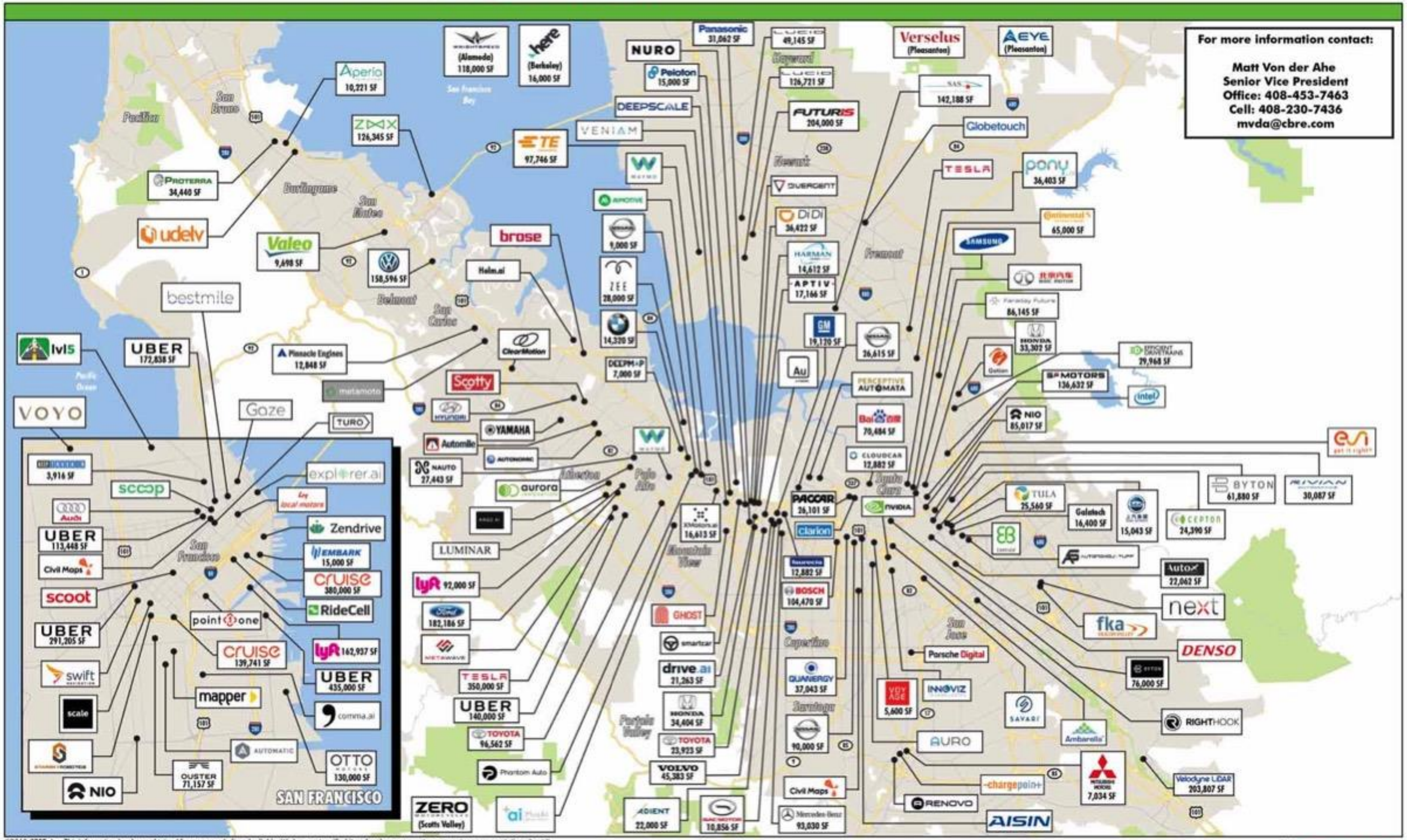
AUTO LAB MAP MAY 2018 SILICON VALLEY



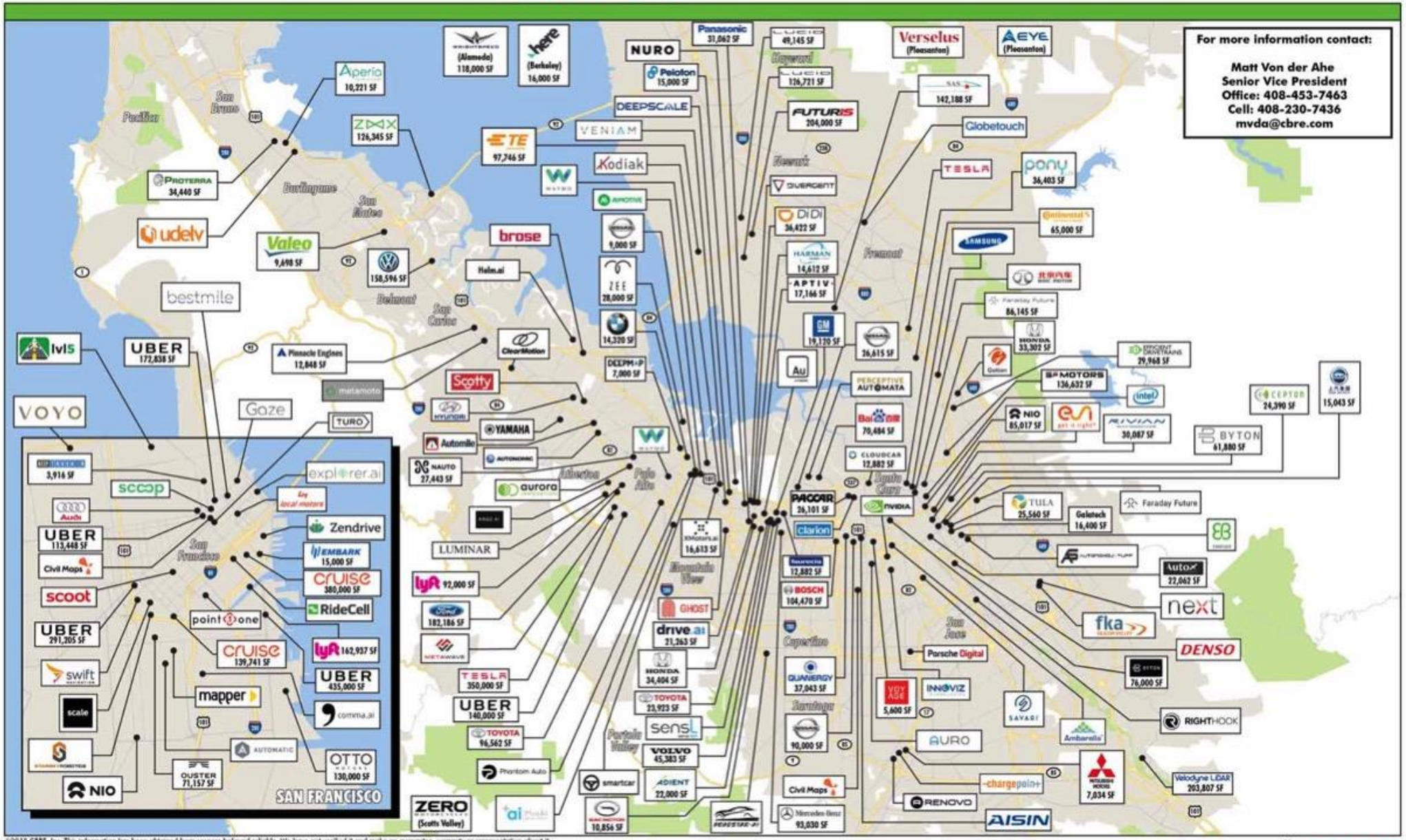
AUTO LAB MAP JUNE 2018 SILICON VALLEY



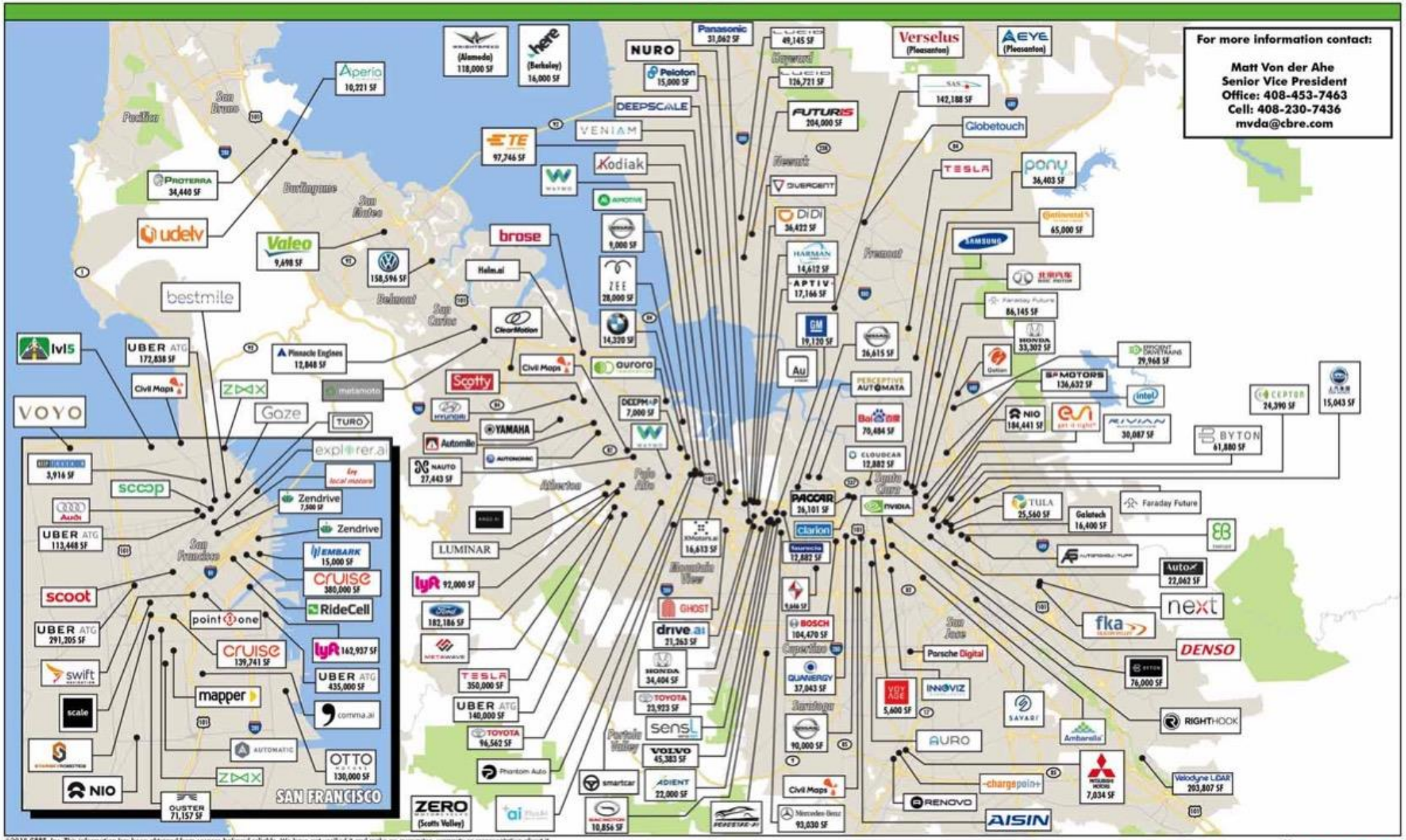
AUTO LAB MAP JULY 2018 SILICON VALLEY



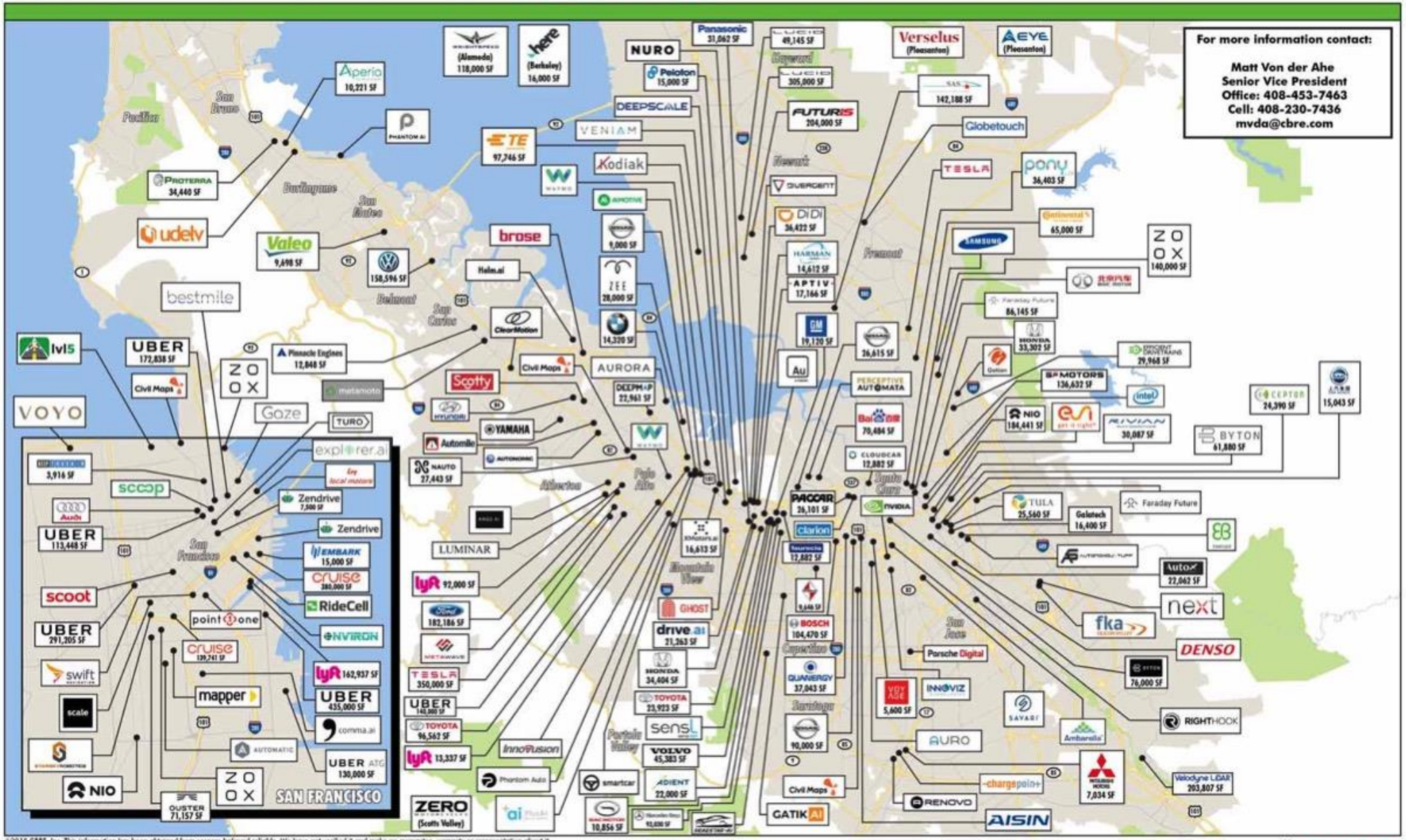
AUTO LAB MAP AUGUST 2018 SILICON VALLEY



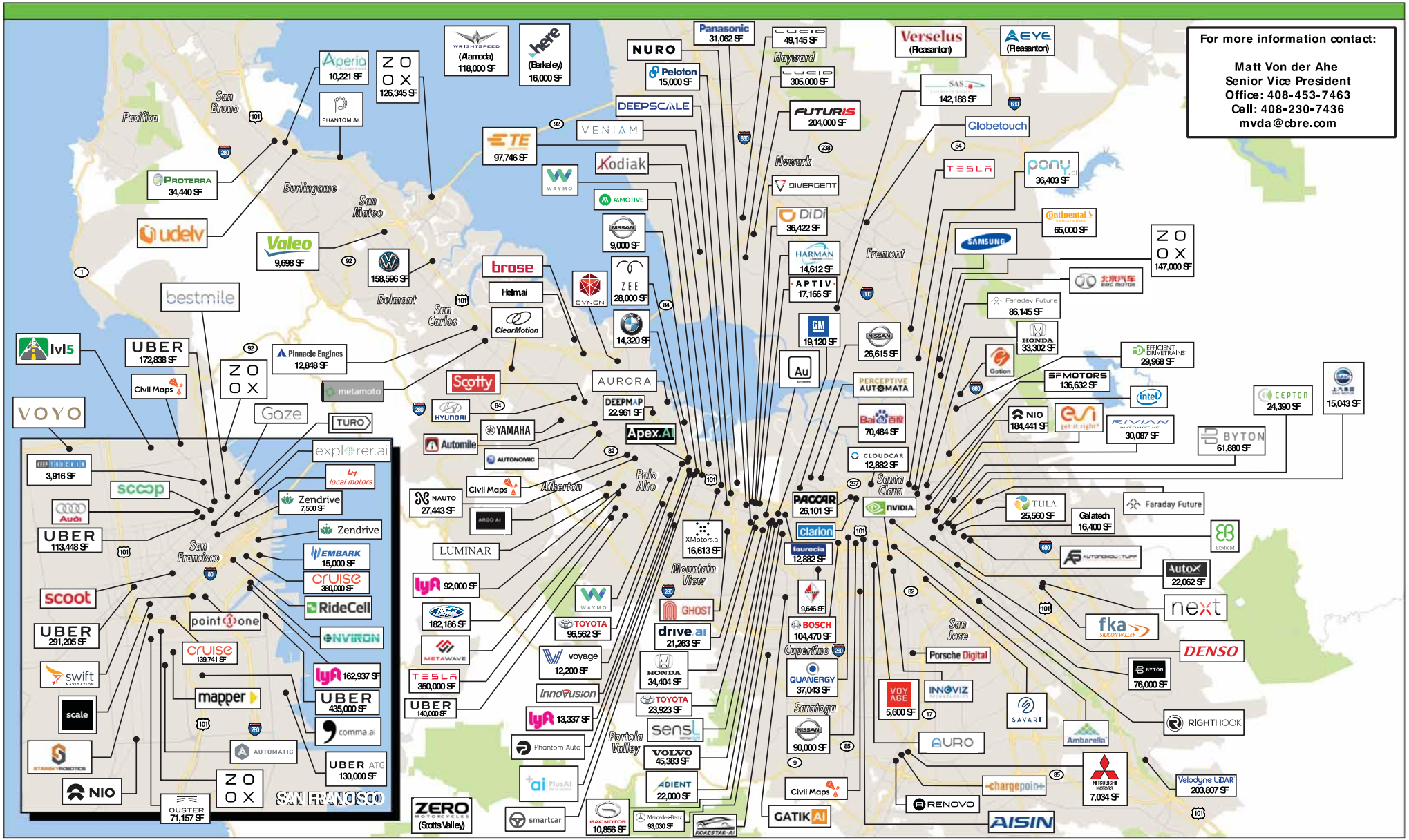
AUTO LAB MAP SEPTEMBER 2018 SILICON VALLEY



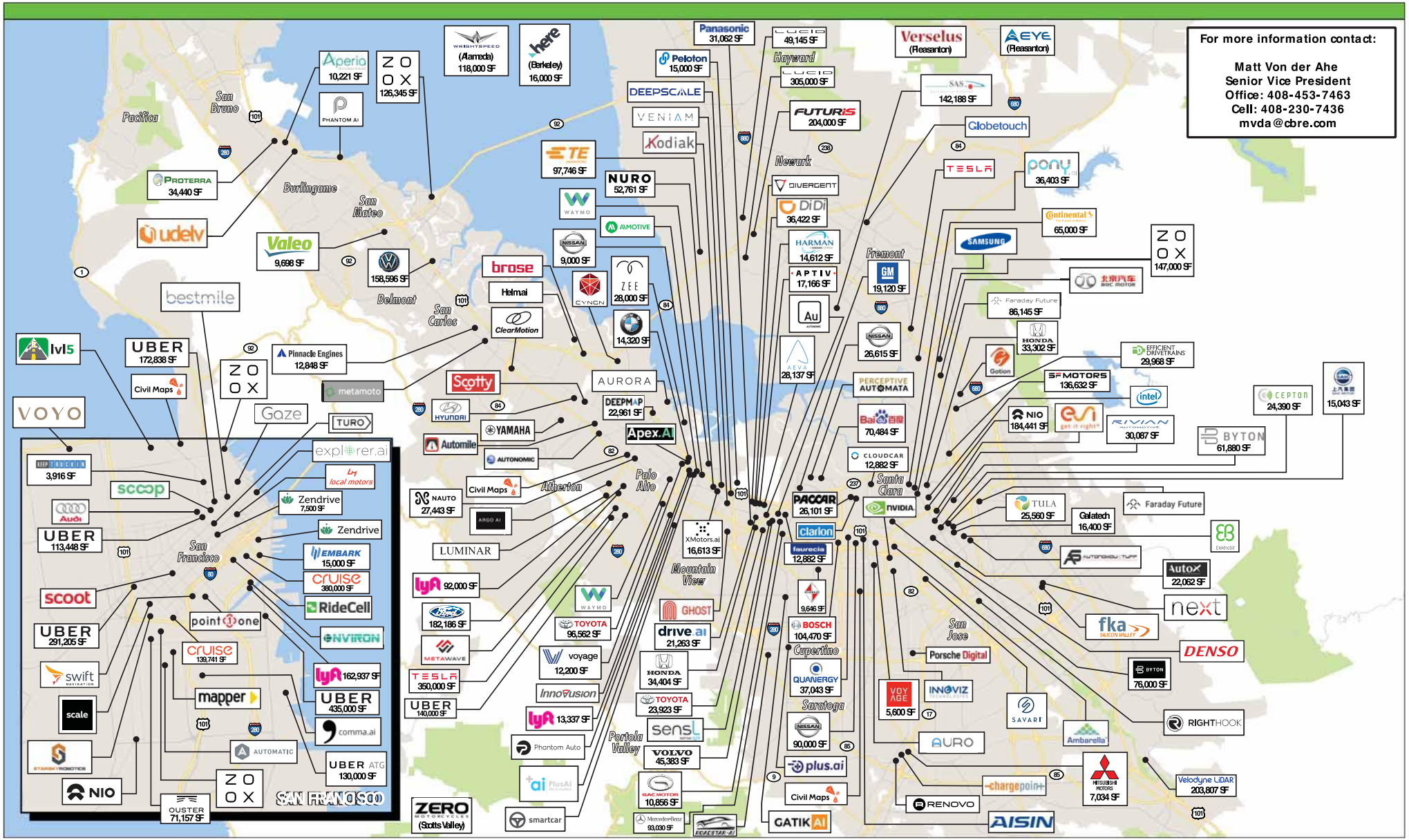
AUTO LAB MAP OCTOBER 2018 SILICON VALLEY



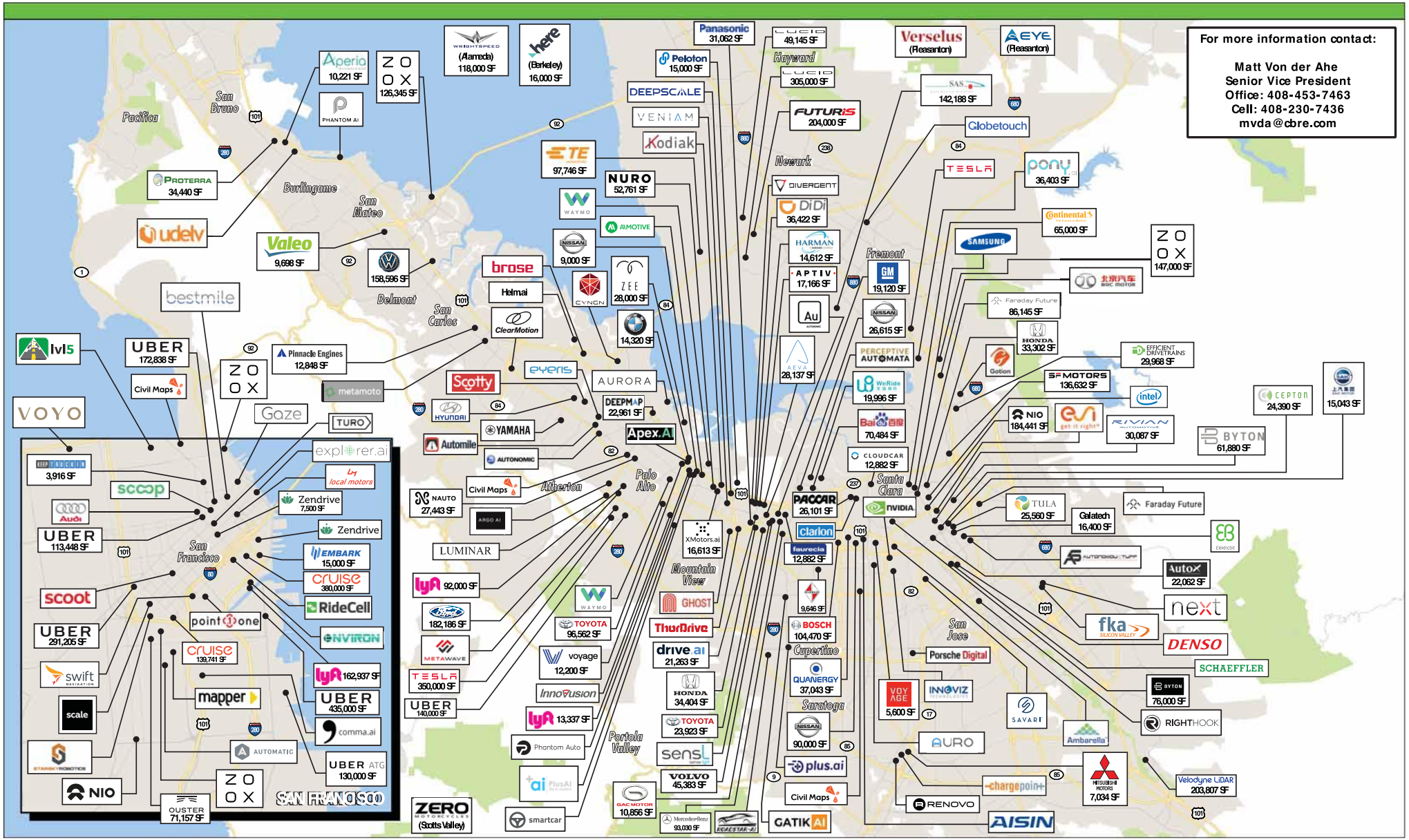
AUTO LAB MAP NOVEMBER 2018 SILICON VALLEY



AUTO LAB MAP DECEMBER 2018 SILICON VALLEY



AUTO LAB MAP JANUARY 2019 SILICON VALLEY



The Drone Market Environment 2019

Hardware

Drone Platforms

Agriculture

Delivery Systems

Safety & Security

Lighter-Than-Air

Drone-in-a-Box

Helicopter

Fixed-Wing

VTOL Fixed-Wing

Recreational

Passenger Drones / eVTOLs / Air Taxis

Counter-Drone Solutions

Components & Systems

Cameras, Imagining and Vision Systems

Launch and Recovery Systems

Navigation & Guidance Systems

Propulsion & Power

Data and Communication

Software

Flight, Fleet & Operation Management

Open Source Infrastructure, SDK

Navigation, CV and AI

UTM, LAANC Suppliers

Data Analytics, Workflow, CV and AI

Services

Drone-as-a-Service Providers

Delivery

Drone Show Providers

Education, Simulation, Training

System Integration, Engineering, Advisory

Maintenance

Market Research & Consulting

User Groups, Networks

Supplier, Retailers

Test Sites

Media, News, Blogs & Magazines

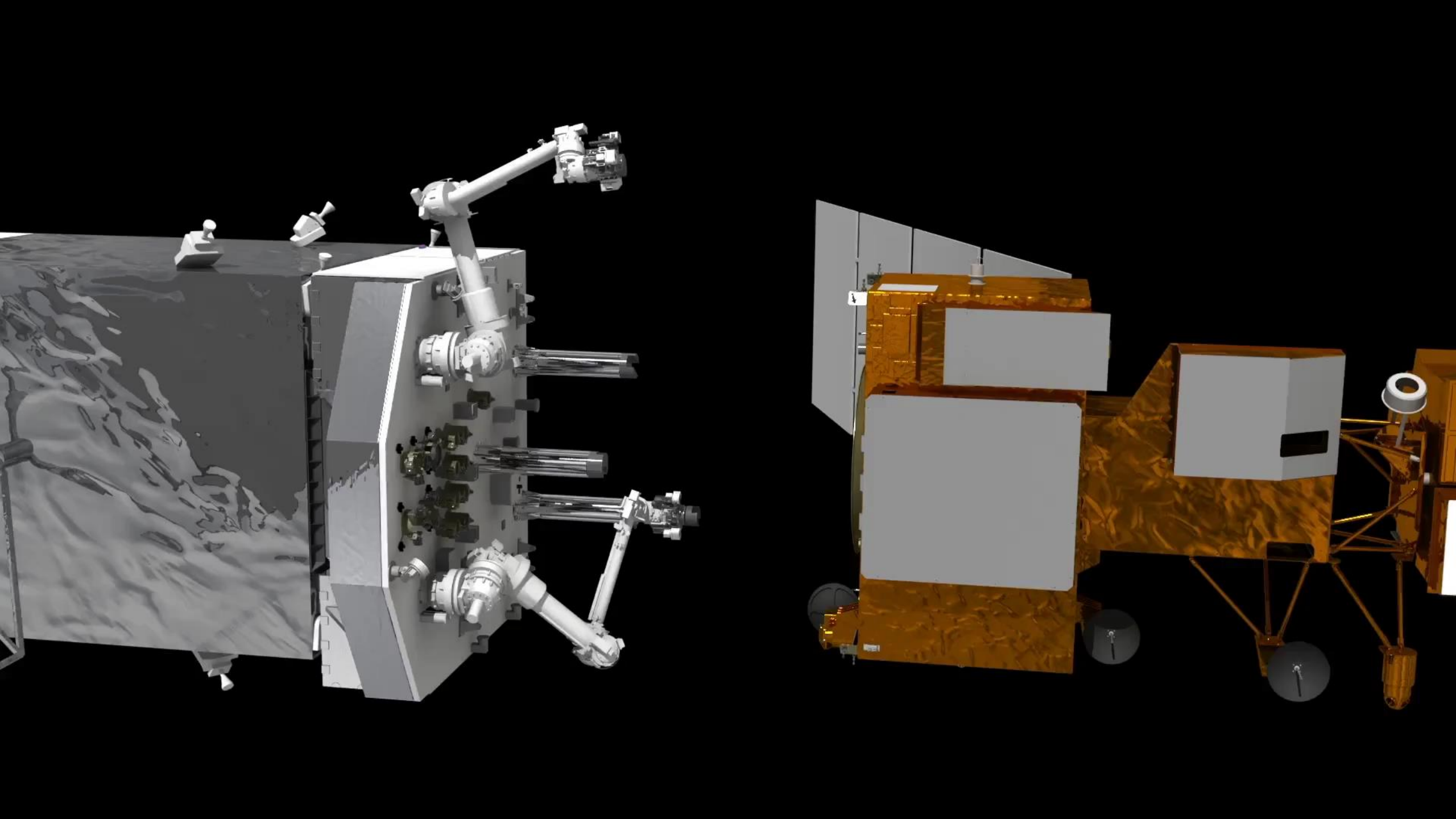
Podcast

Marketplaces

Shows, Conferences, Events

Insurances

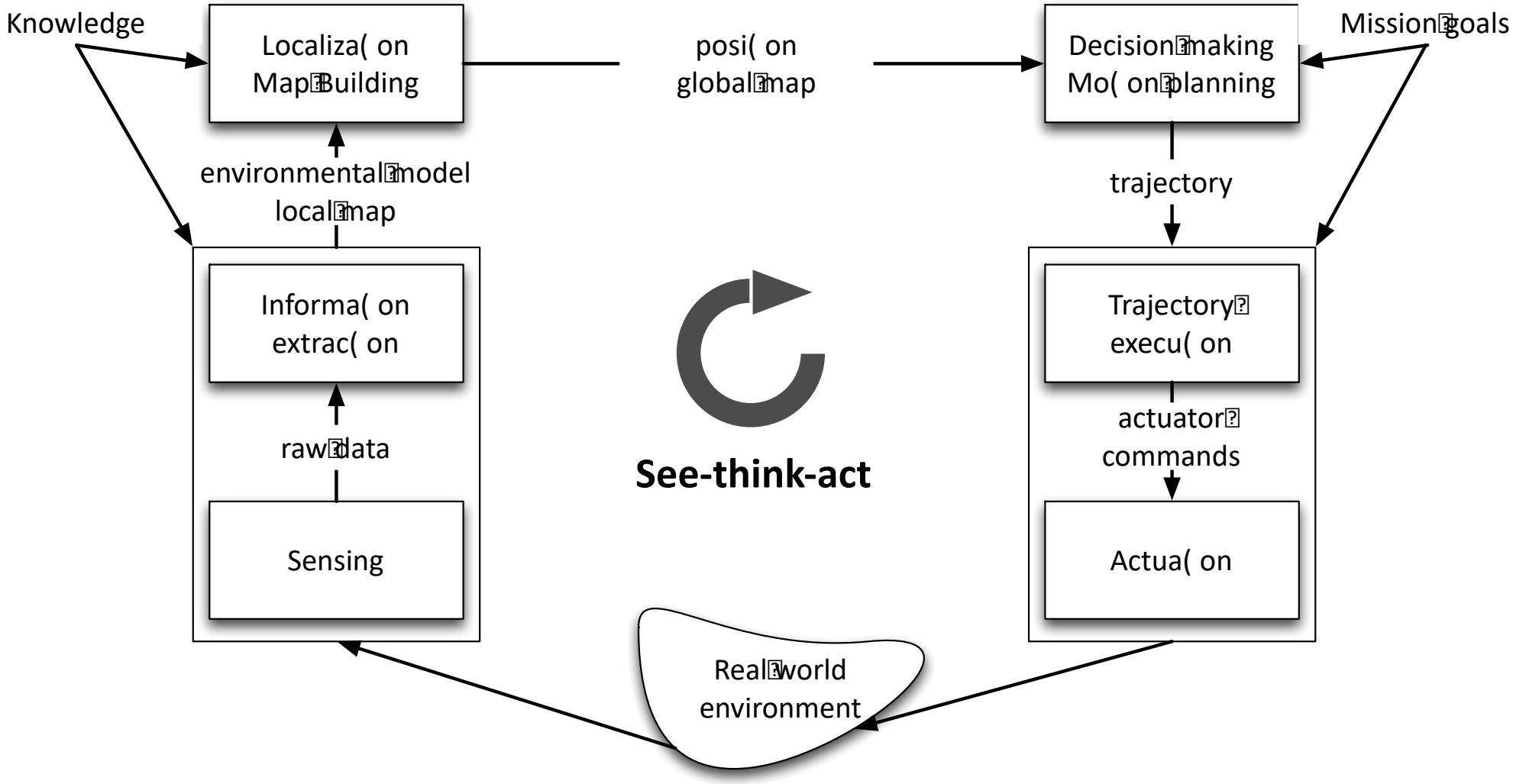
Coalitions, Organizations & Initiatives



Course goals

- To learn the *theoretical*, *algorithmic*, and *implementation* aspects of main techniques for robot autonomy. Specifically, the student will
  1. Gain a fundamental knowledge of the “autonomy stack”
  2. Be able to apply such knowledge in applications using ROS

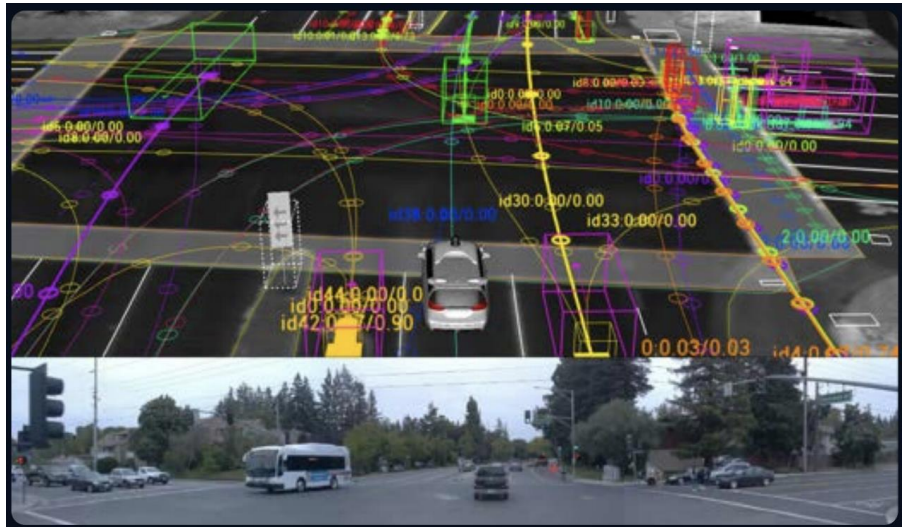
The see-think-act cycle



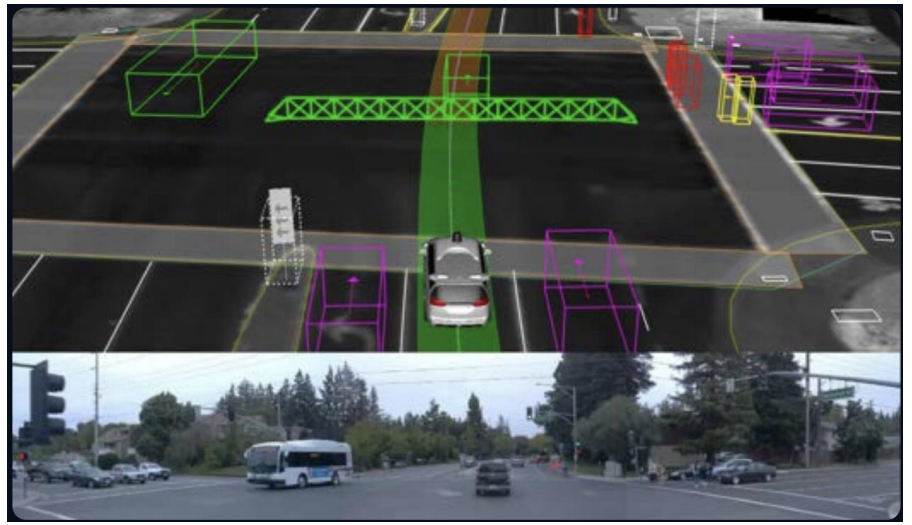
See-think-act cycle for AVs

Note: other architectures are possible and subject of active R&D!

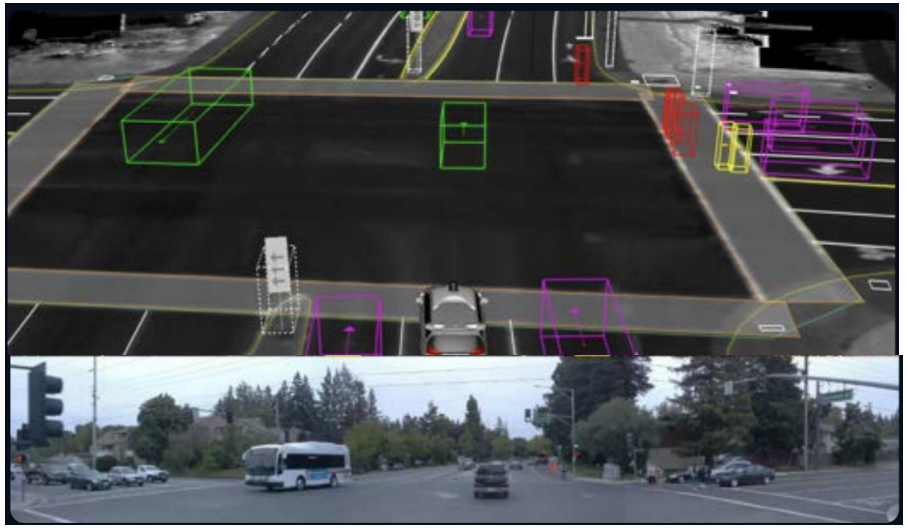
Think



Act



See



Waymo report, 2020

Course structure

- Four main topics:
  1. Robot Operating System (week 1)
  2. Controls & Motion Planning (weeks 2-4) -- *act*
  3. Perception (weeks 5-8) -- *see*
  4. State Estimation, Localization & SLAM (weeks 8-11) -- *think*
- Extensive use of the Robot Operating System (ROS)

- Requirements
  - CS 106A or equivalent
  - CME 100 or equivalent (for calculus, linear algebra)
  - CME 102 or equivalent (for differential equations)
  - CME 106 or equivalent (for probability theory)
  - See also the pre-knowledge quiz on the course website

Logistics

- Lectures:
  - Tuesdays and Thursdays, 10:30am – 11:50am (Lathrop 018)

- Sections
  - 2-hour, once-a-week on Fridays
  - Hands-on exercises that complement the lecture material, build familiarity with ROS, and develop skills necessary for working with hardware
  - Link to the section sign-up sheet
  - Link to hazard acknowledgement and Skilling lab policy

Logistics

- Office hours:
  - Prof. Pavone: Tuesdays, 1:00 – 2:00pm (Durand 261), after class, and by appointment. Office hours will start during the week of Sep 28.
  - Course assistants: Wednesdays, 4:30 – 6:30pm (Durand 270), and Thursdays, 4:30 – 6:30pm (Durand 270)

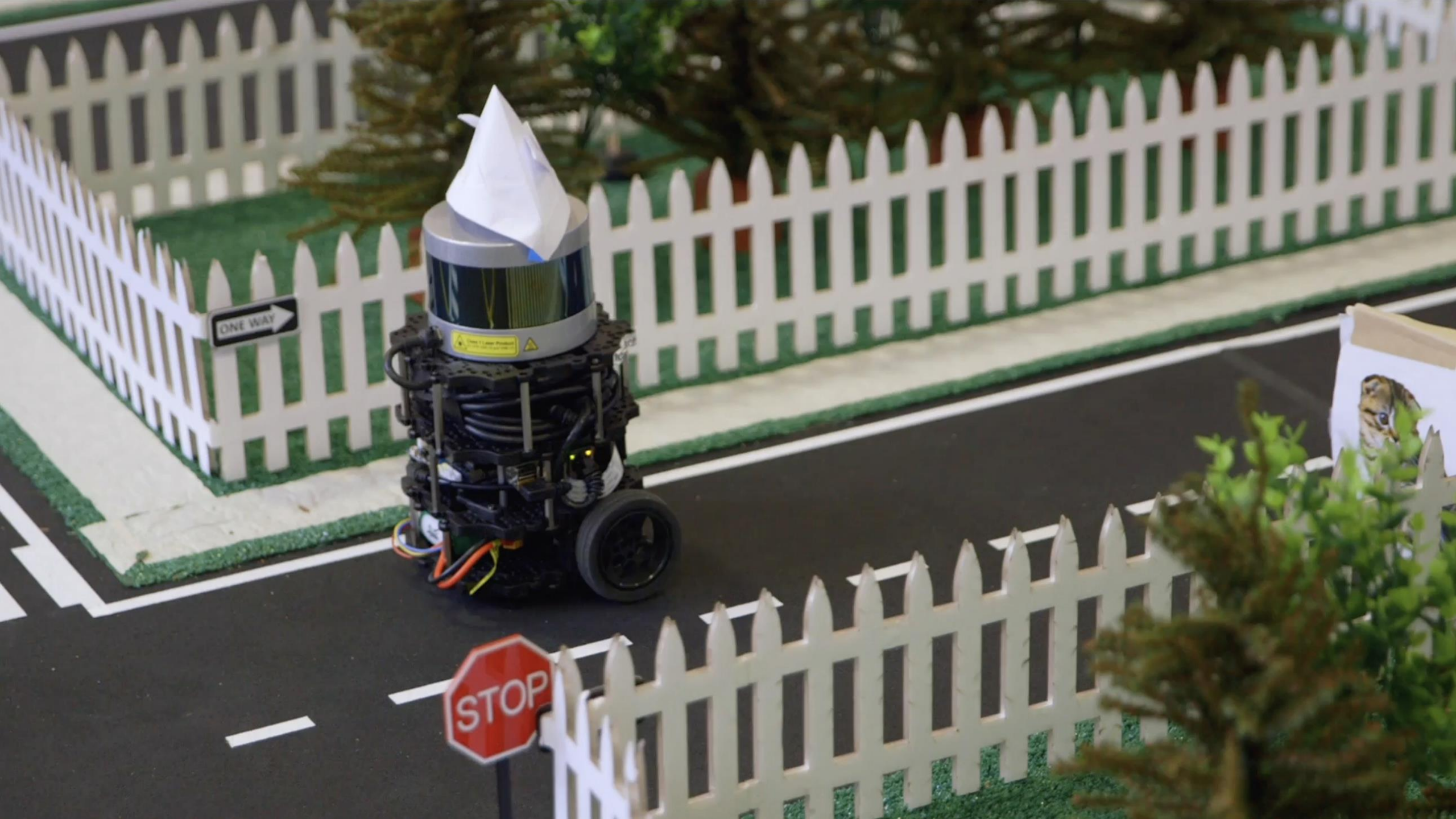
- Course websites:
  - For course content: https://stanfordasl.github.io/PoRA-I/aa174a_aut2627/
  - For course announcements: https://canvas.stanford.edu/courses/231155
  - For course-related questions: https://edstem.org/us/courses/106110/discussion
  - For homework submissions: https://www.gradescope.com/courses/1396967
  - To contact the AA174A staff: aa174a-aut2627-staff@lists.stanford.edu

Grading

- (40%) Homework (4 psets)
- (20%) Midterm exam
- (40%) Sections:
  - (16%) attendance
  - (8%) group participation
  - (16%) final section demo
- (extra 1%) Participation on Edstem
- (extra 4%) Final demo with additional autonomy features
- (extra 5%) In-person classroom attendance; link to record your attendance: https://docs.google.com/forms/d/e/1FAIpQLSeqk3kzr3e8Q2o6d-dpJtnvpVWDPtaVVOLvM_Gano4QQUnboQ/viewform

AI Policy

- AI tools are allowed: Use ChatGPT, coding assistants, and other AI tools freely for homework and projects
- **Use AI to learn, not to replace learning**: Carefully balance AI assistance with developing your own understanding and problem-solving skills
- *You are responsible*: You are responsible for the correctness of your work and for understanding anything you submit, including AI-assisted work
- Midterm: Pen-and-paper exam, with no access to AI tools
- Final demo: We will ask oral questions about your project, implementation, design choices, and underlying concepts
- Bottom line: Use AI as a powerful learning tool—but make sure you are still doing the learning.


ONE WAY
STOP

Team and resources

Instructor

Marco Pavone
Associate Professor AA,
and CS/EE (by courtesy)

CAs

Erik Bauer (Head CA)

Ellie Brosius

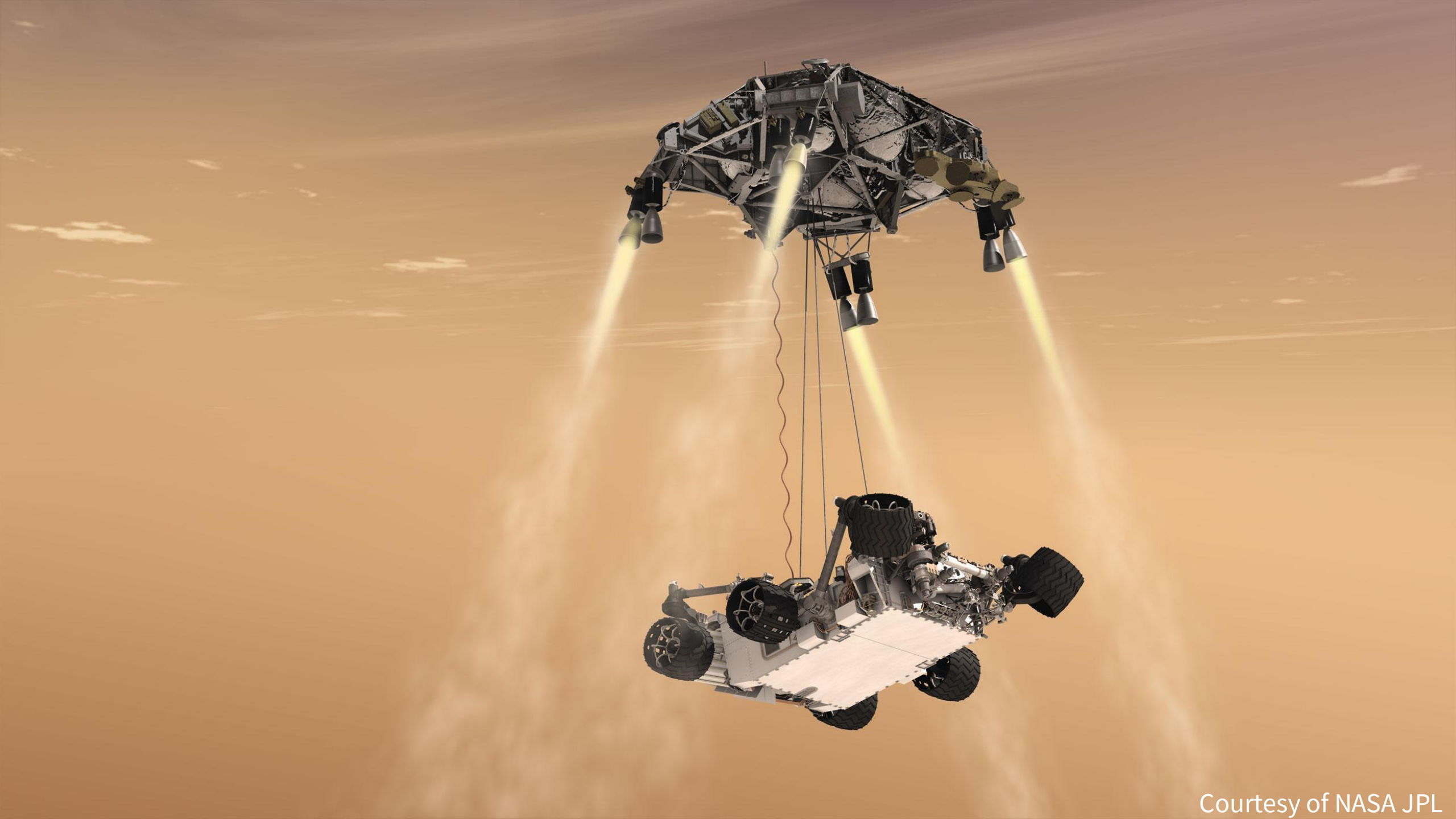
Rohan Garg

Main reference:

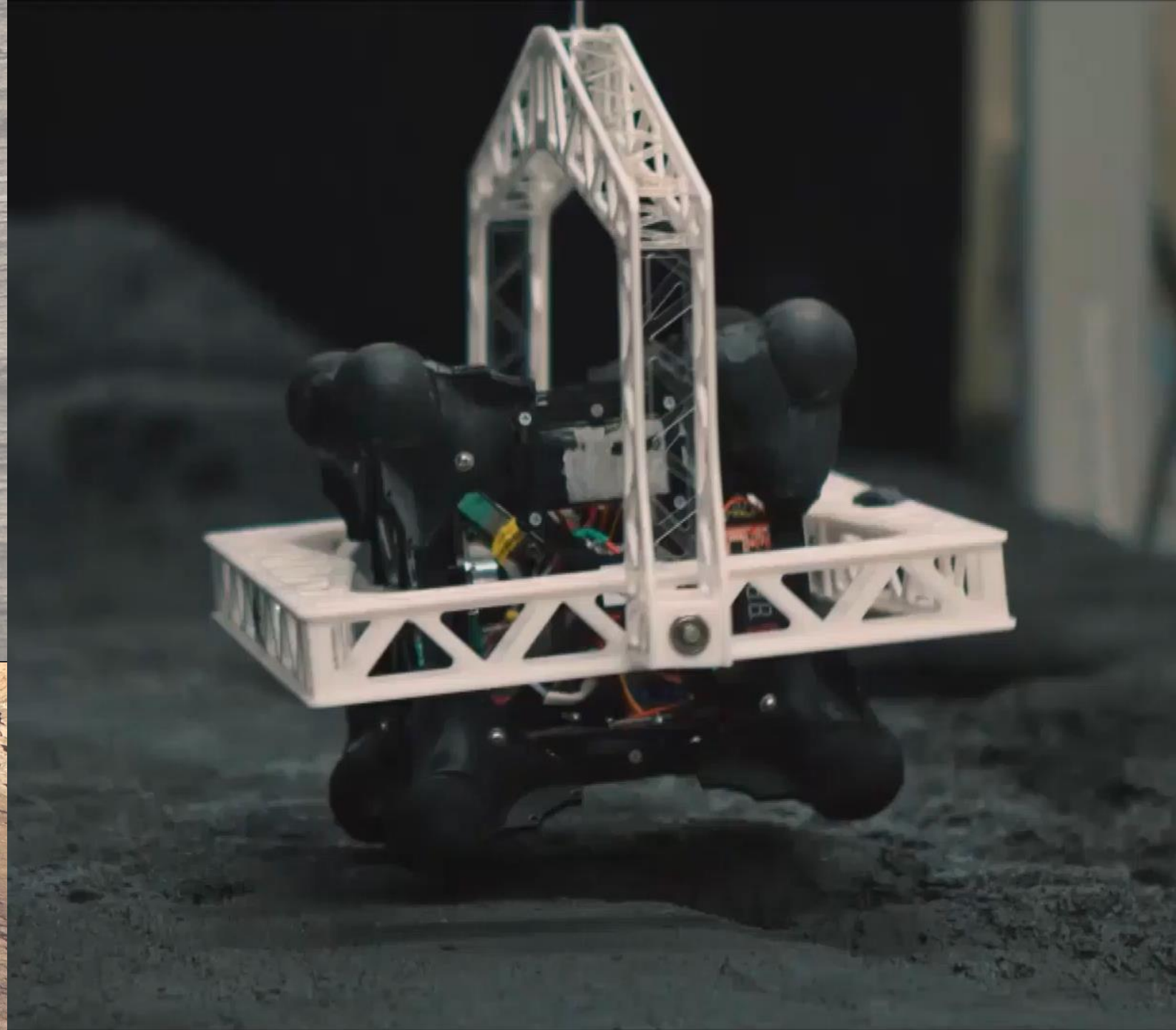
- D. Gammelli, J. Lorenzetti, K. Luo, G. Zardini, M. Pavone. Principles of Robot Autonomy. 2026.
- Book website: https://porabook.com/

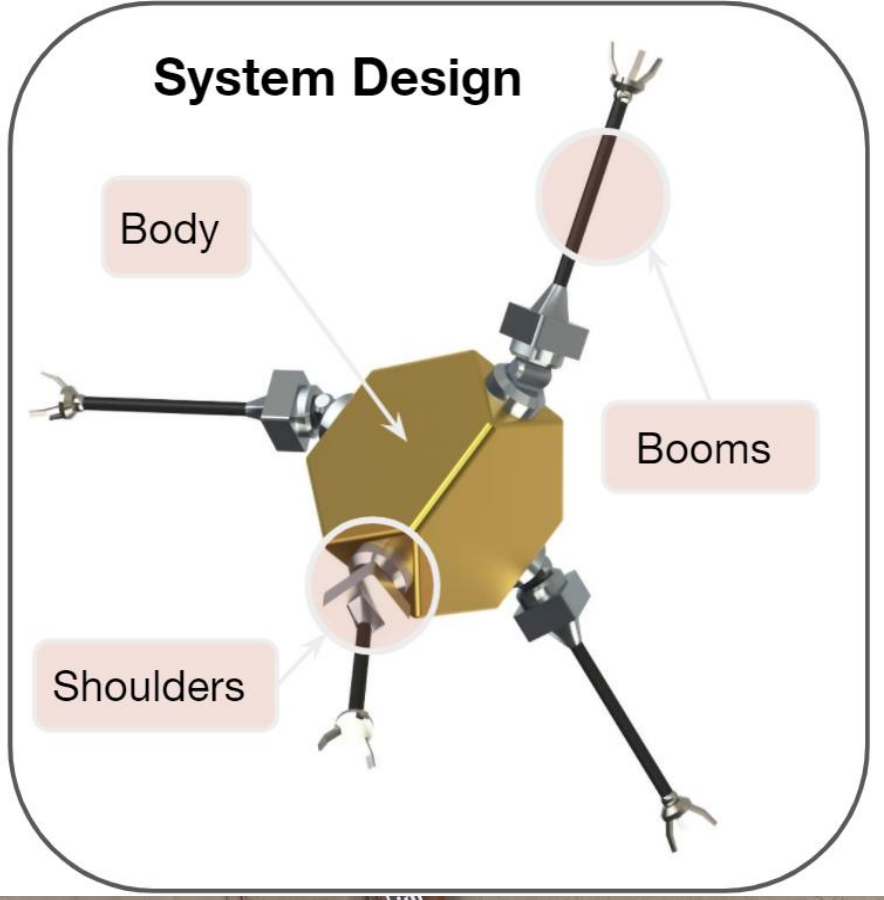
Yousef Liang

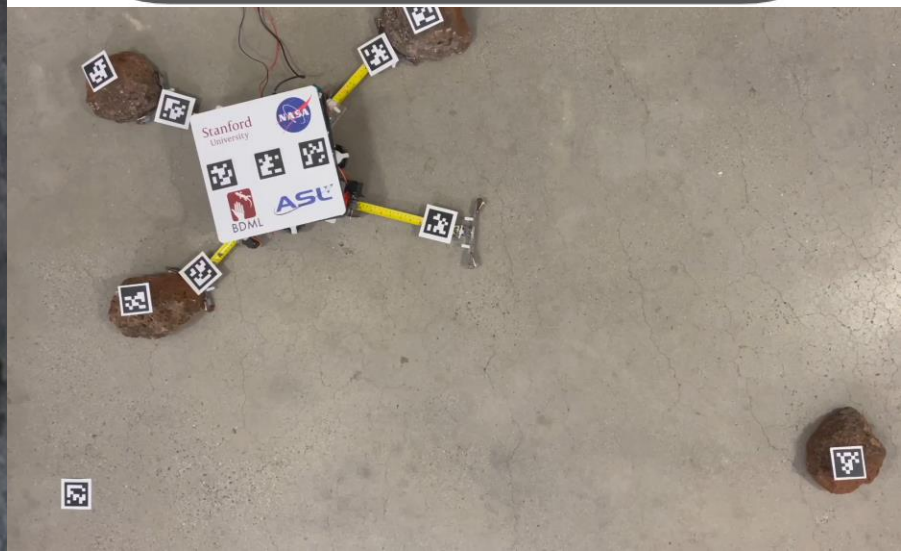
Xilun Zhang



Courtesy of NASA JPL



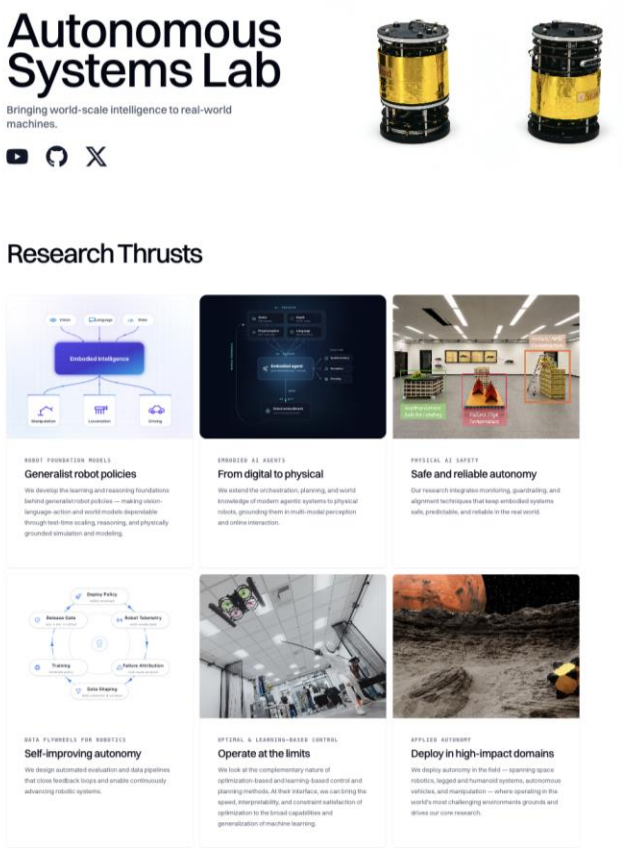

System Design
Body
Booms
Shoulders


Stanford University
NASA
BDML
ASU

FIRE PORT


SP10

Autonomous Systems Lab at Stanford (ASL)



https://stanfordasl.github.io/

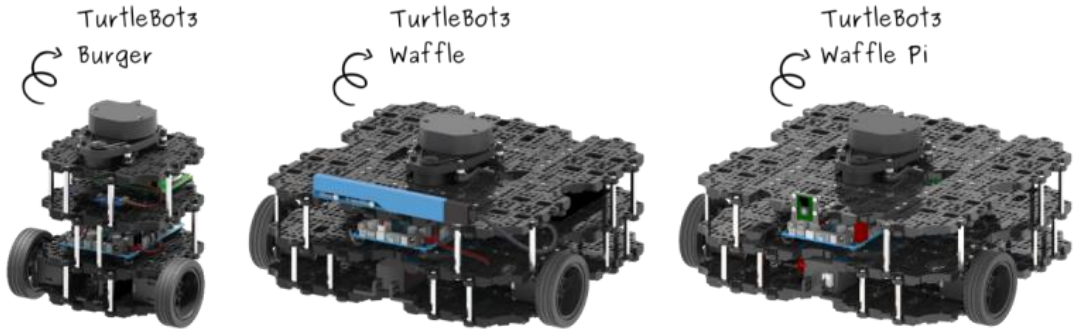
Autonomous Systems and Physical AI Research (ASPIRE)

https://research.nvidia.com/labs/aspire/

Schedule

Date	Topic	Assignments
09/22	Course overview, intro to robotic systems and ROS	
09/24	Fundamentals of ROS & vectorized computation in Python	
09/25	⋆ Section 1 – UNIX, Git, and Python	HW1 out
09/29	State space dynamics – definitions and modeling	
10/01	State space dynamics – computation and simulation	
10/02	⋆ Section 2 – ROS, workspaces, packages, nodes	
10/06	Trajectory optimization	HW2 out
10/08	Trajectory tracking	
10/09	⋆ Section 3 – Launch files & RVIZ	HW1 due
10/13	Motion planning I: graph search algorithms	
10/15	Motion planning II: sampling-based methods	
10/16	⋆ Section 4 – Controller gain tuning in hardware	HW2 due
10/20	Robotic sensors & introduction to computer vision	HW3 out
10/22	Camera models and camera calibration	
10/23	⋆ Section 5 – Running a point-to-point navigator	
10/27	Image processing, feature detection, & feature description	
10/29	Information extraction	
10/30	⋆ Section 5 cont. & Section 6 head start	HW3 due, HW4 (part 1) out
11/03	No Lecture – Democracy Day	
11/05	*In-class midterm*	
11/06	⋆ No Section	
11/10	Deep learning for computer vision	
11/12	Intro to state estimation & filtering theory	
11/13	⋆ Section 6 – Object detection	HW4 (part 2) out
11/17	Parametric filtering (KF and EKF)	HW4 (part 1) due
11/19	Markov localization and EKF-localization	
11/20	⋆ Section 7 – Frontier exploration	
11/24		
11/26	*Thanksgiving*	
11/27		
12/01	Multi-sensor perception & sensor fusion	HW4 (part 2) due
12/03	Simultaneous localization and mapping (SLAM)	
12/04	⋆ Section 7 cont.	

Turtlebot3



- Imagine you have a robot with sensors, motors, and AI.
- How do you make all the pieces talk to each other - reliably, in real time, and in a way others can build on?
- That's where ROS comes in!

Velodyne LiDAR

Quadrotor

Manipulation

TurtleBot

The Challenge of Building Robots

- Robots are complex systems
  - Hardware
    - Sensors (IMU, velocity, LiDAR, Camera, Microphone, …)
    - Actuators (wheels, joints, grippers, valves, ...)
    - Structures (Links, frame, …)
  - Software
    - Control
    - Trajectory planning
    - Object detection
    - Localization
    - …

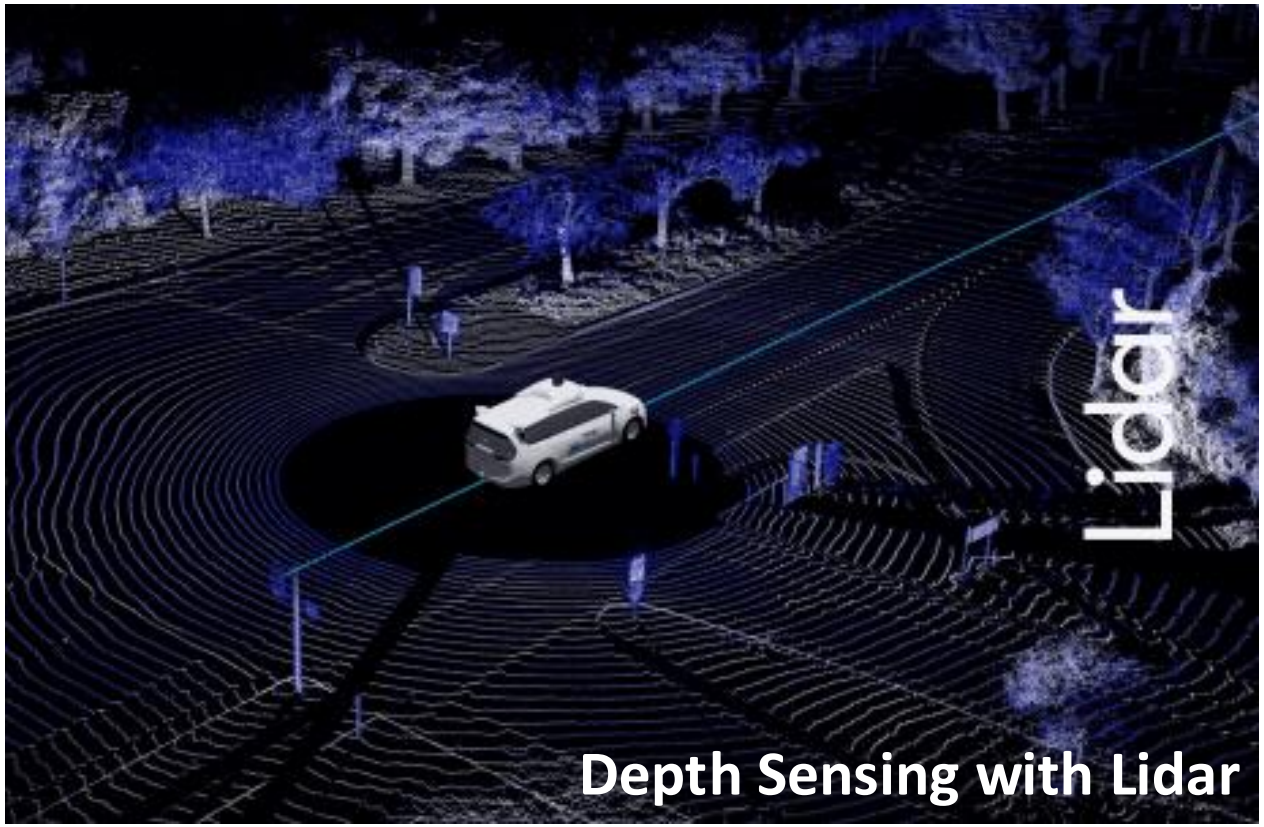
What would we do without ROS?

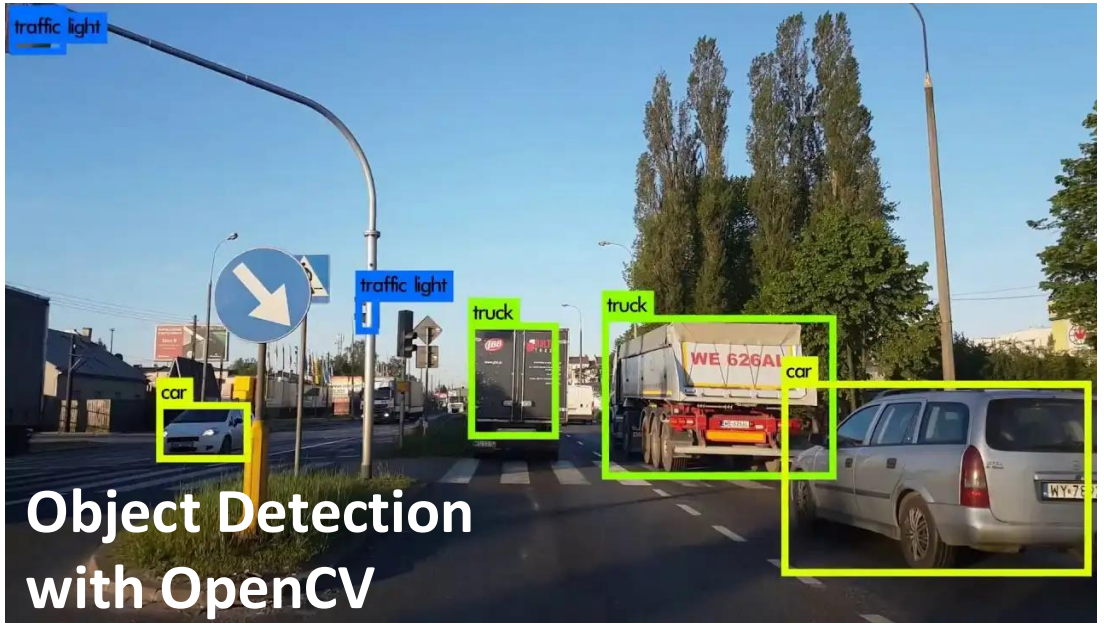
- Write custom code to connect sensors and actuators.
- Reinvent tools for visualization, debugging, and simulation.
- Struggle with versioning, testing, and sharing code.

What is ROS?

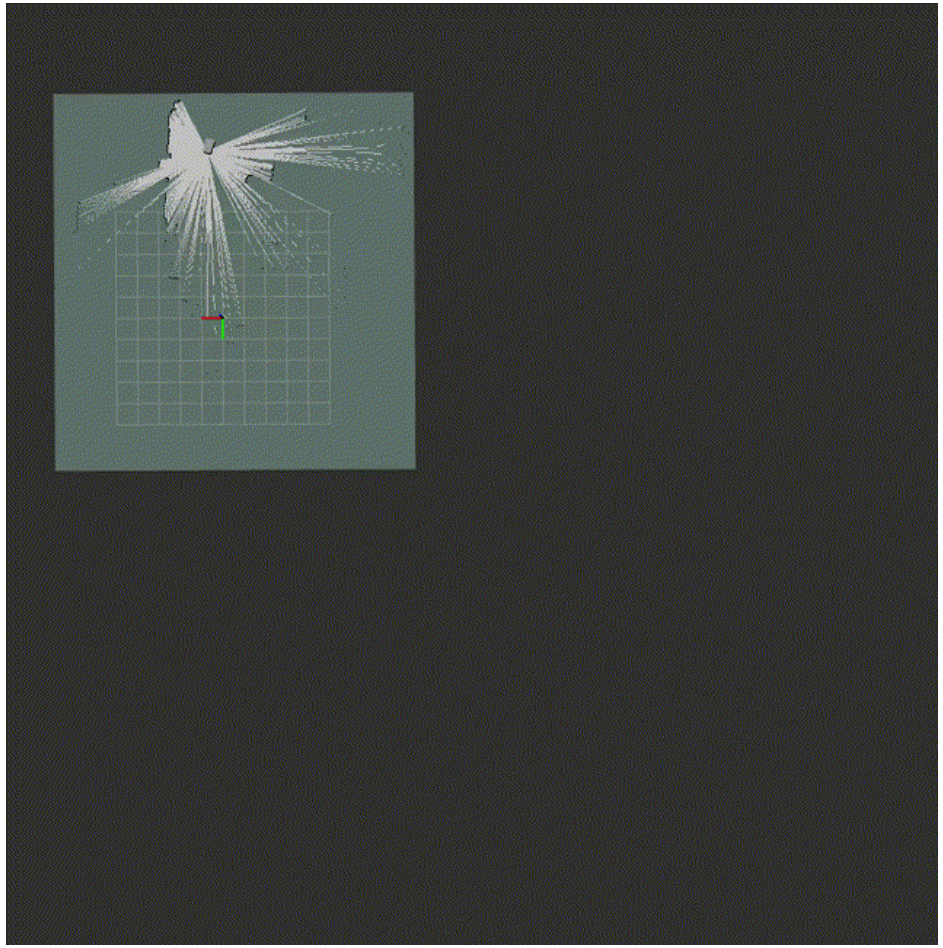
- Middleware (Not an operating system)
- ROS provides:
  - Launching different components
  - Communication between components → Publish/subscribe messaging system.
  - Reusable components → Packages for navigation, vision, motion planning
  - Tools for visualization & debugging → RViz, Gazebo, rqt.
  - Community & ecosystem → Thousands of open-source packages.

Perception


Lidar
Depth Sensing with Lidar


traffic light
traffic light
truck
truck
car
car
Object Detection
with OpenCV

Localization and Mapping



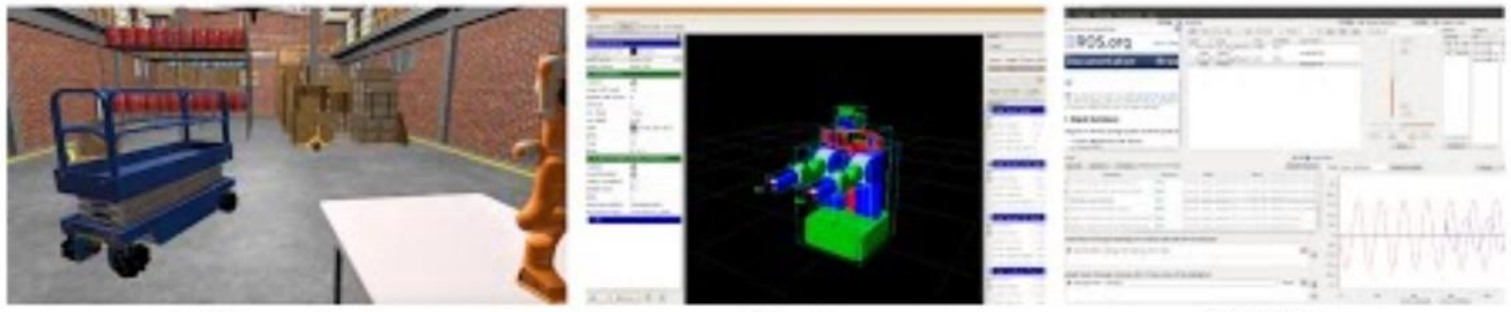
Navigation


Waymo
Navigation
on Freeway


PR2 Navigation around
a Fake Death
MakeAGIF.com

Manipulation

Simulation and Visualization



ROS2 Simulation and Visualization Tools

Multidisciplinary Collaboration

- All disciplines come together
  - Mechanical engineers: Design the robot body and actuators
  - Electrical engineers: Design actuators, sensors, and control
  - Computer scientists / AI researchers: Implement perception and planning
  - Design experts: Build human-robot interfaces via ROS APIs.

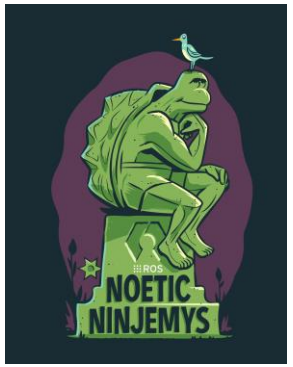
Robot Operating System – History

ROS 1

2014 - 2019

2016 - 2021

2018 - 2023



2020 - 2025

ROS 2

2019 - 2021

2020 - 2023

2022 - 2027

2017 - Present

Key Takeaways

- ROS is a framework that connects all parts of a robot.
- It saves time, encourages collaboration, and accelerates innovation.
- ROS brings robot autonomy to reality.
- The ROS ecosystem is vast and growing.
- Next lecture: *How ROS actually works* — topics, nodes, packages, ...

Next time: fundamentals of ROS

Robot Operating System