Cloud Robotics

- Cloud Robotics: Architecture, Challenges and Applications
- Cloud-Based Robot Grasping with the Google Object Recognition Engine

Definition

- Any robot or automation system that relies on data or code from a network to support its operation.

  --James J. Kuffner Jr. (2010)

- Allow robots to offload compute-intensive tasks like image processing and voice recognition and even download new skills instantly.

- Benefits: make robots "lighter, cheaper, and smarter"


Robot Cloud

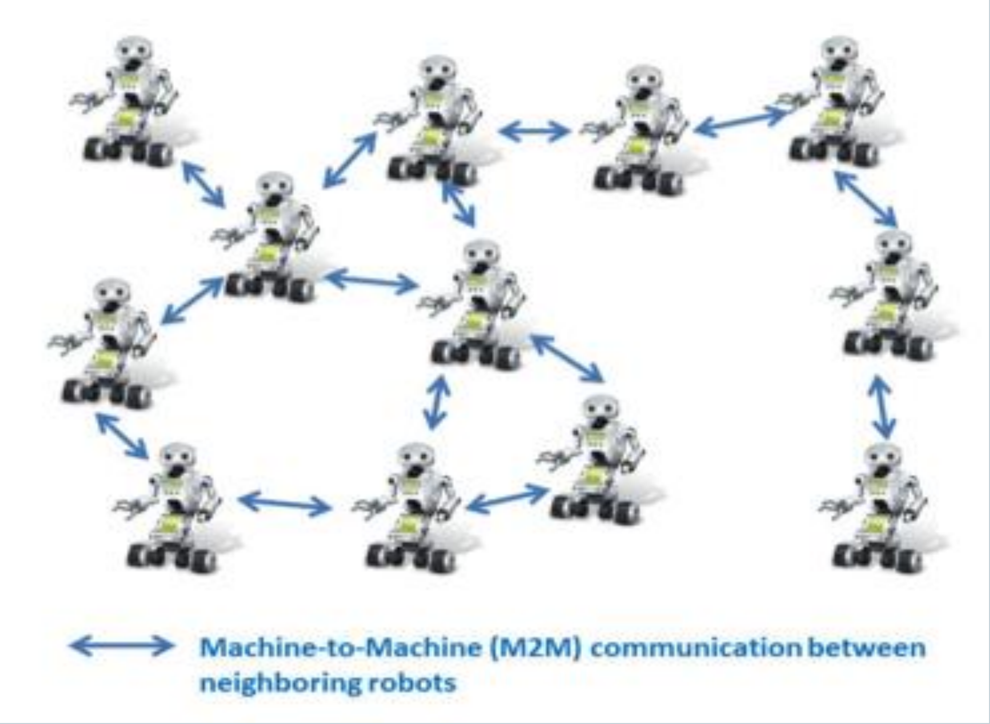
Cloud Robotics:
Architecture, Challenges and Applications

Guoqing Hu, Wee Peng Tay, and Yonggang Wen

Nanyang Technological University

Introduction

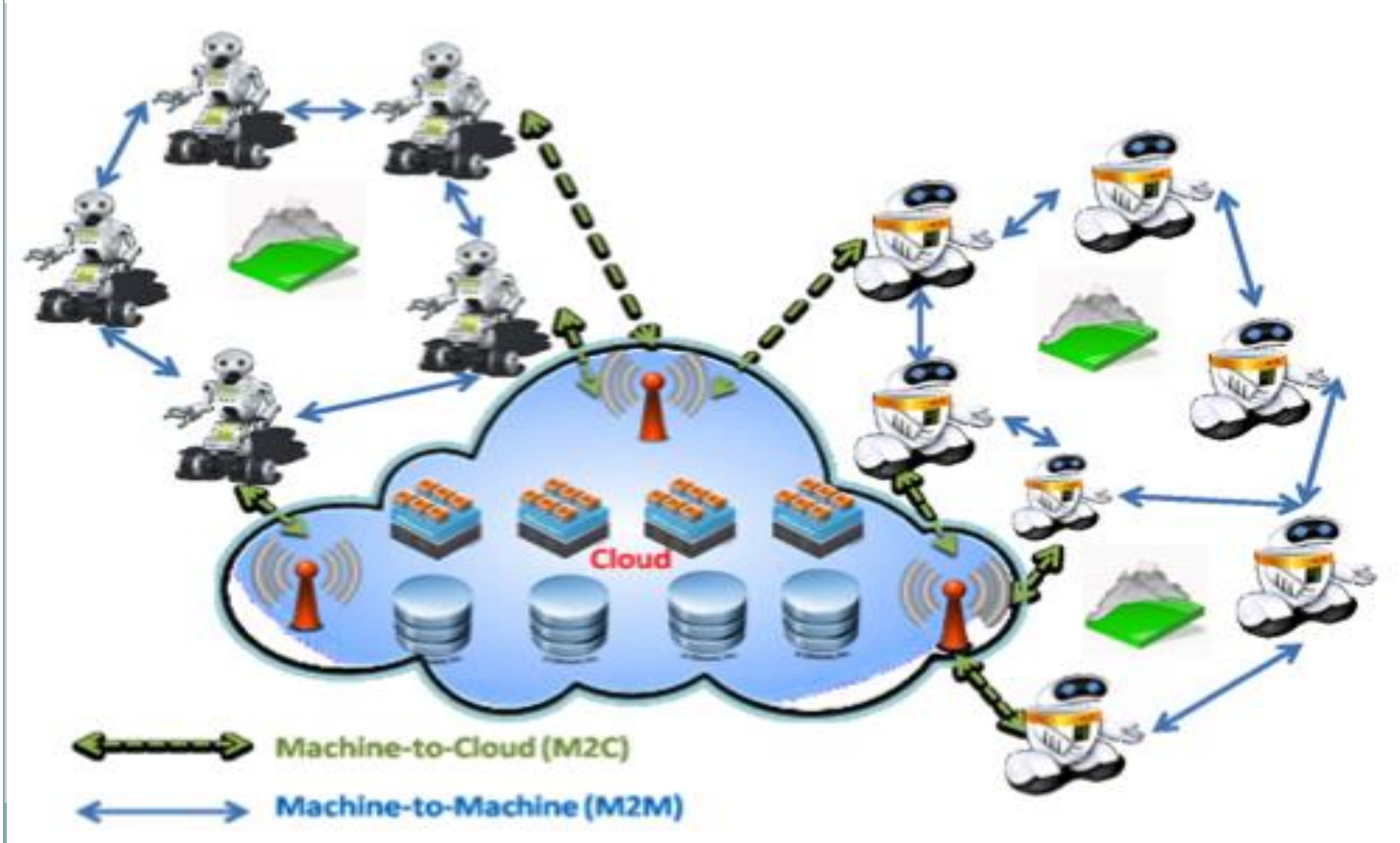
- Network robotic system: a group of robotic devices that are connected via a wired or wireless communication network. (M2M)



Introduction

- Limitation of Networked Robotics:
  - Resource Constraints: Limited by each robot’s resources, including computing units, memories, and storage space
  - Information and Learning Constraints: Limited by the information observed, the scenarios encounters, and collective amount of storage space of the team
  - Communication Constraints: Limitation of proactive routing and ad hoc routing

Cloud Robotics



System Architecture

- System architecture:
  - M2M: machine-to-machine level
  - A group of robots communicate with each other to form a collaborative computing unit.
  - M2C: machine-to-cloud level
  - Resources are dynamically allocated from a shared resource pool in the cloud (Support task offloading & information sharing)

M2M Communication Architecture

- Gossip algorithms: transmit a message from a source to a destination by randomly choosing neighbor
- Benefits: do not require route discoveries and maintenance, highly dynamic!
- Trade-off: high message latency (Not a problem as the cloud serves as a central super node for M2M)
- Discussion: what if there is a fault in the network? Is this better than learning routes at the same time?

Elastic Cloud Computing Architecture

- Three elastic computing models:
  - Peer-Based Model
  - Proxy-Based Model
  - Clone-Based Model

- Choosing factors:
  - Network conditions
  - Application requirements
  - Resource availability

Elastic Cloud Computing Architecture

- Peer-Based Model: each robot and each virtual machine (VM) in the cloud is considered as a computing unit.

(a) Peer-Based Model

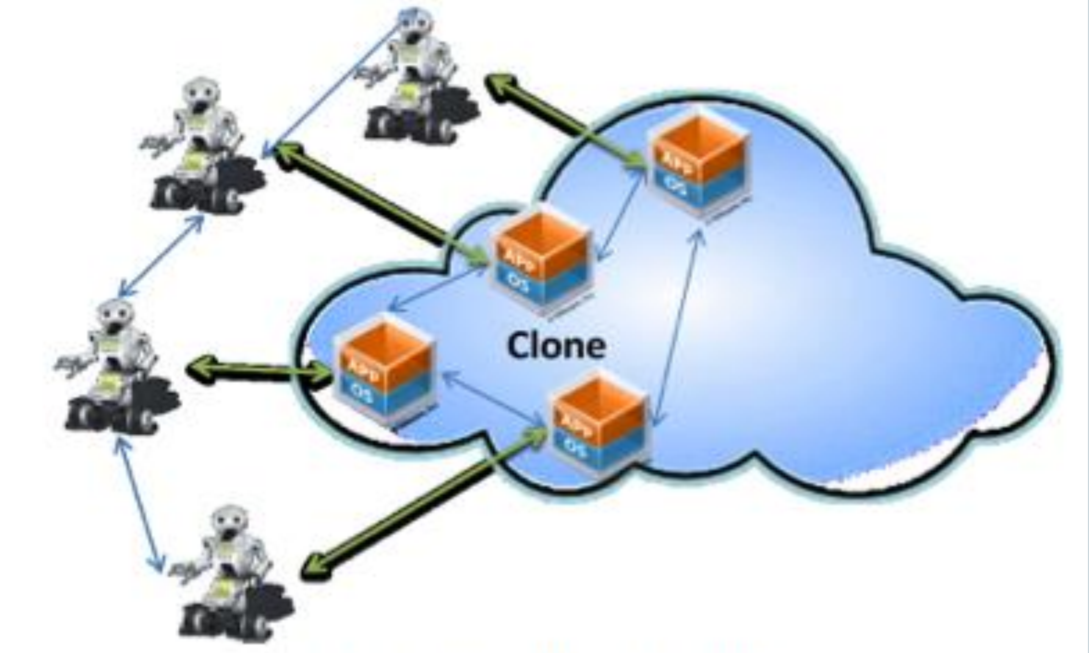
Elastic Cloud Computing Architecture

- Proxy-Based Model: one unit functions as a group leader, communicating with a proxy VM in the cloud infrastructure. (Two-tier hierarchy structure)

(b) Proxy-Based Model

Elastic Cloud Computing Architecture

- Clone-Based Model: each robot corresponds to a system-level clone in the cloud. Peer-to-peer between clones.



(c) Clone-Based Model

Comparisons of Different Computing Models

Model	Robustness	Interoperability	Mobility
Peer-based	Medium	Medium	High
Proxy-based	Low	High	Medium
Clone-based	High	Low	Low

Robustness: network connectivity between the set of networked robots and the centralized cloud infrastructure.

Interoperability: additional complexity required in operating a cloud robotics infrastructure with an existing robotic network.

Mobility: the capability of supporting mobile robots.

Questions: Why the model High for each property?

Technical Challenges

- A. Computation Challenges

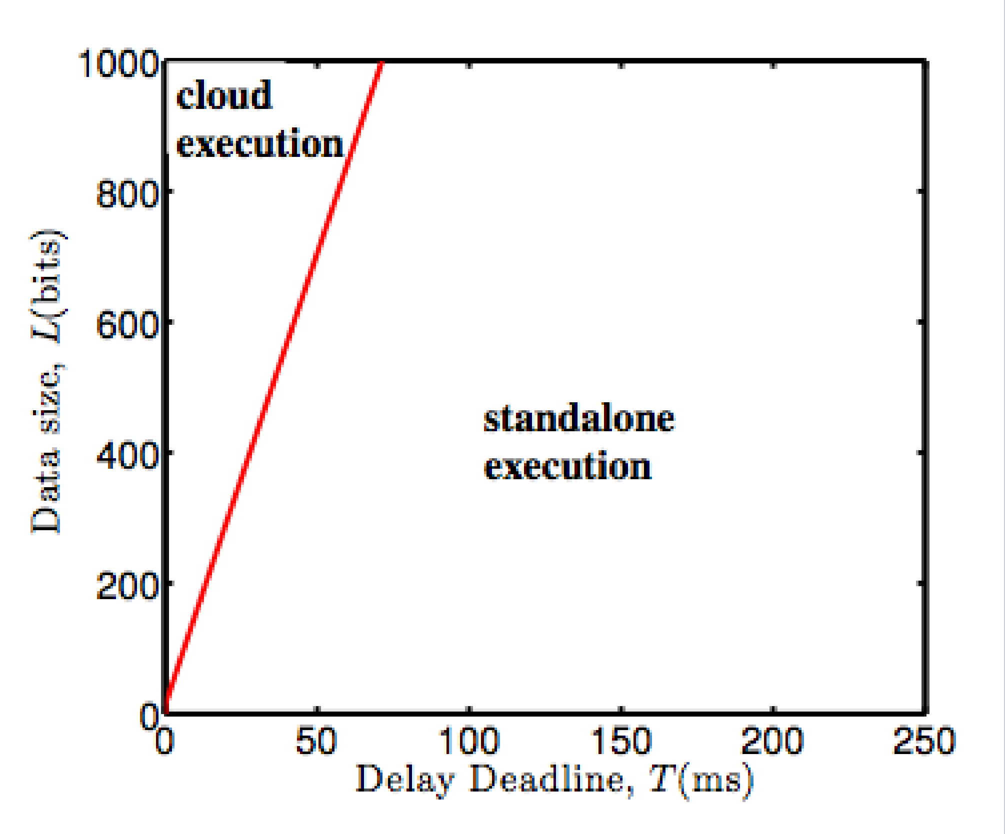
- In terms of offloading computationally intensive tasks to the cloud for execution.

- Objective: minimize the amount of energy consumed by the robot, under the constraint that the task should be completed within a specified deadline.

- Trade-off: the energy consumed for executing the task by the on-board CPU within the robot vs. the energy consumed for transmitting the amount of data to the cloud for remote execution.

Technical Challenges

- A. Computation Challenges



Technical Challenges

- A. Computation Challenges

- Discussion: In terms of offloading computationally intensive tasks to the cloud, besides energy consumed, what other factors might also need to be considered?

Technical Challenges

- B. Communication Challenges (Delay)

- C. Optimization Framework
  - Make decision among three execution strategies:
  - 1) Standalone execution by the individual robot
  - 2) Collaborative execution
  - 3) Cloud execution

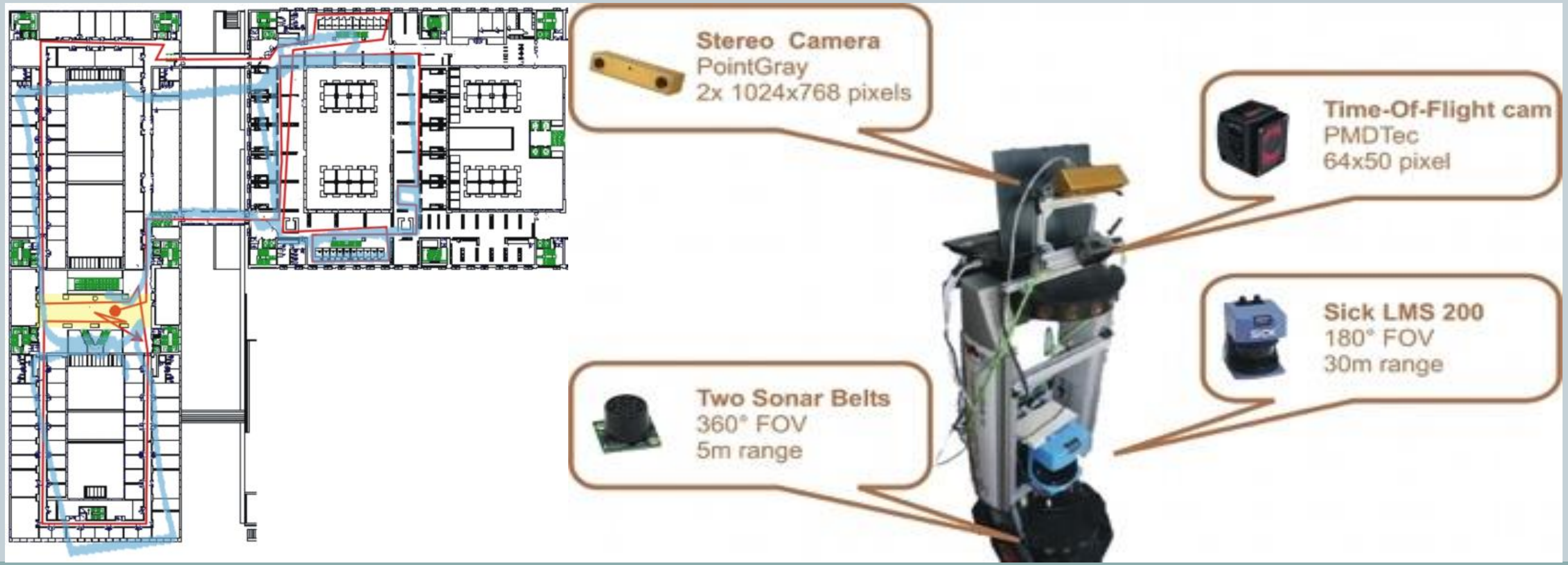
- D. Security Challenges (VM & Robot)

Applications

- Advantages:
- 1) Ability to offload computation-intensive tasks to the cloud.
- 2) Access to vast amounts of data.
- 3) Access to shared knowledge and new skills.

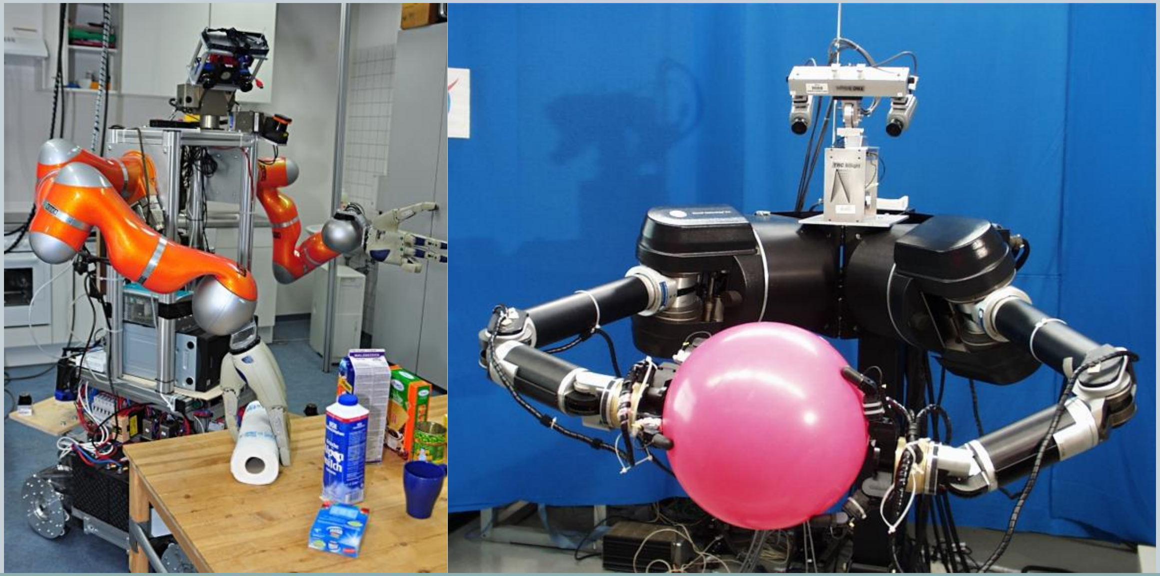
Applications

- SLAM (Simultaneous Localization And Mapping): a technique for a robot to build a map of the environment without a prior knowledge, and to simultaneously localize itself in the unknown environment.



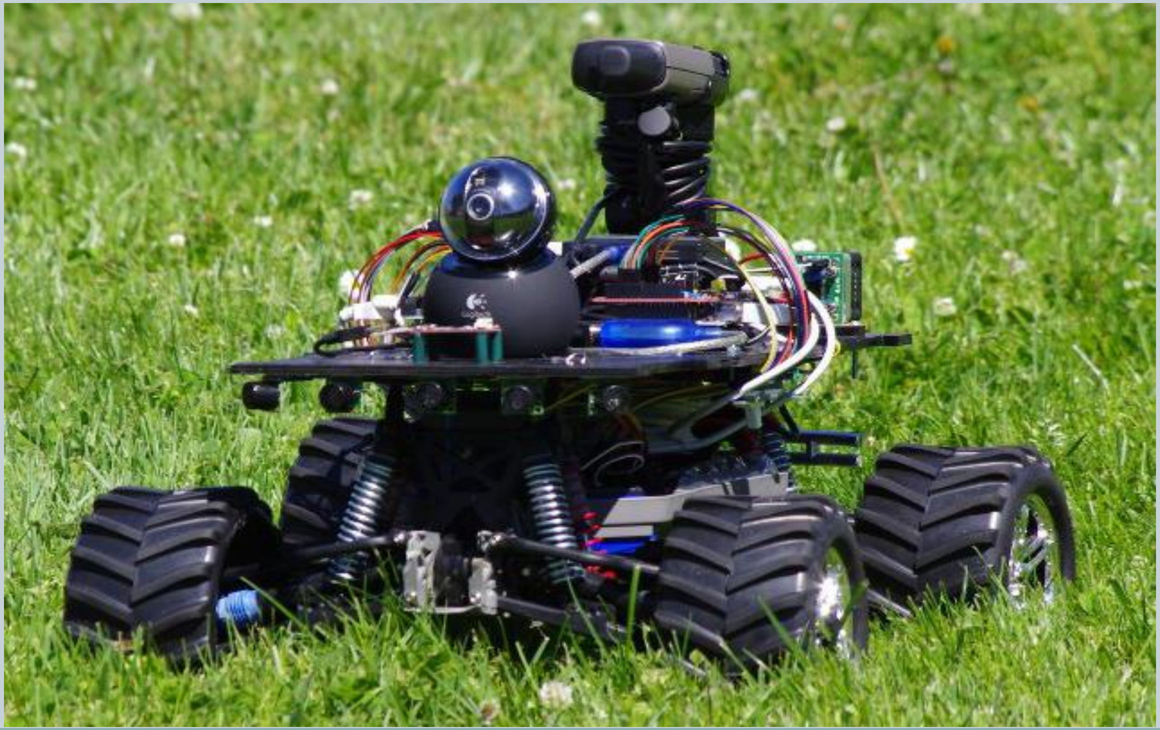
Applications

- Grasping: If the object is unknown or not precisely known, the problem involves the access and preprocessing of vast amounts of data and can be computationally intensive. (Access to large databases & shared knowledge of new objects learned)



Applications

- Navigation: the process of building the map requires large amounts of storage space, the process of searching a map requires access to large amounts of data.



Discussion

Cloud-Based Robot Grasping with the Google Object Recognition Engine

Ben Kehoe, Akihiro Matsukawa, Sal Candido
James Kuffner, Ken Goldberg

University of California, Berkeley
Google

Background

- Robots operating in unstructured environments such as homes and offices have limits on onboard computation and limited data on the objects they encounter.

- Focusing on recognizing and grasping common household objects.

Introduction

- Advantages of Cloud Robotics in grasping:
  - Indexing vast libraries of annotated image and object models.
  - Massively parallel sample-based motion planning and uncertainty modeling.
  - Sharing of outcomes, trajectories, and dynamic control policies.
  - Obtain on-demand human guidance when needed.

Training + Execution

- Steps:

  ① Train an object recognition server on a set of objects;

  ① Tie it to a database of CAD models and candidate grasp sets;

  ① Select the candidate grasp sets based on previous work;

  ② Robot uses the object recognition service;

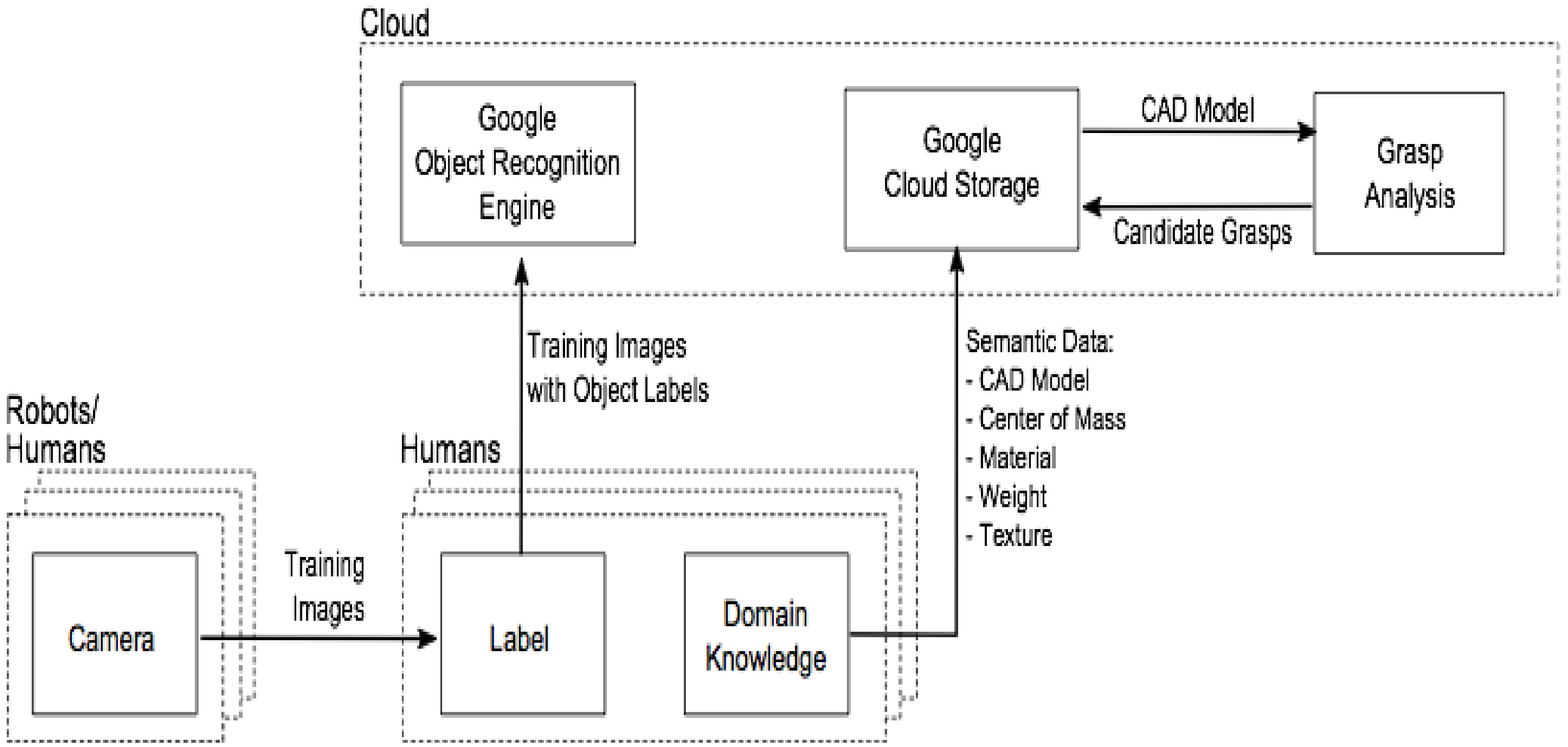
  ③ Robot uses the reference model to perform pose estimation of the object using the robot's 3D sensors;

  ④ Grasp the object.

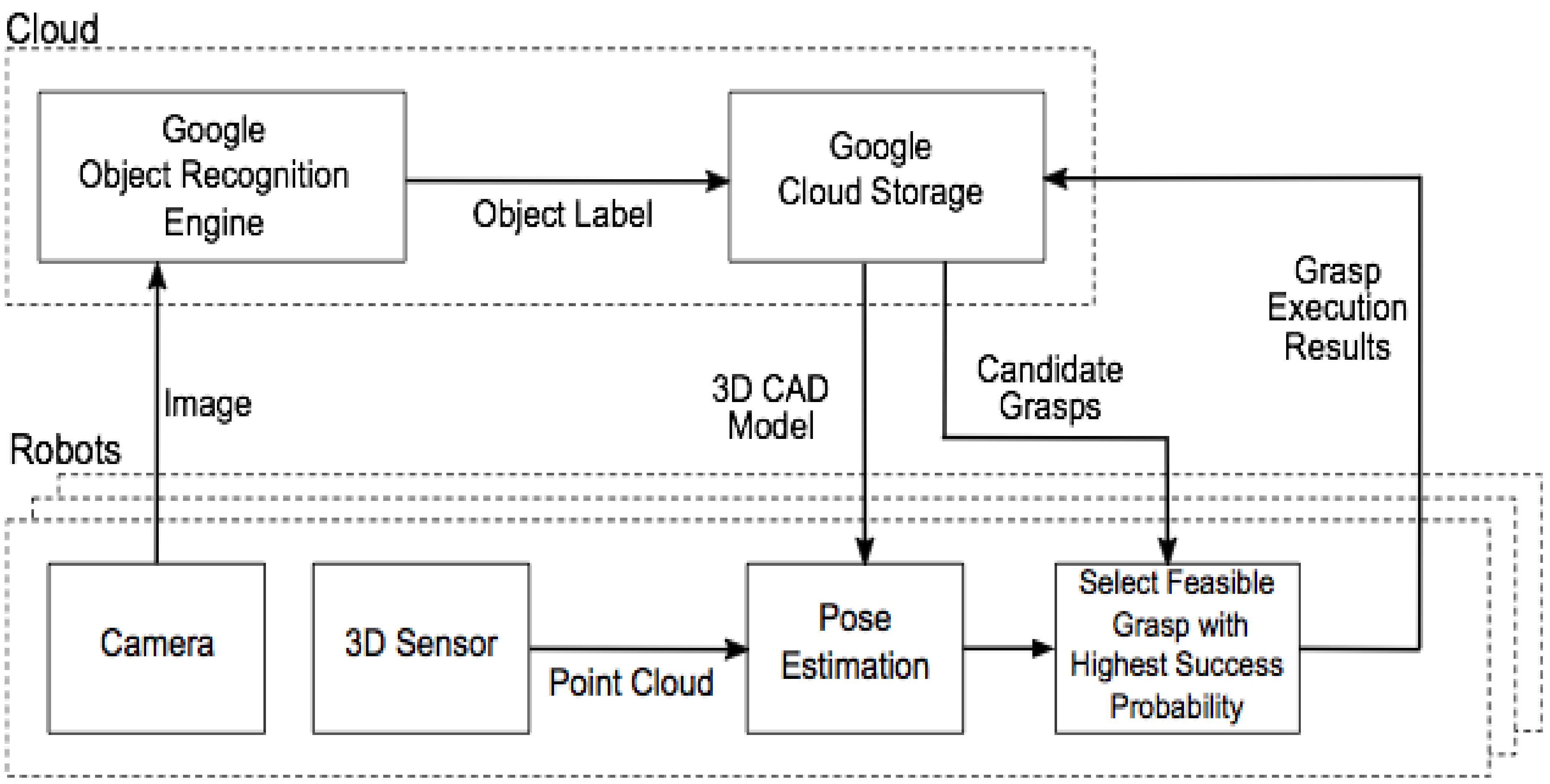
System Architecture

- Two phases:
- 1) Offline:
  - Training of the object recognition server
  - Creation of object reference
  - Creation and analysis of candidate grasp set
- 2) Online:
  - Take a photo of the object
  - Send it to the object recognition server
  - Server returns the stored data for the object
  - Perform pose estimation and selects a grasp from candidate
  - Attempt the grasp, stored the results for future reference

System Architecture for offline phase



System Architecture for online phase

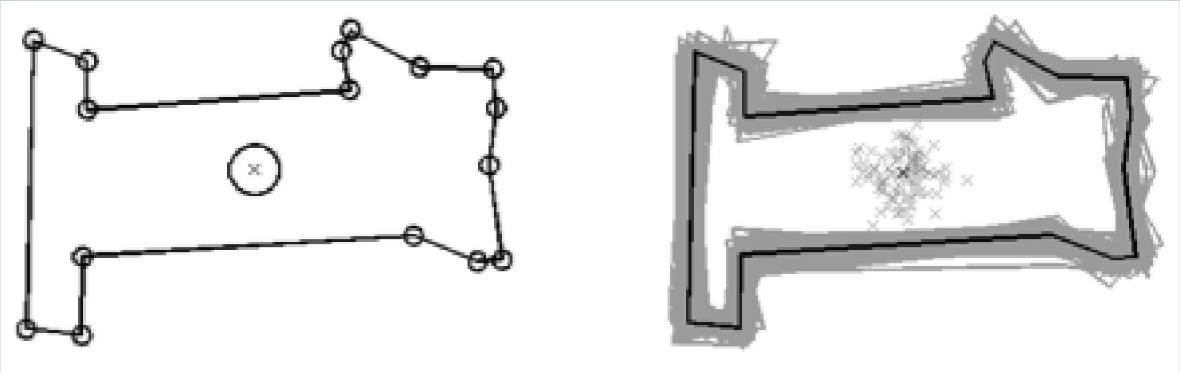


Object Data

- Object Data: both offline and online phases
- Google Goggles server: return the object’s identifying string based on uploaded images.
- Google Cloud Storage: used to query the identifying string for the reference data.
- Microsoft Kinects: scan all sides of the object.
- Point Cloud Library (PCL): filter the point cloud and find out reference point set for the object, create a 3D CAD model.

Robust 3D Grasp Analysis--Offline

- Use Columbia University GraspIt! System
  - Assume a Gaussian uncertainty distribution for each vertex and center of mass
  - Use 2D geometric features to determine if the gripper can push the object into a stable orientation
  - Sample over perturbations in object pose and test candidate grasps in the system
  - Find out the best grasp based on the 3D model



Pose Estimation and Grasp Selection--Online

- Use ICP, Iterative Closest Point method in PCL, to perform a least-squares fit between the measured 3D point cloud and reference point set.
- Use OpenRAVE, a robotics motion-planning library, to plan the grasp.

Experiments

Excperiments

- Object Recognition: different training sets (A, B, C, R), find out the recall of the test sets

Training Set	Size	Recall	Recall Rate	Training Time (s)	Recall Time (s)
R	228	307/387	0.79	0.45	0.29
A	92	247/422	0.59	0.40	0.29
B	52	215/422	0.51	0.39	0.28
A+B	144	317/422	0.75	0.40	0.29
C	49	199/422	0.47	0.39	0.30
A+B+C	193	353/422	0.84	0.40	0.29

Excperiments

- Pose Estimation: use 15 stable poses for each object, failure when the estimated pose is more than 5mm or 5 degrees from the ture pose.

Object	Total Trials	Failures	Failure Rate	Average Time (s)
Air freshener	15	2	0.13	7.4
Candy	15	0	0.00	1.4
Juice	15	1	0.07	10.2
Mustard	15	2	0.13	10.6
Peanut butter	15	2	0.13	2.1
Soap	15	0	0.00	3.6

Excperiments

- Grasp: for cases pose estimation is successful, having the system execute a grasp and attempt to lift the object off the worksurface.

Object	Candidate Grasp Set Size	Total Trials	Failures	Failure Rate
Air freshener	76	13	2	0.15
Candy	30	15	3	0.20
Juice	105	14	1	0.07
Mustard	61	13	3	0.23
Peanut butter	80	13	2	0.15
Soap	30	15	0	0.00

Next Step

- Do more experiments with larger object sets
- Study how more accurate CAD models may affect pose estimation and grasping
- Refine each phase of algorithm

Discussion

Thoughts

- Discussion: Will the Artificial Intelligence (AI) surpasses human intelligence?

  (Background: Stephen Hawking, Elon Musk, and Bill Gates issued warnings about the dangers of AI and robotics.)

Thank You!