DR 8.4: Proceeedings of the NIFTi summer school Yr1 – Reasoning in the Robot World 2014

Tomáš Svoboda, Václav Hlaváč*

*CTU in Prague

⟨svobodat@fel.cvut.cz⟩

Project, project Id:	EU FP7 TRADR / ICT-60963
Project start date:	Nov 1 2013 (50 months)
Due date of deliverable:	December 2014
Actual submission date:	February 2015
Lead partner:	Czech Technical University in Prague
Revision:	final
Dissemination level:	PU

This deliverable describes the first year Summer school organized by CTU in Prague. The main topic of the school was Reasoning in the Robot World. The school took place at CTU Campus in Prague, July 28 – August 1, 18 talks was delivered, each 50 minutes long. Altogether six invited speakers were lecturing.

Executive Summary

This deliverable describes the first TRADR summer school. After some discussions the consortium converged to the topic *Reasoning in the Robot World* as the reasoning about data lies in the very heart of the foreseen TRADR system. The school was organized by CTU and took place in the CTU Prague campus, from July 28 till August 1. Six invited speakers were lecturing. More than 40 participants were present, around half from them were from the TRADR consortion. The summer school inspired very vivid inspiring discussion. The impact was deepened by a one-day meeting of the TRADR consortium members that followed the school immediately.

Role of the summer school in TRADR

The general role of the yearly summer schools is to gain new knowledge and disseminate experience. As this was the first year school, gaining new knowledge was the prevailing goal.

Contribution to the TRADR scenarios and prototypes

The year 1 summer school deliberately helped the members of the TRADR consortium as a discussion starter. The Year1 summer school helped the TRADR consortium to gain new knowledge and stimulated discussions about world representation, storing data for sharing among robots and sorties and about the overall knowledge base architecture.

1 Tasks, objectives, results

1.1 Planned work

The project proposal plans summer schools organized yearly. For the Year 1 it was decided that CTU in Prague organizes it.

1.2 Actual work performed

The complete event took place in Prague at the CTU school campus from Mon 28 July to Fri 1 August. The first afternoon and the last day served as discussion days for the TRADR consortium. Sixe invited speakers were lecturing, see `http://summerschool2014.ciirc.cvut.cz/speakers/`. Mostly, each delivered three talks, 50 minutes each. In addition to that one talk was delivered by prof. Hlaváč (CTU in Prague) and M. Achtelik (Ascending Technologies). All the material are public at the summer school homepage `http://summerschool2014.ciirc.cvut.cz/`

All lecture slides are available on the web, `http://summerschool2014.ciirc.cvut.cz/program/`. For the sake of completeness the slides were put 6 on one page and included in the public version of this deliverable.

Invited Speaker	**Topic**
Peter Abbeel	Motion planning and reinforcement learning in robotics
Roman Barták	Robot world representation and reasoning in it
David Vernon	Cognitive Robotics
Stan Birchfield	Perception – action loop in practice
Hans Georg Stork	Robots, Research and Responsibility
Michael Zilich	Knowing and acting in an uncertain world – a practical view.

Table 1: List of invited speakers and the lectured topics.

2 Annexes

2.1 Summerschool main page

This annex includes the main information page of the summerschool

Summer School – Reasoning in the Robot World 2014

Prague, Czech Republic 29-31 July, 2014. Hosted by the Czech Technical University in Prague, Center for Machine Perception

Co-financed by EU Project TRADR (FP7-ICT-609763). Talks in pdf available, see Program

Home/topic

Photo of ReaRW 2014 participants taken on July 29, 2014

Full resolution photo (3795 x 2112 pixels)

ReaRW aims

The summer school **Reasoning in the Robot World 2014** (abbreviated ReaRW) had **four aims**:

1. **Teaching** its participants the topic of **the robot world perception, representation, reason-**

ing/planning, and learning in it. (more about the topic in the sequel)

2. **Formulating issues** related to ReaRW 2014 topics / **answering some of them**. Some these issues are generated by projects TRADR (long-term multi-robot missions in search & rescue scenarios) and CloPeMa (dual-arm manipulation with soft materials). These EU funded projects co-finance ReaRW 2014.
3. **Establishing personal links and friendships** among ReaRW participants. Participants may display their posters to show others actively what do they do in their own research.
4. **Introducing participants to the relevant research** environment and project results at the Czech Technical University in Prague.

ReaRW topic/task description

The recent development in autonomous robotics has been requiring the ability to perform movements, manipulation, navigation, etc. in a real, noisy and dynamically changing world. Let us call it *ReaRW task*.

The ReaRW task has been popular with the artificial intelligence community since 1960s. The task has penetrated and expanded in pattern recognition / machine learning, computer vision and robotics since. There were many attempts to provide a general method/solution to the task. The bad news is, however, that there has been no general solution to ReaRW task yet. The good news is that there were ReaRW success stories in particular applications. The most advanced and known also by the general public is, probably, the self-driving autonomous automobile, e.g. the Daimler-Benz solution.

There are two extreme approaches to ReaRW task:

- Assuming a full model of the task/environment and performing reasoning in it or
- Assuming no model and exploring the reactive feedback in a perception-action loop.

The implemented ReaRW success stories found a right position in-between the mentioned extreme approaches. The invited speakers of the ReaRW summer school were asked to give talks which might help the audience to orient in the issue. They will also provide theoretical and practical hints how to implement such ReaRW tasks.

Responsible: V. Hlavac, hlavac@ciirc.cvut.cz

Vaclac Hlavac last modified on: 20.11.2014, 16:13

2.2 Programme

This annex includes the program of the school with talk abstracts and list of the speakers. The photos were recorded during the summer school.

Summer School – Reasoning in the Robot World 2014

Prague, Czech Republic 29-31 July, 2014. Hosted by the Czech Technical University in Prague, Center for Machine Perception

Co-financed by EU Project TRADR (FP7-ICT-609763). Talks in pdf available, see Program

Program

Presentations of speakers in pdf

- Pieter Abbeel: Lec1, Lec2, Lec 3
- Roman Barták: Lec1, Lec2, Lec 3
- Stan Birchfield: Lec1+Lec2+Lec 3
- Václav Hlaváč: Lec
- Hans-Georg Stork: Lec, Text
- David Vernon: Lec1, Lec2, Lec 3
- Michael Zillich: Lec1, Lec2

Monday July 28, 2014 – Preparation, Day 0

- *The project TRADR and the project CloPeMa might organize meetings of their project researchers (not open to public)*
- *13:00 (only for member of the TRADR project) Summer school Preparatory meeting (room G205)*
- 18:00 – 20:00 ReaRW 2014 Registration, ReaRW 2014 Ice-breaker party in Room G205.

Tuesday July 29, 2014 – ReaRW Day 1

- *8:30 – 8:50 Registration*
- 9:00 – 9:25 – ReaRW 2014 Opening, brief intro of each participant
- 09:25 – 09:50 (L01) V. Hlavac: Robotic representation, reasoning and learning – ReaRW 2014 assignment
- 10:00 – 10:50 (L02) P. Abbeel 1: Motion planning and reinforcement learning in robotics
- 11:00 – 11:50 Lecture 3 (L03) R. Barták 1: Robot world representation and reasoning in it

- *11:50 – 13:00 Buffet lunch at the venue (provided), poster session*
- 13:00 – 13:50 (L04) P. Abbeel 2: Motion planning and reinforcement learning in robotics
- 14:00 – 14:50 (L05) R. Barták 2: Robot world representation and reasoning in it
- 15:00 – 15:50 (L06) P. Abbeel 3: Motion planning and reinforcement learning in robotics
- 16:00 – 16: 20 Discussion of the day
- *16:20 – Informal program, probably a guided walk in Prague city center.*
- 19:00 – Networking dinner at Strahovsky pivovar

Wednesday July 30, 2014 – ReaRW Day 2

- 09:00 – 09:50 (L07) R. Barták 3: Robot world representation and reasoning in it
- 10:00 – 10:50 (L08) D. Vernon 1: Cognitive robotics
- 11:00 – 11:50 (L09) S. Birchfield 1: Perception – action loop in practice
- *11:50 – 13:00 Buffet lunch at the venue (provided), poster session, host institution demos*
- 13:00 – 13:50 (L10) H.-G. Stork: Robots, Research, and Responsibility
- 14:00 – 14:50 (L11) D. Vernon 2: Cognitive robotics
- 15:00 – 15:50 (L12) M. Zillich 1: Knowing and acting in an uncertain world – a practical view
- 16:00 – 16: 20 Discussion of the day
- *16:20 – Informal program, probably a small trip in the Prague green finished in a nice pub*

Thursday July 31, 2014 – ReaRW Day 3

- 09:00 – 09:50 (L13) D. Vernon 3: Cognitive robotics
- 10:00 – 10:50 (L14) S. Birchfield 2: Perception – action loop in practice
- 11:00 – 11:50 (L15) M. Zillich 2: Knowing and acting in an uncertain world – a practical view
- *11:50 – 13:00 Buffet lunch at the venue (provided), host institutions demos*
- 13:00 – 13:50 (L16) S. Birchfield 3: Perception – action loop in practice
- 14:00 – 14:50 (L17) M. Zillich 3: Knowing and acting in an uncertain world – a practical view
- 15:00 – 15:50 (L18) M. Achtelik: MAVs in real applications – State of the art and limits
- 16:00 – Close of ReaRW 2014

Friday August 1, 2014 – Post Summer school activities – Day 4

- *08:30 – 16:00 (only for member of the TRADR project) Discussion, Summary and Evaluation of the Summer School (Room G205)*

Talk titles / abstracts

Pieter Abbeel

Title: Motion planning and reinforcement learning in robotics

Abstract: This tutorial will cover optimization-based approaches for motion planning, and contrast them with more traditional approaches. This will include a tutorial introduction to sequential convex optimization, and the extension from state space planning to belief space planning. This tutorial will also cover reinforcement learning approaches that are a good fit for robotics, with an emphasis on policy search methods.

Roman Barták

Title: Robot world representation and reasoning in it

Abstract:

- Formal models: predicate and first-order logic, Boolean Satisfiability (SAT), Constraint Satisfaction Problem (CSP), temporal models
- World and planning models: situation calculus, set representation, predicate representation, state-variable representation, finite-state automata, planning with time and resources, task networks
- Reasoning techniques: search and inference, DPLL, constraint propagation
- Planning techniques: state-space planning, plan-space planning, translation to SAT/CSP, HTN planning, heuristics, control rules, adding time and resources
- Practical planning: planning systems, PDDL, ANML
- Challenges: real world vs abstract world, open worlds, uncertainty, plan execution

Stan Birchfield

Title: Perception – action loop in practice

Abstract:

- Traditional sense-plan-act vs. behavior-based approach
- Direct mapping from perception to action
- Interactive perception and active sensing
- Objects of interest come from model (in a logical sense), need for segmentation from percepts
- Evolving (logical) model in changing robot world
- Hybrid approach = putting it all together
- Use cases: mobile robot navigation, piece of garment perception and manipulation, map building in mobile robotics

Vašek Hlaváč

Title: Robotic representation, reasoning and learning – ReaRW 2014 assignment

Abstract (a single 25 min talk):
In this introduction talk, I will formulate the summer school scope and goals.

I will mention the AI and machine learning (pattern recognition) attempts to provide the high-level understanding/control tools to robots. I will also explain why it has been so difficult. I will mention theoretical troubles coming from mathematical logic even for the crisp robot world. It is also rather challenging to induce objects and their semantics from observing real world and acting in it. Dynamical changes and possible interventions of other agents in the robot world make the situation even more complicated.

Finally, I will formulate the expected performance of the high-level modules in projects TRADR (multi-robot-based situation awareness for Urban Search and Rescue scenarios) and CloPeMa (dual-arm manipulation with soft objects as pieces of garments).

Hans-Georg Stork

Title: Robots, Research, and Responsibility

Abstract (a single 50 min talk):

- This is going to be a meta-research talk of a philosophical nature.
- The speaker retired from the European Commission in early 2012. He worked for many years as a project officer in the area of cognitive systems and robotics. He shaped the calls for project proposals in several EU frameworks (i.e. seven years long planning periods).
- Instead of the the abstract, have a look at his contribution.
- The aim of the talk is to stimulate the discussion among the ReaRW participants.

David Vernon

Title: Cognitive Robotics

Abstract:

These three talks look at the link between artificial cognitive systems and robotics. The first talk will examine the capabilities of a cognitive system – goal-directed action, perception, learning, prospection, and adaptation – before explaining the different approaches that people take to modeling cognition, ranging from symbolic artificial intelligence to connectionism and dynamical systems theory. This foundation sets the scene for the second talk which looks at cognitive architectures, the essential foundation of any cognitive system. We will discuss three different architectures – Soar, Darwin, ISAC, and CLARION – focusing on how they have been applied in robotics. The third talk then addresses in more detail some of the core components of a robotic cognitive architecture, including different forms of memory, development and learning, knowledge representation and reasoning, symbol grounding, prospection and anticipatory action, i.e. planning. We will conclude the third talk by looking briefly at human robot interaction and the role of social cognition in developing robots that can interact effectively with people. While these talks mainly deal with issues at a conceptual level, we will refer to specific computational models where possible to illustrate how some researchers have realized various aspects of cognitive robotics and we will highlight the many challenges that remain.

Michael Zillich

Title: Knowing and acting in an uncertain world – a practical view

Abstract:

Lecture 1: Robot knowledge

- Task related knowledge: domain knowledge, process knowledge, and dealing with incoming percepts
- Types of representations: symbolic, graphical models, etc.
- Fully symbolic vs. fully reactive/sensor-motor

Lecture 2: Reasoning and acting

- symbolic planning (Roman: How much about symbolic planning will be covered in your lecture?) Actually, my [Roman] idea was to cover automated (symbolic) action planning in “full” but without talking much about plan execution/acting.
- PDDL, STRIPS, extensions to handle time and uncertainty
- graphical models and probabilistic planning

Lecture 3: Use cases, lessons learned, best practices

- A list of real world example solutions from various collaborative projects, their challenges and problems (magic “probabilities”, open worlds, etc.)
- Possibly an exercise with a symbolic planner in ROS

Michael Achtelik

Title: MAVs in real applications – State of the art and limits

MAVs are already used in many applications like surveying and industrial inspections. The talk will give an overview of possible applications as well as technical limitations. The gap between end users' requirements and todays possible applications links directly to ongoing research in projects like TRADR. In the second part of the talk Ascending Technologies UAV platforms are presented as basis for the TRADR hardware decision.

Responsible: V. Hlavac, hlavac@ciirc.cvut.cz

Vaclac Hlavac last modified on: 31.07.2014, 16:08

Summer School – Reasoning in the Robot World 2014

Prague, Czech Republic 29-31 July, 2014. Hosted by the Czech Technical University in Prague, Center for Machine Perception

Co-financed by EU Project TRADR (FP7-ICT-609763). Talks in pdf available, see Program

Speakers

Pieter Abbeel
University of California, Berkeley, USA

Roman Barták
Charles University in Prague
Czech Republic

Stan Birchfield

Micrososoft Research Redmond

Robotics Group

Hans-Georg Stork

retired EC project officer

in cognitive systems and robotics

David Vernon

University of Skövde, Sweden

Michael Zillich

Vienna University of Technology, Austria

Responsible: V. Hlavac, hlavac@ciirc.cvut.cz

Vaclac Hlavac last modified on: 03.08.2014, 1:47

2.3 Lecture materials

This annex includes all the lecture materials, besides the opening talk of prof. Hlaváč, mainly the slides of the invited speakers.

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Robotic representation, reasoning and learning

Assignment for ReaRW 2014

Václav Hlaváč

Czech Technical University in Prague

- Faculty of Electrical Engineering
- Czech Institute of Informatics, Robotics and Cybernetics

Prague, Czech Republic

hlavac@fel.cvut.cz;
http://cmp.felk.cvut.cz/~hlavac

Three EC projects, our motivation

2

- **NIFTi**, IP, 4 years, finished 2013-12
- TRADR IP, 4 years, takes over since 2013-11
- Both projects led by DFKI Saarbrücken
- Many partners

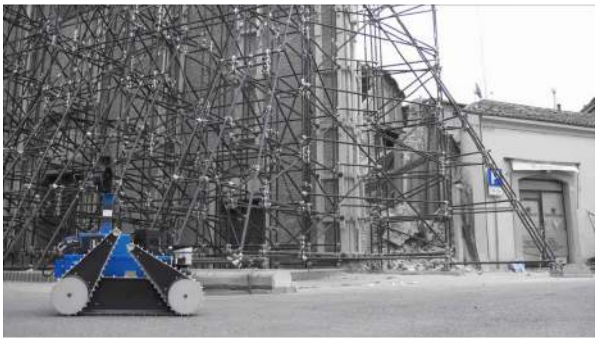
- **CloPeMa** = Clothes perception and manipulation
- STREP, 2012-02 to 2015-01
- CERTH Thessaloniki, U of Genova, U of Glasgow, CTU Prague, Neovision (integration)

NIFTI practical success

3

3

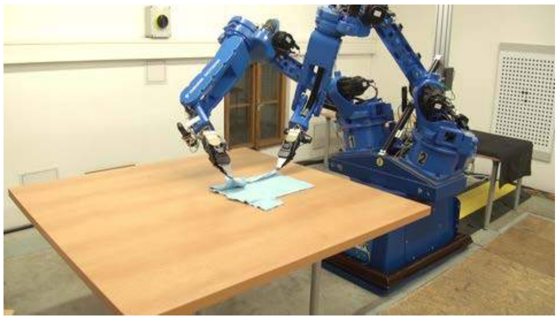
May 20 – June 18, 2012,
Mirandola, Emilia-Romagna region, Italy
246 seismic activities, 3 – 6.1 Richter scale



CloPeMa – current abilities

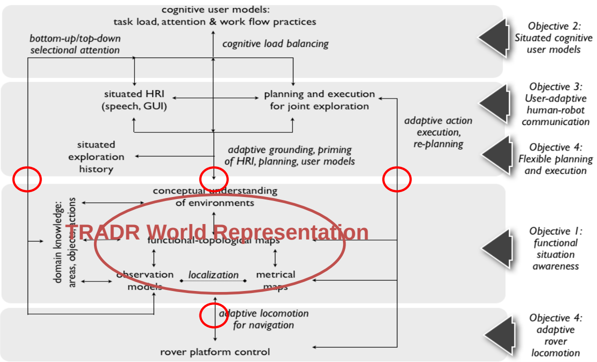
4

- Maria Petrou's project, the coordinator
- Maria died on Dec 15, 2012 of cancer



NIFTi knowledge architecture

5



Intelligence needed

6

Ability to:

- Represent knowledge, including commonsense knowledge.
- Learn
- Reason, plan
- Sense / act
- Salience – the ability to recognize importance

- Metric map
- Topological map
- Functional map
- Detected object (≈ concepts, primitives, function of objects)
- Graph having objects as graph nodes and their relations as graph edges.
- Action representation

Some philosophy

7

Rationalism

- " the criterion of the truth is not sensory but intellectual and deductive".
- Intelligence stems from (logical) reasoning.
- Many rationalists believe that some part of human knowledge is innate.
- *René Descartes (1596–1650), Baruch Spinoza (1632–1677), Gottfried Leibniz (1646–1716).*

Empiricism

- "An empiricist holds that experience is the source of all human knowledge"
- E.g., all knowledge must be grounded or based in the sensory world.
- *Aristotle (384 BC–322 BC), William of Ockham (1288–1348), Francis Bacon (1561-1626), John Locke (1632-1704), George Berkeley (1685-1753), David Hume (1711-1776), Hermann von Helmholtz (1821-1894), Karl Popper (1902-1994).*

A physical symbol system hypothesis

8

Harnad, S. (1990) The Symbol Grounding Problem. Physica D 42: 335-346.

- The rule-governed symbol-token manipulation is based purely on the shape of the symbol tokens (not their "meaning"), i.e., it is purely syntactic.
- There are primitive atomic symbol tokens and composite symbol-token strings.
- The syntax can be systematically assigned a meaning e.g., as standing for objects, as describing states of affairs.
- The symbols in an autonomous hybrid symbolic+sensorimotor system would be grounded. (Importance of the perception action cycle.)

AI issues in robotics

9

- A state space – a set of possible configurations of a robot, usually a discrete one, e.g. a position on a occupancy grid.
- Representation of a robot environment (a practical implementation of the state space), e.g. the robot world model.
- A reasoning engine – usually some type of symbol manipulation mechanism used for search/planning in a state space.
- Ontology (next slide)

Ontology in computer science

10

- *Definition:*
  Ontology formally represents knowledge as a set of concepts within a domain, and the relationships between pairs of concepts.
- It can be used to model a domain and support reasoning about entities.
- Ontologies are the structural frameworks for organizing information.
- Serves as the information architecture as a form of knowledge representation about the world or some part of it.
- Used in AI, robotics, semantic web, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking,

Czech connections

11



- **Golem**
  Late 16th century, Prague Rabi Loew (Judah Loew ben Bezalel).
- **Robot**
  Word invented by Czech playwright Karel Čapek (1980-1938)
- **Incompleteness theorem**
  Kurt Gödel (1906 in Brno - 1978)

ReaRW 2014 – The assignment

12

- A gap remains between low level percepts, robot control ↔ high level reasoning.
- Key issue is scene/knowledge representation.
- Learning is used only within a module to set parameters, not in the entire system to establish structure qualitatively.

- **ReaRW aims at telling: what and how an intelligent robot can be built in practice.**

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Learning from Demonstrations

Pieter Abbeel
UC Berkeley EECS

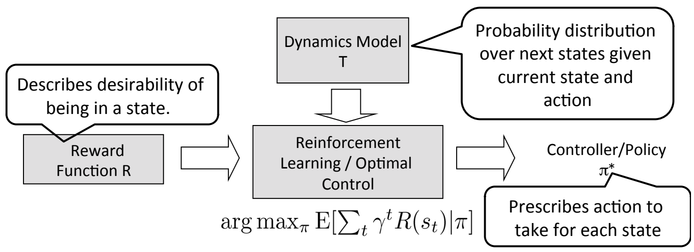
Outline

- Optimization-based motion planning
  - State space
  - Belief space
- **Learning from demonstrations**
- Reinforcement learning

Outline

- Optimization-based motion planning
  - State space
  - Belief space
- **Learning from demonstrations**
  - **Inverse optimal control**
  - **Trajectory learning**
- Reinforcement learning

Inverse Optimal Control



Inverse RL:

Given π^* and T, can we recover R?

More generally, given execution traces, can we recover R?

Motivation for Inverse Optimal Control

- Scientific inquiry
  - Model animal and human behavior
    - E.g., bee foraging, songbird vocalization. [See intro of Ng and Russell, 2000 for a brief overview.]
- Apprenticeship learning/Imitation learning
  - Presupposition: reward function provides the most succinct and transferable definition of the task
  - Has enabled advancing the state of the art in various robotic domains
- Modeling of other agents, both adversarial and cooperative

Outline

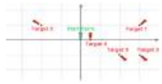
- Example applications
- Inverse optimal control vs. behavioral cloning
- Historical sketch of inverse optimal control
- Mathematical formulations for inverse optimal control
- Case studies

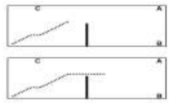
Examples

- Simulated highway driving
  - Abbeel and Ng, ICML 2004,
  - Syed and Schapire, NIPS 2007
- Aerial imagery based navigation
  - Ratliff, Bagnell and Zinkevich, ICML 2006
- Parking lot navigation
  - Abbeel, Dolgov, Ng and Thrun, IROS 2008

Examples (ctd)

- Human path planning
  - Mombaur, Truong and Laumond, AURO 2009
- Human goal inference
  - Baker, Saxe and Tenenbaum, Cognition 2009
- Quadruped locomotion
  - Ratliff, Bradley, Bagnell and Chestnutt, NIPS 2007
  - Kolter, Abbeel and Ng, NIPS 2008





Urban navigation

- Reward function for urban navigation?

Ziebart, Maas, Bagnell and Dey AAAI 2008

Lecture outline

- Example applications
- *Inverse RL vs. behavioral cloning*
- Historical sketch of inverse RL
- Mathematical formulations for inverse RL
- Case studies

Problem setup

- Input:
  - State space, action space
  - Transition model $P_{sa}(s_{t+1} \mid s_t, a_t)$
  - *No* reward function
  - Teacher's demonstration: $s_0, a_0, s_1, a_1, s_2, a_2, \ldots$ (= trace of teacher's policy π^*)
- Inverse optimal control:
  - Can we recover R ?
- Apprenticeship learning via inverse optimal control
  - Can we then use this R to find a good policy ?
- Behavioral cloning
  - Can we directly learn the teacher's policy using supervised learning?

Behavioral cloning

- Formulate as standard machine learning problem
  - Fix a policy class
    - E.g., support vector machine, neural network, decision tree, deep belief net, …
  - Estimate a policy (=mapping from states to actions) from the training examples $(s_0, a_0), (s_1, a_1), (s_2, a_2), \ldots$
- Two of the most notable success stories:
  - Pomerleau, NIPS 1989: ALVINN
  - Sammut et al., ICML 1992: Learning to fly (flight sim)

Inverse optimal control vs. behavioral cloning

- **Which has the most succinct description: π^* vs. R*?**
- Especially in planning oriented tasks, the reward function is often much more succinct than the optimal policy.

Inverse Optimal Control History

- 1964, Kalman posed the inverse optimal control problem and solved it in the 1D input case
- 1994, Boyd+al.: a linear matrix inequality (LMI) characterization for the general linear quadratic setting
- 2000, Ng and Russell: first MDP formulation, reward function ambiguity pointed out and a few solutions suggested
- 2004, Abbeel and Ng: inverse RL for apprenticeship learning---reward feature matching
- 2006, Ratliff+al: max margin formulation

Inverse RL history

- 2007, Ratliff+al: max margin with boosting---enables large vocabulary of reward features
- 2007, Ramachandran and Amir [R&A], and Neu and Szepesvari: reward function as characterization of policy class
- 2008, Kolter, Abbeel and Ng: hierarchical max-margin
- 2008, Syed and Schapire: feature matching + game theoretic formulation
- 2008, Ziebart+al: feature matching + max entropy
- 2008, Abbeel+al: feature matching -- application to learning parking lot navigation style
- 2009, Baker, Saxe, Tenenbaum: same formulation as [R&A], investigation of understanding of human inverse planning inference
- 2009, Mombaur, Truong, Laumond: human path planning
- Active inverse RL? Inverse RL w.r.t. minmax control, partial observability, learning stage (rather than observing optimal policy), … ?

Outline

- Example applications
- Inverse optimal control vs. behavioral cloning
- Historical sketch of inverse optimal control
- *Mathematical formulations for inverse RL*
- Case studies

Three broad categories of formalizations

- ***Max margin***
- Feature expectation matching
- Interpret reward function as parameterization of a policy class

Basic principle

- Find a reward function R* which explains the expert behaviour.
- Find R* such that

$$\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*] \geq \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi] \quad \forall \pi$$

- In fact a convex feasibility problem, but many challenges:
  - R=0 is a solution, more generally: reward function ambiguity
  - We typically only observe expert traces rather than the entire expert policy π^* --- how to compute left-hand side?
  - Assumes the expert is indeed optimal --- otherwise infeasible
  - Computationally: assumes we can enumerate all policies

Feature-based Reward Function

- Let $R(s) = w^\top \phi(s)$, where $w \in \mathbb{R}^n$, and $\phi : S \to \mathbb{R}^n$.

$$\begin{aligned} \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi] &= \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t w^\top \phi(s_t)|\pi] \\ &= w^\top \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t)|\pi] \\ &= w^\top \underbrace{\mu(\pi)}_{} \end{aligned}$$

Expected cumulative discounted sum of feature values or "feature expectations"

- $\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*] \geq \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi] \quad \forall \pi$ becomes:

$$\text{Find } w^* \text{ such that } {w^*}^\top \mu(\pi^*) \geq {w^*}^\top \mu(\pi) \qquad \forall \pi$$

Feature-based Reward Function

$$\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*] \geq \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi] \qquad \forall \pi$$

Let $R(s) = w^\top \phi(s)$, where $w \in \Re^n$, and $\phi : S \to \Re^n$.
Find w^* such that ${w^*}^\top \mu(\pi^*) \geq {w^*}^\top \mu(\pi) \qquad \forall \pi$

- Feature expectations can be readily estimated from sample trajectories.
- The number of expert demonstrations required scales with the number of features in the reward function.
- The number of expert demonstration required does *not* depend on
  - Complexity of the expert's optimal policy ¼*
  - Size of the state space

Recap of Challenges

Let $R(s) = w^\top \phi(s)$, where $w \in \mathbb{R}^n$, and $\phi : S \to \mathbb{R}^n$.

$${w^*}^\top \mu(\pi^*) \geq {w^*}^\top \mu(\pi) \qquad \forall \pi$$

- Challenges:
  - Assumes we know the entire expert policy π^*→ assumes we can estimate expert feature expectations
  - R=0 is a solution (now: w=0), more generally: reward function ambiguity
  - Assumes the expert is indeed optimal---became even more of an issue with the more limited reward function expressiveness!
  - Computationally: assumes we can enumerate all policies

Ambiguity

- Standard max margin:

$$\begin{aligned} \min_w \quad & \|w\|_2^2 \\ \text{s.t.} \quad & w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + 1 \qquad \forall \pi \end{aligned}$$

- "Structured prediction" max margin:

$$\begin{aligned} \min_w \quad & \|w\|_2^2 \\ \text{s.t.} \quad & w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + m(\pi^*, \pi) \qquad \forall \pi \end{aligned}$$

- Justification: margin larger for policies very different from π^*.
- Example: $m(\pi, \pi^*)$= number of states in which disagree

Expert Suboptimality

- Structured prediction max margin with slack variables:

$$\begin{aligned} \min_{w,\xi} \quad & \|w\|_2^2 + C\xi \\ \text{s.t.} \quad & w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + m(\pi^*, \pi) - \xi \qquad \forall \pi \end{aligned}$$

Complete max-margin formulation

$$\begin{aligned} \min_w \quad & \|w\|_2^2 + C \sum_i \xi^{(i)} \\ \text{s.t.} \quad & w^\top \mu(\pi^{(i)*}) \geq w^\top \mu(\pi^{(i)}) + m(\pi^{(i)*}, \pi^{(i)}) - \xi^{(i)} \qquad \forall i, \pi^{(i)} \end{aligned}$$

[Ratliff, Zinkevich and Bagnell, 2006]

- Resolved: access to π^*, ambiguity, expert suboptimality
- One challenge remains: very large number of constraints
  - Ratliff+al use subgradient methods.
  - In this lecture: constraint generation

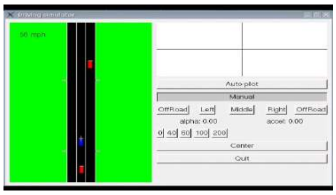
Simulated highway driving

Abbeel and Ng, ICML 2004; Syed and Schapire, NIPS 2007

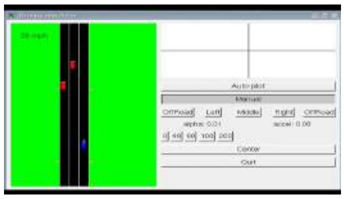
Highway driving

[Abbeel and Ng 2004]

Teacher in Training World



Learned Policy in Testing World

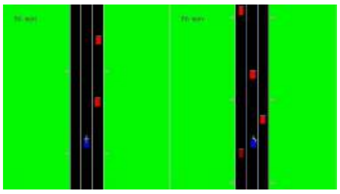


- Input:
  - Dynamics model / Simulator $P_{sa}(s_{t+1} \mid s_t, a_t)$
  - Teacher's demonstration: 1 minute in "training world"
  - R* is unknown. Reward features: 5 features corresponding to lanes/shoulders; 10 features corresponding to presence of other car in current lane at different distances

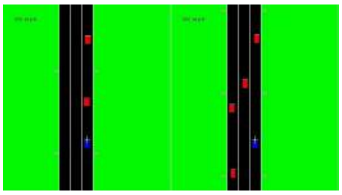
More driving examples

[Abbeel and Ng 2004]

Driving demonstration | Learned behavior

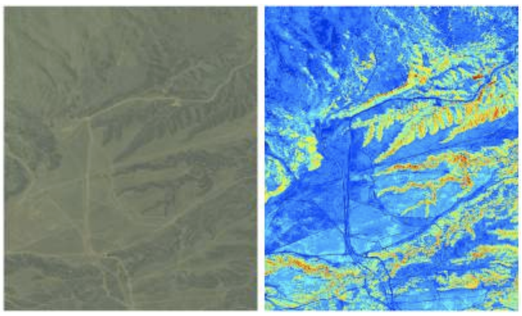


Driving demonstration | Learned behavior



In each video, the left sub-panel shows a demonstration of a different driving "style", and the right sub-panel shows the behavior learned from watching the demonstration.

Max margin



[Ratliff + al, 2006/7/8]

Quadruped

- Reward function trades off 25 features.

Hierarchical max margin [Kolter, Abbeel & Ng, 2008]

Experimental setup

- Demonstrate path across the “training terrain”

- Run inverse optimal control to find the reward function
- Receive “testing terrain”---height map.

- Find the optimal policy with respect to the *learned reward function* for crossing the testing terrain.

Hierarchical max margin [Kolter, Abbeel & Ng, 2008]

Without learning

With learned reward function

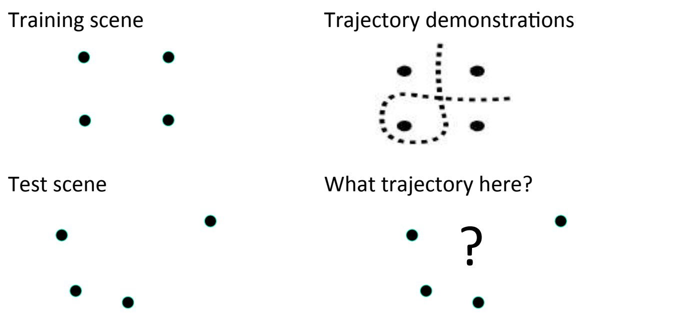
Summary Inverse Optimal Control

- Example applications
- Inverse optimal control vs. behavioral cloning
- Sketch of history of inverse optimal control
- Mathematical formulations for inverse optimal control
- Case studies
- Open directions: Active inverse optimal control, Inverse minmax control, partial observability, learning stage (rather than observing optimal policy), … ?

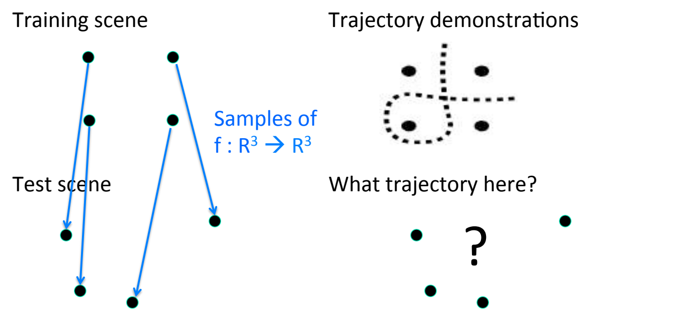
Learning from Demonstrations

- The problem
  - Human demonstrated knot-tie in this rope
  - Robot has to tie a knot in this rope

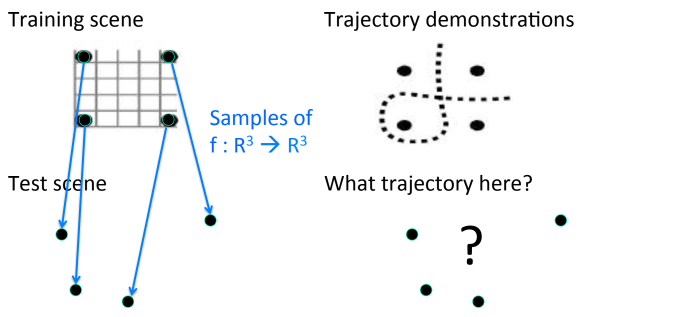
Cartoon Problem Setting



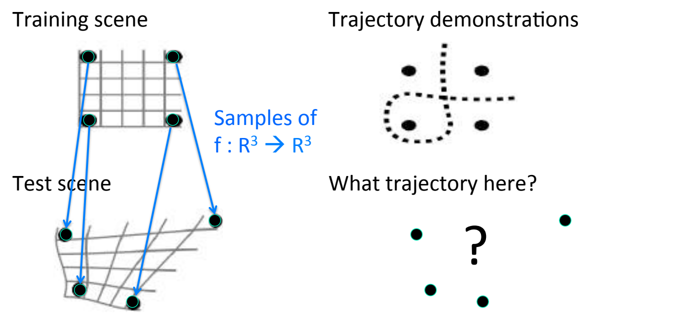
Cartoon Problem Setting



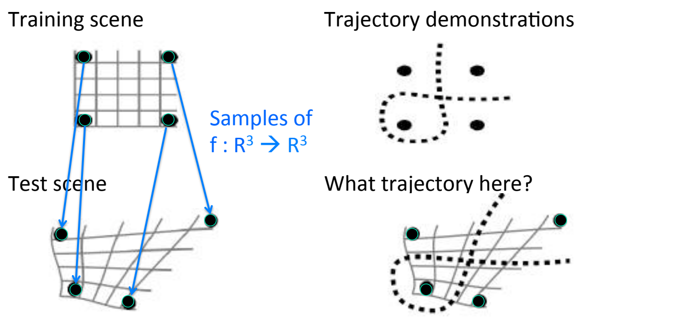
Cartoon Problem Setting



Cartoon Problem Setting



Cartoon Problem Setting



Learning f : $R^3 \rightarrow R^3$ from Samples

$$\min_{f \in \{\mathbb{R}^3 \rightarrow \mathbb{R}^3\}} \int_{x \in \mathbb{R}^3} \|D^2 f(x)\|^2_{\text{Frob}} dx$$

$$\text{s.t.} \quad f(x^{(i)}_{\text{train}}) = x^{(i)}_{\text{test}} \quad \forall i \in 1, \ldots, m$$

- Observations
  - Translations, rotations and scaling are FREE
  - Can be solved efficiently manipulating matrices of size of number of examples

Learning f : $R^3 \rightarrow R^3$ from Samples

$$\min_{f \in \{\mathbb{R}^3 \to \mathbb{R}^3\}} \int_{x \in \mathbb{R}^3} \|D^2 f\|^2_{\text{Frob}}(x)dx$$
$$\text{s.t.} \quad f(x^{(i)}_{\text{train}}) = x^{(i)}_{\text{test}} \quad \forall i \in 1, \ldots, m$$

- Solution has form:

$$f(x) = \sum_{i=1}^{m} a_i K(x_i, x) + b^\top x + c,$$
$$K(x,y) = \begin{cases} c_0 r^{4-d} \ln r, \ d = 2 \text{ or } d = 4 \\ c_1 r^{4-d}, \ \text{otherwise} \end{cases} \quad \text{with} \ \ r = \|x - y\|_2.$$

Wahba, Spline models for observational data. Philadelphia: Society for Industrial and Applied Mathematics. 1990.
Evgeniou, Pontil, Poggio, Regularization Networks and Support Vector Machines. Advances in Computational Mathematics. 2000.
Hastie, Tibshirani, Friedman, Elements of Statistical Learning, Chapter 5. 2008.

Point Cloud Registration

- Need to match up points in source and target point clouds

- Thin Plate Spline Robust Point Matching (TPS-RPM) algorithm (Chui & Ragnaran, 2003)
  - Alternate between soW-assignment and fiYng thin plate spline transforma/on

Trajectory Transfer Procedure

- Using non-rigid registration, find a transformation **f** from training scene to test scene
- Apply **f** to the demonstrated end-effector trajectory

$$\mathbf{p}_t \to \mathbf{f}(\mathbf{p}_t)$$
$$\mathbf{R}_t \to \text{orth}\left(\mathbf{J}_\mathbf{f}(\mathbf{p}_t)\mathbf{R}_t\right)$$

- Convert the end-effector trajectory to a joint trajectory

$$\underset{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_T}{\text{minimize}} \left[\sum_{t=1}^{T-1} \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 + \mu \sum_{t=1}^{T} \left\|\text{err}\left(\tilde{\mathbf{T}}_t^{-1} \cdot \text{fk}(\boldsymbol{\theta}_t)\right)\right\|_{\ell_1}\right]$$

subject to

No collisions, with safety margin d_{safe}

$\boldsymbol{\theta}_{\min} \leq \boldsymbol{\theta}_{1:T} \leq \boldsymbol{\theta}_{\max}$ (Joint limits)

Theoretical Guarantees

- Can be expected to work if the dynamics of the system are approximately covariant under sufficiently smooth warpings.

$$\begin{tikzcd}
\{\text{STATE}^t_{\text{train}}\} \arrow[r, "\Pi_{\text{TRAJ}}"] \arrow[d, "\mathbf{f}"'] & \{\text{STATE}^{t+1}_{\text{train}}\} \arrow[d, "\mathbf{f}"] \\
\{\text{STATE}^t_{\text{test}}\} \arrow[r, "\Pi_{\mathbf{f}(\text{TRAJ})}"] & \{\text{STATE}^{t+1}_{\text{test}}\}
\end{tikzcd}$$

Nearest-Neighbor Policy for Tasks

- Repeat
  - Acquire new point cloud X_{test}
  - Using non-rigid registration compute distance between X_{test} and point clouds $X_{train,i}$ from demonstrations including initial states and "done" states

$$i^* = \arg\min_i \text{Distance}(X_{\text{test}}, X_{\text{train},i})$$

  - If i* is a "done" state, **break**
  - Apply trajectory transfer to generate new trajectory

$$\text{Traj}_{\text{test}} = \text{TrajectoryTransfer}(\text{State}_{i^*}, \text{Traj}_{i^*}, \text{State}_{\text{test}})$$

Experimental Validation

- Knots tied
  - Overhand
  - Figure-eight
  - Double-overhand
  - Square
  - Clove-hitch

Experiment: Knot-Tie

Schulman et al., ISRR 2013

Evaluation

Knot type	Segments	Success ($d_{pert} = 3\,\text{cm}$)	($d_{pert} = 10\,\text{cm}$)
Overhand	3	5/5	4/5
Figure-eight	4	5/5	1/5
Dbl-overhand	5	3/5	3/5
Square	6	5/5	3/5
Clove hitch	4	1/5	0/5

Limitations of the Nearest-Neighbor Policy

- Doesn't account for demonstration quality
- Doesn't prefer moves that make progress
- Doesn't account for reachability of trajectory

Learning to Choose Better Actions

s: state of system (robot, rope, …)
a: chosen demonstration

policy: $\hat{a} = \arg\max_a \phi(s, a) \cdot \mathbf{w}$

features in ϕ

- action bias
- registration cost
- distance to "landmark" states

Max-Margin Policy Learning

$$\underset{\mathbf{w}}{\text{minimize}}\, L(\mathbf{w})$$

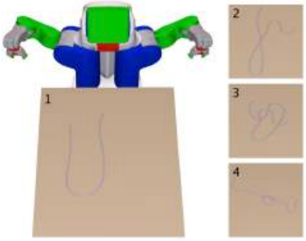
$$L(\mathbf{w}) = \underbrace{\frac{1}{2}\|\mathbf{w}\|^2}_{\text{Regularizer}} + \sum_i \left| \max_a \underbrace{[\phi(s_i, a) \cdot \mathbf{w} + \Delta(a, a_i)]}_{\text{score + margin of action a}} - \underbrace{\phi(s_i, a_i) \cdot \mathbf{w}}_{\text{score of demonstrated action } a_i \text{ from state } s_i} \right|^+$$

Max-Margin Q-Function Learning

$$\underset{\mathbf{w}}{\text{minimize}}\, L(\mathbf{w})$$

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i \left| \max_a [\phi(s_i, a) \cdot \mathbf{w} + \Delta(a, a_i)] - \phi(s_i, a_i) \cdot \mathbf{w} \right|^+ + \sum_i \underbrace{|\phi(s_i, a_i) \cdot \mathbf{w} - (R(s_i, a_i) + \phi(s_i', a_i') \cdot \mathbf{w})|}_{\substack{\text{On-policy Bellman error} \\ |Q(s,a) - (R(s,a) + Q(s',a'))|}}$$

Experiments



- 148 Demonstrated trajectories (on real robot)
- 1000 human-guided trajectory transfers in physics simulation
- Test on 500 new initial configurations

Results

Policy	Success rate
Nearest neighbor	68.8%
argmax_a Q(s,a)	85.6%
Lookahead (d=1, w=10)	93.6%
Lookahead (d=2, w=5)	95.2%

Experiment: Suturing

Schulman et al., IROS 2013

On-going Work on Learning from Demonstrations

- Indexing into large sets of demonstrations
- Improving registration
  - Visual appearance
  - Weighting based on relevance
- Reinforcement learning to further improve performance
- Forces and torques (to extend to non-kinematic tasks)

Trajectory Transfer with Non-Rigid Registration

- Transfer of manipulation motions from training scene to test scene
- Q-value learning
- Experiments on knot-tying, and also

Thank you

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Nonlinear Optimization for Motion Planning in State Space and in Belief Space

Pieter Abbeel
UC Berkeley EECS

Prague – July 29th, 2014

Personal Robotics Hardware

PR2
Willow Garage
$400,000
2009

Baxter
Rethink Robotics
$30,000
2012

UBR-1
Unbounded Robotics
$35,000
2013

?

?
$2,000 ?
2017?

Concrete Tasks

- Unstructured environments – homes, offices, surgery
  - High variability
  - Uncertainty
- Challenge tasks: manipulation of deformable objects
  - Robotic laundry
  - Robotic suturing

Robotic Laundry

Technical Contributions

- Low-level perception cues for clothing articles (ICRA 2010, ICRA 2011, IROS 2011)
- Perception – Action loop for clothing disambiguation (ICRA 2011)
- Hierarchical motion planning for mobile manipulation (ISER 2012)
- Motion planning for folding (WAFR 2010, IJRR 2011)

Current Research Directions

- Optimization-based motion planning
  - State space
  - Belief space
- Learning from demonstrations
- Reinforcement learning

Motion Planning

- Sampling-based methods (e.g., RRTs)
- Graph search methods (e.g., A*)
- Optimization-based methods
  - Reactive control
    - Potential-based methods (Khatib '86)
  - Optimize over entire trajectory
    - Elastic bands (Quinlan and Khatib '93)
    - CHOMP (Ratliff et al. '09) and variants (STOMP, ITOMP)

Trajectory Optimization

$$\min_{\theta_{1:T}} \sum_t \|\theta_{t+1} - \theta_t\|^2 + \text{other costs}$$

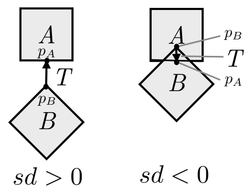
subject to θ_0 = start state, θ_T in goal set

joint limits

for all robot parts, for all obstacles:
no collision ⟶ ***non-convex***

Solution method: sequential convex optimization

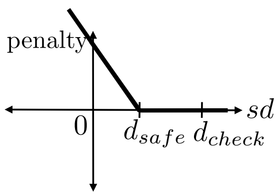
Collision Constraints



$$sd_{AB}(\theta) \approx \hat{n} \cdot (p_B - p_A(\theta))$$
$$\approx sd_{AB}(\theta_0) - \hat{n}^\top J_{P_A}(\theta_0)(\theta - \theta_0)$$

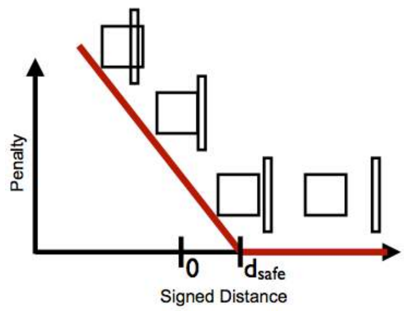
[SD from: Gilbert-Johnson-Keerthi (GJK) algorithm and Expanding Polytope Algorithm (EPA)]

Penalty for Collision Constraints

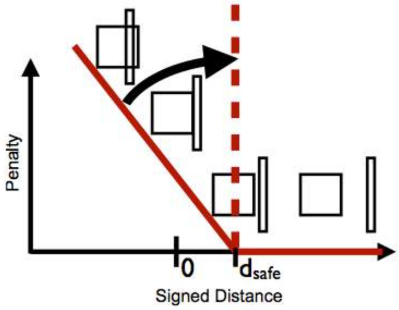


$$sd_{AB}(\theta) \approx \hat{n} \cdot (p_B - p_A(\theta))$$
$$\approx sd_{AB}(\theta_0) - \hat{n}^\top J_{P_A}(\theta_0)(\theta - \theta_0)$$

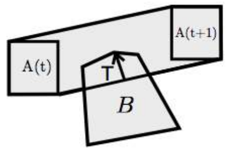
Collision Constraint as L1 Penalty



Collision Constraint as L1 Penalty



Continuous-Time Safety

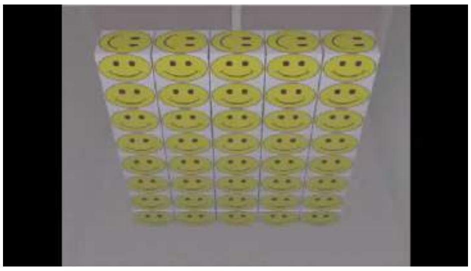


Collision check against swept-out volume

- Allows coarsely sampling trajectory
  - Overall faster
- Finds better local optima

Collision-free Path for Dubin's Car

Experiments: Industrial Box Picking

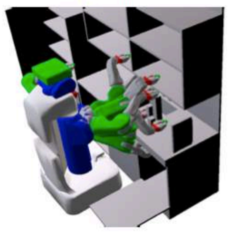


Experiments: DRC Robot

Benchmark

7 DOF (one arm)
198 problems

18 DOF (two arms + base + torso)
96 problems



example scene (taken from MoveIt collection)

example scene (imported from Trimble 3d Warehouse / Google Sketchup)

Benchmark Results

Arm planning (7 DOF) 10s limit			
	Trajopt	BiRRT (*)	CHOMP
success	99%	97%	85%
time (s)	0.32	1.2	6.0
path length	1.2	1.6	2.6

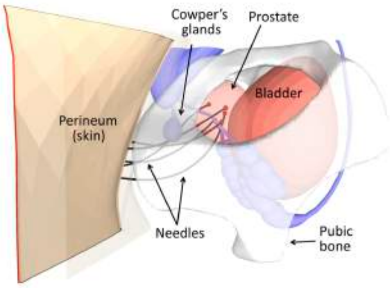
Full body (18 DOF) 30s limit			
	Trajopt	BiRRT (*)	CHOMP (**)
success	84%	53%	N/A
time (s)	7.6	18	N/A
path length	1.1	1.6	N/A

(*) Top-performing algorithm from MoveIt/OMPL
(**) Not supported in available implementation

[RSS 2013]

Experiments: PR2

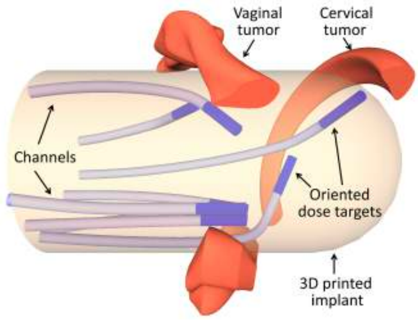
Medical Application 1: Needle Steering



Resulting paths: (i) shorter, (ii) less twist (i.e., less tissue carved up), (iii) found more quickly—replanning!

[ICRA 2014]

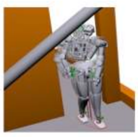
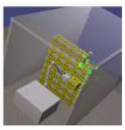
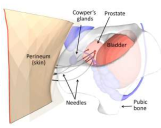
Medical Application 2: Channel Planning



[ICRA 2014]

Try It Yourself

- Code and docs: rll.berkeley.edu/trajopt
- Benchmark: github.com/joschu/planning_benchmark

Experiments: PR2

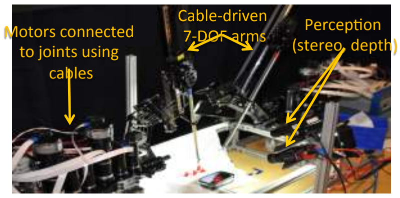
Current Research Directions

- Optimization-based motion planning
  - State space
  - Belief space
- Learning from demonstrations
- Reinforcement learning

Goal: Reliable Autonomous Execution

Baxter (Rethink)

Low-cost arm (Quigley et al.)



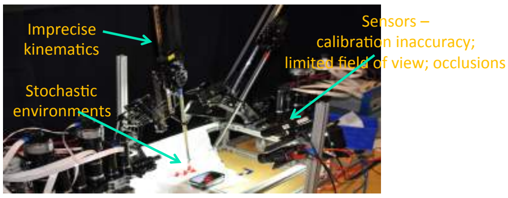
Raven surgical robot (Rosen et al.)

Cost-effective, less precise robots

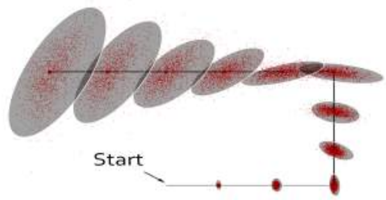
Robots are not perfect

Sources of <u>uncertainty</u> :

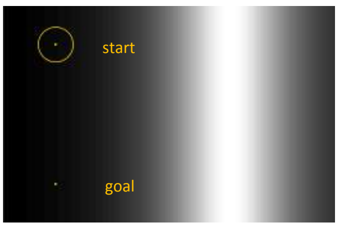
- Imprecise actuation
- Sensing errors
- Stochastic environments
- Modeling errors



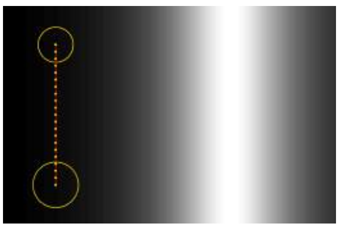
Uncertainty Propagation



Dark-Light Domain



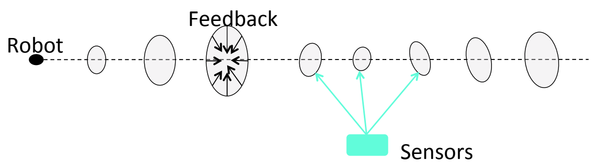
Problem Setup



State space plan

[Example from Platt, Tedrake, Kaelbling, Lozano-Perez, 2010]

Dealing with Uncertainty



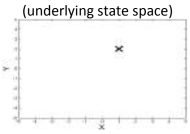
This is a Partially Observable Markov Decision Process (POMDP) – in general intractable

(Gaussian) Belief Space Planning

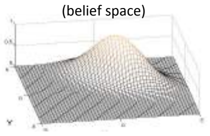
- Redefine problem

$"x_t" = (\mu_t, \Sigma_t)$

(underlying state space)



(belief space)



- Convert underlying dynamics to belief space dynamics
  – Bayesian filter (e.g., extended Kalman filter)
  – Stochastic – Determinize and re-plan

(Gaussian) Belief Space Planning

<u>Minimize</u>
t = 1 .. T,
[means, covariances, controls]

Cost(means, covariances, controls)

<u>Constraints</u>:

- [mean, covariance]$_{t+1}$ = KF([mean, covariance]$_t$, control$_t$)
- mean$_T$ = goal
- Feasible control inputs

Solved by Sequential Convex Programming

(Gaussian) Belief Space Planning

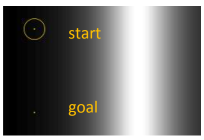
$$
\begin{aligned}
min_{\mu,\Sigma,u} \quad & \sum_{t=0}^{H} c(\mu_t, \Sigma_t, u_t) \\
\text{s.t.} \quad & (\mu_{t+1}, \Sigma_{t+1}) = xKF(\mu_t, \Sigma_t, u_t, w_t, v_t) \\
& \mu_H = \text{goal} \\
& u \in \mathcal{U}
\end{aligned}
$$

(Gaussian) Belief Space Planning

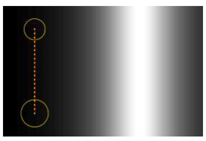
$$
\begin{aligned}
min_{\mu,\Sigma,u} \quad & \sum_{t=0}^{H} c(\mu_t, \Sigma_t, u_t) \\
\text{s.t.} \quad & (\mu_{t+1}, \Sigma_{t+1}) = xKF(\mu_t, \Sigma_t, u_t, 0, 0) \\
& \mu_H = \text{goal} \\
& u \in \mathcal{U}
\end{aligned}
$$

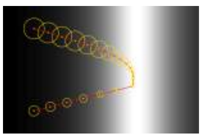
= maximum likelihood assumption for observations
Can now be solved by Sequential Convex Programming
[Platt, Tedrake, Kaelbling, Lozano-Perez, 2010; also Roy et al ; van den Berg et al.]

Dealing with Uncertainty

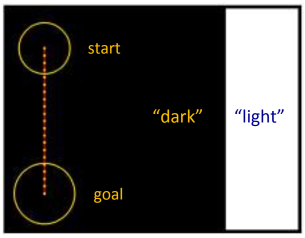


Problem setting


State space plan

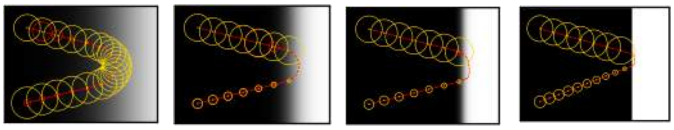

Belief space plan

Dealing with Discontinuities



Zero gradient, hence local optimum

Dealing with Discontinuities



Increasing difficulty

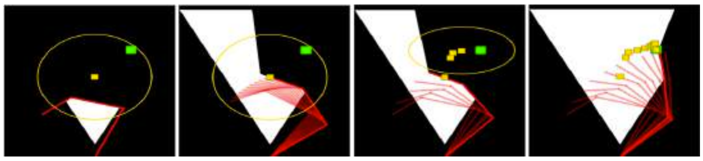
Noise level determined by

signed distance to sensing region (computed with GJK/EPA)

homotopy iteration

[ICRA 2014]

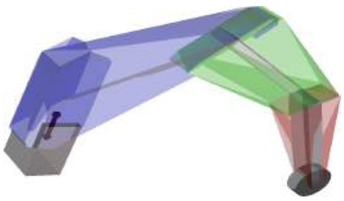
Arm Occluding Camera



Initial belief | State space plan execution | Belief space plan execution (way-point) | (end)

[ICRA 2014]

Sigma-Hull Collision Checking



Use Unscented Kalman Filter (UKF) to represent belief space dynamics

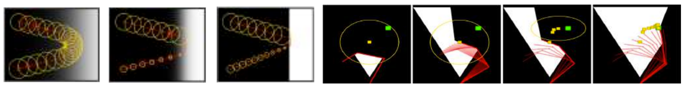
Collision cost computed for z-sigma-hulls for each link of the robot

[IROS 2013]

Recap

- High DOF motion planning

- Belief space planning



Sequential Convex Optimization

- Convex optimization problems
- Sequential convex optimization

Convex Functions

- A function is f is convex if and only if
  $\forall x_1, x_2 \in \mathrm{Domain}(f), \forall t \in [0,1]:$
  $$f(tx_1 + (1-t)x_2) \le tf(x_1) + (1-t)f(x_2)$$

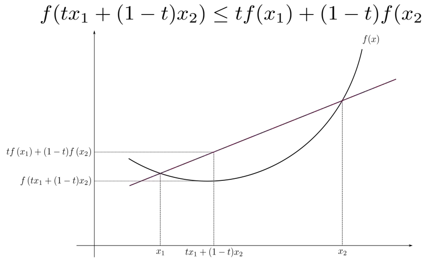
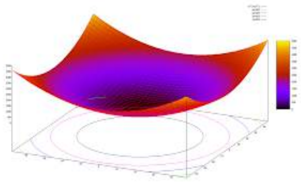


Image source: wikipedia

Convex Functions



- Unique minimum
- Set of points for which f(x) <= a is convex

Source: Thomas Junglut's Blog

Convex Optimization Problems

- Convex optimization problems are a special class of optimization problems, of the following form:

$$\min_{x \in \mathbb{R}^n} f_0(x)$$
$$\text{s.t.} \quad f_i(x) \leq 0 \qquad i = 1, \ldots, n$$
$$Ax = b$$

with $f_i(x)$ convex for i = 0, 1, …, n

- Can be solved efficiently. E.g., Boyd and Vandenberghe Ch. 9 -11.

Non-Convex Optimization

- Reminder: Convex optimization:

$$\min_x \quad f_0(x)$$
$$\text{s.t.} \quad f_i(x) \leq 0 \quad \forall i$$
$$A(j,:)x - b_j = 0 \quad \forall j$$

with f_0, f_i convex

- Non-convex optimization:

$$\min_x \quad g_0(x)$$
$$\text{s.t.} \quad g_i(x) \leq 0 \quad \forall i$$
$$h_j(x) = 0 \quad \forall j$$

with:
g_i non-convex
h_j nonlinear

Sequential Convex Programming

- *To solve:*

$$\min_x \quad g_0(x) \qquad \text{(1)}$$
$$\text{s.t.} \quad g_i(x) \leq 0 \quad \forall i$$
$$h_j(x) = 0 \quad \forall j$$

merit function

- *Solve:* $$\min_x \quad g_0(x) + \mu \sum_i |g_i(x)|^+ + \mu \sum_j |h_j(x)| \quad = \min_x \quad f_\mu(x) \qquad \text{(2)}$$

and increase μ in an outer loop until the two sums equal zero.

- *To solve (2), repeatedly solve the convex program:*

$$\min_x \quad g_0(\bar{x}) + \nabla_x g_0(\bar{x})(x - \bar{x}) + \mu \sum_i |g_i(\bar{x}) + \nabla_x g_i(\bar{x})(x - \bar{x})|^+ + \mu \sum_j |h_j(\bar{x}) + \nabla_x h_j(\bar{x})(x - \bar{x})|$$

$\text{s.t.} \quad \|x - \bar{x}\|_2 \leq \varepsilon$ (trust region constraint) $\bar{x}$: current point

Inputs: $\bar{x}, \mu = 1, \varepsilon_0, \alpha \in (0.5, 1), \beta \in (0,1), t \in (1, \infty)$

While ($\sum_i |g_i(\bar{x})|^+ + \sum_j |h_j(\bar{x})| \geq \delta$ AND $\mu < \mu_{\mathrm{MAX}}$)

$\mu \leftarrow t\mu, \quad \varepsilon \leftarrow \varepsilon_0$ // increase penalty coefficient for constraints; re-init trust region size

While (1) // [2] loop that optimizes

Compute terms of first-order approximations: $g_0(\bar{x}), \nabla_x g_0(\bar{x}), g_i(\bar{x}), \nabla_x g_i(\bar{x}), h_j(\bar{x}), \nabla_x h_j(\bar{x}), \quad \forall i, j$

While (1) // [3] loop that does trust-region size search

Call convex program solver to solve:

$$(\bar{f}_\mu(\bar{x}_{\text{next?}}, \bar{x}_{\text{next?}}) = \min_x \quad g_0(\bar{x}) + \nabla_x g_0(\bar{x})(x - \bar{x}) + \mu \sum_i |g_i(\bar{x}) + \nabla_x g_i(\bar{x})(x - \bar{x})|^+$$
$$+\mu \sum_j |h_j(\bar{x}) + \nabla_x h_j(\bar{x})(x - \bar{x})| \quad \text{s.t.} \quad \|x - \bar{x}\|_2 \leq \varepsilon$$

If $\frac{f_\mu(\bar{x}) - f_\mu(\bar{x}_{\text{next?}})}{\bar{f}_\mu(\bar{x}) - \bar{f}_\mu(\bar{x}_{\text{next?}})} \geq \alpha$

Then shrink trust region:

Else Update $\bar{x} \leftarrow \bar{x}_{\text{next?}}$, Grow trust region: $\varepsilon \leftarrow \varepsilon / \beta$, and Break out of while [3]

If below some threshold, Break out of while [3] and while [2]

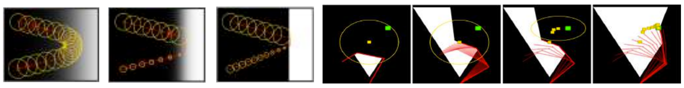
To Learn More

- Detailed tutorial / homework exercises
  - UC Berkeley's CS 287 Advanced Robotics
  - http://www.cs.berkeley.edu/~pabbeel/cs287-fa13/
- Complete treatise on numerical optimization:
  - Nocedal and Wright, Numerical Optimization

Recap

- High DOF motion planning

- Belief space planning



Thank you

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Reinforcement Learning – Policy Search

Pieter Abbeel
UC Berkeley EECS

Outline

- Optimization-based motion planning
  - State space
  - Belief space
- Learning from demonstrations
- **Reinforcement learning**

Reinforcement Learning -- Policy Search

- Consider control policy parameterized by parameter vector θ

$$\max_{\theta} \quad \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

- Often stochastic policy class (smooths out the problem):
  - $\pi_\theta(a|s)$ probability of taking action a in state s

Learning to Trot/Run

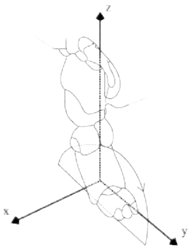
Before learning (hand-tuned)

After learning

[Policy search was done through trials on the actual robot.] Kohl and Stone, ICRA2004

Learning to Trot/Run

- 12 parameters define the Aibo's gait:
  - The front locus (3 parameters: height, x-pos., y-pos.)
  - The rear locus (3 parameters)
  - Locus length
  - Locus skew multiplier in the x-y plane (for turning)
  - The height of the front of the body
  - The height of the rear of the body
  - The time each foot takes to move through its locus
  - The fraction of time each foot spends on the ground



Kohl and Stone, ICRA2004

[Policy search was done in simulation] [Ng + al, ISER 2004]

Learning to Hover

x, y, z: x points forward along the helicopter, y sideways to the right, z downward.

n_x, n_y, n_z: rotation vector that brings helicopter back to "level" position (expressed in the helicopter frame).

$$u_{collective} = \theta_1 \cdot f_1(z^* - z) + \theta_2 \cdot \dot{z}$$

$$u_{elevator} = \theta_3 \cdot f_2(x^* - x) + \theta_4 f_4(\dot{x}) + \theta_5 \cdot q + \theta_6 \cdot n_y$$

$$u_{aileron} = \theta_7 \cdot f_3(y^* - y) + \theta_8 f_5(\dot{y}) + \theta_9 \cdot p + \theta_{10} \cdot n_x$$

$$u_{rudder} = \theta_{11} \cdot r + \theta_{12} \cdot n_z$$

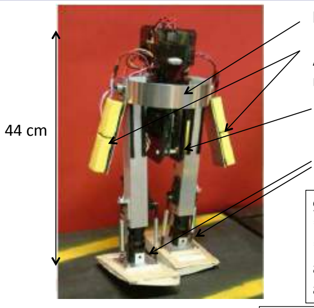
Ball-In-A-Cup

[Kober and Peters, 2009]

Learning to Walk in 20 Minutes

[Tedrake, Zhang, Seung 2005]

Learning to Walk in 20 Minutes



9DOFs:
* 6 internal DOFs
* 3 DOFs for the robot's orientation (always assumed in contact with ground at a single point, absolute (x,y) ignored)

[Tedrake, Zhang, Seung 2005]

Natural gait down 0.03 radians ramp: 0.8Hz, 6.5cm steps

Learning to Walk in 20 Minutes

[Tedrake, Zhang, Seung 2005]

Outline

- Stochastic policy classes
- Gradient computation
  - Known dynamics model
    - Perturbation analysis
  - Unknown dynamics model
    - Finite differences
    - Likelihood ratio
- Step-sizing and KL-divergence trust regions ("natural gradient")
- Guided policy search

Gradient Computation – Known Model

- Reminder of optimization objective:

$$\max_\theta \ U(\theta) = \max_\theta \ \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

- With dynamics model known, can directly compute gradient estimate along sample roll-out:

$$\frac{\partial U}{\partial \theta_i} = \sum_{t=0}^{H} \frac{\partial R}{\partial s}(s_t)\frac{\partial s_t}{\partial \theta_i}$$
$$\frac{\partial s_t}{\partial \theta_i} = \frac{\partial f}{\partial s}(s_{t-1}, u_{t-1})\frac{\partial s_{t-1}}{\partial \theta_i} + \frac{\partial f}{\partial s}(s_{t-1}, u_{t-1})\frac{\partial u_{t-1}}{\partial \theta_i}$$
$$\frac{\partial u_t}{\partial \theta_i} = \frac{\partial \pi_\theta}{\partial \theta_i}(s_t, \theta) + \frac{\partial \pi_\theta}{\partial s}(s_t, \theta)\frac{\partial s_t}{\partial \theta_i}$$

Gradient Computation – Unknown Model – Finite Differences

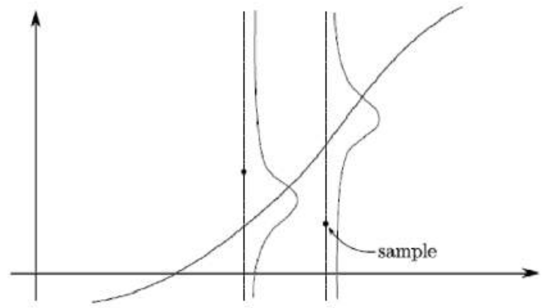
We can compute the gradient g using standard finite difference methods, as follows:

$$\frac{\partial U}{\partial \theta_j}(\theta) = \frac{U(\theta + \epsilon e_j) - U(\theta - \epsilon e_j)}{2\epsilon}$$

Where:

$$e_j = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{ } j\text{'th entry}$$

Noise Can Dominate



Finite Differences and Noise

- Solution 1: Average over many samples
- Solution 2: Fix the randomness (if possible)
  - Intuition by example: wind influence on a helicopter is stochastic, but if we assume the same wind pattern across trials, this will make the different choices of θ more readily comparable
  - General instantiation: Fix the random seed; and the result is deterministic system
  - Ng & Jordan, 2000 provide theoretical analysis of gains from fixing randomness

Gradient Computation – Unknown Model – Likelihood Ratio

We let τ denote a state-action sequence $s_0, u_0, \ldots, s_H, u_H$. We overload notation: $R(\tau) = \sum_{t=0}^{H} R(s_t, u_t)$.

$$U(\theta) = \mathrm{E}[\sum_{t=0}^{H} R(s_t, u_t); \pi_\theta] = \sum_\tau P(\tau; \theta) R(\tau)$$

In our new notation, our goal is to find θ:

$$\max_\theta U(\theta) = \max_\theta \sum_\tau P(\tau; \theta) R(\tau)$$

Likelihood Ratio Gradient

$$U(\theta) = \sum_\tau P(\tau; \theta) R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_\tau P(\tau; \theta) R(\tau) \\
&= \sum_\tau \nabla_\theta P(\tau; \theta) R(\tau) \\
&= \sum_\tau \frac{P(\tau; \theta)}{P(\tau; \theta)} \nabla_\theta P(\tau; \theta) R(\tau) \\
&= \sum_\tau P(\tau; \theta) \frac{\nabla_\theta P(\tau; \theta)}{P(\tau; \theta)} R(\tau) \\
&= \sum_\tau P(\tau; \theta) \nabla_\theta \log P(\tau; \theta) R(\tau)
\end{aligned}$$

Approximate with the empirical estimate for m sample paths under policy π_θ:

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)}; \theta) R(\tau^{(i)})$$

[Note: Can also be derived/generalized through an importance sampling derivation – Tang and Abbeel, 2011]

Likelihood Ratio Gradient Estimate

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta \log \left[\prod_{t=0}^{H} \underbrace{P(s_{t+1}^{(i)}|s_t^{(i)},u_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_\theta(u_t^{(i)}|s_t^{(i)})}_{\text{policy}} \right]$$

$$= \nabla_\theta \left[\sum_{t=0}^{H} \log P(s_{t+1}^{(i)}|s_t^{(i)},u_t^{(i)}) + \sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)}|s_t^{(i)}) \right]$$

$$= \nabla_\theta \sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)}|s_t^{(i)})$$

$$= \sum_{t=0}^{H} \underbrace{\nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)})}_{\text{no dynamics model required!!}}$$

Likelihood Ratio Gradient Estimate

The following expression provides us with an unbiased estimate of the gradient, and we can compute it without access to a dynamics model:

$$\hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta) R(\tau^{(i)})$$

Here:

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \sum_{t=0}^{H} \underbrace{\nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)})}_{\text{no dynamics model required!!}}$$

Unbiased means:

$$\mathrm{E}[\hat{g}] = \nabla_\theta U(\theta)$$

Likelihood Ratio Gradient Estimate

- As formulated thus far: unbiased but very noisy
- Fixes that lead to real-world practicality
  - Baseline
  - Temporal structure
  - Also: KL-divergence trust region / natural gradient (= general trick, equally applicable to perturbation analysis and finite differences)

Likelihood Ratio with Baseline

- Gradient estimate with baseline:

$$\hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta)(R(\tau^{(i)}) - b)$$

- Crudely, increasing log-likelihood of paths with higher than baseline reward and decreasing log-likelihood of paths with lower than baseline reward
- Still unbiased? Yes! $\mathrm{E}\left[\frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta) b\right] = 0$

Likelihood Ratio and Temporal Structure

- Current estimate:

$$\hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta)(R(\tau^{(i)}) - b)$$
$$= \frac{1}{m}\sum_{i=1}^{m} \left(\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)})\right)\left(\sum_{t=0}^{H-1} R(s_t^{(i)},u_t^{(i)}) - b\right)$$

- Future actions do not depend on past rewards, hence can lower variance by instead using:

$$\frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)}) \left(\sum_{k=t}^{H-1} R(s_k^{(i)},u_k^{(i)}) - b\right)$$

Actor-Critic Variant

- Current estimate:

$$\frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)}) \left(\sum_{k=t}^{H-1} R(s_k^{(i)},u_k^{(i)}) - b\right)$$

Sample based estimate of $Q(s_k^{(i)}, u_k^{(i)})$

- Actor-critic algorithms in parallel run an estimator for the Q-function, and substitute in the estimated Q value

Outline

- Stochastic policy classes
- Gradient computation
  - Known dynamics model
    - Perturbation analysis
  - Unknown dynamics model
    - Finite differences
    - Likelihood ratio
- ***Step-sizing and KL-divergence trust regions ("natural gradient")***
- Guided policy search

Step-sizing and Trust Regions

- Naïve step-sizing: Line search
  - Step-sizing necessary as gradient is only first-order approximation
  - Line search in the direction of gradient
    - Simple, but expensive (evaluations along the line)
    - Naïve: ignores where the first-order approximation is good/poor

Step-sizing and Trust Regions

- Advanced step-sizing: Trust regions
- First-order approximation from gradient is a good approximation within "trust region"

→ Solve for best point within trust region:

$$\max_{\delta\theta} \; \hat{g}^\top \delta\theta$$
$$\text{s.t.} \; KL(P(\tau;\theta)||P(\tau;\theta+\delta\theta)) \leq \varepsilon$$

KL Trust Region (a.k.a. natural gradient)

- Solve for best point within trust region:

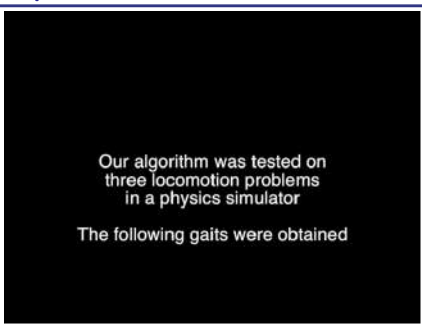
$$\max_{\delta\theta} \; \hat{g}^\top \delta\theta$$
$$\text{s.t.} \; KL(P(\tau;\theta)||P(\tau;\theta+\delta\theta)) \leq \varepsilon$$

- KL can be approximated efficiently with 2^{nd} order expansion:

$$\begin{aligned} KL(P(\tau;\theta)||P(\tau;\theta+\delta\theta)) &\approx \sum_\tau P(\tau;\theta)\delta\theta^\top \nabla_\theta \log P(\tau;\theta) \nabla_\theta \log P(\tau;\theta)^\top \delta\theta \\ &= \delta\theta^\top \left(\sum_\tau P(\tau;\theta) \nabla_\theta \log P(\tau;\theta) \nabla_\theta \log P(\tau;\theta)^\top \right) \delta\theta \\ &= \delta\theta^\top G(\theta) \delta\theta \end{aligned}$$

G: Fisher Information Matrix

Experiments in Locomotion



[Schulman, Levine, Abbeel, 2014]

Outline

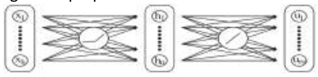
- Stochastic policy classes
- Gradient computation
  - Known dynamics model
    - Perturbation analysis
  - Unknown dynamics model
    - Finite differences
    - Likelihood ratio
- Step-sizing and KL-divergence trust regions ("natural gradient")
- ***Guided policy search***

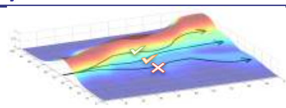
Guided Policy Search

- Get best of both worlds:
  - Policy search
    - Parameterized policy class
    - Can be difficult to "get going"
  - Optimization-based planning
    - Can find trajectories directly (assuming a model)
    - But needs to compute a new trajectory for each new situation

Guided Policy Search

general-purpose neural network controller



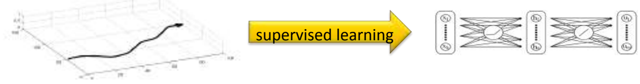


policy search (RL) complex dynamics complex policy HARD

supervised learning ~~complex dynamics~~ complex policy EASY

trajectory optimization complex dynamics ~~complex policy~~ EASY

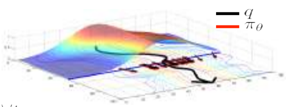
trajectory optimization



Constrained Guided Policy Search

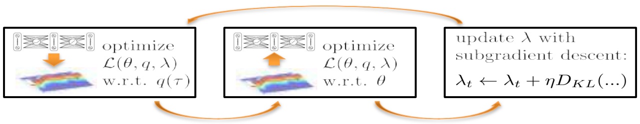
$\min_\theta E_{\pi_\theta}[c(\tau)]$

$\min_{\theta, q(\tau)} E_q[c(\tau)]$



s.t. $E_{q(\mathbf{x}_t)}[D_{KL}(\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\|q(\mathbf{u}_t|\mathbf{x}_t))] = 0 \ \forall t$

$$\mathcal{L}(\theta, q, \lambda) = E_q[c(\tau)] + \sum_{t=1}^{T} \lambda_t E_{q(\mathbf{x}_t)}[D_{KL}(\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\|q(\mathbf{u}_t|\mathbf{x}_t))]$$



[Levine & Koltun ICML 2014]

Experiments Constrained Guided Policy Search

2D Insertion

unobserved slot positions

[neural network]

Outline

- Stochastic policy classes
- Gradient computation
  - Known dynamics model
    - Perturbation analysis
  - Unknown dynamics model
    - Finite differences
    - Likelihood ratio
- Step-sizing and KL-divergence trust regions ("natural gradient")
- Guided policy search

Thank you

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Robot world representation and reasoning in it

1

Roman Barták
Charles University in Prague, Faculty of Mathematics and Physics

Knowledge Representation and Reasoning: SAT and CSP

Introduction

- Assume a rational agent that can form **representations** of a complex world, use a process of **inference** to derive new information about the world, and use that information to **deduce** what to do.
- Such agent is called a **knowledge-based agent** – combines and recombines information about the world with current observations to uncover hidden aspects of the world and use them for action selection.
- We need to know:
  – how to represent **knowledge?**
  – how to **reason** over that knowledge?

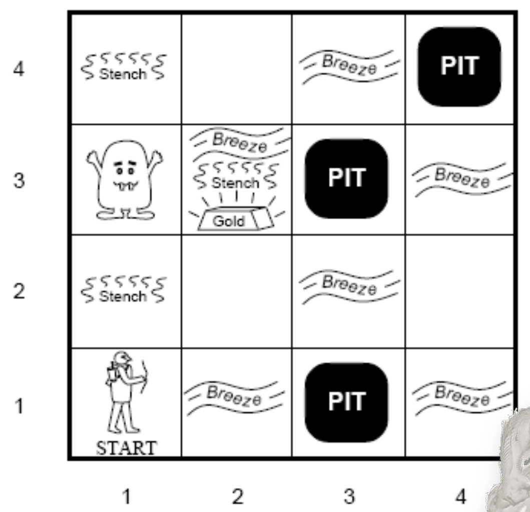
- A knowledge-based agent uses a **knowledge base** – a set of sentences expressed in a given language – that can be updated by operation TELL and can be queried about what is known using operation ASK.
- Answers to queries may involve **inference** – that is deriving new sentences from old (inserted using the TELL operations).

```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time
    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action
```

knowledge base contains information about observations as well as about own actions

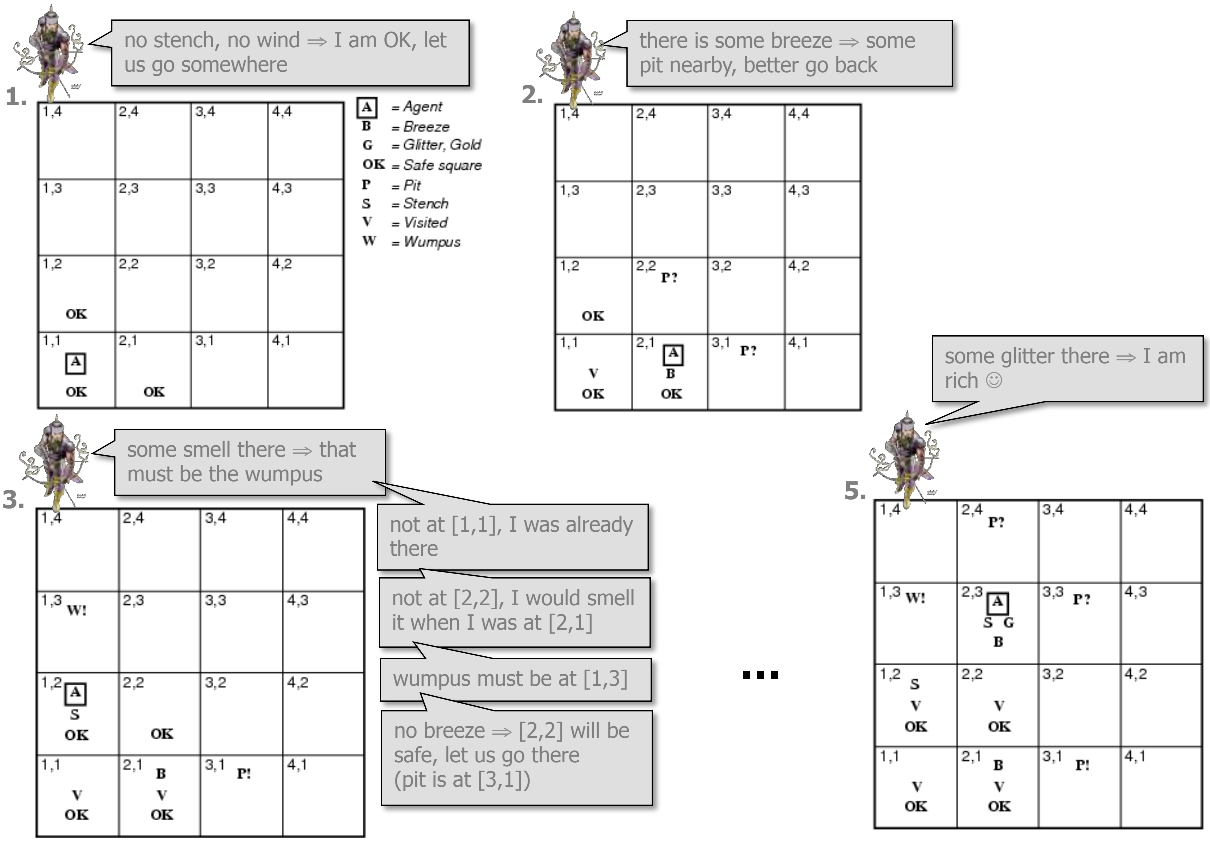
inference will help the agent to select an action even if information about the world is incomplete

- A cave consisting of rooms connected by passageways, inhabited by the terrible **wumpus**, a beast that eats anyone who enters its room, containing rooms with bottomless **pits** that will trap anyone, and a room with a heap of **gold**.



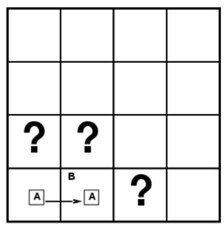
– The agent will perceive a **Stench** in the directly (not diagonally) adjacent squares to the square containing the wumpus.
– In the squares directly adjacent to a pit, the agent will perceive a **Breeze**.
– In the square where the gold is, the agent will perceive a **Glitter**.
– When an agent walks into a wall, it will perceive a **Bump**.
– The wumpus can be shot by an agent, but the agent has only one arrow.
  - Killed wumpus emits a woeful **Scream** that can be perceived anywhere in the cave.

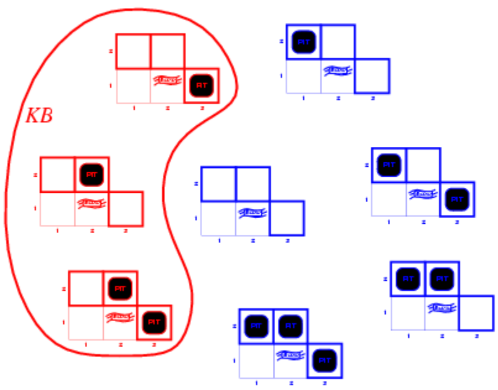
The Wumpus world – the quest for gold



The Wumpus world – possible models

- Assume a situation when there is no percept at [1,1], we went right to [2,1] and feel Breeze there.





  - For pit detection we have 8 ($=2^3$) possible models (states of the neighbouring world).
  - Only three of these models correspond to our knowledge base, the other models conflict the observations:
    - no percept at [1,1]
    - Breeze at [2,1]

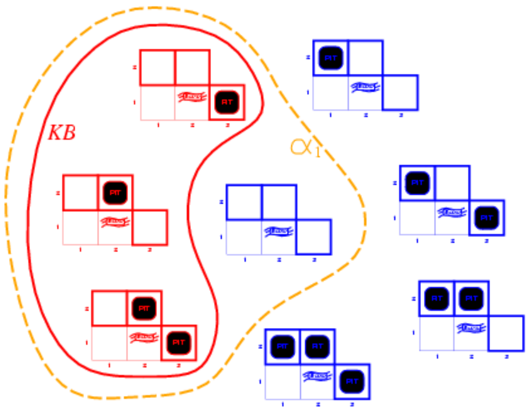
The Wumpus world – some consequences

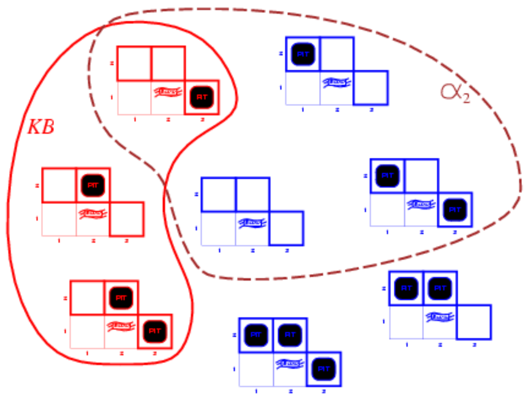
Let us ask whether the room [1,2] is safe.

- Is information α_1 = „[1,2] is safe" entailed by our representation?
- we compare models for KB and for α_1
- every model of KB is also a model for α_1 so α_1 is entailed by KB

And what about room [2,2]?

- we compare models for KB and for α_2
- some models of KB are not models of α_2
- α_2 is not entailed by KB and we do not know for sure if room [2,2] is safe



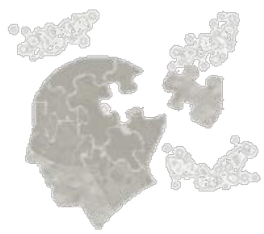


Inference in general

How to implement inference in general?

We can use **propositional logic**. Sentences are propositional expression and a knowledge base will be a conjunction of these expressions.

- **Propositional variables** describe the properties of the world
  - $P_{i,j}$ = **true** iff there is a pit at [i, j]
  - $B_{i,j}$ = **true** if the agent perceives Breeze at [i, j]
- **Propositional formulas** describe
  - known information about the world
    - $\neg P_{1,1}$ no pit at [1, 1] (we are there)
  - general knowledge about the world (for example, Breeze means a pit in some neighbourhood room)
    - $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
    - $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
    - …
  - observations
    - $\neg B_{1,1}$ no Breeze at [1, 1]
    - $B_{2,1}$ Breeze at [2, 1]
- We will be using **inference** for propositional logic.



- **Syntax** defines the allowable sentences.
  - a propositional variable (and constants true and false) is an (atomic) sentence
  - two sentences can be connected via logical connectives ¬, ∧, ∨, ⇒, ⇔ to get a (complex) sentence
- **Semantics** defines the rules for determining the truth of a sentence with respect to a particular model.
  - **model** is an assignment of truth values to all propositional variables
  - an atomic sentence P is true in any model containing P=true
  - semantics of complex sentences is given by the truth table

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

- M is a **model** of sentence α, if α is true in M.
  - The set of models for α is denoted M(α).
- **entailment: KB $\models \alpha$**
  means that α is a logical consequence of KB
  - KB entails α iff M(KB) $\subseteq$ M(α)
- We are interested in **inference methods**, that can find/verify consequences of KB.
  - KB $\vdash_i \alpha$ means that algorithm i infers sentence α from KB
  - the algorithm is **sound** iff KB $\vdash_i \alpha$ implies KB $\models \alpha$
  - the algorithm is **complete** iff KB $\models \alpha$ implies KB $\vdash_i \alpha$

Inference methods

- There are basically two classes of inference algorithms.
  - **model checking**
    - based on enumeration of a truth table
    - **Davis-Putnam-Logemann-Loveland (DPLL)**
    - local search (minimization of conflicts)
  - **inference rules**
    - theorem proving by applying inference rules
    - a resolution algorithm

A bit of logic

- Sentence (formula) is **satisfiable** if it is true in, or satisfied by, *some* model.
  Example: $A \vee B$, C
- Sentence (formula) is **unsatisfiable** if it is not true in *any* model.
  Example: $A \wedge \neg A$
- Entailment can then be implemented as checking satisfiability as follows:
  KB $\models \alpha$ if and only if **(KB $\wedge \neg\alpha$) is unsatisfiable.**
  - proof by **refutation**
  - proof by **contradiction**
- Verifying if α is entailed by KB can be implemented as the satisfiability problem for the formula (KB $\wedge \neg\alpha$).
  Usually the formulas are in a **conjunctive normal form** (CNF)
  - **literal** is an atomic variable or its negation
  - **clause** is a disjunction of literals
  - **formula** in CNF is a conjunction of clauses

  Example: $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

  Each propositional sentence (formula) can be represents in CNF.

$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$
$$(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$$
$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$$
$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$$
$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$$

DPLL

Davis, Putnam, Logemann, Loveland

– a sound and complete algorithm for verifying satisfiablity of formulas in a CNF (finds its model)

```
function DPLL-SATISFIABLE?(s) returns true or false
   inputs: s, a sentence in propositional logic

   clauses ← the set of clauses in the CNF representation of s
   symbols ← a list of the proposition symbols in s
   return DPLL(clauses, symbols, [ ])

function DPLL(clauses, symbols, model) returns true or false
   if every clause in clauses is true in model then return true
   if some clause in clauses is false in model then return false
   P, value ← FIND-PURE-SYMBOL(symbols, clauses, model)
   if P is non-null then return DPLL(clauses, symbols–P, [P = value|model])
   P, value ← FIND-UNIT-CLAUSE(clauses, model)
   if P is non-null then return DPLL(clauses, symbols–P, [P = value|model])
   P ← FIRST(symbols); rest ← REST(symbols)
   return DPLL(clauses, rest, [P = true|model]) or
          DPLL(clauses, rest, [P = false|model])
```

Early termination for partial models
- clause is true if any of its literals is true
- formula is not true if any of its clauses is not true

Pure symbol heuristics
- a pure symbol always appears with the same „sign" in all clauses
- the corresponding literal set to true

Unit clause heuristics
- a unit clause is a clause with just one literal
- the corresponding literal set to true

branching for backtracking

The Wumpus world – knowledge base

- For simplicity we will represent only the „physics" of the wumpus world.
  - we know that
    - $\neg P_{1,1}$
    - $\neg W_{1,1}$
  - we also know why and where breeze appears
    - $B_{x,y} \Leftrightarrow (P_{x,y+1} \lor P_{x,y-1} \lor P_{x+1,y} \lor P_{x-1,y})$
  - and why a smell is generated
    - $S_{x,y} \Leftrightarrow (W_{x,y+1} \lor W_{x,y-1} \lor W_{x+1,y} \lor W_{x-1,y})$
  - and finally one "hidden" information that there is a single Wumpus in the world
    - $W_{1,1} \lor W_{1,2} \lor \ldots \lor W_{4,4}$
    - $\neg W_{1,1} \lor \neg W_{1,2}$
    - $\neg W_{1,1} \lor \neg W_{1,3}$
    - …
- We should also include information about the **agent**.
  - where the agent is
    - $L^1_{1,1}$
    - $FacingRight^1$
  - and what happens when agent performs actions
    - $L^t_{x,y} \land FacingRight^t \land Forward^t \Rightarrow L^{t+1}_{x+1,y}$
    - we need an upper bound for the number of steps and still it will lead to a huge number of formulas

The Wumpus world – an agent

function PL-WUMPUS-AGENT(*percept*) **returns** an *action*
  inputs: *percept*, a list, [*stench*, *breeze*, *glitter*]
  static: *KB*, initially containing the "physics" of the wumpus world
    x, *y*, *orientation*, the agent's position (init. [1,1]) and orient. (init. *right*)
    visited, an array indicating which squares have been visited, initially *false*
    action, the agent's most recent action, initially null
    plan, an action sequence, initially empty

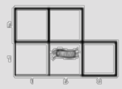
  update *x*,*y*,*orientation*, *visited* based on *action*
  if *stench* **then** TELL($KB, S_{x,y}$) **else** TELL($KB, \neg S_{x,y}$)
  if *breeze* **then** TELL($KB, B_{x,y}$) **else** TELL($KB, \neg B_{x,y}$)
  if *glitter* **then** *action* ← *grab*
  else if *plan* is nonempty **then** *action* ← POP(*plan*)
  else if for some fringe square [*i*,*j*], ASK($KB, (\neg P_{i,j} \wedge \neg W_{i,j})$) is *true* **or**
    for some fringe square [*i*,*j*], ASK($KB, (P_{i,j} \vee W_{i,j})$) is *false* **then do**
   plan ← A*-GRAPH-SEARCH(ROUTE-PB([*x*,*y*], *orientation*, [*i*,*j*], *visited*))
   action ← POP(*plan*)
  else *action* ← a randomly chosen move
  return *action*

Include information about observations

Try to find a safe room located in thefringe.

Or at least a room that is no provably unsafe.

Find a sequence of actions moving the agent to the selected room via known rooms.

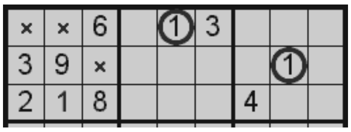


Logical frameworks - a survey

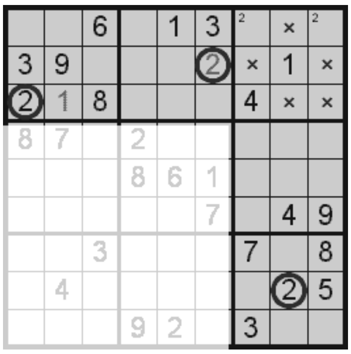
Propositional logic	facts that hold or not
First-order logic	objects and relations between them
Temporal logic	facts and times when they hold
Higher-order logic	relations between objects are used as objects (there are claims on relations)

Combinatorial puzzle, whose goal is to enter digits 1-9 in cells of 9×9 table in such a way, that no digit appears twice or more in every row, column, and 3×3 sub-grid.

Solving Sudoku

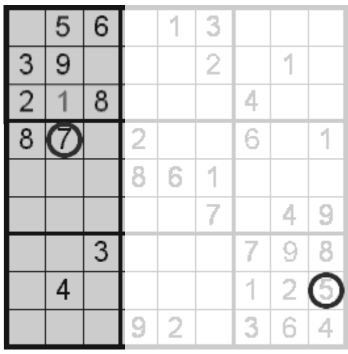


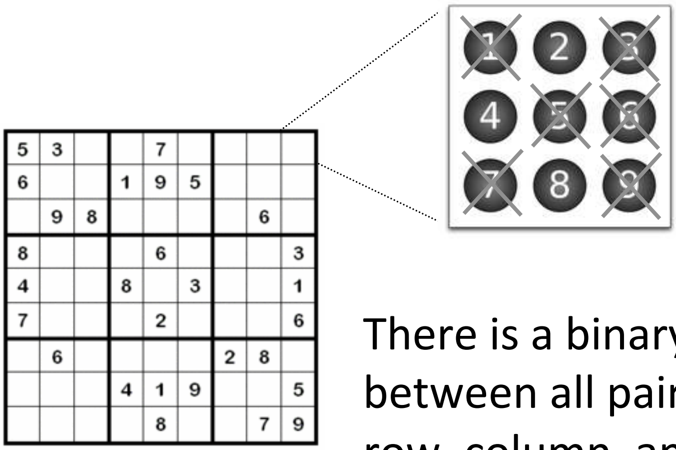
Use information that each digit appears exactly once in each row, column and sub-grid.



- If neither rows and columns provide enough information, we can note allowed digits in each cell.

- The position of a digit can be inferred from positions of other digits and restrictions of Sudoku that each digit appears one in a column (row, sub-grid)





We can see every cell as a **variable** with possible values from **domain** {1,…,9}.

There is a binary inequality **constraint** between all pairs of variables in every row, column, and sub-grid.

Such formulation of the problem is called a **constraint satisfaction problem.**

Constraint satisfaction problem consists of:

- a finite set of **variables**
  - describe some features of the world state that we are looking for, for example positions of queens at a chessboard
- **domains** – finite sets of values for each variable
  - describe “options” that are available, for example the rows for queens
  - sometimes, there is a single common “superdomain” and domains for particular variables are defined via unary constraints
- a finite set of **constraints**
  - a constraint is a *relation* over a subset of variables for example rowA ≠ rowB
  - a constraint can be defined *in extension* (a set of tuples satisfying the constraint) or using a *formula* (see above)

A feasible solution of a constraint satisfaction problem is a complete consistent assignment of values to variables.

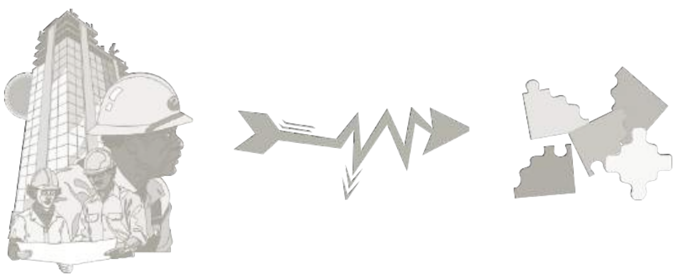
- **complete** = each variable has assigned a value
- **consistent** = all constraints are satisfied

Sometimes we may look for all the feasible solutions or for the number of feasible solutions.

An optimal solution of a constraint satisfaction problem is a feasible solution that minimizes/maximizes a value of some objective function.

- **objective function** = a function mapping feasible solutions to real numbers

- **Problem Modelling**
  How to describe a problem as a constraint satisfaction problem?



- **Solving Techniques**
  How to find values for the variables satisfying all the constraints?

Solving CSPs

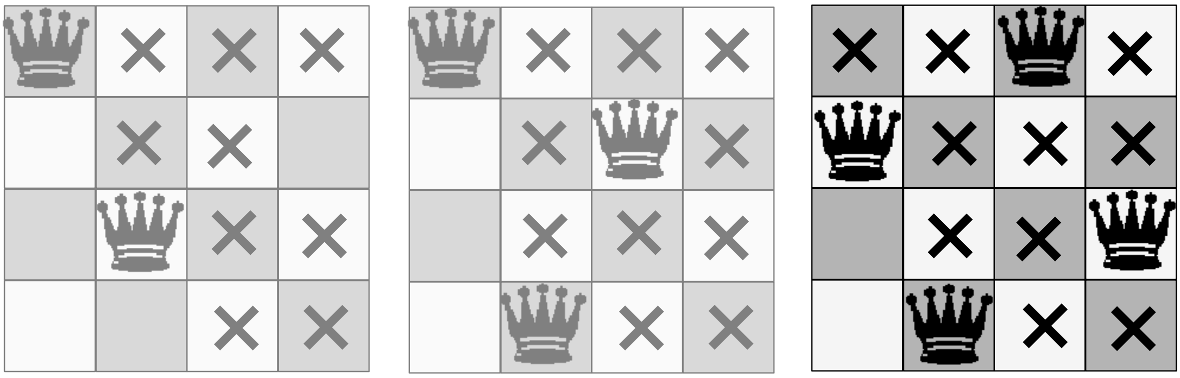
N-queens: allocate N queens to a chess board of size N×N in a such way that no two queens attack each other

the modelling decision: each queen is located in its own column

variables: N variables r(i) with the domain {1,…,N}

constraints: no two queens attack each other

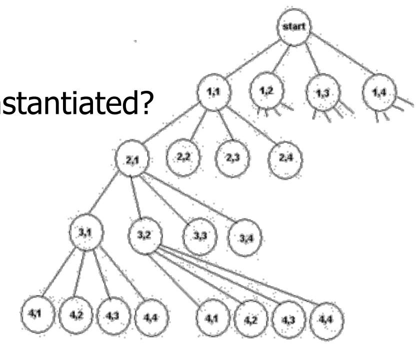
$$\forall i \neq j \quad r(i) \neq r(j) \land |i-j| \neq |r(i)-r(j)|$$



Backtracking

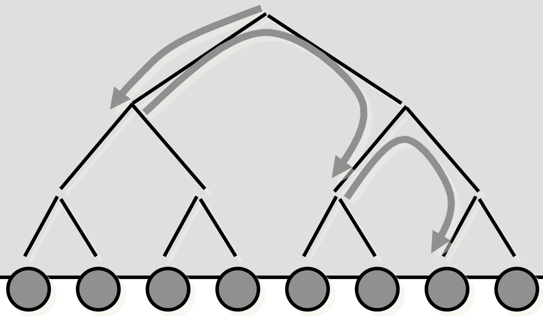
- Probably the most widely used systematic search algorithm that **verifies the constraints as soon as possible**.
  - upon failure (any constraint is violated) the algorithm goes back to the last instantiated variable and tries a different value for it
  - depth-first search
- The core principle of applying backtracking to solve a CSP:
  1. assign values to variables one by one
  2. after each assignment verify satisfaction of constraints with known values of all constrained variables

  Open questions:
    - What is the order of variables being instantiated?
    - What is the order of values tried?
- **Backtracking explores partial consistent assignments until it finds a complete (consistent) assignment.**



Chronological Backtracking (a recursive version)

```
procedure BT(X:variables, V:assignment, C:constraints)
    if X={} then return V
    x ← select a not-yet assigned variable from X
    for each value h from the domain of x do
       if constraints C are consistent with V ∪ {x/h} then
           R ← BT(X – {x}, V ∪ {x/h}, C)
           if R ≠ fail then return R
    end for
    return fail
end BT
Call as BT(X, {}, C)
```



Note:

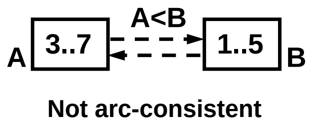
If it is possible to perform the test stage for a partially generated solution candidate then BT is always better than GT, as BT does not explore all complete solution candidates.

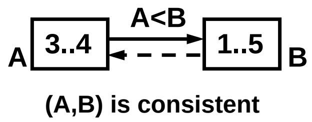
Constraint consistency

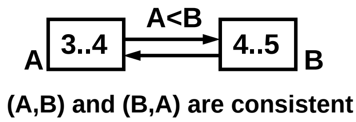
Example:

A in [3,..,7], B in [1,..,5], A<B

Constraint can be used to **prune the domains** actively using a dedicated filtering algorithm!







Some definitions:

The arc (Vi,Vj) is **arc consistent** iff for each value x from the domain Di there exists a value y in the domain Dj such that the assignment Vi =x a Vj = y satisfies all the binary constraints on Vi, Vj.

CSP is **arc consistent** iff every arc (Vi,Vj) is arc consistent (in both directions).

Algorithm AC-3

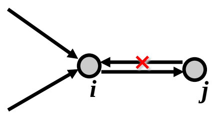
```
function AC-3( csp) returns the CSP, possibly with reduced domains
   inputs: csp, a binary CSP with variables {X1, X2, ..., Xn}
   local variables: queue, a queue of arcs, initially all the arcs in csp

   while queue is not empty do
      (Xi, Xj) ← REMOVE-FIRST(queue)
      if RM-INCONSISTENT-VALUES(Xi, Xj) then
         for each Xk in NEIGHBORS[Xi] do
            add (Xk, Xi) to queue

function RM-INCONSISTENT-VALUES( Xi, Xj) returns true iff remove a value
   removed ← false
   for each x in DOMAIN[Xi] do
      if no value y in DOMAIN[Xj] allows (x,y) to satisfy constraint(Xi, Xj)
         then delete x from DOMAIN[Xi];   removed ← true
   return removed
```

The algorithm can be applied incrementally during search – when X is instantiated put to queue all constraints related to X.

If the domain of variable X_i changed then verify all arcs (constraints) leading to the variable except the arc from the variable X_j.

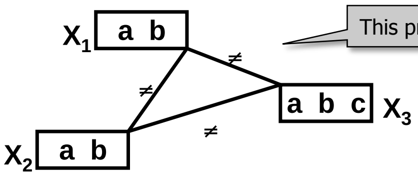


Domain filtering for variable X_i removes values that have no support in the variable X_j, also, if any value is deleted this information is passed to the calling procedure. Knowing constraint semantics can speedup constraint checking (for example X<Y).

Time complexity of AC-3 is $O(ed^3)$, where e is the number of constraints and d is the size of domain – we need to repeatedly (ed) check the constraints (d^2). This is not optimal, we can remember the result of consistency checks - AC-4, AC-3.1, AC-2001 with time complexity $O(ed^2)$.

Stronger consistency

- We can generally define **k-consistency**, as the consistency check where for a consistent assignment of (k-1) variables we require a consistent value in one more given variable.
  - arc consistency (AC) = 2-consistency
  - path consistency (PC) = 3-consistency



This problem is AC, but not PC.

- If the problem is i-consistent ∀i=1,..,n (n is the number of variables), then we can solve it in a backtrack-free way.
  - DFS can always find a value consistent with the assignment of previous variables
- Unfortunately, the time complexity of k-consistency is exponential in k.

Régin (1994)

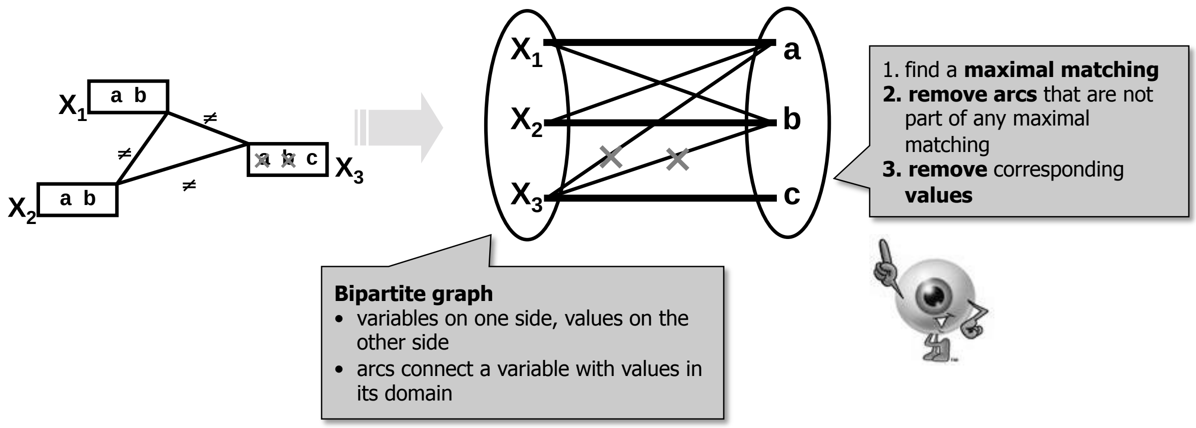
Global constraints

Instead of stronger consistency techniques (expensive) usually **global constraints** are used – a global constraint encapsulates a sub-problem with a specific structure that can be exploited in the ad-doc domain filtering procedure.

Example:

global constraint **all_different({X_1,…, X_k})**

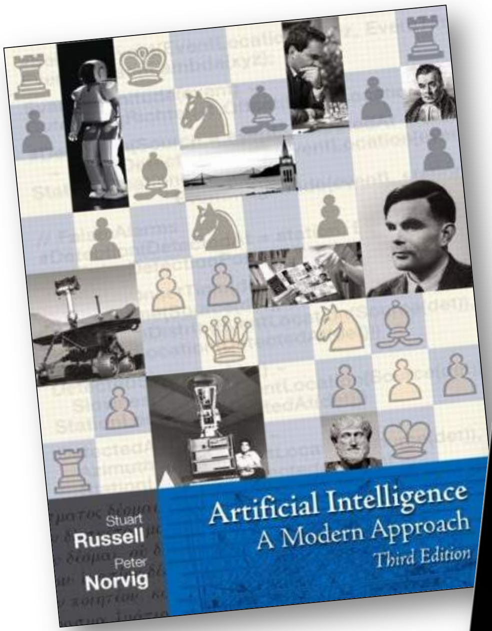
- encapsulates a set of binary inequalities $X_1 \neq X_2$, $X_1 \neq X_3$, …, $X_{k-1} \neq X_k$
- **all_different({X_1,…, X_k})** = $\{(d_1,\dots, d_k) \mid \forall i \ d_i \in D_i \ \land \ \forall i \neq j \ d_i \neq d_j\}$
- the filtering procedure is based on matching in bipartite graphs

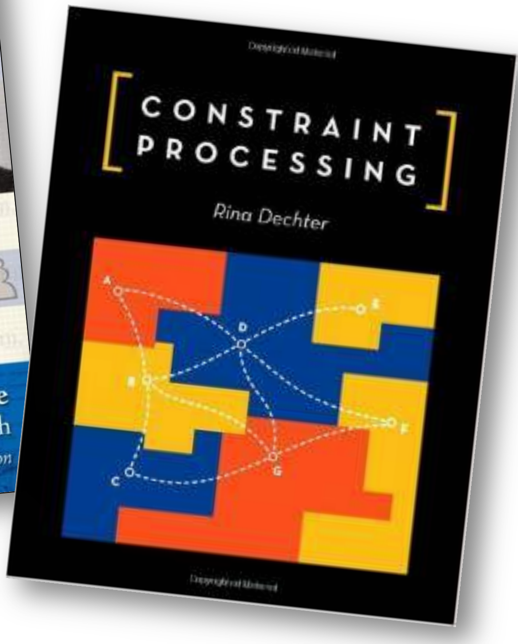


SAT vs. CSP

Based on a similar core reasoning principle – backtracking + inference

- **SAT (Boolean satisfiability)**
  - uniform modeling formalism (CNF)
  - big efficiency leap of SAT solvers (clause learning)
  - grounding (formula size)
- **CSP (Constraint Satisfaction Problem)**
  - flexible modeling framework
  - “good old solvers”
  - integration of ad-hoc solvers (global constraints)





Charles University in Prague, Faculty of Mathematics and Physics
bartak@ktiml.mff.cuni.cz

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

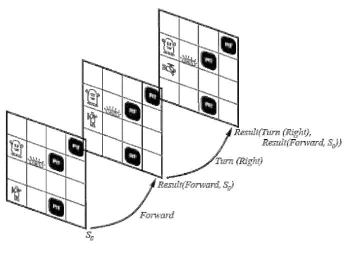
Robot world representation and reasoning in it 2

Roman Barták
Charles University in Prague, Faculty of Mathematics and Physics

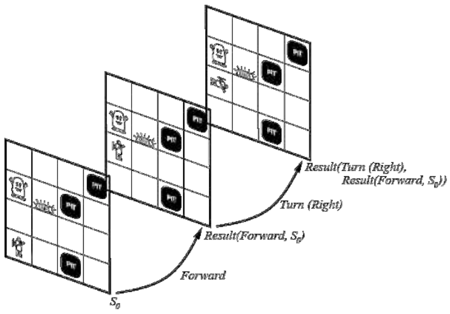
Planning and Temporal Models

Actions and situations

- So far we modelled a static world only.
- **How to reason about actions and their effects in time?**
- **In propositional logic** we need a copy of each action for each time (situation):
  - $L^{t}_{x,y} \land FacingRight^{t} \land Forward^{t} \Rightarrow L^{t+1}_{x+1,y}$
  - We need an upper bound for the number of steps to reach a goal but this will lead to a huge number of formulas.
- Can we do it better in **first order logic**?
  - We do not need copies of axioms describing state changes; this can be implemented using a universal quantifier for time (situation)
  - ∀t P is the result of action A in time t+1



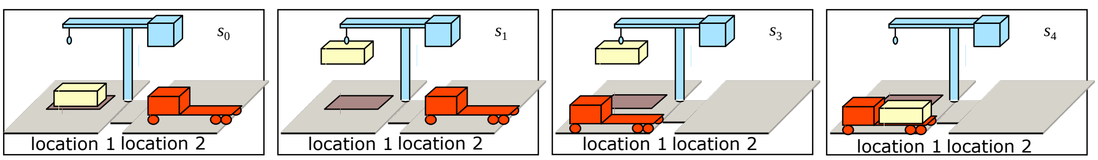
Situation calculus



- **actions** are represented by terms
  - Go(x,y)
  - Grab(g)
  - Release(g)
- **situation** is also a term
  - initial situation: S_0
  - situation after applying action *a* to state *s*: Result(a,s)
- **fluent** is a predicates changing with time
  - the situation is in the last argument of that predicate
  - Holding(G, S_0)
- **rigid** (eternal) **predicates**
  - Gold(G)
  - Adjacent(x,y)

Situation calculus: plans

- We need to reason about sequences of actions – about **plans**.
  - Result([],s) = s
  - Result([a|seq],s) = Result(seq, Result(a,s))
- What are the typical tasks related to plans?
  - **projection task** – what is the state/situation after applying a given sequence of actions?
    - At(Agent, [1,1] , S_0) ∧ At(G, [1,2], S_0) ∧ ¬Holding(o, S_0)
    - At(G, [1,1], Result([Go([1,1],[1,2]),Grab(G),Go([1,2],[1,1])], S_0))
  - **planning task** – which sequence of actions reaches a given state/situation?
    - ∃seq At(G, [1,1], Result(seq, S_0))



Situation calculus: actions

- Each **action** can be described using two axioms:
  - **possibility axiom:** Preconditions ⇒ Poss(a,s)
    - At(Agent,x,s) ∧ Adjacent(x,y) ⇒ Poss(Go(x,y),s)
    - Gold(g) ∧ At(Agent,x,s) ∧ At(g,x,s) ⇒ Poss(Grab(g),s)
    - Holding(g,s) ⇒ Poss(Release(g),s)
  - **effect axiom:** Poss(a,s) ⇒ Changes
    - Poss(Go(x,y),s) ⇒ At(Agent,y,Result(Go(x,y),s))
    - Poss(Grab(g),s) ⇒ Holding(g,Result(Grab(g),s))
    - Poss(Release(g),s) ⇒ ¬Holding(g,Result(Release(g),s))
- Beware! This is not enough to deduce that a plan reaches a given goal.
  - we can deduce At(Agent, [1,2], Result(Go([1,1],[1,2]), S_0))
  - but we **cannot deduce** At(G, [1,2], Result(Go([1,1],[1,2]), S_0))
  - Effect axioms describe what has been changed in the world but say nothing about the property that everything else is not changed!
  - This is a so called **frame problem.**

Frame problem

- We need to represent properties that are not changed by actions.
- A simple **frame axiom** says what is not changed:

  At(o,x,s) ∧ o≠Agent ∧ ¬Holding(o,s) ⇒
    At(o,x,Result(Go(y,z),s))

  - for F fluents and A actions we need O(FA) frame axioms
  - This is a lot especially taking in account that most predicates are not changed.

Frame problem: better axioms

Can we use less axioms to model the frame problem?

- **successor-state axiom**

  Poss(a,s) ⇒
  (fluent holds in Result(a,s) ⇔
  fluent is effect of a ∨ (fluent holds in s ∧ a does not change fluent))

  – We get F axioms (F is the number of fluents) with O(AE) literals in total (A is the number of actions, E is the number of effects).

  Examples:

  $Poss(a,s) \Rightarrow$
  $(At(Agent,y,Result(a,s)) \Leftrightarrow a=Go(x,y) \lor (At(Agent,y,s) \land a \neq Go(y,z)))$

  $Poss(a,s) \Rightarrow$
  $(Holding(g,Result(a,s)) \Leftrightarrow a=Grab(g) \lor (Holding(g,s) \land a \neq Release(g)))$

  – **Beware of implicit effects!**
    - If an agent holds some object and the agent moves then also the object moves.
    - This is called a **ramification problem**.

  $Poss(a,s) \Rightarrow$
  $(At(o,y,Result(a,s)) \Leftrightarrow$
  $(a=Go(x,y) \land (o=Agent \lor Holding(o,s))) \lor$
  $(At(o,y,s) \land \neg\exists z\ (y \neq z \land a=Go(y,z) \land (o=Agent \lor Holding(o,s)))))$

Frame problem: even better axioms

- Successor-state axiom is still too big with O(AE/F) literals in average.
  – To solve the projection task with t actions, the time complexity depends on the total number of actions – O(AEt) – rather than on the actions in plan.
  – If we know each action, cannot we do it better say O(Et)?
- **classical successor-state axiom:**

  $Poss(a,s) \Rightarrow$
  $(F_i(Result(a,s)) \Leftrightarrow (a=A_1 \lor a=A_2 \lor \ldots) \lor (F_i(s) \land a \neq A_3 \land a \neq A_4 \ldots))$

  actions having F_i among effects   **actions having $\neg F_i$ among effects**

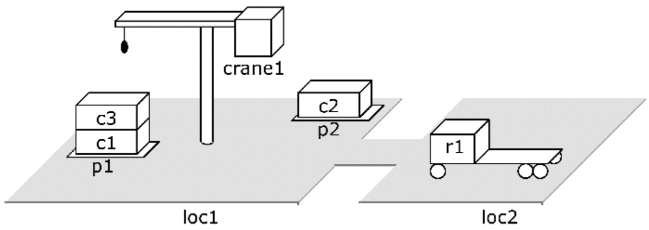
- We can introduce **positive** and **negative effects** of actions:
  – **PosEffect(a, F_i)** action a causes F_i to become true
  – **NegEffect(a, F_i)** action a causes F_i to become false
- **modified successor state axiom:**

  $Poss(a,s) \Rightarrow (F_i(Result(a,s)) \Leftrightarrow PossEffect(a, F_i) \lor (F_i(s) \land \neg NegEffect(a,F_i)))$
  $PosEffect(A_1, F_i)$
  $PosEffect(A_2, F_i)$
  $NegEffect(A_3, F_i)$
  $NegEffect(A_4, F_i)$

Classical representation: states

We can simplify the full FOL model into a so called **classical representation** of planning problems.

State is a set of instantiated atoms (no variables). There is a finite number of states!



{attached(p1,loc1), in(c1,p1), in(c3,p1), top(c3,p1), on(c3,c1), on(c1,pallet), attached(p2,loc1), in(c2,p2), top(c2,p2), on(c2,pallet), belong(crane1,loc1), empty(crane1), adjacent(loc1,loc2), adjacent(loc2,loc1), at(r1,loc2), occupied(loc2), unloaded(r1)}.

- The truth value of some atoms is changing in states:
  - **fluents**
  - *example: at(r1,loc2)*
- The truth value of some state is the same in all states
  - **rigid atoms**
  - *example: adjacent(loc1,loc2)*

We will use a classical **closed world assumption**.
An atom that is not included in the state does not hold at that state!

Classical representation: operators

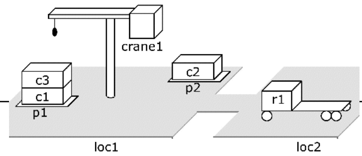
operator o is a triple (name(o), precond(o), effects(o))

- **name(o): name of the operator** in the form n(x_1,…,x_k)
  - n: a symbol of the operator (a unique name for each operator)
  - x_1,…,x_k: symbols for variables (operator parameters)
    - Must contain all variables appearing in the operator definition!
- **precond(o):**
  - literals that must hold in the state so the operator is applicable on it
- **effects(o):**
  - literals that will become true after operator application (only fluents can be there!)

take(k, l, c, d, p)
;; crane k at location l takes c off of d in pile p
precond: belong(k, l), attached(p, l), empty(k), top(c, p), on(c, d)
effects: holding(k, c), $\neg$ empty(k), $\neg$ in(c, p), $\neg$ top(c, p), $\neg$ on(c, d), top(d, p)

An action is a fully instantiated operator

– substitute constants to variables



operator

take(k, l, c, d, p)
;; crane k at location l takes c off of d in pile p
precond: belong(k, l), attached(p, l), empty(k), top(c, p), on(c, d)
effects: holding(k, c), $\neg$ empty(k), $\neg$ in(c, p), $\neg$ top(c, p), $\neg$ on(c, d), top(d, p)

action

take(crane1,loc1,c3,c1,p1)
;; crane crane1 at location loc1 takes c3 off c1 in pile p1
precond: belong(crane1,loc1), attached(p1,loc1), empty(crane1), top(c3,p1), on(c3,c1)
effects: holding(crane1,c3), ¬empty(crane1), ¬in(c3,p1), ¬top(c3,p1), ¬on(c3,c1), top(c1,p1)

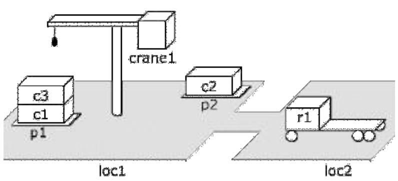
Notation:

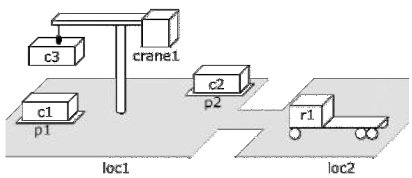
- S^+ = {positive atoms in S}
- S^- = {atoms, whose negation is in S}

Action **a** is **applicable** to state **s** if any only
precond$^+$(**a**) $\subseteq$ **s** $\wedge$ precond$^-$(**a**) $\cap$ **s** = $\emptyset$

The result of application of action a to **s** is
γ(**s**,**a**) = (**s** – effects$^-$(**a**)) $\cup$ effects$^+$(**a**)

take(crane1,loc1,c3,c1,p1)
;; crane crane1 at location loc1 takes c3 off c1 in pile p1
precond: belong(crane1,loc1), attached(p1,loc1), empty(crane1), top(c3,p1), on(c3,c1)
effects: holding(crane1,c3), ¬empty(crane1), ¬in(c3,p1), ¬top(c3,p1), ¬on(c3,c1), top(c1,p1)





Classical representation: planning domain

Let L be a language and O be a set of operators.

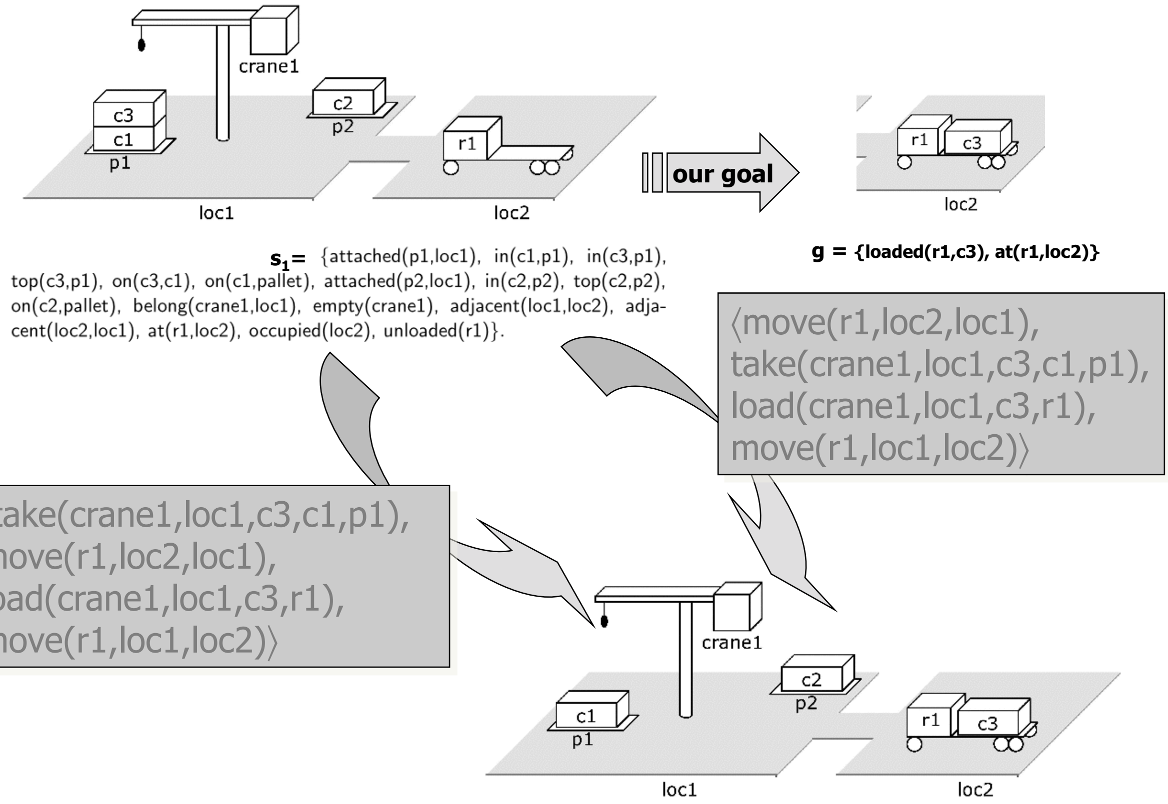
Planning domain Σ over language L with operators O is a triple (S,A,γ):

- **states** $S \subseteq P$({all instantiated atoms from L})
- **actions** A = {all instantiated operators from O over L}
  - action **a** is **applicable** to state **s** if $precond^+(\mathbf{a}) \subseteq \mathbf{s} \;\wedge\; precond^-(\mathbf{a}) \cap \mathbf{s} = \emptyset$
- **transition function** γ:
  - $\gamma(\mathbf{s},\mathbf{a}) = (\mathbf{s} - effects^-(\mathbf{a})) \cup effects^+(\mathbf{a})$, if **a** is applicable on **s**
  - S is closed with respect to γ (if $\mathbf{s} \in S$, then for every action **a** applicable to **s** it holds $\gamma(\mathbf{s},\mathbf{a}) \in S$)

Classical representation: planning problem

- **Planning problem** P is a triple (Σ,s_0,g):
  - $\Sigma = (S,A,\gamma)$ is a planning domain
  - s_0 is an initial state, $s_0 \in S$
  - g is a set of instantiated literals
    - state **s** satisfies the goal condition **g** if and only if $\mathbf{g}^+ \subseteq \mathbf{s} \;\wedge\; \mathbf{g}^- \cap \mathbf{s} = \emptyset$
    - $S_g = \{\mathbf{s} \in S \mid \mathbf{s}$ satisfies $\mathbf{g}\}$ – a set of goal states
- **Plan** is a sequence of actions $\langle a_1,a_2,\ldots,a_k\rangle$**.**
- Plan $\langle a_1,a_2,\ldots,a_k\rangle$ is a **solution plan** for problem P iff $\gamma^*(s_0,\pi)$ satisfies the goal condition g.
- Usually the planning problem is given by a triple (O,s_0,g).
  - O defines the the operators and predicates used
  - s_0 provides the particular constants (objects)

Classical representation: example plan



Multi-valued state variables

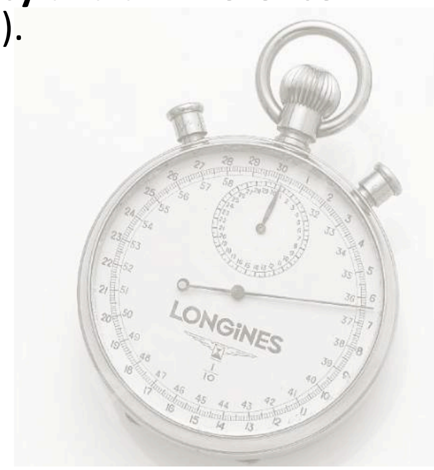
- **Multi-valued state variables** describe the properties of objects that are changing between the states (by actions).
  - rloc: robots × S → locations
  - rload: robots × S → containers ∪ {nil}
  - cpos: containers × S → locations ∪ robots
- **Rigid relations** are (still) represented using relations.
  - adjacent(loc1,loc2)
  - robots(r1) ;; describes the types of constants
- **Operators** describe changes of state variables.
  - **move(r,l,m)**
    ;; robot r at location l moves to an adjacent position m
    precond: rloc(r)=l, adjacent(l,m)
    effects: rloc(r)←m
  - **load(c,r,l)**
    ;; robot r loads container c at location l
    precond: rloc(r)=l, cpos(c)=l, rload(r)=nil
    effects: rload(r)←c, cpos(c)←r
  - **unload(c,r,l)**
    ;; robot r unloads container c at location l
    precond: rloc(r)=l, rload(r)=c
    effects: rload(r)←nil, cpos(c)←l

What is time?

The core mathematical structure for describing time is a **set with transitive and asymmetric ordering** relation.

The set can be continuous (real numbers) or discrete (integer numbers).

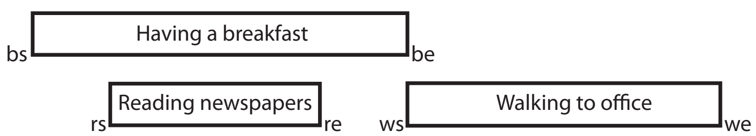
The planning system will use a **database of temporal references** with a procedure for **verifying consistency** and an **inference mechanism** (to deduce new information).



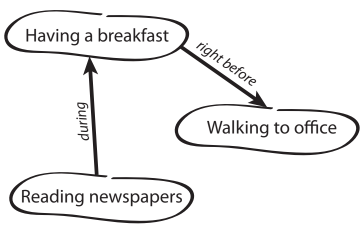
We can model time in two ways:

- **qualitative**
  - relative relations (A finished before B)
- **quantitative**
  - metric (numerical) relations (A started 23 minutes after B)

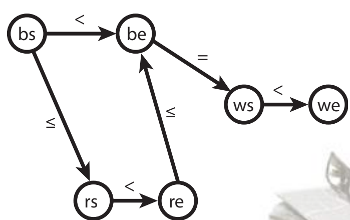
- Based on **relative temporal relations** between temporal references.
- *"I read newspapers during breakfast and after breakfast I walked to my office"*



Temporal intervals (activities)



Time points (important events)



When **modeling time** we are interested in:

- **temporal references**
  (when something happened or hold)
  - **time points** (instants) when a state is changed
    instant is a variable over the real numbers
  - **time periods** (intervals) when some proposition is true
    interval is a pair of variables (x,y) over the real numbers, such that $x<y$
- **temporal relations** between temporal references
  - **ordering** of temporal references

Typical problems solved:

- verifying **consistency** of the temporal database
- asking **queries** (*"Did I read newspapers when entering the office?"*)
- finding **minimal networks** to deduce inevitable relations

Symbolic calculus modelling qualitative relations between instants.

- There are three possible **primitive relations** between instants t_1 and t_2:
  - $[t_1 < t_2]$
  - $[t_1 > t_2]$
  - $[t_1 = t_2]$

  Relations P = {<,=,>} are called **primitive relations**.
- Partially known relation between two instants can be modelled using a set (disjunction) of primitive relations:
  - {}, {<}, {=}, {>}, {<,=}, {>,=}, {<,>}, {<,=,>}
- **Relation** r between temporal instants t and t‘ is denoted **[t r t‘]**
- Point algebra allows us to **work with relative relations** without placing the instants to particular (numeric) times.

Point algebra - operations

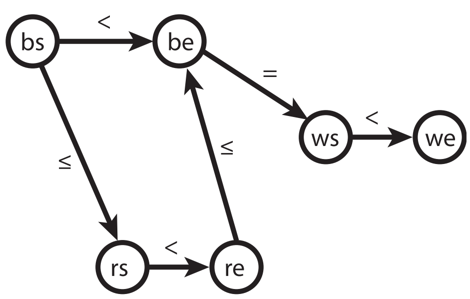
- Let R be a set of all possible relations between two instants
  - {{}, {<}, {=}, {>}, {<,=}, {>,=}, {<,>}, {<,=,>}}
- Symbolic operations over R:
  - **set operations ∩, ∪**
    - they express conjunction and disjunction of relations
  - **composition operation •**
    - transitive relation for a pair of connected relations
    - $[t_1\ r\ t_2]$ and $[t_2\ q\ t_3]$ gives $[t_1\ r{\bullet}q\ t_3]$ using the table

•	<	=	>
<	<	<	P
=	<	=	>
>	P	>	>

- The most widely used operations are ∩ and •, that allow combining existing and inferred relations:
  - $[t_1\ r\ t_2]$ and $[t_1\ q\ t_3]$ and $[t_3\ s\ t_2]$ gives $[t_1\ r{\cap}(q{\bullet}s)\ t_2]$

Point algebra – inference

"I read newspapers during breakfast and after breakfast I walked to my office"



- Query: *"Did I read newspapers when entering the office?"*
- [rs < we] ∧ [we < re]

$(r_{re,be} \bullet r_{be,ws} \bullet r_{ws,we}) \cap (r_{re,we})$

$= (\{=,<\} \bullet \{=\} \bullet \{<\}) \cap \{>\}$

$= \{<\} \cap \{>\} = \{\}$

•	<	=	>
<	<	<	P
=	<	=	>
>	P	>	>

Point algebra - consistency

- A set of instants X together with the set of (binary) temporal relations $r_{i,j} \in R$ over these instants C forms a **PA network** (X,C).
  - If some relation is not explicitly assumed in C then we assume universal relation P.
- The **PA network** consisting of instants and relations between them is **consistent** if it is possible to assign a real number to each instant in such a way that all the relations between instants are satisfied.

Claim:

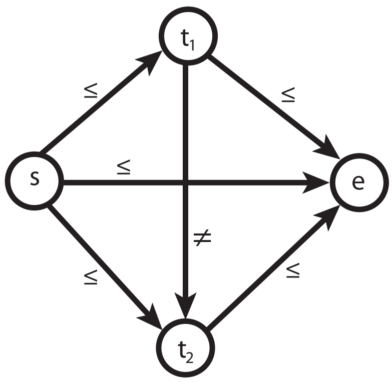
The PA network (X,C) is consistent if and only if there exists a set of primitive relations $p_{i,j} \in r_{i,j}$ such that for any triple of such relations $p_{i,j} \in p_{i,k} \bullet p_{k,j}$ holds.

Efficient consistency checking:

To make the PA network consistent it is enough to make its transitive closure, for example using techniques of **path consistency**.

- for each k: for each i,j: do $r_{i,j} \leftarrow r_{i,j} \cap (r_{i,k} \bullet r_{k,j})$
- obtaining {} means that the network is inconsistent

Point algebra – minimal networks



- PC verifies consistency but does not remove redundant constraints.
- Primitive constraint $p_{i,j}$ is **redundant** if there does not exist any solution where $[t_i \ p_{i,j} \ t_j]$ holds.

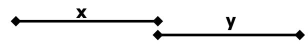
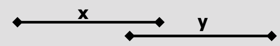
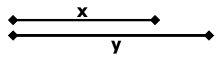
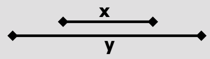
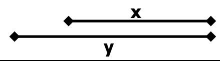
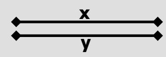
- **PA network is minimal** if it has no primitive constraints that are redundant.
- To make the network minimal we need 4-consistency.

Allen (1983)

Interval algebra

Symbolic calculus modelling relations between intervals
(interval is defined by a pair of instants i^- and i^+, $[i^- < i^+]$)

- There are thirteen primitive relations:

x **b**efore y	$x^+ < y^-$	
x **m**eets y	$x^+ = y^-$	
x **o**verlaps y	$x^- < y^- < x^+ \wedge x^+ < y^+$	
x **s**tarts y	$x^- = y^- \wedge x^+ < y^+$	
x **d**uring y	$y^- < x^- \wedge x^+ < y^+$	
x **f**inishes y	$y^- < x^- \wedge x^+ = y^+$	
x **e**quals y	$x^- = y^- \wedge x^+ = y^+$	
bi,mi,oi,si,di,fi	symmetrical relations	

Interval algebra – consistency

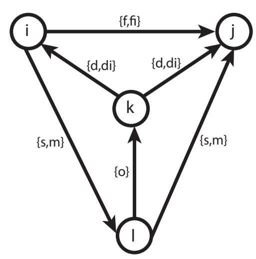
- Primitive relations can be again combined in sets (2^{13} relations).
  - Sometimes we select only a subset of possible relations that are useful for a particular application.
    - for example {b,m,bi,mi} means no-overlaps and it is useful to model unary resources
- set operations ∩, ∪ and the composition operation •
- The **IA network** is **consistent** when it is possible to assign real numbers to x_i^-, x_i^+ of each interval x_i in such a way that all the relations between intervals are satisfied.

Claim:

The IA network (X,C) is consistent if and only if there exists a set of primitive relations $p_{i,j} \in r_{i,j}$ such that for any triple of such relations $p_{i,j} \in p_{i,k} \bullet p_{k,j}$ holds.

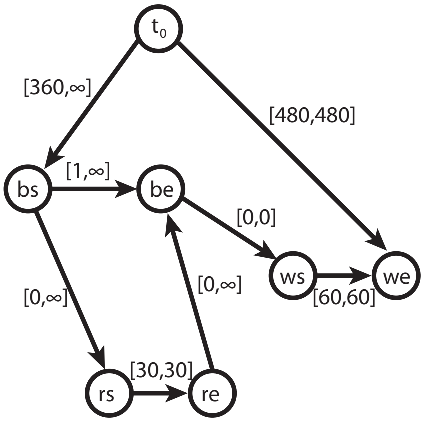
Notes:

- Path consistency is not a complete consistency technique for interval algebra.
- Consistency-checking problem for IA networks is an NP-complete problem.
- Intervals can be converted to instants but some interval relations will not be binary relations among the instants.



"I got up at 6 o'clock. I read newspapers for 30 minutes during the breakfast. After the breakfast I walked to my office which took me one hour. I entered the office at 8:00AM".

When did I start my breakfast?



- 360 =< bs, *"I got up at 6 o'clock"*
- bs =< rs, re =< be, *"I read newspapers during breakfast"*
- re-rs = 30, "I read newspapers for 30 minutes"
- be = ws, *"after breakfast I walked to my office"*
- we-ws = 60, *"[walking] took me one hour"*
- we = 480, *"I entered the office at 8:00AM"*

bs =< rs = re-30 =< be-30 = ws-30 = (we-60)-30 = 390

I started my breakfast between 6:00AM and 6:30AM.

- The basic temporal primitives are again **time points**, but now the relations are numerical.
- Simple **temporal constraints** for instants t_i and t_j:
  - unary: $a_i \leq t_i \leq b_i$
  - binary: $a_{ij} \leq t_i - t_j \leq b_{ij}$,

  where a_i, b_i, a_{ij}, b_{ij} are (real) constants

Notes:

- Unary relation can be converted to a binary one, if we use some fix origin reference point t_0.
- $[a_{ij},b_{ij}]$ denotes a constraint between instants t_i a t_j.
- It is possible to use disjunction of simple temporal constraints.

Simple Temporal Network (STN)

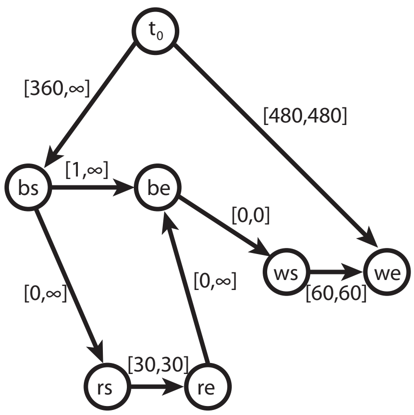
- only simple temporal constraints $r_{ij} = [a_{ij}, b_{ij}]$ are used
- **operations**:
  - composition: $r_{ij} \bullet r_{jk} = [a_{ij}+a_{jk}, b_{ij}+b_{jk}]$
  - intersection: $r_{ij} \cap r'_{ij} = [\max\{a_{ij}, a'_{ij}\}, \min\{b_{ij}, b'_{ij}\}]$
- **STN** is **consistent** if there is an assignment of values to instants satisfying all the temporal constraints.
- **Path consistency** is a complete technique making STN consistent (all inconsistent values are filtered out, one iteration is enough). Another option is using all-pairs minimal distance **Floyd-Warshall algorithm**.

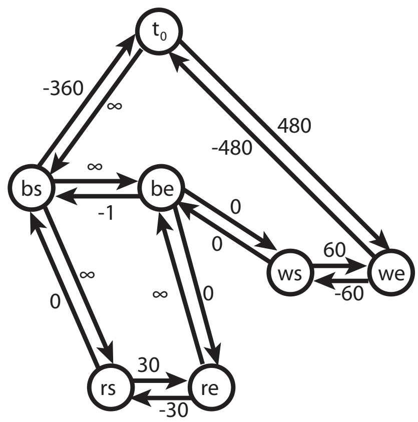
Relations $a_{ij} \leq t_i - t_j \leq b_{ij}$ can be expressed as maximal distances between the time points:

- $t_i - t_j \leq b_{ij}$
- $t_j - t_i \leq -a_{ij}$

This gives a **distance graph**.

- Negative cycle in the distance graph means inconsistency.





Algorithms

- **Path consistency**
  - finds a transitive closure of binary relations r
  - one iteration is enough for STN (in general, it is iterated until any domain changes)
  - works incrementally

one iteration for STN

```
PC(X, C)
    for each k : 1 ≤ k ≤ n do
        for each pair i,j : 1 ≤ i < j ≤ n,, i ≠ k, j ≠ k do
            r_ij ← r_ij ∩ [r_ik • r_kj]
            if r_ij = ∅ then exit(inconsistent)
end
```

general

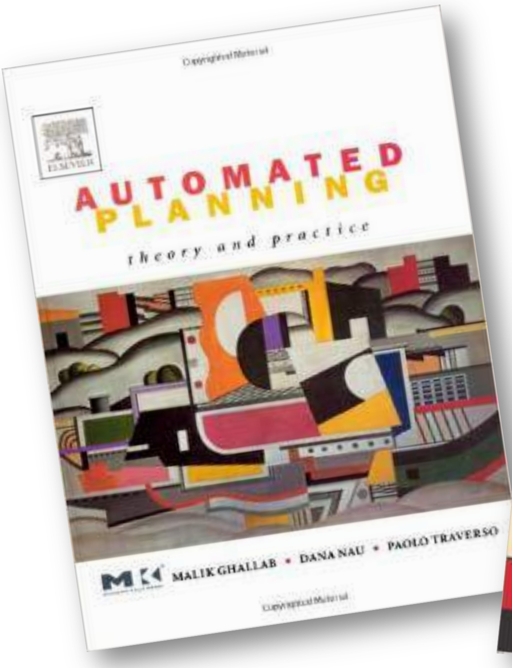
```
PC(C)
    until stabilization of all constraints in C do
        for each k : 1 ≤ k ≤ n do
            for each pair i,j : 1 ≤ i < j ≤ n, i ≠ k, j ≠ k do
                c_ij ← c_ij ∩ [c_ik • c_kj]
                if c_ij = ∅ then exit(inconsistent)
end
```

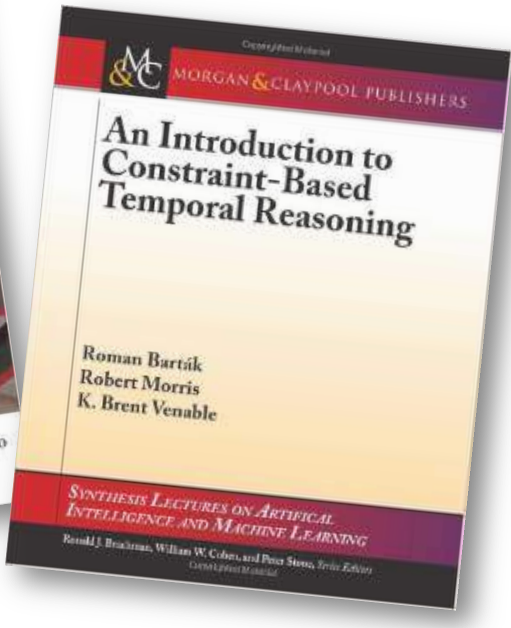
- **Floyd-Warshall algorithm**
  - finds minimal distances between all pairs of nodes
  - First, the temporal network is converted into a **distance graph**
    - there is an arc from i to j with distance b_{ij}
    - there is an arc from j to i with distance $-a_{ij}$.
  - STN is consistent iff there are no negative cycles in the graph, that is, d(i,i)≥0

```
Floyd-Warshall(X, E)
    for each i and j in X do
        if (i,j) ∈ E then d(i,j) ← l_ij else d(i,j) ← ∞
        d(i,i) ← 0
    for each i,j,k in X do
        d(i,j) ← min{d(i,j), d(i,k) + d(k,j)}
end
```

Summary of temporal models

	name	approach	temporal reference	temporal propositions	complexity
PA	point algebra	qualitative	time points	{<,=,>}	tractable
IA	interval algebra	qualitative	intervals	{b,m,o,s,d,f,e,bi, mi,oi,si,di,fi}	NP-c
QA	qualitative algebra	qualitative	time points, intervals	IA, PA, interval-to-point	NP-c
STP	simple temporal problem	quantitative	time points	binary difference	tractable
TCSP	temporal CSP	quantitative	time points	binary disjunctive difference	NP-c
DTP	disjunctive temporal problem	quantitative	time points	n-ary disjunctive difference	NP-c
TNA	temporal network with alternatives	quantitative	time points	precedence, logical	NP-c
	general temporal CSP	qualitative, quantitative	time points, intervals	TCSP, QA	NP-c





Charles University in Prague, Faculty of Mathematics and Physics
bartak@ktiml.mff.cuni.cz

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Robot world representation and reasoning in it

3

Roman Barták
Charles University in Prague, Faculty of Mathematics and Physics

Automated Planning

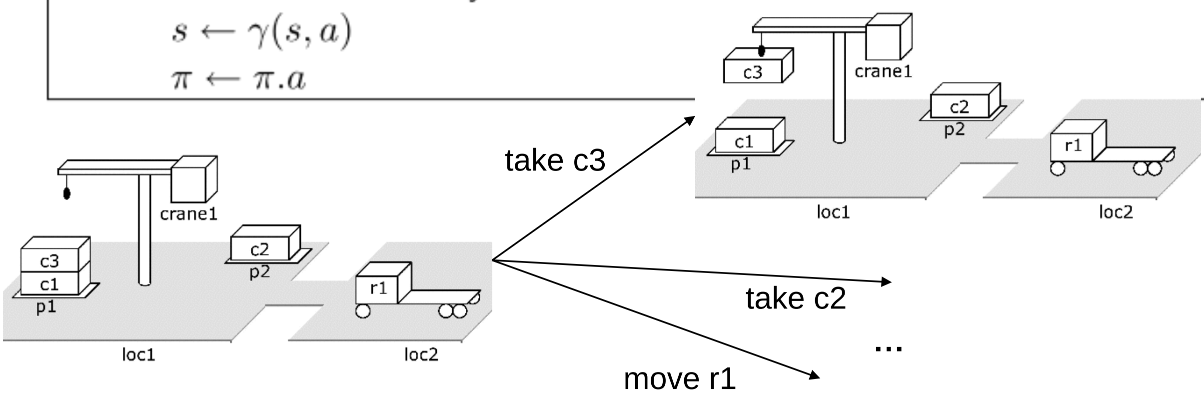
Introduction

Today we will explore techniques for **action planning** – how to find a sequence of actions to reach a given goal.

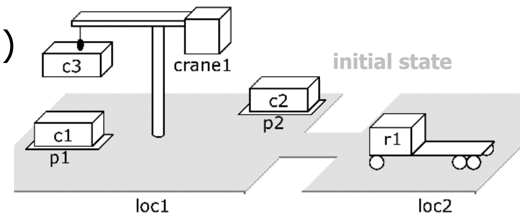
- **problem representation**
  - situation calculus (pure logical representation)
  - using sets of predicates (instead of formulas)
  - planning domain vs. planning problem
- **planning techniques**
  - state-space planning
    - forward and backward
  - plan-space planning
    - partially ordered plans
  - hierarchical planning

- **The search space corresponds to the state space of the planning problem.**
  - search nodes correspond to world states
  - arcs correspond to state transitions by means of actions
  - the task is to find a path from the initial state to some goal state
- **Basic approaches**
  - forward search (progression)
    - start in the initial state and apply actions until reaching a goal state
  - backward search (regression)
    - start with the goal and apply actions in the reverse order until a subgoal satisfying the initial state is reached
    - lifting (actions are only partially instantiated)

Forward-search(O, s_0, g)
  $s \leftarrow s_0$
  $\pi \leftarrow$ the empty plan
  loop
    if s satisfies g then return π
    $E \leftarrow \{a | a$ is a ground instance of an operator in *O*,
        and $\mathrm{precond}(a)$ is true in $s\}$
    if $E = \emptyset$ then return failure
    nondeterministically choose an action $a \in E$
    $s \leftarrow \gamma(s, a)$
    $\pi \leftarrow \pi . a$



Forward planning: an example



{belong(crane1,loc1), adjacent(loc2,loc1)
holding(crane1,c3), unloaded(r1),
at(r1,loc2), ¬occupied(loc1),
occupied(loc2),…}

move(r1,loc2,loc1)

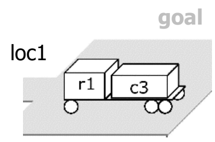
move(r,l,m)
;; robot r moves from location l to location m
precond: adjacent(l,m), at(r,l), ¬occupied(m)
effects: at(r,m), occupied(m), ¬occupied(l), ¬at(r,l)

{belong(crane1,loc1),
adjacent(loc2,loc1), holding(crane1,c3), unloaded(r1),
at(r1,loc1), occupied(loc1), …}

load(crane1,loc1,c3,r1)

load(k,l,c,r)
;; crane k at location l loads container c onto robot r
precond: belong(k,l), holding(k,c), at(r,l), unloaded(r)
effects: empty(k), ¬holding(k,c), loaded(r,c), ¬unloaded(r)

{belong(crane1,loc1), adjacent(loc2,loc1),
empty(crane1), loaded(r1,c3),
at(r1,loc1), occupied(loc1), …}



Goal = {at(r1,loc1),loaded(r1,c3)}

Backward planning

Start with a goal (not a goal state as there might be more goal states) and through sub-goals try to reach the initial state.

Action a is relevant for a goal g if and only if:

- action **a** contributes to goal **g**: **g** ∩ effects(**a**) ≠ ∅
- effects of action **a** are not conflicting goal **g**:
  - $g^- \cap \text{effects}^+(a) = \emptyset$
  - $g^+ \cap \text{effects}^-(a) = \emptyset$

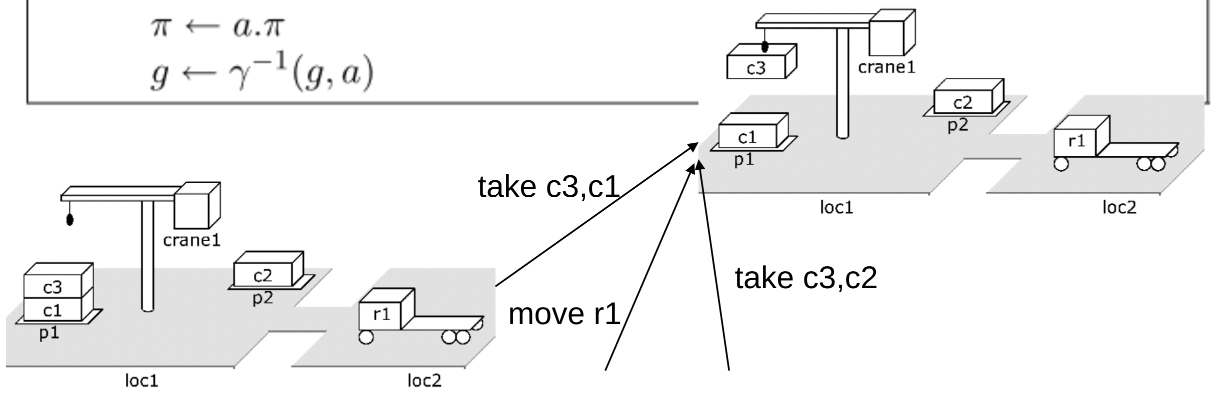
A **regression set** of the goal **g** for (relevant) action **a** is
$\gamma^{-1}(g,a) = (g - \text{effects}(a)) \cup \text{precond}(a)$

Example:
goal: **{on(a,b), on(b,c)}**
action **stack(a,b)** is relevant
by backward application of the action we get a new goal:
{holding(a), clear(b), on(b,c)}

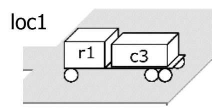
stack(*x*,*y*)
Precond: holding(*x*), clear(*y*)
Effects: ~holding(*x*), ~clear(*y*), on(*x*,*y*), clear(*x*), handempty

Backward planning: algorithm

Backward-search(O, s_0, g)
  $\pi \leftarrow$ the empty plan
  loop
    if s_0 satisfies g then return π
    $A \leftarrow \{a | a$ is a ground instance of an operator in O
        and $\gamma^{-1}(g, a)$ is defined$\}$
    if $A = \emptyset$ then return failure
    nondeterministically choose an action $a \in A$
    $\pi \leftarrow a.\pi$
    $g \leftarrow \gamma^{-1}(g, a)$



Backward planning: an example



Goal = {at(r1,loc1),loaded(r1,c3)}

load(crane1,loc1,c3,r1)

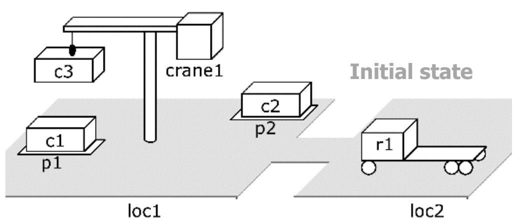
load(k, l, c, r)
;; crane k at location l loads container c onto robot r
precond: belong(k, l), holding(k, c), at(r, l), unloaded(r)
effects: empty(k), $\neg$ holding(k, c), loaded(r, c), $\neg$ unloaded(r)

{at(r1,loc1), belong(crane1,loc1),
holding(crane1,c3), unloaded(r1)}

move(r1,loc2,loc1)

move(r, l, m)
;; robot r moves from location l to location m
precond: adjacent(l, m), at(r, l), $\neg$ occupied(m)
effects: at(r, m), occupied(m), $\neg$ occupied(l), $\neg$ at(r, l)

{belong(crane1,loc1), holding(crane1,c3),
unloaded(r1),
adjacent(loc2,loc1),
at(r1,loc2),
¬ occupied(loc1)}



Lifted-backward-search(O, s_0, g)
  $\pi \leftarrow$ the empty plan
  loop
    if s_0 satisfies g then return π
    $A \leftarrow \{(o, \theta) | o$ is a standardization of an operator in O,
      θ is an mgu for an atom of g and an atom of effects (o),
      and $\gamma^{-1}(\theta(g), \theta(o))$ is defined$\}$
    if $A = \emptyset$ then return failure
    nondeterministically choose a pair $(o, \theta) \in A$
    $\pi \leftarrow$ the concatenation of $\theta(o)$ and $\theta(\pi)$
    $g \leftarrow \gamma^{-1}(\theta(g), \theta(o))$

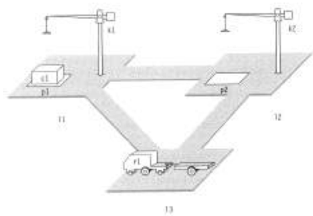
Notes:

- standardization = a copy with fresh variables
- mgu = most general unifier
- by using the variables we can decrease the branching factor but the trade off is more complicated loop check

- The principle of plan space planning is similar to backward planning:
  - start from an **„empty" plan** containing just the description of initial state and goal
  - **add other actions** to satisfy not yet covered (open) goals
  - if necessary **add other relations** between actions in the plan
- Planning is realised as **repairing flaws in a partial plan**
  - go from one partial plan to another partial plan until a complete plan is found

Plan space planning: an example

- Assume a partial plan with the following two actions:
  - take(k1,c1,p1,l1)
  - load(k1,c1,r1,l1)



- **Possible modifications of the plan:**
  - **adding a new action**
    - to apply action **load**, robot r1 must be at location l1
    - action **move**(r1,l,l1) moves robot r1 to location l1 from some location l
  - **binding the variables**
    - action **move** is used for the right robot and the right location
  - **ordering some actions**
    - the robot must **move** to the location before the action **load** can be used
    - the order with respect to action **take** is not relevant
  - **adding a causal relation**
    - new action is added to move the robot to a given location that is a precondition of another action
    - the causal relation between **move** and **load** ensures that no other action between them moves the robot to another location

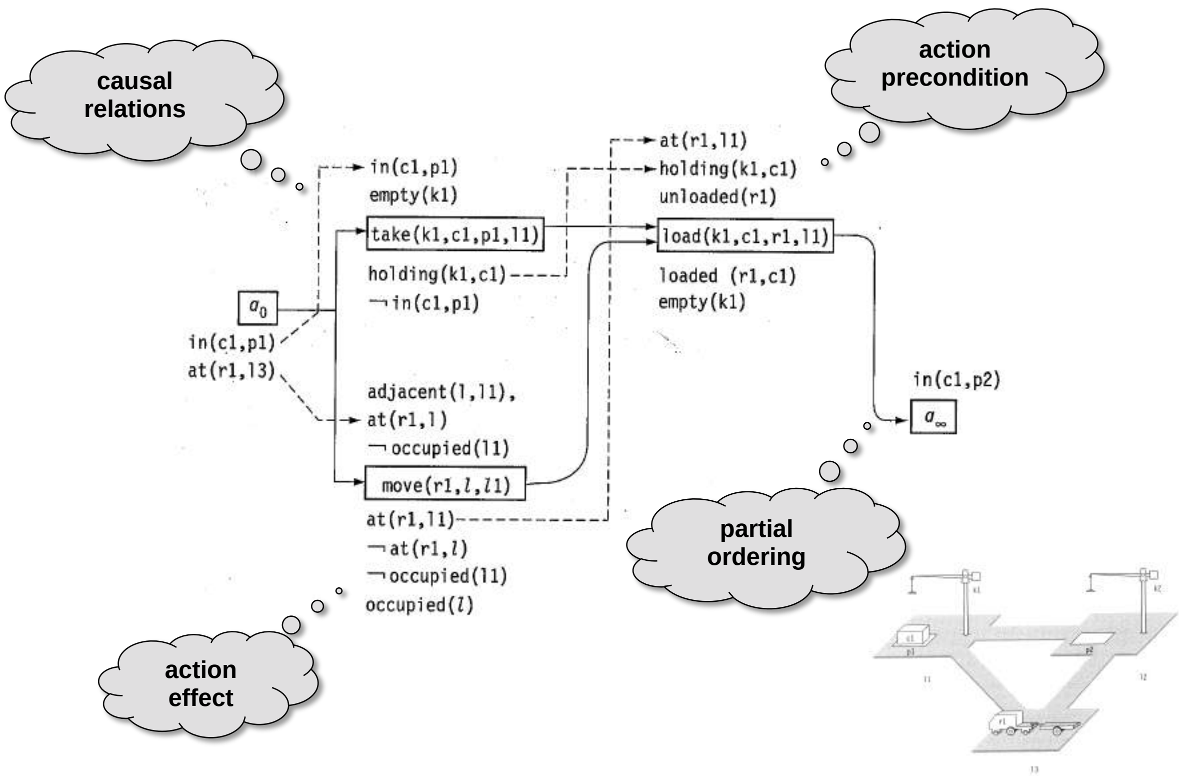
Plan space planning: the initial plan

- **The initial state and the goal** are encoded using two **special actions** in the initial partial plan:
  - **Action a_0 represents the initial state** in such a way that atoms from the initial state define effects of the action and there are no preconditions. This action will be before all other actions in the partial plan.
  - **Action a_∞ represents the goal** in a similar way – atoms from the goal define the precondition of that action and there is no effect. This action will be after all other actions.
- **Planning** is realised by **repairing flaws** in the partial plan.

The search nodes correspond to partial plans.

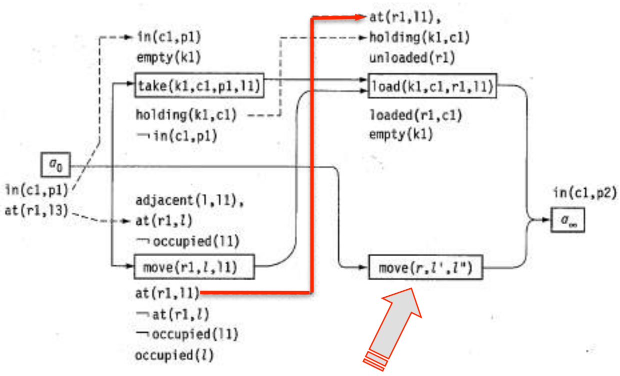
A partial plan Π is a tuple (A,<,B,L), where

- A is a set of partially instantiated planning operators $\{a_1,\ldots,a_k\}$
- < is a partial order on A ($a_i<a_j$)
- B is set of constraints in the form x=y, x≠y or $x \in D_i$
- L is a set of causal relations ($a_i \rightarrow^p a_j$)
    - a_i,a_j are ordered actions $a_i<a_j$
    - p is a literal that is effect of a_i and precondition of a_j
    - B contains relations that bind the corresponding variables in p



- **Open goal** is an example of a **flaw**.
- This is a precondition **p** of some operator **b** in the partial plan such that no action was decided to satisfy this precondition (there is no causal relation $a_i \rightarrow^p b$).
- **The open goal p of action b can be resolved by:**
  - finding an operator **a** (either present in the partial plan or a new one) that can give **p** (**p** is among the effects of **a** and **a** can be before **b**)
  - binding the variables from **p**
  - adding a causal relation $\mathbf{a} \rightarrow^p \mathbf{b}$

- **Threat** is another example of **flaw**.
- This is action that can influence existing causal relation.
  - Let $a_i \rightarrow^p a_j$ be a causal relation and action **b** has among its effects a literal unifiable with the negation of **p** and action **b** can be between actions a_i and a_j. Then **b** is threat for that causal relation.
- We can **remove the threat** by one of the ways**:**
  - ordering **b** before $\mathbf{a_i}$
  - ordering **b** after $\mathbf{a_j}$
  - binding variables in **b** in such a way that **p** does not bind with the negation of **p**



Solution plan

- Partial plan Π = (A,<,B,L) is a **solution plan** for the problem P = (Σ,s_0,g) if:
  - partial ordering < and constraints B are globally consistent
    - there are no cycles in the partial ordering
    - we can assign variables in such a way that constraints from B hold
  - Any linearly ordered sequence of fully instantiated actions from A satisfying < and B goes from s_0 to a state satisfying g.
- Hmm, but this definition **does not say how** to verify that a partial plan is a solution plan!

Claim: Partial plan Π = (A,<,B,L) is a solution plan if:

- there are no flaws (no open goals and no threats)
- partial ordering < and constraints B are globally consistent

Plan-space planning: algorithm

- **PSP = Plan-Space Planning**

```
PSP(π)
    flaws ← OpenGoals(π) ∪ Threats(π)
    if flaws = ∅ then return(π)
    select any flaw φ ∈ flaws
    resolvers ← Resolve(φ, π)
    if resolvers = ∅ then return(failure)
    nondeterministically choose a resolver ρ ∈ resolvers
    π' ← Refine(ρ, π)
    return(PSP(π'))
end
```

Notes:

- The selection of flaw is deterministic (all flaws must be resolved).
- The resolvent is selected non-deterministically (search in case of failure).

Hierarchical Task Network Planning

Classical planning assumes primitive actions connected via causal relations.

In real-life we can frequently use “**recipes**” to solve a particular task.

- recipe is a set of operations to achieve a sub-goal

HTN planning is based on performing a set of tasks (instead of achieving goals).

- **primitive task**: performed by a classical planning operator
- **non-primitive task**: decomposed by a **method** to other tasks (can use recursion)

Task networks

How to describe a recipe to perform a given task?

- specify sub-tasks and their relations

A **task network** is a pair (U,C), where U is a set of tasks and C is a set of constraints.

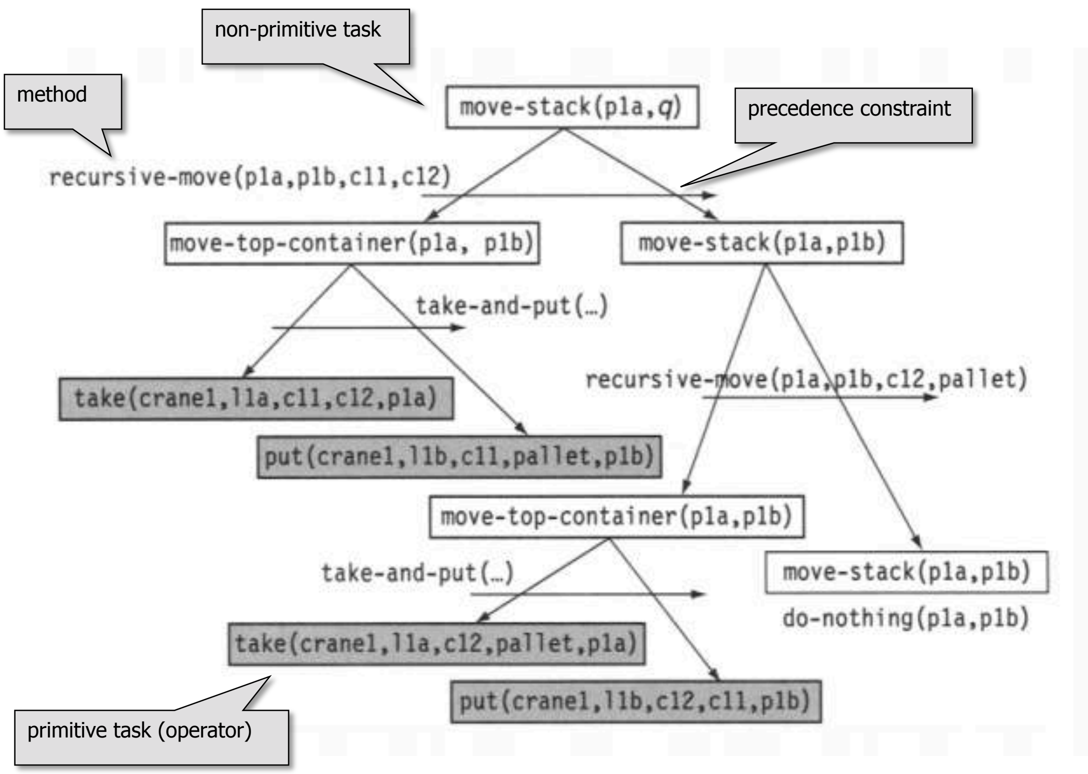
- **tasks** are named similarly to operators: $t(r_1,…,r_n)$
- constraints are in the form:
  - **precedence constraint**: u < v (task u is performed before task v)
  - **before-constraint**: before(U’,l) (literal l is true right before the set of tasks U’)
  - **after-constraint**: after(U’,l) (literal l is true right after the set of tasks U’)
  - **between-constraint**: between(U’,U’’,l) (literal l must be true right after U’, right before U’’ and in all states in between)

To perform non-primitive tasks, we need to decompose them to other tasks using a method.

An **HTN method** is a tuple
m = (*name, task, subtasks, constr*)

– *name* is $n(x_1,…,x_n)$, where $\{x_1,…,x_n\}$ are all variables in *m* and n is a unique name of the method,
– *task* is a non-primitive task,
– (*subtasks*, *constr*) is a task network.

There may be more methods for a single non-primitive task.



HTN Problem

Now, the planning problem is specified somehow differently from classical planning as a process to obtain a plan from decomposition of tasks in a given task network.

An **HTN planning domain** is a pair (O,M)

- O is a set of operators
- M is a set of HTN methods

An **HTN planning problem** is a 4-tuple (s_0,w,O,M)

- s_0 is the initial state
- w is the initial task network
- (O,M) is the HTN planning domain

Solution plan

When is a plan π a **solution for problem** P?

- If w = (U,C) is **primitive** then $\pi = <a_1,\ldots,a_k>$ is a solution for P, if (U',C') is a ground instance of (U,C) with total ordering $<u_1,\ldots,u_k>$ of nodes in U':
  - the names of tasks $<u_1,\ldots,u_k>$ are actions $<a_1,\ldots,a_k>$
  - the plan π is executable in the state s_0
  - all constraints C' are satisfied by $<a_1,\ldots,a_k>$
- If w = (U,C) is **non-primitive** then π is a solution for P if there is a sequence of task decompositions applied to w and giving a primitive task network w' (all tasks are primitive) that is a solution for P.

HTN Planning

```
Abstract-HTN(s, U, C, O, M)
    if (U, C) can be shown to have no solution
        then return failure
    else if U is primitive then
        if (U, C) has a solution then
            nondeterministically let π be any such solution
            return π
        else return failure
    else
        choose a nonprimitive task node u ∈ U
        active ← {m ∈ M | task(m) is unifiable with t_u}
        if active ≠ ∅ then
            nondeterministically choose any m ∈ active
            σ ← an mgu for m and t_u that renames all variables of m
            (U′, C′) ← δ(σ(U, C), σ(u), σ(m))
            (U′, C′) ← apply-critic(U′, C′) ;; this line is optional
            return Abstract-HTN(s, U′, C′, O, M)
        else return failure
```

decomposition of a task

performing application-specific computations

Modeling languages

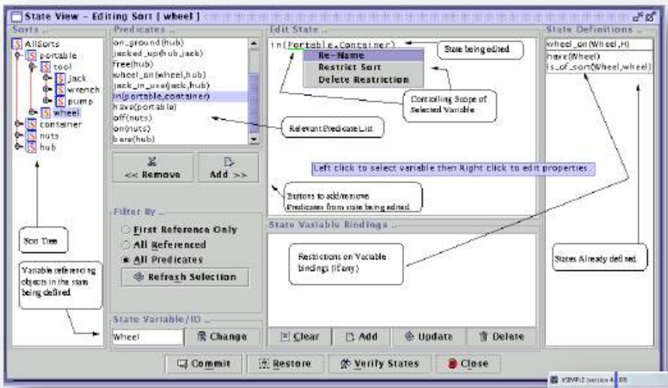
- **PDDL** (Planning Domain Definition Language)
  - mainly for classical planning with many extensions
  - de-facto standard used in planning competitions

```
(:action load-cargo
   :parameters (?s - Ship ?c - Cargo ?loc – Location)
   :precondition (and
      (at ?s ?loc)
      (cargo-at ?c ?loc)
      (>= (free-cargo-cap ?s) (cargo-weight ?c))
      (isDocked ?s ?loc))
   :effect (and
      (not (cargo-at ?c ?loc))
      (cargo-at ?c ?s)
      (decrease (free-cargo-cap ?s) (cargo-weight ?c))))
```

- **ANML** (Action Notation Modeling Language)
  - rich temporal constraints, state variables, numeric fluents, functions, and HTN methods

…

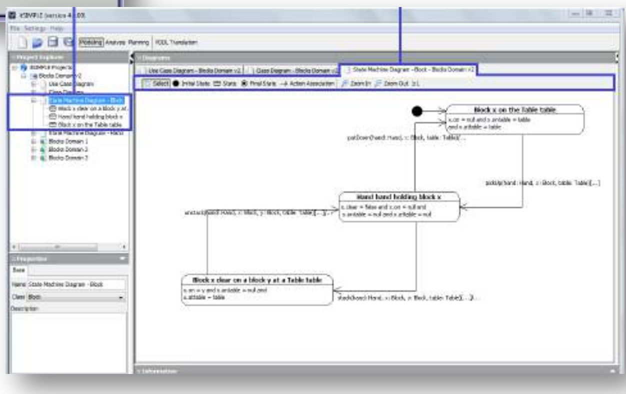
Modeling environments



GIPO

Graphical Interface for Planning with Objects:

- PDDL export/import
- hyHtn + external planners
- debugger
- operator induction



itSIMPLE

Integrated Tools Software Interface for Modeling Planning Environments:

- UML, XML, Petri Nets and PDDL
- support for planners
- analysis

Planners

- **Fast Forward** (FF) (heuristic search)
- **Fast Downward** (heuristic search)
- **SGPlan** (Subgoal Partitioning and Resolution in Planning)
- **CPT** (partial order temporal planner)
- **SHOP2** (hierarchical task networks)
- **EUROPA** (timeline-based planning)

…

`http://ipc.icaps-conference.org`

Flexible Acting and Planning Environment

- interleaving planning and acting
- map actions into low-level commands
- executing commands

Filuta 2 planner

- partial order planner
- simple temporal constraints
- resource constraints
- hierarchical planning
- support for ANML language

Autonomous Robots and Agents

A Special Track at the 28-th International FLAIRS Conference

`http://ktiml.mff.cuni.cz/~bartak/FLAIRS2015/`

Hollywood, Florida, USA
May 18-20, 2015

Important dates:

- Paper submission deadline: 17th November 2014
- Notification of paper decisions: 19th January 2015

Charles University in Prague, Faculty of Mathematics and Physics
bartak@ktiml.mff.cuni.cz

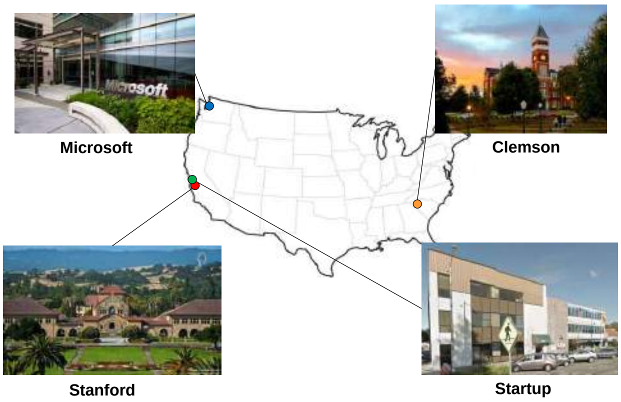
Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Perception-Action Loop in Practice

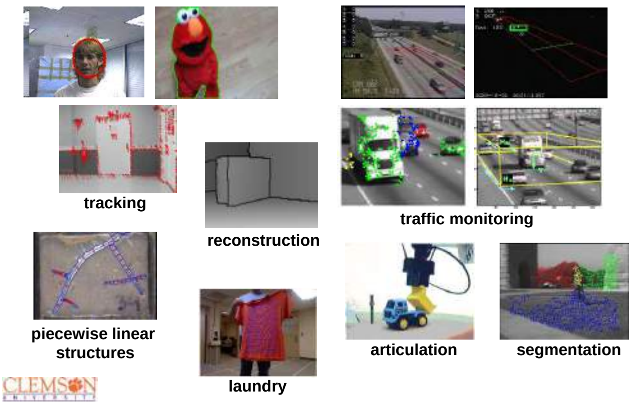
Stan Birchfield
Microsoft Robotics
Clemson University

Joint work with:
Z. Chen, Y. Li, V. Murali, B. Peasley, B. Willimon, A. Cunningham, F. Dellaert, I. Nourbakhsh, R. Powers, H. Dang, A. Feniello, M. Jalobeanu, H. Kikkeri, G. Parent, M. Revow, G. Shirakyan, B. Thibodeau C. Sprunk, J. Röwekämper, L. Spinello, G. D. Tipaldi, W. Burgard

My background



Computer vision research



Vision *in* Robotics

A robot must perceive its surroundings in order to interact

Object
- Pose
- Type
- Identity
- Affordance

Obstacles
- Location
- Velocity
- Type
- Affordance

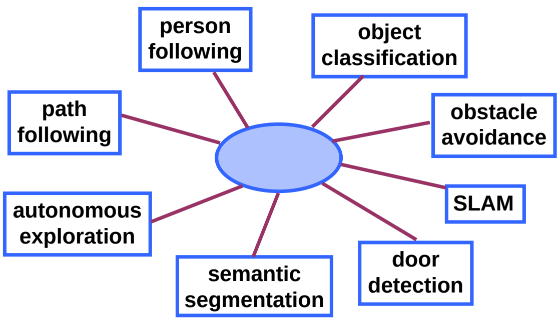
Target
- Location (for servoing)

Closed-loop robot can adapt to
- Uncertainties in system
- Uncertainties in environment

People
- Location
- Velocity
- Gestures

Advantages of vision:
- Non-contact (passive) measurement
- Low cost
- Low power
- Rich capturing ability

Vision *for* Robotics

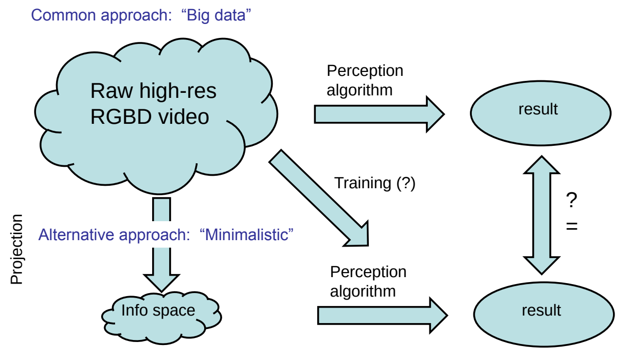


Modular capabilities

The case for simplicity

- **Occam's razor: "The simplest explanation is the best"**
- **Einstein's razor: "Everything should be made as simple as possible, but no simpler"**
- **KISS principle: "Keep it sensibly simple"**

Simplicity applied to perception



[O'Kane and LaValle, *TROB* 2007]

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution [AIMag 1995]
  - Indoor mapping [IROS 2012]
  - Door detection [RAS 2011]
- Behavior-Based Robotics
  - Person and path following [ICRA 2006, IROS 2007]
  - Low-resolution navigation [AR 2012]
  - Obstacle avoidance [WoRV 2013]
- Interactive Perception
  - Handling highly flexible objects [IROS 2010, IROS 2011, IROS 2012, ICRA 2013]
  - Classification of laundry [ICRA 2011]
  - Articulated reconstruction [RAS 2014]
- Hybrid Approach
  - Complete indoor navigation system [ICRA 2014, ISER 2014]
  - Graph optimization for loop closure [ICRA 2014]
  - Learning-by-demonstration [IROS 2014]
- Conclusion

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

SRI's Shakey (1960s)

- High-level STRIPS planner
  - Predicts using action model
  - Translates English sentence into predicate calculus
    ("Use Box1 to block Door4 from Room2")
- Intermediate-level planner
  - Primitive actions (GOTO, GOTHRU, PUSH)
  - Augments with learned complex actions
  - Recover from errors – replan at either level
- Low-level control
  - Computer vision to correct odometry
  - World knowledge may be incomplete, inaccurate

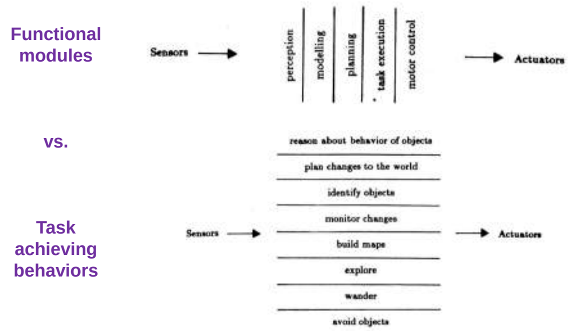
[Nilsson et al. 1969]

Stanford Cart (1970s)

- "Slider stereo" – single camera moved laterally to take 9 images
- Autonomously navigates room, avoiding obstacles
- Traditional "sense-plan-act" paradigm:
  - 15 minutes for image processing and planning
  - Then 1-meter movements

[Moravec 1980]

Subsumption architecture (1980s)



[Brooks 1986] -- 8915 citations

Subsumption architecture (1980s)

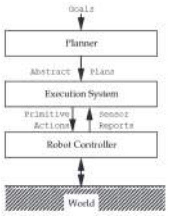
- "Intelligent creatures"
  - interact with world
  - pursue multiple goals
  - no plans for specific mission
- Each layer is a finite state machine
  - connect directly
  - suppression and inhibition
- No world model. The world is its own model.
- Complex behavior emerges from simple program

[Brooks-Flynn 1989, Brooks 1991] -- 6815 citations

Three-layer architecture (1990s)

- Limitations of previous approaches:
  - SPA: Internal state out of sync with reality it represents → plan executed in wrong context
  - Reactive: Sensors are noisy, unreliable, limited view → world model is unobservable
- Three layers:
  - Time-consuming deliberation (future state)
    - Performs compute-intensive operations
    - Generates long-term plans
  - Reactive plan execution (past state)
    - Select which behavior
    - Supply parameters to behavior
  - Reactive feedback control (present – no state)
    - Tightly couples sensors to actuators
    - Library of hand-crafted transfer functions (behaviors)

Higher layers provide advice to lower layers



[Firby 1989]

[Gat 1998]

Working systems (2000s)

- DARPA Grand Challenge 2005
  - ~130 miles (209 km)
  - ~3000 GPS coordinates
  - Nearly 7 hours of driving
  - Max speed 38 mph (61 km/h)
- Requirements: Reliability, precision, speed in unstructured env.
- Architecture:
  - Related to three-layer architecture
  - No centralized master
  - 30 modules running concurrently (Perception, collision avoidance, stable vehicle control, …)
  - All modules communicate via publish/subscribe
  - Special modules monitoring health (auto restart if failure)

[Thrun et al. 2006]

Latest developments (2010s)

- Highly capable mobile manipulators (PR2)
- Powerful software architectures / libraries (ROS, PCL, etc.)
- Advances in perception (RGBD sensors, KinectFusion, person detection, skeletal tracking, …)
- Big data vs. interactive perception

[Katz-Brock 2008]

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Topological navigation

Goal: Navigate indoor office building using topological map

- Standard SPA approach:
  - Sonars indicate door / hallway
  - Plan in topological space
  - Execute in Manhattan world with probabilistic state
- Novel sensor configuration
  - Some sonars point upward
  - No sonars point backward

[Nourbakhsh et al., AI Mag 1995]

Finite state machine

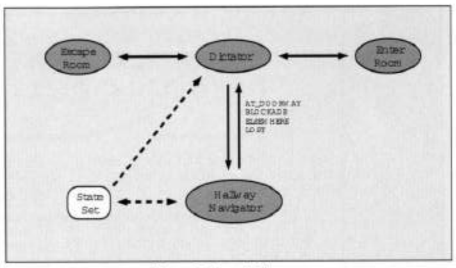


Figure 2. Control Diagram.

Assumptive planning

- Robot maintains probabilistic state (HMM)
- Relies on collapsing of states before replanning

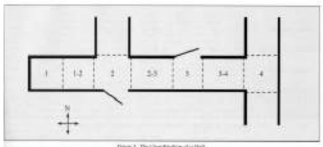


Figure 3. The Quantization of a Hall.

	Wall	Closed Door	Open Door	Open Hallway	Foyer
Nothing detected	.70	.40	.05	.001	.30
Closed door detected	.30	.60	0	0	.05
Open door detected	0	0	.90	.10	.15
Open hallway detected	0	0	.001	.90	.50

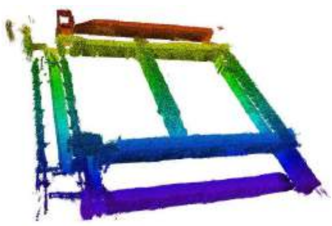
Table 1. Confusion Matrix for the Computer Science Department at Stanford University.

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Building Large-Scale Indoor Maps

Goal: Build *on-line* accurate 3D map of large environment

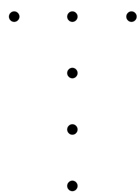


Key challenge: Avoid drift

[Peasley et al., IROS 2012]

Occupancy grids

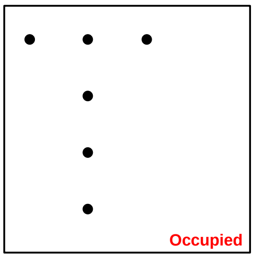
[Moravec and Elfes 1985]



An occupancy grid can represent any point cloud with arbitrary precision

Occupancy grids

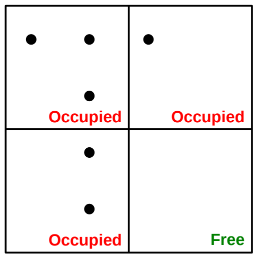
[Moravec and Elfes 1985]



An occupancy grid can represent any point cloud with arbitrary precision

Occupancy grids

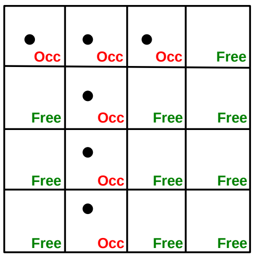
[Moravec and Elfes 1985]



An occupancy grid can represent any point cloud with arbitrary precision

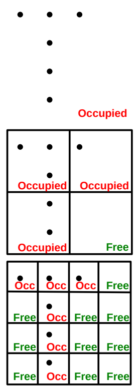
Occupancy grids

[Moravec and Elfes 1985]

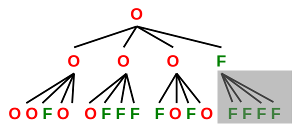


An occupancy grid can represent any point cloud with arbitrary precision

Occupancy grids



Leads naturally to a hierarchical representation: QuadTree (2D)

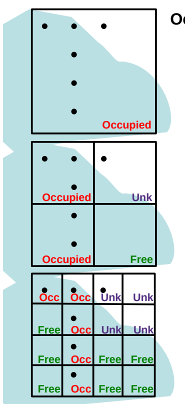


… efficient encoding:

1 1110 1101100001010000

0 = Free
1 = Occupied

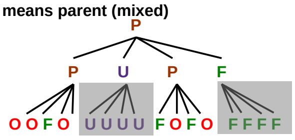
Encoding “I don’t know”



Occupancy grids can explicitly represent unknown areas (which sensor has not yet seen)

Make representation symmetric:

- O means *confidently completely* occupied
- F means *confidently completely* free
- U means *unconfident*
- P means parent (mixed)



… which is more efficient:

11 11001101 1010011001100110

00 = Unknown
01 = Free
10 = Occupied
11 = Parent

Issue of drift

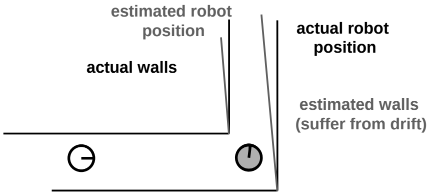
- Occupancy grids
  - discretize space
    → lose information
  - assume fixed coordinate system
    → cannot be corrected
- SLAM (simultaneous localization and mapping)
  - estimate of sensor pose is constantly adjusted (especially during loop closure)

How to *correct* or *avoid* drift?

Manhattan world assumption

[Furukawa et al. 2009]

- **Manhattan world assumes all planes are axis-aligned (parallel or perpendicular):**

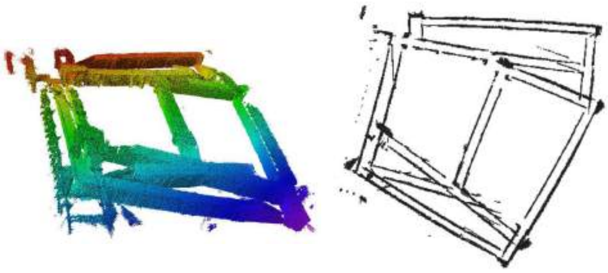


- **Rotational drift leads to severe error:**
  1 degree angle error yields 2 m position error (after 100 m)

  But Manhattan → *zero rotational drift!*

Results

- **Large building (no loop closure)**

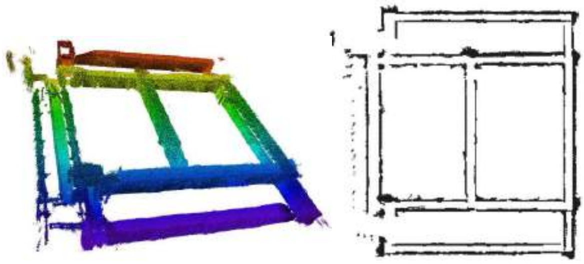


RO+VR

(52.6 x 53.2 m)

Results

- **Large building (no loop closure)**

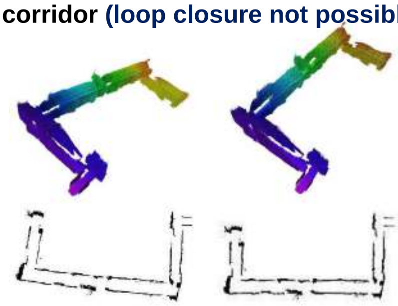


RO+VR+M

(52.6 x 53.2 m)

Results

- **Long corridor (loop closure not possible)**

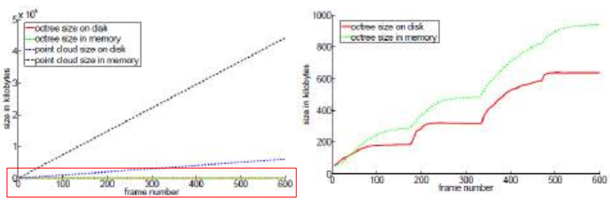


RO+VR **RO+VR+M**

(23.9 x 47.8 m)

Size of OctoMaps

- **Octree reduces storage requirements by 3 orders of magnitude:**



	in memory	on disk
Point Cloud	4,350,000 kB	590,000 kB
Octree	941 kB	363 kB
Compression	4622:1	1625:1

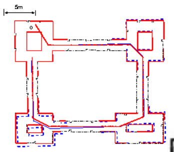
Outline

- **Historical overview**
- **Sense-Plan-Act**
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- **Behavior-Based Robotics**
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- **Interactive Perception**
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- **Hybrid Approach**
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- **Conclusion**

Door Detection

Metric map

Topological map



Doors are *semantically meaningful* landmarks

[Chen and Birchfield, VLMP/CVPR 2008; Chen et al., RAS 2011]

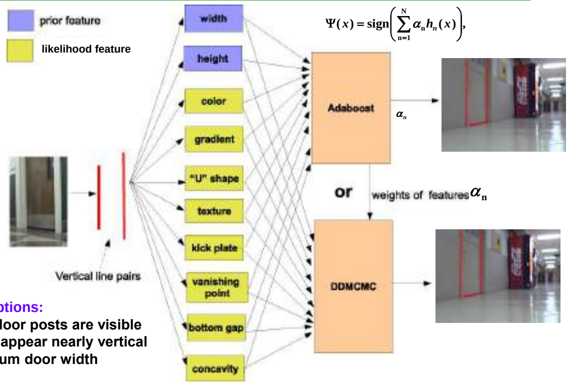
What is Lintel-Occluded?

- post-and-lintel architecture
- camera is low to ground
- cannot point upward b/c obstacles

lintel

post

Our Approach to Door Detection



Assumptions:
- **Both door posts are visible**
- **Posts appear nearly vertical**
- **Minimum door width**

Key idea: Multiple cues are necessary for robustness

Adaboost

- **Bayes decision rule: declare a door if the *a posteriori* probability of the predicate is greater than that of its complement:**

$$\frac{P(D_\ell \mid I)}{P(\neg D_\ell \mid I)} = \exp(\psi(D_\ell \mid I)) > 1,$$

where

$$\psi(D_\ell \mid I) = \sum_{i=1}^{n} \alpha_i h_i(D_\ell \mid I)$$

- **The strong classifier is given by a weighted sum of the weak classifiers:**

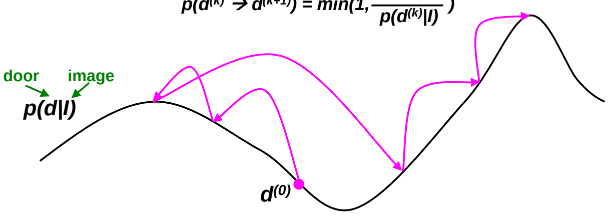
$$\psi(x) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n h_n(x)\right)$$

where weight $\alpha_n = \frac{1}{2}\left(\ln \frac{1-\varepsilon_n}{\varepsilon_n}\right)$, **error** $\varepsilon_n = P_{r_i \sim D_n}[h_n(x_i) \neq y_i] = \sum_{i: h_n(x_i) \neq y_i} D_n(i)$

MCMC

Markov Chain Monte Carlo (MCMC) maximizes probability by random walk through state space of doors

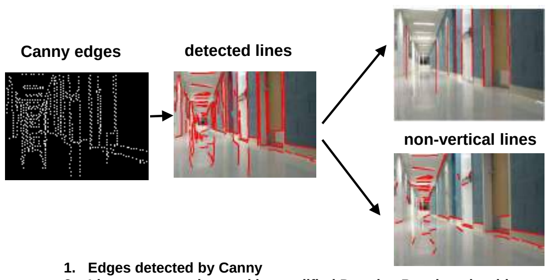
$$p(d^{(k)} \rightarrow d^{(k+1)}) = \min\left(1, \frac{p(d^{(k+1)} \mid I)}{p(d^{(k)} \mid I)}\right)$$



Data-driven MCMC speeds up computation by choosing jump direction based on data

- **doors appear more frequently at the position close to the vertical lines**
- **the top of the door is often occluded or a horizontal line closest to the top**
- **the bottom of the door is often close to the wall/floor boundary.**

Pairs of Vertical Lines



1. **Edges detected by Canny**
2. **Line segments detected by modified Douglas-Peucker algorithm**
3. **Clean up (merge lines across small gaps, discard short lines)**
4. **Separate vertical and non-vertical lines**
5. ***Door candidates* given by all the vertical line pairs whose spacing is within a given range**

Likelihood Features


Image gradient along edges (g_1)


Placement of top and bottom edges (g_2, g_3)


Color (g_4)

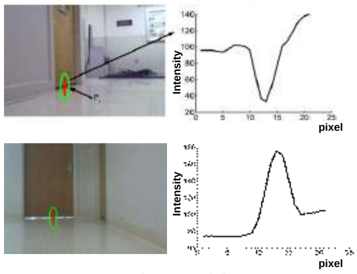

texture (g_5)


Kick plate (g_6)

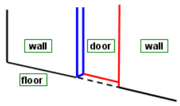
Vanishing point (g_7)

and two more…

Likelihood Features (cont.)

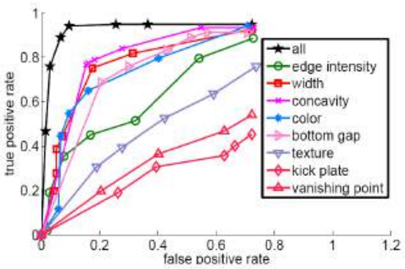


bottom gap (g_8)



concavity (g_9)

Results

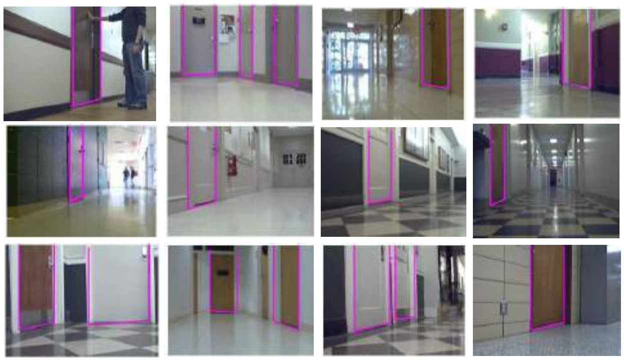


25 different buildings
600 images:
- *100 training*
- *500 testing*

91.1% accuracy with 0.09 FP per image

Speed: 5 fps 1.6GHz (unoptimized)

Results



False Negatives and Positives


strong reflection
distracting reflection

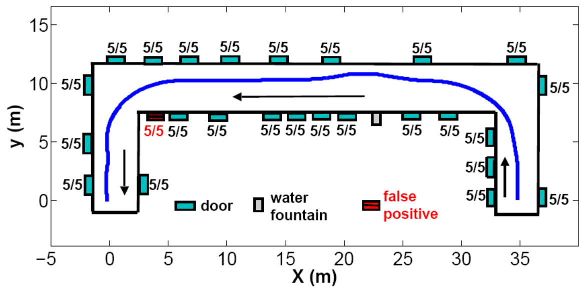

two vertical lines unavailable


concavity and bottom gap tests fail
distracting reflection


concavity erroneously detected

Navigation in a Corridor

- **Doors were detected and tracked from frame to frame (some temporal smoothing)**
- **Repeated 5 times**



Video

Outline

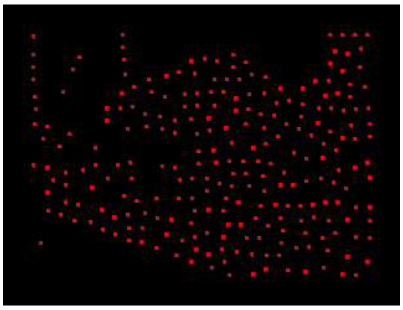
- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Feature Tracking

- Automatic processing
- All information is 2D
- Dramatic reduction in information
- Retains
  - number of objects
  - depth ordering
  - occlusion relationships
  - 3D structure
  - categories of scene, objects, motions
- Real-time processing



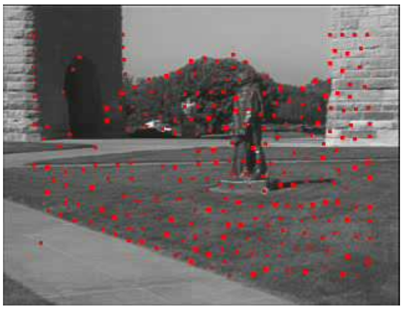
Original RGB image: 640 x 480 x 3 x 30 = 27 MB/sec

Ratio: 500:1 (discard 99.8% of data)

Feature Tracking

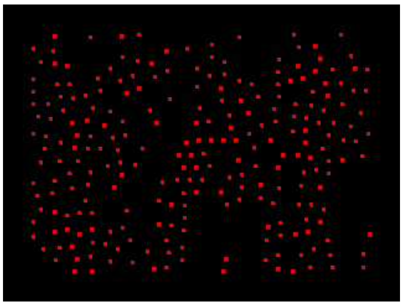
Kanade-Lucas-Tomasi (KLT) algorithm:

- Features are detected in regions with intensity changes in all directions ("corners")
- Features are tracked by searching for locally similar image locations



A Harder Example

- Biological motion gives instant cue to identifying motion

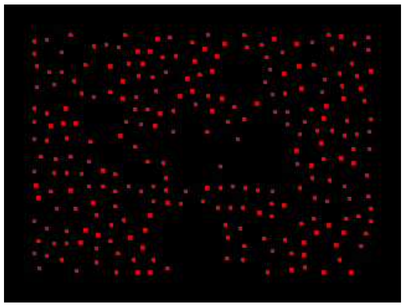


A Harder Example

From the video, we can observe:
- Objects: People
- Activity: Walking
- Number of objects
- Depth ordering

An Easy Example

Classification is instantaneous!



An Easy Example

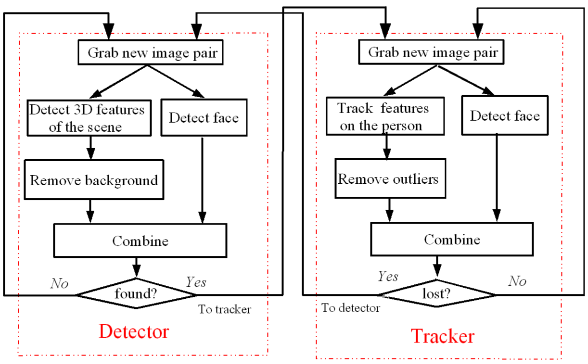
Person Following

- *Goal:* Autonomously follow a person in a cluttered indoor environment
- *Our approach:* Sparse stereo based on KLT feature tracking
  Handles:
  - Dynamic backgrounds
  - Out-of-plane rotation
  - Similar disparity b/w person and background
  - Similar color b/w person and background

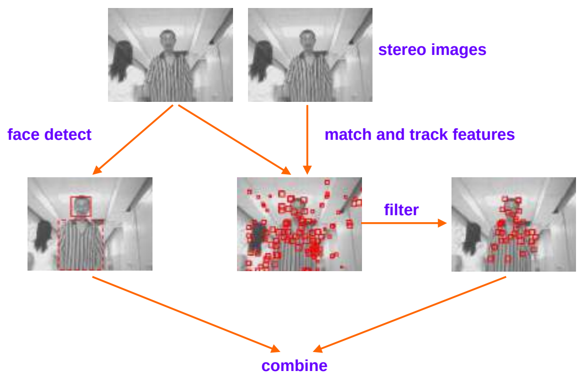
Previous approaches:
- Appearance properties: color, edges
  [Sidenbladh et al. 1999, Tarokh and Ferrari 2003, Kwon et al. 2005]
  - Person has different color from background or faces camera
  - Lighting changes
- Optical flow [Piaggio et al 1998, Chivilò et al. 2004]
  - Drift as the person moves with out-of-plane rotation
- Dense stereo and odometry [Beymer and Konolige 2001]
  - Difficult to predict the movement of the robot (uneven surfaces, slippage in the wheels)

[Chen and Birchfield, IROS 2007]

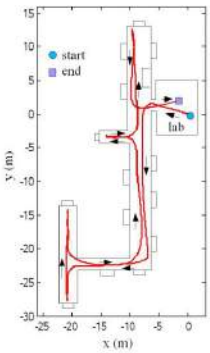
Person Following System Overview



Person Following System Overview



Experimental Results



Video

Path Following

- *Goal:* **Traverse a desired trajectory**
- *Applications:* **courier, delivery, tour guide, scout**

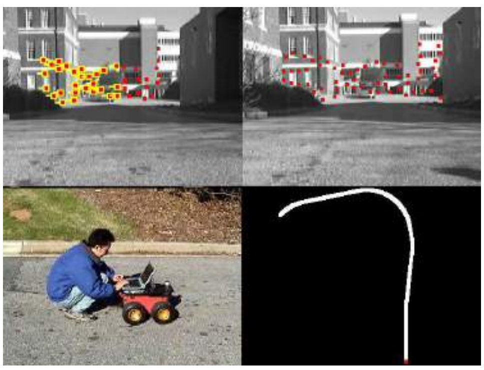
Our Approach to Path Following

- *Key intuition:* **Vastly overdetermined system**
  (Dozens of feature points, one control decision)
- *Key result:* **Simple control algorithm**
  - **Teach / replay approach using sparse feature points**
  - **Single, off-the-shelf camera**
  - **No calibration for camera or lens**
  - **Easy to implement (no homographies or Jacobians)**

    Previous approaches:
    - Image Jacobian [Burschka and Hager 2001]
    - Homography [Sagues and Guerrero 2005]
    - Homography (flat ground plane) [Liang and Pears 2002]
    - Man-made environment [Guerrero and Sagues 2001]
    - Calibrated camera [Atiya and Hager 1993]
    - Stereo cameras [Shimizu and Sato 2000]
    - Omni-directional cameras [Adorni et al. 2003]

[Chen and Birchfield, ICRA 2006; Chen and Birchfield, TROB 2009]

Preview of Results



current image **keyframe**
overview **top-down view**

Tracking Feature Points

Kanade-Lucas-Tomasi (KLT) feature tracker

- Automatically selects features using eigenvalues of 2x2 gradient covariance matrix

$$Z = \iint_W \underbrace{\mathbf{g}(\mathbf{x})\mathbf{g}^T(\mathbf{x})}_{\text{gradient of image}}$$

- Automatically tracks features by minimizing sum of squared differences (SSD) between consecutive image frames

$$\iint_W \left[I\left(\mathbf{x} - \frac{\mathbf{d}}{2}\right) - J\left(\mathbf{x} + \frac{\mathbf{d}}{2}\right) \right]^2 d\mathbf{x}$$

unknown displacement

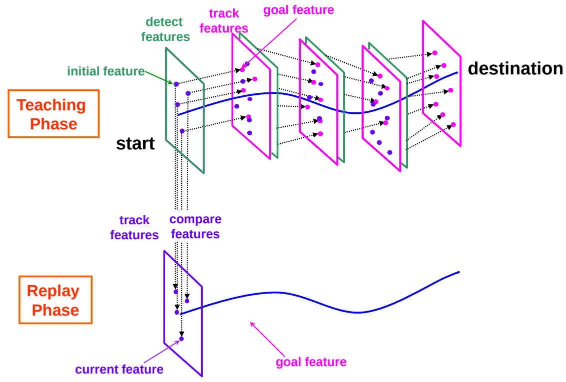
gray-level images

- Augmented with gain and bias to handle lighting changes

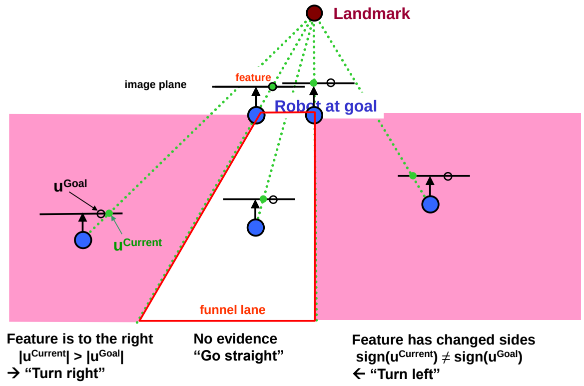
$$\iint_W \left[I\left(\mathbf{x} - \frac{\mathbf{d}}{2}\right) - \left(\alpha J\left(\mathbf{x} + \frac{\mathbf{d}}{2}\right) + \beta\right) \right]^2 d\mathbf{x}$$

- Open-source implementation http://www.ces.clemson.edu/~stb/klt

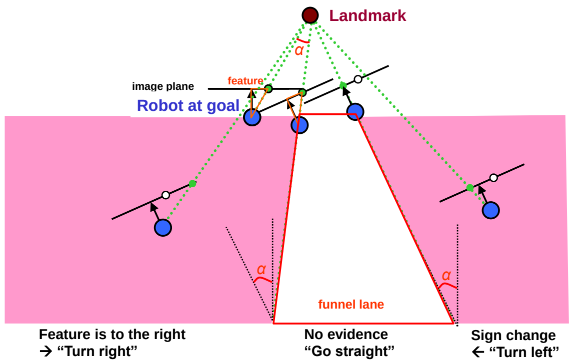
Teach-Replay



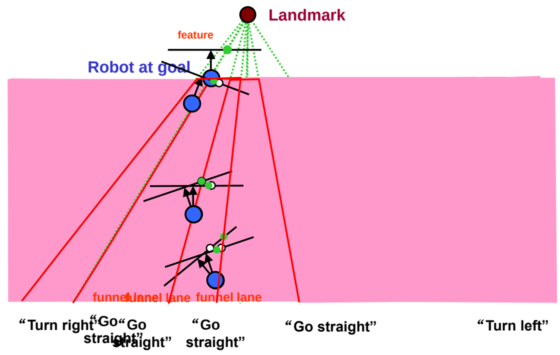
Qualitative Decision Rule



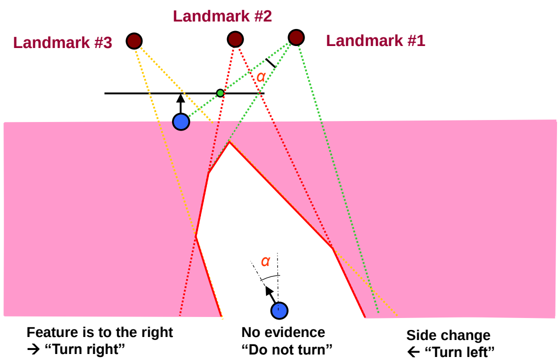
The Funnel Lane at an Angle



A Simplified Example



The Funnel Lane Created by Multiple Feature Points



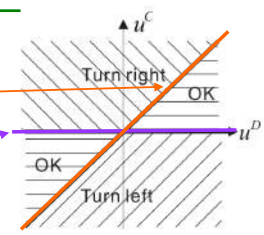
Qualitative Control Algorithm

Funnel constraints:

$$|u^C| < |u^D|$$

and

$$\mathrm{sign}(u^C) = \mathrm{sign}(u^D)$$



Desired heading

$$\theta_d^i = \begin{cases} \gamma \min\{u^C, \varphi(u^C, u^D)\} & \text{if}(u^C > 0 \text{ and } u^C > u^D) \\ \gamma \max\{u^C, \varphi(u^C, u^D)\} & \text{if}(u^C > 0 \text{ and } u^C < u^D) \\ 0 & \text{otherwise} \end{cases}$$

where φ is the signed distance between the u^C and u^D

Incorporating Odometry

Desired heading

$$\theta_d = \beta \frac{1}{N} \sum_{i=1}^{N} \theta_d^i + (1-\beta)\theta_o$$

Desired heading from *i*th feature point

Desired heading from odometry

N = number of features

$\beta \in [0,1]$

Overcoming Practical Difficulties

To deal with rough terrain:

Prior to comparison, feature coordinates are warped to compensate for a non-zero roll angle about the optical axis by applying the RANSAC algorithm.

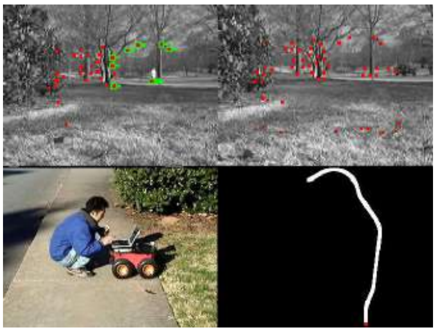
To avoid obstacles:

The robot detects and avoids an obstacle by sonar, and the odometry enables the robot to roughly return to the path. Then the robot converges to the path using both odometry and vision.

Experimental Results

current image

overview



keyframe

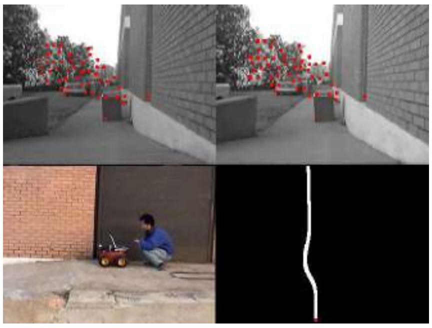
top-down view

Videos available at http://www.ces.clemson.edu/~stb/research/mobile_robot

Experimental Results

current image

overview

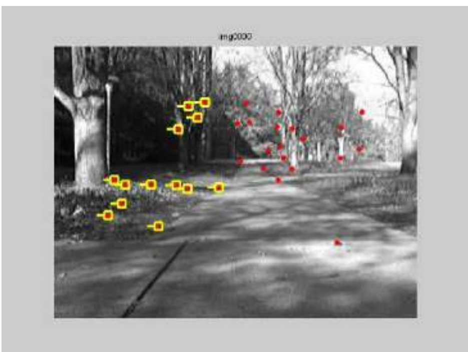


keyframe

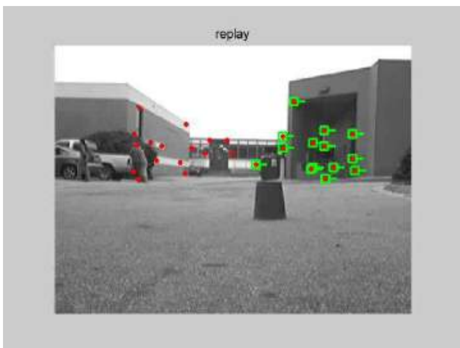
top-down view

Videos available at http://www.ces.clemson.edu/~stb/research/mobile_robot

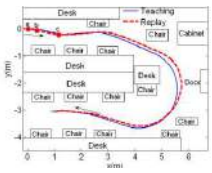
Results: Rough Terrain



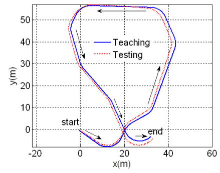
Results: Avoiding an Obstacle



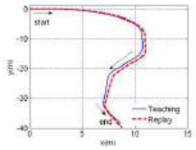
Results: 2D paths



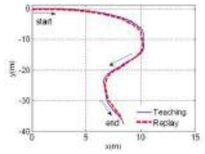
Indoor



Outdoor



Imaging Source Firewire camera



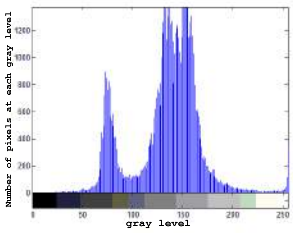
Logitech Pro 4000 webcam

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Low-Resolution Exploration

Image



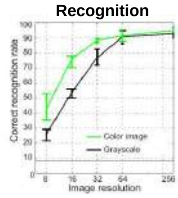
Gray-level histogram

Resolution:

[Murali and Birchfield, VLMP/CVPR 2008; Murali and Birchfield, AR 2012]

What Resolution is Needed?

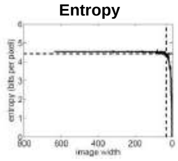
Recognition

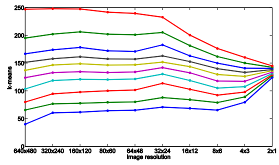


[Torralba et al. 2008]

Discernible segments

Entropy

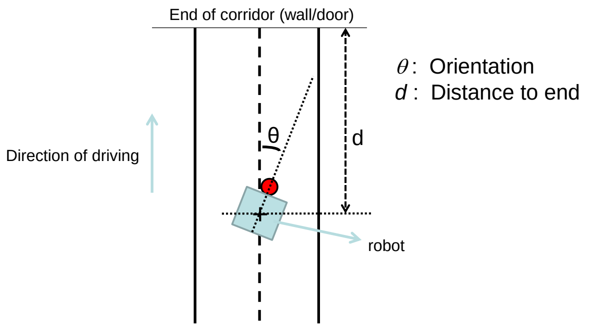




→ 32 x 24

Ratio: 1000:1 (discard 99.9% of data)

Variables to be Estimated



Preview of Results

Estimating the Robot Heading

- Maximum Entropy
- Symmetry by Mutual Information
- Aggregate phase
- Median of bright pixels
- Vanishing points using self-similarity

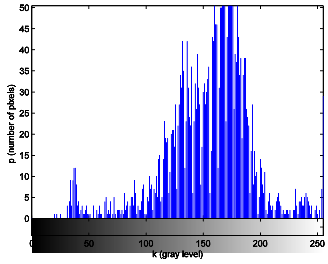
Maximum Entropy

- Entropy - measure of information content

*v*th histogram bin

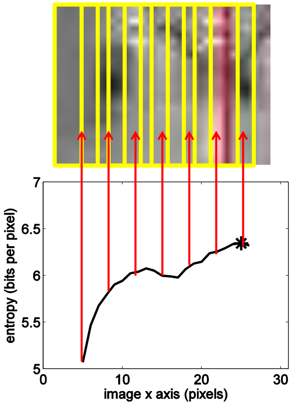
$$H(V) = \sum_{v \in \mathcal{V}} -p(v) \log p(v)$$

{0, 1, 2, …, 255}



Entropy of the gray level histogram

Maximum Entropy

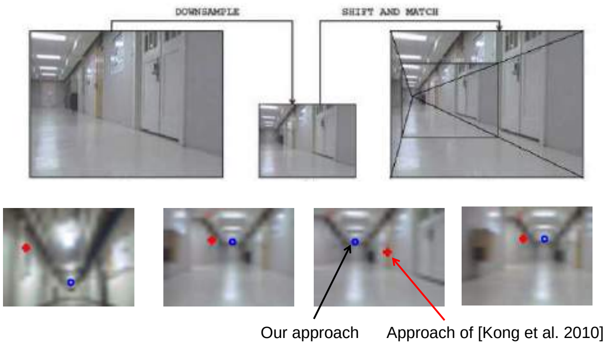


Estimating the Robot Heading

- Maximum Entropy
- Symmetry by Mutual Information
- Aggregate phase
- Median of bright pixels
- Vanishing points using self-similarity

Vanishing Points Using Self-Similarity

Image is locally self-similar under scaling around the vanishing point.
[Stentiford 2006, Kogan et al. 2009]



Estimating Distance to End of Corridor

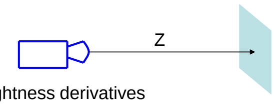
- Time-to-collision
- Jeffrey Divergence
- Entropy
- *Mutual Information*
- *Aggregate optical flow*

Tried, but was not effective

Time To Collision

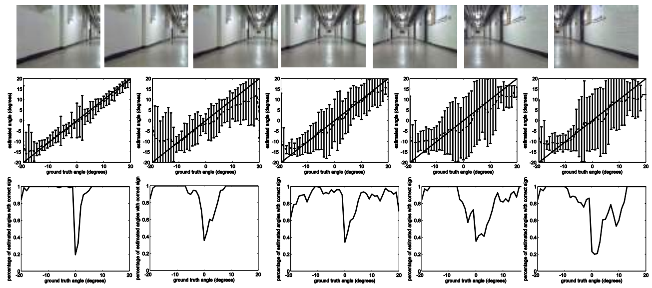
- TTC – Time taken by the camera to reach the surface being viewed
- Assumptions:
  - Brightness constancy
  - Camera moving such that optical axis is perpendicular to a planar surface
- Approach of [Horn et al. 2007] requires no calibration, no tracking:

$$\tau_{TTC} = \frac{-\sum (G(x,y))^2}{\sum G(x,y)\, E_t}$$



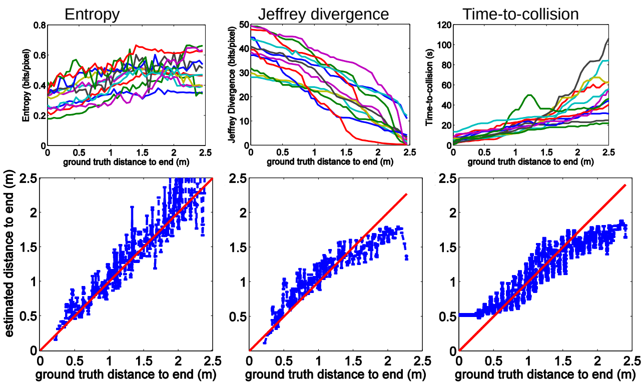
  - E_x and E_y : spatial image brightness derivatives
  - E_t : temporal image derivative
  - $G = xE_x + yE_y$

Orientation

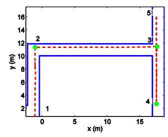


ceiling | entropy | symmetry | vanishing point | aggregate phase

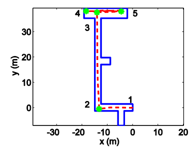
Distance to the End



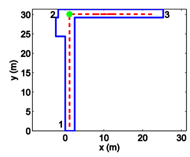
Autonomous Exploration



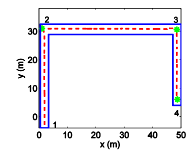
Freeman 0



Riggs 0

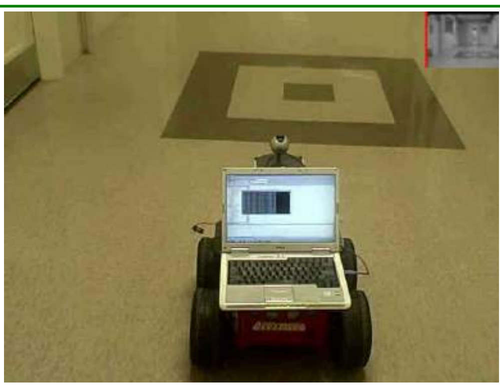


Lowry 0

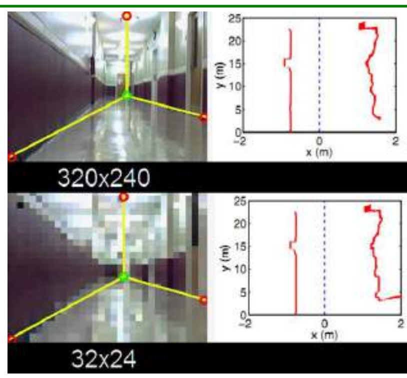


Sirrine 2

Video: Autonomous Exploration



Video (Minimalistic Corridor Geometry)



[Li et al., ICIRA 2012]

Outline

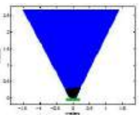
- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Obstacle Avoidance

Goal: Wander aimlessly while avoiding obstacles

Two problems with 3D sensors:

Specular surfaces



close objects

Both yield invalid depth readings

[Peasley and Birchfield, WORV 2013]

Handling invalid depth readings

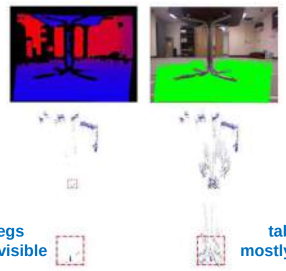
Requirement #1: Do not ever wander into region with no information.

Requirement #2: Do not be overly conservative.

Need to achieve safe compromise

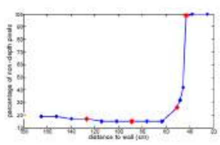
Infinite poles for specular surfaces

For any invalid depth reading,
- Hypothesize a thin, vertical, infinitely tall pole if any neighbor is floor
- Otherwise ignore it



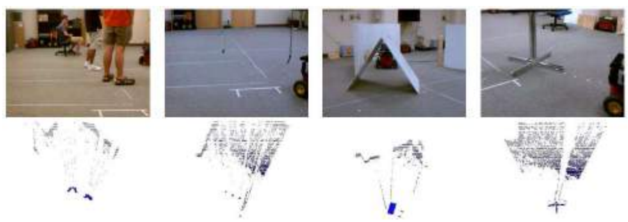
Close objects

If percentage of invalid pixels exceeds a threshold, then stop and turn continually until path is clear.



Experiments

Robust behavior at 30 Hz:



Outline

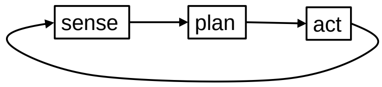
- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

What is interactive perception?

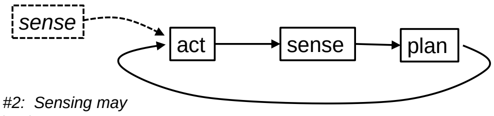
Interactive Perception – gathering information through altering the environment

Traditional approach:

Caveat #1: Reality involves loop



Interactive perception:



Caveat #2: Sensing may be used to jumpstart

Example #1

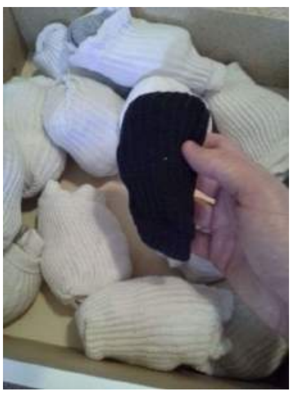
Raccoons and cats learn about their environment using their front paws

Example #2

What is in the can?

Example #3

Where are the dark socks?



Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Types of objects

Non-Rigid

Articulated (6+K DOF)

Example: stapler, cabinet

Example: shirt, towel

Flexible (∞ DOF)

degrees of freedom (DOF)

Example: brick, table

rigid (6 DOF)

Bendable (6+M DOF)

Example: shoe, paper

We have come a long way

Fetch water → boil → soak → beat → scrub → rinse → wring → hang → fold / iron

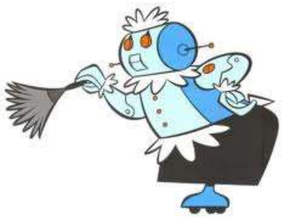
~2000 BC
laundry bat

1937
first automatic
washing machine

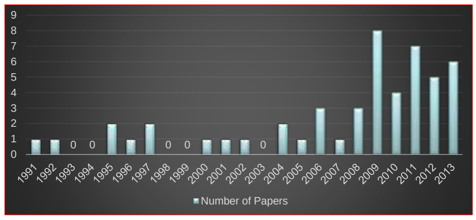
1800s
washboard

The Problem

A Solution



History of robotic laundry research



Total: Just 50 papers

1962
Jetsons popularizes
robotic household servant

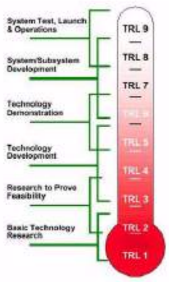
1991
first paper on
robotic cloth handling

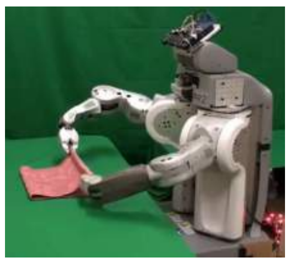
2010
robot folding
towels, shirts, pants

The 50 papers

- Ono-Ichijo-Aisaka, ICAR 1991
- Patton et al., JDSMC 1992
- Ono-Kita-Sakane, IROS 1995
- Paraschidis et al., CI 1995
- Kaneko-Kakikura, ACRSJ 1996
- Howard-Bekey, ICAR 1997
- Bordegoni-Frugoli-Rizzi, HCI 1997
- Hamajima-Kakikura, RAS 2000
- Kaneko-Kakikura, SATP 2001
- Kita-Kita, WACV 2002
- Kita-Saito-Kita, ICRA 2004
- Salleh et al., ICMA 2004
- Cannata-Maggiali, ICHR 2005
- Salleh et al., JRM 2006
- Salleh et al., SICE-ICASE 2006
- Osawa-Seki-Kamiya, JRM 2006
- Osawa-Seki-Kamiya, JACIII 2007
- Salleh et al., ALR 2008
- Shibata et al., ICASE 2008
- Kobayashi et al., IECON 2008
- Hata et al., ICMA 2009
- Yoshida et al., ISIE 2009
- Kita et al., IROS 2009
- Kita et al., ICRA 2009
- Fong, IJRR 2009
- Gibbons-Culverhouse-Bugmann, TAROS 2009
- Willimon, 2009
- Yamakazi-Inaba, MVA, 2009
- Bell, 2010
- Van den Berg et al., WAFR 2010
- Molfino et al., ROBOTIK 2010
- Maitin-Shepard et al., ICRA 2010
- Willimon-Birchfield-Walker, IROS 2010
- Willimon-Birchfield-Walker, ICRA 2011
- Willimon-Birchfield-Walker, IROS 2011
- Cusumano-Towner et al., ICRA 2011
- Elbrechter-Haschke-Ritter, IROS 2011
- Miller et al., ICRA 2011
- Bersch-Pitzer-Kammel, IROS 2011
- Wang et al., IROS 2011
- Lakshmanan et al., ISER 2012
- Ramisa et al., ICRA 2012
- Miller et al., IJRR 2012
- Aragon-Camarasa et al., SICSA 2012
- Willimon-Birchfield-Walker, IJARS 2012
- Willimon et al., IROS 2012
- Willimon-Birchfield-Walker, ICRA 2013
- Willimon-Birchfield-Walker, ICRA 2013
- Willimon-Birchfield-Walker, ISRNR 2013
- Willimon, 2013
- Schulman et al., ICRA 2013
- Sun et al., TAROS 2013

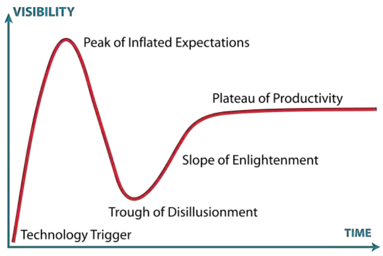
Where are we, really?





Technology readiness levels

Hype “cycle”



Special-purpose machines?

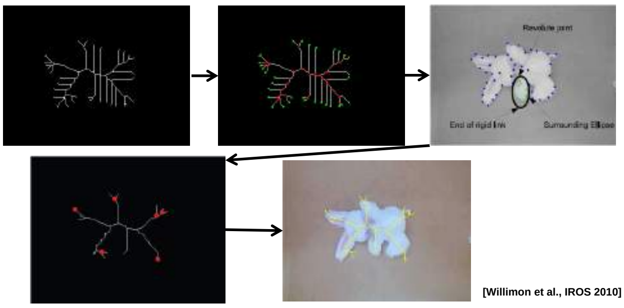
www.foldimate.com

Challenges

- Clothing is infinite-dimensional:
  - difficult to model
  - difficult to perceive
  - difficult to manipulate
- But compliance also makes some tasks easier (singulation)

Our early work on interactive perception

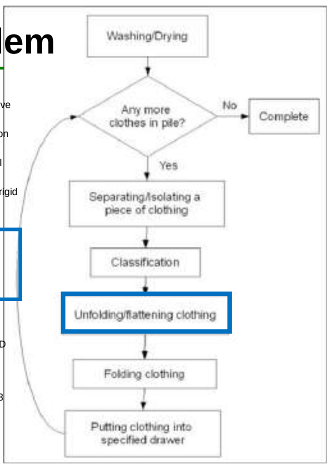
- Compute skeleton
- Push/poke object at interaction/end points (green)
- Revolute joint (red) given by major axis of ellipse



[Willimon et al., IROS 2010]

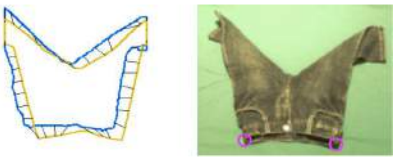
Laundry problem

- Clothing Classification
  - Willimon et al., Classification of clothing using interactive perception, ICRA 2011
  - Willimon et al., A new approach to clothing classification using mid-level layers, ICRA 2013
  - Willimon et al., Classification of clothing using midlevel layers, ISRN Robotics, 2013
  - Willimon et al., Interactive perception of rigid and non-rigid objects, IJARS 2012
- Unfolding Clothing
  - Willimon et al., Model for Unfolding Laundry using Interactive Perception, IROS 2011
- Pose Estimation
  - Willimon et al.,, An energy minimization approach to 3D non-rigid deformable surface estimation using RGBD data, IROS 2012
  - Willimon et al., 3D non-rigid deformable surface estimation without feature correspondence, ICRA 2013



Related Work on Unfolding Clothing

- Cusumano-Towner et al., ICRA 2011, flattened a piece of crumpled clothing by disambiguation and reconfiguration phases



Unfolding System

Two phases:

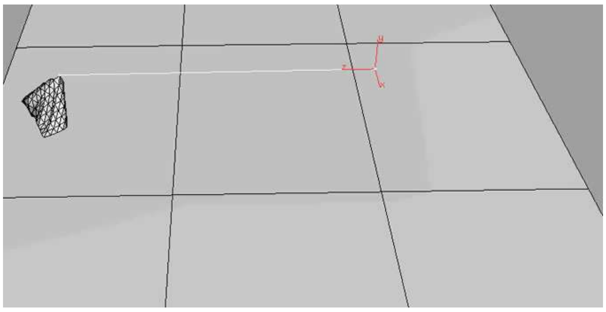
- Tug in random directions
- **Use energy model to identify folds and peaks**

Experimental Results

- The proposed approach was applied to a 3D simulated cloth
- **Together, the first and second phase of the algorithm obtained a flattened percentage greater than 95%.**

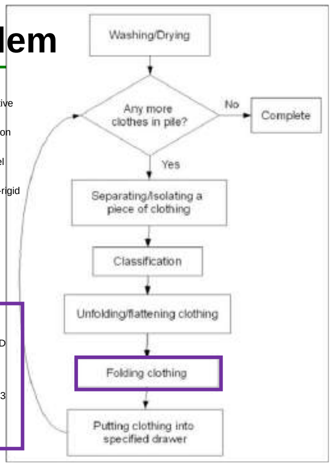
Iteration	1	7	13	19	25	31	37	43	49
Depth Image									
Cloth Model									
PC_{Flat}(%)	3.21	22.61	45.27	72.32	46.36	4.54	23.93	0.01	95.57

Video

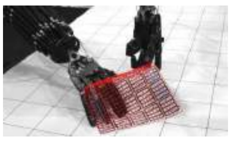


Laundry problem

- Clothing Classification
  - Willimon et al., Classification of clothing using interactive perception, ICRA 2011
  - Willimon et al., A new approach to clothing classification using mid-level layers, ICRA 2013
  - Willimon et al., Classification of clothing using midlevel layers, ISRN Robotics, 2013
  - Willimon et al., Interactive perception of rigid and non-rigid objects, IJARS 2012
- Unfolding Clothing
  - Willimon et al., Model for Unfolding Laundry using Interactive Perception, IROS 2011
- Pose Estimation
  - Willimon et al., An energy minimization approach to 3D non-rigid deformable surface estimation using RGBD data, IROS 2012
  - Willimon et al., 3D non-rigid deformable surface estimation without feature correspondence, ICRA 2013



Related Work on Pose Estimation



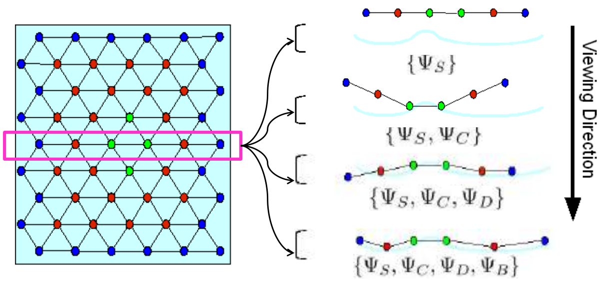
- Elbrechter et al., IROS 2011, use a soft-body-physics model with visual tracking to manipulate a piece of paper

- Bersch et al., IROS 2011, bring a T-shirt into a desired configuration by alternately grasping the item with two hands, using a fold detection algorithm

Both approaches require predefined fiducial markers

Pose Estimation by Energy Minimization

- **We minimize the energy equation of a mesh model that involves 4 terms:** Smoothness, correspondence, depth, boundary



[Pilet et al., IJCV 2008]

Experimental Results

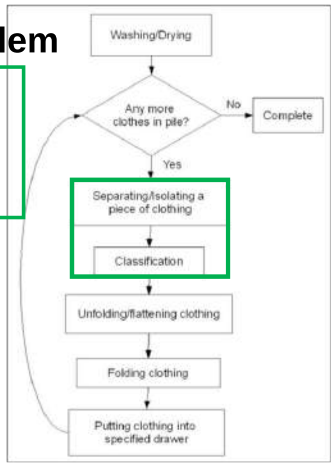
Video

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Laundry problem

- Clothing Classification
  - Willimon et al., ICRA 2011
  - Willimon et al., ICRA 2013
  - Willimon et al., ISRN Robotics, 2013
  - Willimon et al., IJARS 2012
- Unfolding Clothing
  - Willimon et al., IROS 2011
- Pose Estimation
  - Willimon et al., IROS 2012
  - Willimon et al., ICRA 2013



Related Work on Clothing Classification

- Kita et al., ICRA 2009 recognize the state of clothes (pants, shirt)

- Cusumano-Towner et al., ICRA 2011, classified 7 articles (short-sleeve shirt, long-sleeve shirt, pants, skirt, towel, 2 infant clothes)

Classification system

Given pile of clothes, individuate and classify the items:

- **Isolation**
  - **Graph-based segmentation**
  - **Stereo matching**
  - **Determining grasp point**

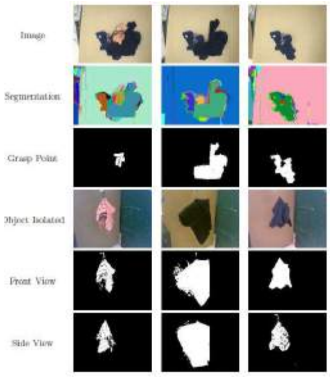
- **Classification**
  - **Binary silhouettes**
  - **Visual-based shape and appearance information**

Classification system

For each article of clothing:
- Multiple views
- Multiple grasp points

Details:
- **6 categories**
- 5 items per category
- **10 grasps per item**
- 2 views per grasp
  (front and side)

→ **20 images per item**
600 total images

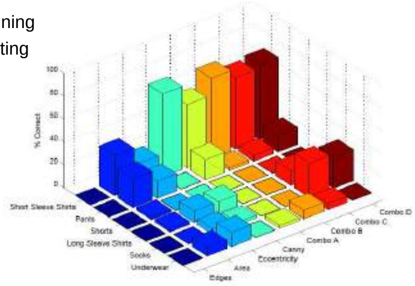


Results

For each category,
- 3 items used for training
- 2 items used for testing

Given single image,
find category of
nearest neighbor

→ Results are poor



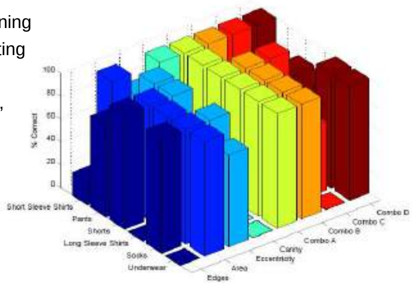
Combo A: area + ecc
Combo B: area + ecc + Canny
Combo C: ecc + Canny
Combo D: area + Canny

Results

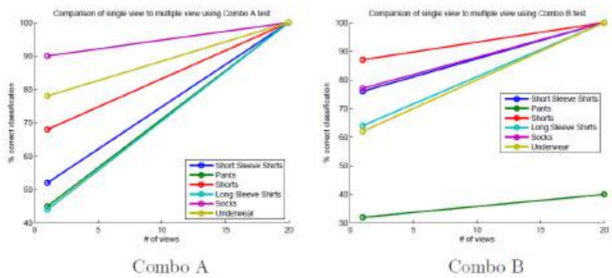
For each category,
- 3 items used for training
- 2 items used for testing

Given multiple images,
find category of
nearest neighbor

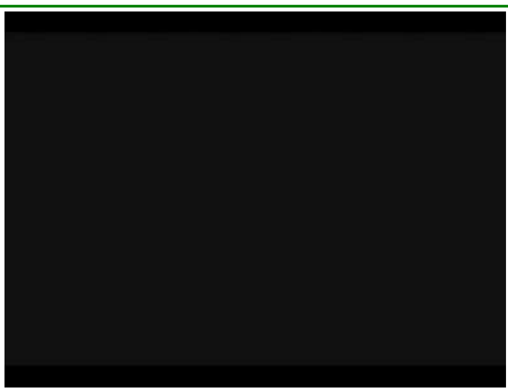
→ Interactive
perception greatly
improves results



Results



Video



What is in a real laundry basket?

Shirts
SS_Shirt (SSS) - 28
LS_Shirt (LSS) - 11
LS_Sweatshirt (SWS) - 1
LS_Dress_Shirt (LSD) - 1
LS_Polo_Shirt (LSP) - 1
SS_Polo_Shirt (POS) - 5
Tanktop (TAT) - 1
Scrub_Shirt (SCS) - 1
Sleeveless_Shirt (SLS) - 4
Undershirt (UND) - 1

Cloths
Washcloth (WCL) - 14
Dishtowel (DTO) - 6
Dishcloth (DSC) - 1
Handtowel (HTO) - 2
Dinner_Cloth (DNC) - 4

Pants
Bluejean_Pants (BJP) - 8
Non_Bluejean_Pants (NBP) - 5
Sweatpants (SWP) - 3
Scrub_Pants (SCP) - 1
Dress_Pants (DRP) - 3

Shorts
Bluejean_Shorts (BJS) - 4
Bluejean_Cargo_Shorts (BCS) - 1
Dress_Cargo_Shorts (DCS) - 2
Gym_Shorts (GYS) - 3
Cargo_Shorts (CRS) - 4
Dress_Shorts (DRS) - 6

Dress/Skirt/Bloomer
Sleeveless_Collar_Dress (SCD) - 1
Bloomers (BLO) - 1
Skirt (SKR) - 1

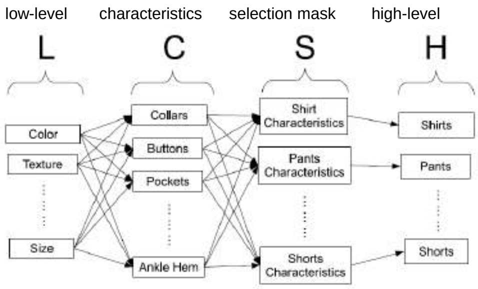
Socks
Sock (WSK) - 7
Ankle_Sock (ASK) - 7
Dress_Sock (DSK) - 4

Jacket
Dress_Jacket (DRJ) - 1
Hooded_Jacket (HOJ) - 2
LS_Pullover_Jacket (LPJ) - 1

Other
Overalls (OVR) - 1
Baby_Blanket (BBB) - 3
Pajama_Top (PJT) - 2
Pajama_Bottom (PJB) - 2

7 major categories, 39 subcategories

LCSH hierarchy



intermediate layers improve performance

(cf. Kumar et al., ICCV 2009)

Dataset

RGBD images for

> 200 articles

> 1000 configurations

7 categories

> 36 subcategories

Articles lie on table

Example data

Low-level features

<u>Global</u>

- **Color histogram**
- **Histogram of line lengths**
- **Table point feature histogram**
- **2D boundary**

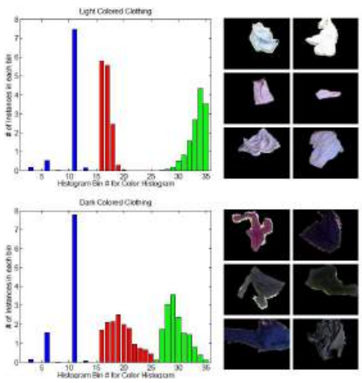
→ Concatenated to create 334D histogram (35 + 20 + 263 + 16 = 334)

<u>Local</u>

- **SIFT**
- **Fast point feature histogram**

Two separate bag-of-words, then concatenated to create 200D vector (100 + 100 = 200)

Color histogram



Histogram of line lengths

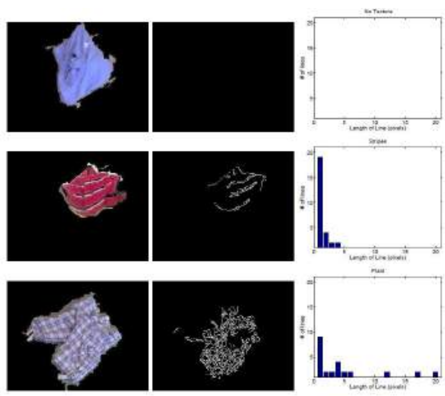
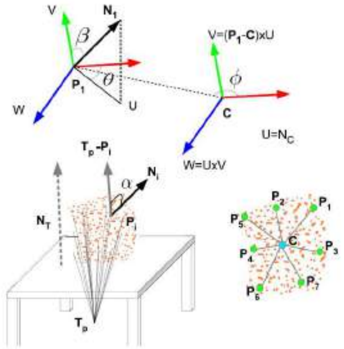
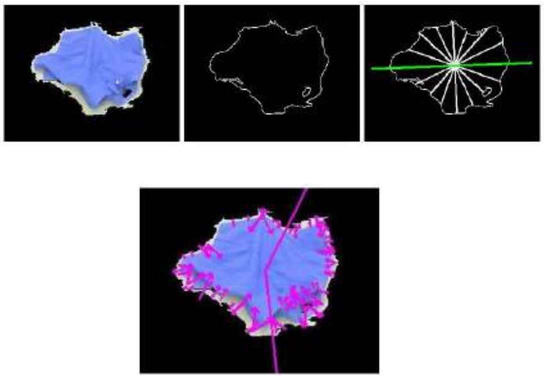


Table point feature histogram

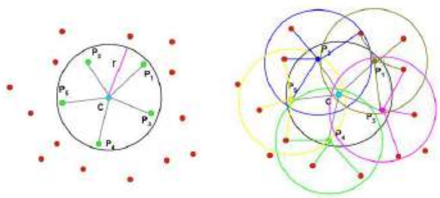


(variant of viewpoint feature histogram, Rudu et al., IROS 2010)

2D boundary / SIFT



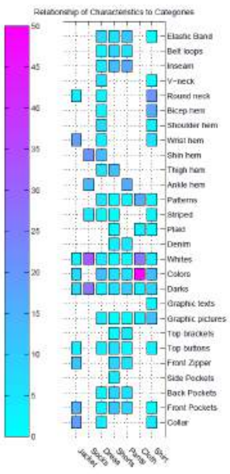
Fast point feature histogram



(Rudu et al., IAS 2009)

27 Characteristics

Collar (21)	the part of a garment that fastens around or frames the neck
Front Pockets (39)	pockets on the front of any garment
Back Pockets (26)	pockets on the back of any garment
Side Pockets (8)	pockets on the side of any garment
Elastic Band (32)	any sort of stretchy band, mainly around the waist
Front Zipper (28)	zippers on the front of any garment
Top buttons (30)	button on the top of any garment, like polo shirts or pants
Belt loops (25)	loops around the waist for a belt
Top brackets (10)	brackets on the top of any garment, like pants
Graphic pictures (47)	any types of pictures or logos on a garment
Graphic texts (7)	any type of text or word on a garment
Dark Colored (70)	clothing that is darker in color, close to black
Colored (94)	any clothing with color that is not whites or darks
White Colored (38)	any clothing that is totally white
V-neck (7)	neckline protrudes down the chest and to a point
Denim (13)	a rugged cotton twill textile
Plaid (5)	a pattern consisting of crossed horizontal and vertical bands
Striped (11)	a pattern consisting of horizontal or vertical bands
Patterns (18)	any repetitious design that doesn't fall under plaid or striped
Round neck (73)	neckline that encircle the neck in a curved shape
Ankle hem (30)	a hem located in the ankle region
Thigh hem (28)	a hem located in the thigh region
Inseam (52)	the seam that binds the length of the inner trouser leg
Shin hem (15)	a hem located in the shin region
Wrist hem (26)	a hem located in the wrist region
Shoulder hem (15)	a hem located in the shoulder region
Bicep hem (59)	a hem located in the bicep region



Results (without characteristics)

Local and Global Features			
	Shirt	Dress	Socks
Shirt	**40.00**	0.00	60.00
Dress	40.00	**0.00**	60.00
Socks	18.18	27.27	**54.55**
Average TPR = 31.52%			

(3 categories)

Local and Global Features							
	Shirt	Cloth	Pants	Shorts	Dress	Socks	Jacket
Shirt	**1.18**	0.00	4.71	34.12	0.00	60.00	0.00
Cloth	0.00	**0.00**	13.33	43.33	0.00	43.33	0.00
Pants	0.00	0.00	**8.00**	32.00	0.00	60.00	0.00
Shorts	0.00	0.00	8.00	**32.00**	0.00	60.00	0.00
Dress	0.00	0.00	0.00	40.00	**0.00**	60.00	0.00
Socks	0.00	0.00	4.55	45.45	0.00	**50.00**	0.00
Jacket	0.00	0.00	0.00	40.00	0.00	60.00	**0.00**
Average TPR = 13.03%							

(7 categories)

Results (with characteristics)

Local and Global Features			
	Shirt	Dress	Socks
Shirt	**21.18**	78.82	0.00
Dress	20.00	**80.00**	0.00
Socks	36.36	63.64	**0.00**
Average TPR = 33.73%			

(3 categories)

Local and Global Features							
	Shirt	Cloth	Pants	Shorts	Dress	Socks	Jacket
Shirt	**21.18**	0.00	0.00	0.00	78.82	0.00	0.00
Cloth	10.00	**0.00**	0.00	0.00	90.00	0.00	0.00
Pants	24.00	0.00	**0.00**	0.00	76.00	0.00	0.00
Shorts	28.00	0.00	0.00	**0.00**	72.00	0.00	0.00
Dress	20.00	0.00	0.00	0.00	**80.00**	0.00	0.00
Socks	36.36	0.00	0.00	4.55	59.09	**0.00**	0.00
Jacket	20.00	0.00	0.00	0.00	80.00	0.00	**0.00**
Average TPR = 14.45%							

(7 categories)

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

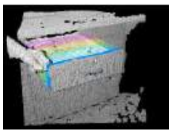
Related Work (articulated reconstruction)

[Yan et al., PAMI 2008]

[Katz et al., ISER 2010]

[Ross et al., IJCV 2010]

[Sturm et al., IJCAI 2009, IROS 2010]



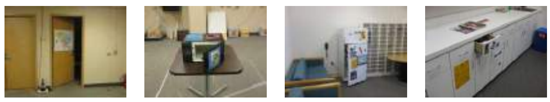
[Sturm et al., ICRA 2010]

Motivation for articulated recon.

Domestic robots in many applications require manipulation of articulated objects

- Tools: scissors, shears, pliers, stapler
- Furniture: cabinets, drawers, doors, windows, fridge
- Devices: laptop, cell phone
- Toys: truck, puppet, train, tricycle

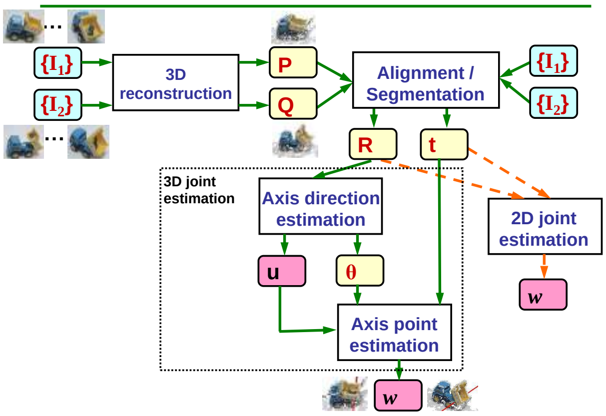
Important problem: Learning kinematic models



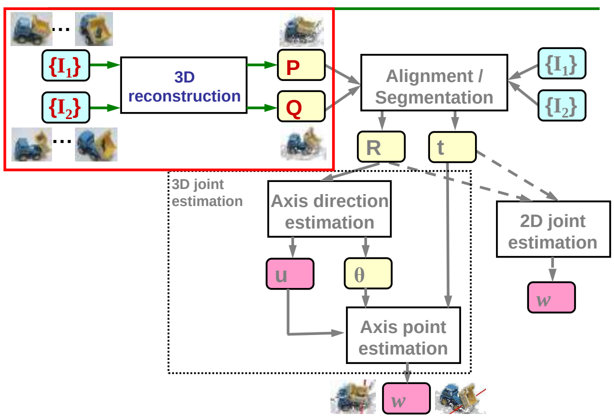
key idea: “occlusion aware”

[Huang and Birchfield, ICRA 2012]

Procrustes-Lo-RANSAC (PLR)



Procrustes-Lo-RANSAC (PLR)



3D Model Reconstruction

Toy truck

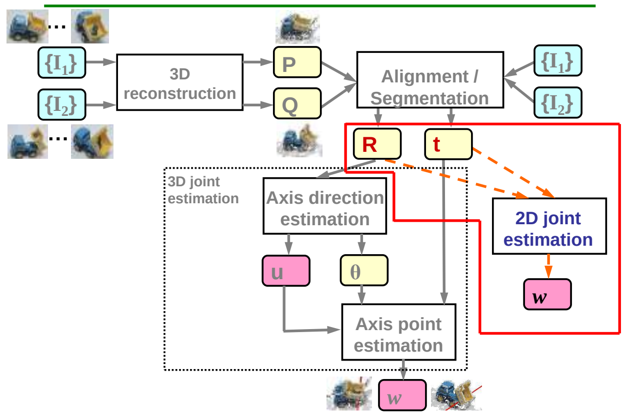
3D model with 147 images

http://phototour.cs.washington.edu/bundler/

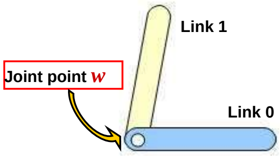
3D model from PMVS

http://grail.cs.washington.edu/software/pmvs/

Procrustes-Lo-RANSAC (PLR)



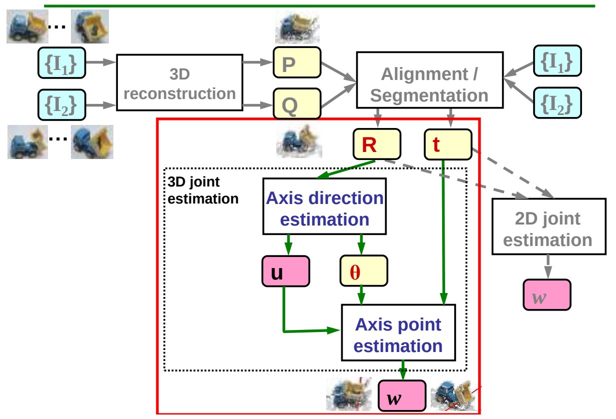
2D Joint Estimation



Configuration 1

3D model

Procrustes-Lo-RANSAC (PLR)



3D Joint

- Revolute joint

Axis direction **u**

Axis point *w*

- Prismatic joint

Axis direction **u**

Axis point *w*

u = t/|t|

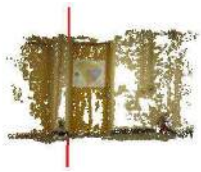
w = mean({p_i})

Joint is classified using R

Experimental Results(1)

1 out of 22

1 out of 19



Red line is the estimated axis

Experimental Results(2)

1 out of 17 | 1 out of 20

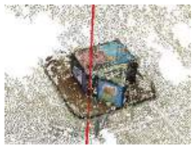
 

Red line is the estimated axis

Experimental Results (3)

1 out of 99 | 1 out of 94

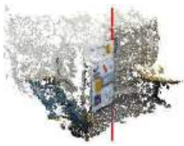
 

Red line is the estimated axis

Experimental Results (4)

1 out of 24 | 1 out of 25

Red line is the estimated axis

Experimental Results (5)

1 out of 13 | 1 out of 18

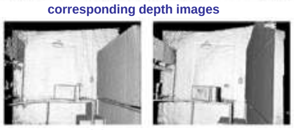
Red line is the estimated axis

Experimental Results (7)

Build a 3D model of a microwave

color images

corresponding depth images



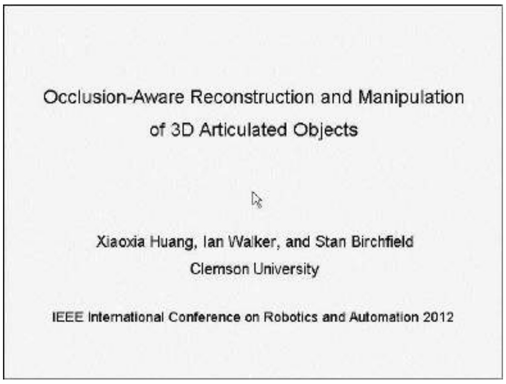
3D model

Experimental Results (8)

Estimate rotation axis of the microwave

Video



Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Microsoft robotics

Mission:

- Advance state-of-art in robotics through research
- Develop commercially viable robotic solutions

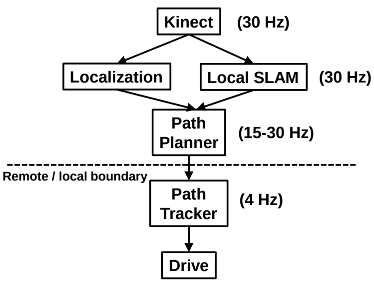

A ROBOT IN EVERY HOME

The leader of the PC revolution predicts that the next hot field will be robotics

By Bill Gates

Indoor navigation

Goal: Reliable, real-time indoor navigation using single forward-facing Kinect sensor



Video

Evaluating navigation system

Want to compare:
- Different algorithms / systems
- Different robots / locations / times

Traditional approaches:
- Simulation – accuracy?
- Dataset – not for closed-loop system
- Contest – co-located participants, only achieve ranking

[Sprunk et al. ISER 2014]

Proposed benchmark

Specify:
- Standardized environment
- Set of challenges
- Test grid
- Reference robot
- Inexpensive, flexible ground-truth system

Standardized environment

Environment contains 4 areas:
- Atrium
- Lounge
- Office
- Hallway

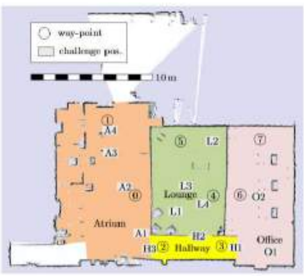
Ideally, should contain
- Doorways
- Multiple surface types (carpet, tile)
- Ability to control lighting

Standardized environment

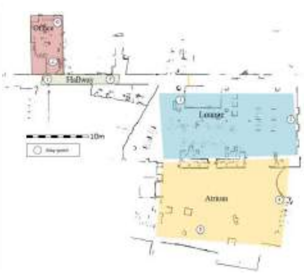
Freiburg

Microsoft

Standardized environment



Freiburg



Microsoft

Set of challenges

Challenges:
- Appearance (A)
  - Picture hanging on walls
  - Lighting conditions
- Geometry (G)
  - Doors opening / closing
  - Moving chairs, desks, tables
- Moving obstacles (O)
  - Dynamic people
  - People gathering in groups
  - Blocking the path

Frequency of change:
- Hourly (H)
- Daily (D)
- Monthly (M)
- Yearly (Y)

Test grid

Simulate a year of activity during test

Area	#	Challenge	Cat.[a]	Freq.[b]	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6
All Areas	1	Artificial Lighting	A	D	Off	On	On	Off	Off	On
	2	Lamps On/Off	A	D	On	On	On	On	On	On
	3	Blinds or Drapes Open/Closed	A	D	All Closed	All Open	50/50	All Closed	All Open	50/50
	4	Wall Art Changes	A	Y	Wall Art 1					
	5	Door Open/Closed	G	H	Constantly	Constantly	Constantly	Constantly	Constantly	Constantly
	6	Wall Color Changes	A	Y	Color 1					
Atrium	7	Large Display Monitors Change Content	A	D	Image 1	Image 2	Image 3	Image 1	Image 2	Image 3
	8	Person in Path (ample room to avoid)	O	D		Position 1	Position2	Position 3	Position 4	
	9	Small Group in Path (ample room to avoid)	O	D		Position 1	Position 2	Position 3		Position 1
	10	Person Pushing Cart (ample room to avoid)	O	D		Position 1	Position 2		Position 1	Position 2
Hallway	11	Shipping Boxes on Floor	G	D		1 Box	2 Boxes	3 Boxes	2 Boxes	1 Box
	12	Cart Moves	G	D	Position 1	Position 2	Position 3	Position 4	Position 1	Position 2
	13	Ladders, Tools, Cables	G	Y			Position 1			
	14	Two People Blocking Path (no room to avoid)	O	D	Position 1				Position 2	
	15	Path Completely Blocked (door) for 1 Minute	O	D		Position 1				Position 2
	16	Path Completely Blocked (people) for 1 Minute	O	D			Position 1			
	17	Person Pushing Cart (no room to avoid)	O	D				Position 1		
Lounge	18	Dining Chairs Shift	G	H	Neat	25% Messy	50% Messy	75% Messy	100% Messy	Neat
	19	Coats/Jackets on Coat Racks	G	D		1/2 Full	Full		1/2 Full	Full
	20	Cart Moves	G	D	Position 1	Position 2	Position 3	Position 4	Position 1	Position 2
	21	Caution Sign (Janitor)	G	D		Position 1	Position 2		Position 2	Position 1
	22	Garbage/Recycling Bags	G	D		Black	White			2 Black
	23	Reconfigure Furniture	G	Y	Configuration 1					
	24	Person Vacuuming or Mopping	O	D		Position 1	Position 2		Position 1	Position 2
	25	Large Work/Social Gathering (20-30 people)	O	M			Position 1	Position 1		
Office	26	Whiteboard Contents Change	A	D	Clean	5%	10%	20%	30%	40%
	27	Desk Chairs Shift (less than 1.5 meters)	G	H	Neat	25% Messy	50% Messy	75% Messy	100% Messy	Neat
	28	Coats/Jackets on Chairs	G	D			5%		10%	2%
	29	Bags on Floor Near Desks	G	D		20%	40%	60%		20%
	30	Loose Paper on Floor	G	D	0 Pieces	3 Pieces	0 Pieces	5 Pieces	0 Pieces	5 Pieces
	31	Shelves Contents Change	G	M	20% Full		40% Full		60% Full	
	32	Shelves Move	G	Y	Position 1					
	33	Small Gathering in Work Area (4-8 people)	O	D		Position 1	Position 2		Position 1	Position 2
	34	Social Gathering (10-15 people)	O	M			Position 1			

[a] Challenge category. A: Appearance, G: Geometry, O: (moving) Obstacle
[b] Frequency of occurrence. H: Hourly, D: Daily, M: Monthly, Y: Yearly

Test grid (cont.)

Area	#	Challenge	Cat.[a]	Freq.[b]	Month 7	Month 8	Month 9	Month 10	Month 11	Month 12
All Areas	1	Artificial Lighting	A	D	On	Off	Off	On	On	Off
	2	Lamps On/Off	A	D	Off	Off	Off	Off	Off	Off
	3	Blinds or Drapes Open/Closed	A	D	All Closed	All Open	50/50	All Closed	All Open	50/50
	4	Wall Art Changes	A	Y	Wall Art 2					
	5	Door Open/Closed	G	H	Constantly	Constantly	Constantly	Constantly	Constantly	Constantly
	6	Wall Color Changes	A	Y	Color 2					
Atrium	7	Large Display Monitors Change Content	A	D	Image 1	Image 2	Image 3	Image 1	Image 2	Image 3
	8	Person in Path (ample room to avoid)	O	D	Position 1	Position 2	Position 3	Position 4		Position 1
	9	Small Group in Path (ample room to avoid)	O	D	Position 2	Position 3		Position 1	Position 2	Position 3
	10	Person Pushing Cart (ample room to avoid)	O	D		Position 1	Position 2		Position 1	Position 2
Hallway	11	Shipping Boxes on Floor	G	D		1 Box	2 Boxes	3 Boxes	2 Boxes	1 Box
	12	Cart Moves	G	D	Position 3	Position 4	Position 1	Position 2	Position 3	Position 4
	13	Ladders, Tools, Cables	G	Y		Position 2				
	14	Two People Blocking Path (no room to avoid)	O	D			Position 3			
	15	Path Completely Blocked (door) for 1 Minute	O	D				Position 3		
	16	Path Completely Blocked (people) for 1 Minute	O	D	Position 2				Position 3	
	17	Person Pushing Cart (no room to avoid)	O	D		Position 1				Position 3
Lounge	18	Dining Chairs Shift	G	H	25% Messy	50% Messy	75% Messy	100% Messy	Neat	25% Messy
	19	Coats/Jackets on Coat Racks	G	D		1/2 Full	Full		1/2 Full	Full
	20	Cart Moves	G	D	Position 3	Position 4	Position 1	Position 2	Position 3	Position 4
	21	Caution Sign (Janitor)	G	D		Position 1	Position 2		Position 2	Position 1
	22	Garbage/Recycling Bags	G	D	2 White			2 Black	2 White	
	23	Reconfigure Furniture	G	Y	Configuration 2					
	24	Person Vacuuming or Mopping	O	D		Position 1	Position 2		Position 1	Position 2
	25	Large Work/Social Gathering (20-30 people)	O	M			Position 2	Position 2		
Office	26	Whiteboard Contents Change	A	D	50%	60%	70%	80%	90%	100%
	27	Desk Chairs Shift (less than 1.5 meters)	G	H	25% Messy	50% Messy	75% Messy	100% Messy	Neat	25% Messy
	28	Coats/Jackets on Chairs	G	D	20%	4%	30%	6%	40%	8%
	29	Bags on Floor Near Desks	G	D	40%	60%		20%	40%	60%
	30	Loose Paper on Floor	G	D	0 Pieces	5 Pieces	6 Pieces	5 Pieces	0 Pieces	5 Pieces
	31	Shelves Contents Change	G	M	20% Full		40% Full		60% Full	
	32	Shelves Move	G	Y	Position 2					
	33	Small Gathering in Work Area (4-8 people)	O	D		Position 1	Position 2		Position 1	Position 2
	34	Social Gathering (10-15 people)	O	M				Position 2		

[a] Challenge category. A: Appearance, G: Geometry, O: (moving) Obstacle
[b] Frequency of occurrence. H: Hourly, D: Daily, M: Monthly, Y: Yearly

Test grid instantiation

- Select waypoints
- Select path between waypoints
- Decide which challenges to implement, when/where

Loop Segment	Area	Challenge/Configuration	Row in Tab. 1, 2
$0 \to 2$	Atrium	person with cart at A2	10
$2 \to 4$	Hallway	2 boxes on the floor	11
	Hallway	2 people block at H2 for 1 min	16
	Lounge	move chairs by 0.2 m	18
	Lounge	cart at L3	20
	Lounge	1 garbage bag on the floor	22
$4 \to 6$	Office	move chairs by 0.2 m	27
	Office	1 jacket on chair	28
	Office	2 bags next to desks	29
	Office	group of 4 people at O2	33
	Office	shelves 40% filled	31
$6 \to 3$			
$3 \to 5$	Lounge	group of 8 people at L1	25
	Lounge	person vacuuming at L2	24
$5 \to 1$	Atrium	group of 4 people at A2	9
$1 \to 7$	Atrium	person at A2	8
	Office	group of 8 people at O1	34
$7 \to 0$			

Reference robot

Reference robot is used to normalize results across environments

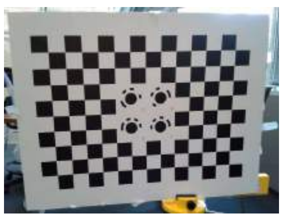
P3-DX (reference robot) omniRob MSR-P1

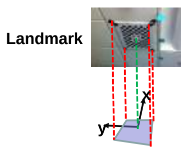
Ground Truth System

- Ground truth system should scale
  - in space (large environments)
  - in time (long runs)
  - in variety (different types of environments)
- *Simplicity* is key to scalability:
  - Low setup time
  - Easy calibration
  - Inexpensive components
  - Non-intrusive

[Kikkeri et al. ICRA 2014]

Ground Truth System





- Checkerboard pattern
- Yields 3D pose of camera relative to target
- Convert to 2D pose of robot on floor

A Useful Instrument

Laser level:

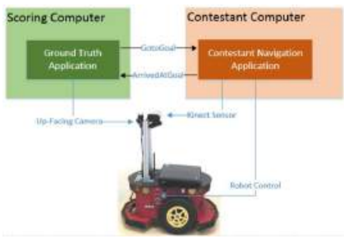
- Upward facing laser provides plumb-up line
- Downward facing laser provides plumb-down line
- Horizontal laser (not used)
- Self-leveling, so plumb lines are parallel to gravity
- Used to determine point on ground directly below origin of target

For more info

- Compare your navigation system: http://research.microsoft.com/brin

Navigation contest

- Microsoft and Adept are organizing Kinect Autonomous Mobile Robot Contest at IROS 2014 in Chicago



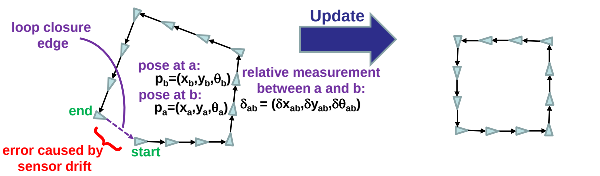
http://www.iros2014.org/program/kinect-robot-navigation-contest

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

PoseSLAM

- Problem: Given a sequence of robot poses and loop closure(s), update the poses

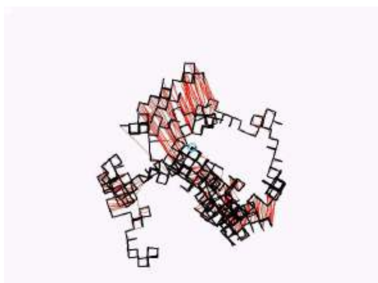


Initial (noisy) pose estimates

Final pose estimates

[Peasley and Birchfield, ICRA 2014]

Video



PoseSLAM as graph optimization

- **Key: Formulate as graph**
  - **Nodes = poses**
  - **Edges = relative pose measurements**

$$\mathcal{G} = (\underset{\{\mathbf{p}_i\}_{i=0}^{n}}{\mathcal{P}}, \mathcal{E})$$

- **Solution: Minimize an objective function**

state

residual for single edge

total error

$$\epsilon(\mathbf{x}) \equiv \sum_{(a,b)\in\mathcal{E}} \mathbf{r}_{ab}^T(\mathbf{x})\Omega_{ab}\mathbf{r}_{ab}(\mathbf{x})$$

info matrix

where

$$\mathbf{r}_{ab}(\mathbf{x}) \equiv \delta_{ab} - \mathbf{f}_{ab}(\mathbf{x})$$

observed measurement

predicted by model

Question: How does state x relate to poses p?

PoseSLAM as nonlinear minimization

- **With a little math,**

$$\epsilon(\mathbf{x}) \equiv \sum_{(a,b)\in\mathcal{E}} \mathbf{r}_{ab}^T(\mathbf{x})\Omega_{ab}\mathbf{r}_{ab}(\mathbf{x})$$

- **becomes**

$$\underbrace{\boxed{J^T(\tilde{\mathbf{x}})\Omega J(\tilde{\mathbf{x}})}}_{A}\ \underbrace{\boxed{\Delta\mathbf{x}}}_{x} = \underbrace{\boxed{J^T(\tilde{\mathbf{x}})\Omega\mathbf{r}(\tilde{\mathbf{x}})}}_{b}$$

Jacobian

current state estimate

increment to state estimate

Repeatedly solve this linear system till convergence

How to solve linear system?

Use sparse linear algebra

- **Solve the equation directly**

$$\Delta\mathbf{x} = M^{-1}J^T(\tilde{\mathbf{x}})\Omega\mathbf{r}(\tilde{\mathbf{x}})$$

where $M = J^T(\tilde{\mathbf{x}})\Omega J(\tilde{\mathbf{x}})$ is preconditioning matrix

- **Drawback:**
  - **Solving equation requires external sparse linear algebra package (e.g., CHOLMOD)**

Example: g2o
(Kummerle et al. ICRA 2011)

Perform gradient descent

- **Update based on single edge**

$$\Delta\mathbf{x} = \lambda_{ab}M^{-1}J_{ab}^T(\tilde{\mathbf{x}})\Omega_{ab}\mathbf{r}_{ab}(\tilde{\mathbf{x}})$$

where λ_{ab} is learning rate

- **Drawback:**
  - **Requires many iterations***

***With global state space**

Example: TORO
(Olson et al. ICRA 2006; Grisetti et al. RSS 2007)

No linear algebra needed! ~100 lines of C++ code

Choosing the state space

Global state space (GSS)
allows fine adjustments

state

$$\mathbf{x_i} \equiv \mathbf{p_i} \equiv \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix}$$

pose

$$\mathbf{p_i} = \mathbf{x_i}$$

Incremental state space (ISS)
simple Jacobian, decoupled parameters, slow convergence
(Olson et al. ICRA 2006)

$$\mathbf{x_i} \equiv \mathbf{p_i} - \mathbf{p_{i-1}} \equiv \begin{bmatrix} x_i - x_{i-1} \\ y_i - y_{i-1} \\ \theta_i - \theta_{i-1} \end{bmatrix}$$

$$\mathbf{p_i} = \mathbf{p_{i-1}} + \mathbf{x_i}$$

$$J_{ab} = \begin{bmatrix} \cdots & 0 & I_{(3\times3)} & 0 & \cdots \end{bmatrix}$$

Relative state space (RSS)
simple Jacobian, coupled parameters, fast convergence
(our approach)

$$\mathbf{x_i} \equiv \begin{bmatrix} x_i' \\ y_i' \\ \theta_i' \end{bmatrix}$$

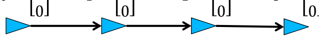
$$\mathbf{p_i} = \mathbf{p_{i-1}} + \underbrace{\begin{bmatrix} \cos\theta_{i-1} & -\sin\theta_{i-1} & 0 \\ \sin\theta_{i-1} & \cos\theta_{i-1} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R(\theta_{i-1})} \underbrace{\begin{bmatrix} x_i' \\ y_i' \\ \theta_i' \end{bmatrix}}_{\mathbf{x}_i}$$

$$J_{ab} = \begin{bmatrix} \cdots & 0 & I_{(3\times3)} & 0 & \cdots \end{bmatrix}$$

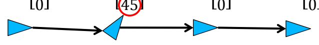
Global state space (GSS)

Global State Space (x, y, θ)

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad x_1 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \quad x_2 = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} \quad x_3 = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$$



$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad x_1 = \begin{bmatrix} 2 \\ 0 \\ 45 \end{bmatrix} \quad x_2 = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} \quad x_3 = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$$

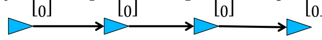


$$\mathbf{p}_i = x_i$$

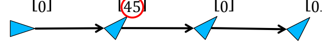
Incremental state space (ISS)

Incremental State Space (Δx, Δ y, Δ θ)

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad x_1 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \quad x_2 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \quad x_3 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$



$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad x_1 = \begin{bmatrix} 2 \\ 0 \\ 45 \end{bmatrix} \quad x_2 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \quad x_3 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

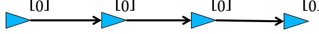


$$\mathbf{p}_i = \sum_{j=0}^{i-1} x_j$$

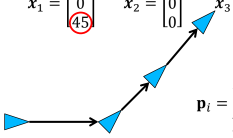
Relative state space (RSS)

Relative State Space (Δx, Δ y, Δ θ)

$$x_0 = \begin{bmatrix}0\\0\\0\end{bmatrix} \quad x_1 = \begin{bmatrix}2\\0\\0\end{bmatrix} \quad x_2 = \begin{bmatrix}2\\0\\0\end{bmatrix} \quad x_3 = \begin{bmatrix}2\\0\\0\end{bmatrix}$$



$$x_0 = \begin{bmatrix}0\\0\\0\end{bmatrix} \quad x_1 = \begin{bmatrix}2\\0\\45\end{bmatrix} \quad x_2 = \begin{bmatrix}2\\0\\0\end{bmatrix} \quad x_3 = \begin{bmatrix}2\\0\\0\end{bmatrix}$$

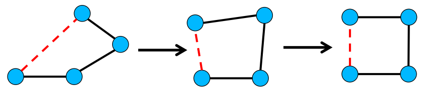


$$\mathbf{p}_i = \sum_{j=0}^{i-1} R(\sum_{k=0}^{j} \theta_k) x_j$$

Proposed approach

Two Phases:

1. **Non-Stochastic Gradient Descent with Relative State Space (POReSS)**
   - RSS allows many poses to be affected in each iteration
   - nSGD prevents being trapped in local minima
   - Result: Quickly gets to a "good" solution
   - Drawback: Long time to convergence
2. **Gauss-Seidel with Global State Space (Graph-Seidel)**
   - Initialize using output from POReSS
   - GSS only changes one pose at a time in each iteration
   - Gauss-Seidel allows for quick calculations
   - Result: Fast "fine tuning"



Pose Optimization using a Relative State Space (POReSS)

$$\Delta x = (J^T(\tilde{x})\Omega J(\tilde{x}))^{-1} J^T(\tilde{x})\Omega r(\tilde{x})$$

nSGD only considers one edge at a time:

$$\Delta x = M^{-1} J_{ab}^T(\tilde{x})\Omega_{ab} r_{ab}(\tilde{x})$$

ISS

$$J_{ab} = [\mathbf{0} \quad \cdots \quad \mathbf{I} \quad \cdots \quad \mathbf{I} \quad \mathbf{0} \quad \cdots]$$

a+1 b

RSS

$$J_{ab} = [\mathbf{0} \quad \cdots \quad \mathbf{I} \quad \cdots \quad \mathbf{I} \quad \mathbf{0} \quad \cdots]$$

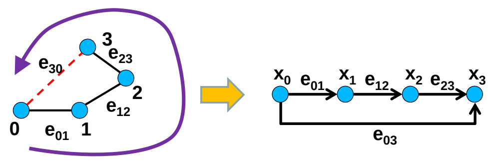
a+1 b

(between consecutive nodes)

$$J_{ab} = \left[\cdots \quad \mathbf{0} \quad {}^a_b B_{a+1} \quad {}^a_b B_{a+2} \quad \cdots \quad {}^a_b B_b \quad \mathbf{0} \quad \cdots\right]$$

(between non-consecutive nodes)

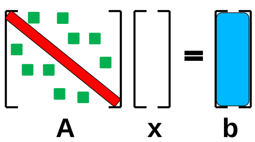
How edges affect states



e_{01} affects x_1
e_{12} affects x_2
e_{23} affects x_3
e_{03} affects x_0, x_1, x_2, x_3

Gauss-Seidel

Repeat: Linearize about current estimate, then solve Ax=b.
→ A is updated each time



Row i:

$$a_{ii}x_i + \sum_{j \neq i} a_{ij}x_j = b_i$$

A is sparse, so solve

$$x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j \neq i} a_{ij}x_j\right)$$

Graph-Seidel

Do not linearize! Instead assume θs are constant
→ A remains constant

$$\begin{bmatrix} \bar{\Omega}_1 & -\bar{\Omega}_{12} & \cdots & -\bar{\Omega}_{1n} \\ -\bar{\Omega}_{21} & \bar{\Omega}_2 & \cdots & -\bar{\Omega}_{2n} \\ \vdots & & \ddots & \vdots \\ -\bar{\Omega}_{n1} & -\bar{\Omega}_{n2} & \cdots & \bar{\Omega}_n \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \underbrace{\sum_{(a,i)\in\mathcal{E}_i^{in}} R(\theta_a)\Omega_{ai}\delta_{ai}}_{\text{edges in}} - \underbrace{\sum_{(i,b)\in\mathcal{E}_i^{out}} R(\theta_i)\Omega_{ib}\delta_{ib}}_{\text{edges out}}$$

A **x** **b**

A: **Diagonal: Connectivity of node (sum of off-diagonals)**
Off-diagonal: Connectivity between nodes
$R(\theta_a)\Omega_{ab}R^T(\theta_a)$ **if connected, 0 otherwise)**

b: **Sum of edges in minus sum of edges out**

Refining the estimate using a Global State Space (Graph-Seidel)

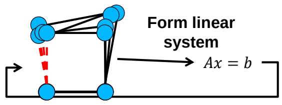
$$x_i^{(k+1)} = \boxed{\frac{1}{a_{ii}}}\left(\boxed{b_i} - \boxed{\sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}}\right), \quad i,j = 1,2,\dots,n.$$

$$diag\left(\sum_{(i,b)\in\varepsilon_i}\Omega_{ib}\right)^{-1}$$

$$\sum_{(i,b)\in\varepsilon_i} -\Omega_{ib}x_b$$

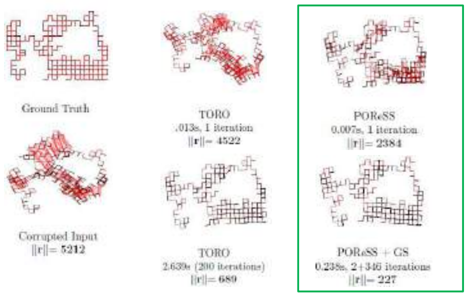
$$\underbrace{\sum_{(a,i)\in\mathcal{E}_i^{in}} R(\theta_a)\Omega_{ai}\delta_{ai}}_{\text{edges in}} - \underbrace{\sum_{(i,b)\in\mathcal{E}_i^{out}} R(\theta_i)\Omega_{ib}\delta_{ib}}_{\text{edges out}}$$

Refining the estimate using a Global State Space (Graph-Seidel)



Compute new state using Graph-Seidel

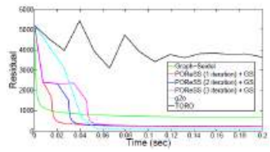
Results: Manhattan world



On Manhattan World, our approach is
- faster and more powerful than TORO
- faster and comparable with g2o

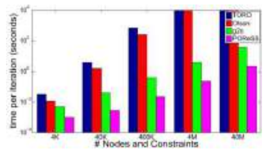
Performance comparison

Residual vs. time



(Manhattan world dataset)

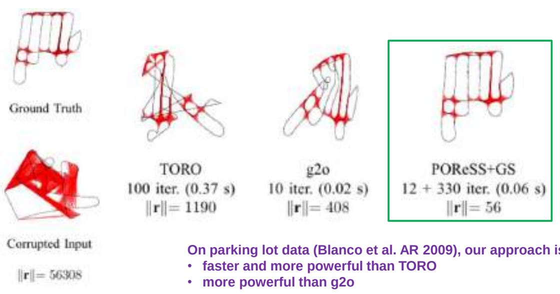
Time vs. size of graph



(corrupted square dataset)

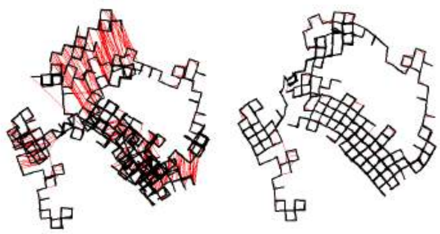
Our approach combines the
- Minimization capability of g2o (with faster convergence)
- Simplicity of TORO

Results: Parking lot

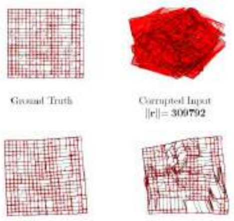


On parking lot data (Blanco et al. AR 2009), our approach is
- faster and more powerful than TORO
- more powerful than g2o

Graph-Seidel



Results: Simple square



Ground Truth

Corrupted Input
$||\mathbf{r}||= \mathbf{309792}$

PORcSS + GS
3 + 773 iterations (3.33s)
$||\mathbf{r}||= \mathbf{1209}$

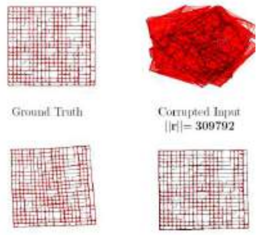
TORO
200 iterations (13.5862s)
$||\mathbf{r}||= \mathbf{7058}$

Rotational version (rPOReSS)

<u>**Majority of drift is caused by rotational errors**</u>

- **Remove rotational drift**
- **Treat (x,y) components of poses as constants**
- **Greatly reduces complexity of Jacobian derivation**

$$J_{ab} = \left[\cdots \; 0 \; \underbrace{1}_{a} \; \cdots \; \underbrace{1}_{b-1} \; 0 \; \cdots \right]$$



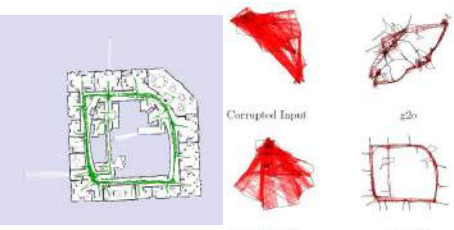
Ground Truth

Corrupted Input
$||\mathbf{r}||= \mathbf{309792}$

Full PORcSS
$||\mathbf{r}||= \mathbf{1209}$

rPORcSS
$||\mathbf{r}||= \mathbf{512}$

Results of rPOReSS: Intel dataset



Corrupted Input

g2o

Full PORcSS

rPORcSS

Incremental version (irPOReSS)

<u>**Incremental**</u>**:**

- **Rotational POReSS can be run incrementally**
- **Goal is to keep graph “close” to optimized**
- **Run Graph-Seidel whenever a fully optimized graph is desired**
- **(Note: Further iterations of POReSS will undo fine adjustments made by Graph-Seidel)**

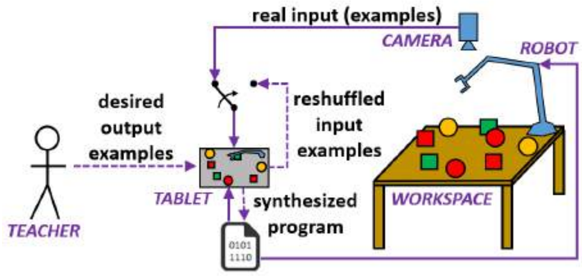
Video

Video

Outline

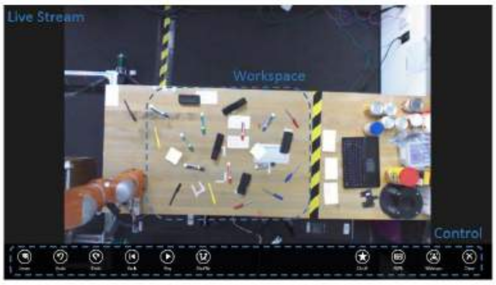
- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Learn-by-demonstration



[Feniello et al., IROS 2014]

Interface



Domain-specific language (DSL)

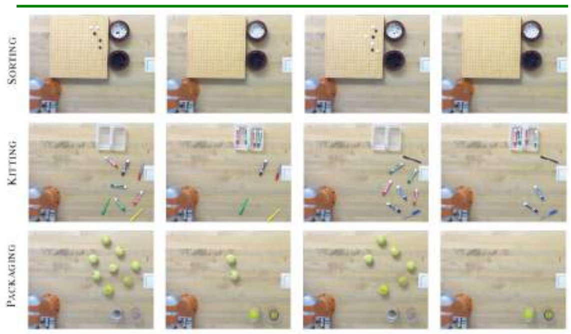
- Stack-based concatenative language
- Concisely represents complex programs
- Programs are composed by concatenation
- Implementation is extremely concise

$\text{things}\ [\text{'color}\ @\ \text{'red}\ =]\ \text{filter}\ 13\ 27\ 1.57\ \text{moveall}$

Searching for programs

- Search for filters (continuous or discrete)
  $[\text{'}\mathit{property}\ @\ \mathit{value}\ =]$
  $[\text{'}\mathit{property}\ @\ \mathit{min}\ \mathit{max}\ \text{between}]$
- Combine filters
  $[[\mathit{filter}_1]\ [\mathit{filter}_2]\ \text{bi and}]$
  $[[\mathit{filter}_1]\ [\mathit{filter}_2]\ \text{bi or}]$
- Search for groupings
  $\text{things}\ [\text{'color}\ @]\ 2\ \text{takebygroup}$
  $[[x_1\ y_1\ \theta_1]\ [x_2\ y_2\ \theta_2]\ [x_3\ y_3\ \theta_3]]\ \text{distribute}$
- Synthesize programs using AST (abstract search tree)

Results



initial final initial final
TRAIN TEST

Video

Outline

- Historical overview
- Sense-Plan-Act
  - Interleaving planning and execution
  - Indoor mapping
  - Door detection
- Behavior-Based Robotics
  - Person and path following
  - Low-resolution navigation
  - Obstacle avoidance
- Interactive Perception
  - Handling highly flexible objects
  - Classification of laundry
  - Articulated reconstruction
- Hybrid Approach
  - Complete indoor navigation system
  - Graph optimization for loop closure
  - Learning-by-demonstration
- Conclusion

Conclusion

- The perception-action loop is key to robotics:
  - Sense-Plan-Act
  - Behavior-based
  - Three-tiered architecture
  - Concurrent modules running in parallel
- Simplicity is key to success:
  - Minimalistic sensing
  - Interactive perception
  - System evaluation
- Future:
  - Interesting, useful behavior will be possible by combining simple capabilities
  - Robust vision is very difficult, even with RGBD sensors
  - Brace yourself for growth in robotics

Thanks!

Meta-Research and Meta-Robotics*

Hans-Georg Stork
(h-gATcikon.de)

Avant-propos

I must begin with an apology. I am not a roboticist. That is to say, I have never been directly involved in activities that are in one way or another linked to the design and building of the kind of machines that this summer school is about. And I have not made the slightest contribution to the advancement of the science underlying the design and building of these machines. So I am not a robot (let alone rocket) scientist either. (I am aware of the ambiguity.)

So be prepared for a largely non-technical, non scientific interlude (as announced). It won't be quite as non-technical and certainly not as literary as Karel Čapek's famous 1920 play *Rosumovi Univerzální Roboti* (*Rossum's Universal Robots*)[1] which allegedly introduced not only the term robot to the world of Science Fiction but also made its robots reason in a real, noisy and dynamically changing world. In fact, the Czech word rozum, if I am not mis-informed, means just that: reason or common sense. Thus R.U.R. predates the "ReaRW task" by nearly one century and represents a fitting genius loci for this summer school.

As for me there is but one justification for speaking to you that I can claim: I have over the last eight years before my departure from the European civil service (two-and-a-half years ago) been involved in a kind of meta-research that was indeed strongly related to what you are doing or learning to do.

Introduction

As the term "*meta*" suggests this has been research about your research: finding out what the burning scientific and technical issues are, who is tackling

*Transcript of a lecture given at the Summer School – Reasoning in the Robot World, 2014 Prague, Czech Republic, 29-31 July, 2014. Hosted by the Czech Technical University in Prague, Center for Machine Perception (`http://summerschool2014.ciirc.cvut.cz/`)

[1] `http://en.wikipedia.org/wiki/R.U.R.`

these issues, what feasible approaches are taken, et cetera, and perhaps most importantly, to what end this research should be done and hence, financially supported.

I did this sort of research jointly with a fair number of more or less like-minded colleagues in my capacity as "Research Program Officer" working for the European Commission's Directorate General "Information Society (INFSO)" (which not long after my departure, has been renamed "Communications Networks, Content and Technology (CNECT)"). It is that department which is in charge of financially and otherwise supporting your projects.

Apart from this "*second-order*" research a "Research Program Officer" is engaged in, he or she has a number of more mundane, clerical, and bureaucratic tasks to attend to. As most of you probably know these include the preparation of Calls for Proposals, finding competent peers – not in cahoots with proposers – to assess and rank proposals, negotiating contracts (and associated work plans) with successful proposers, and last but not least, monitoring running projects and conducting periodic reviews. To the best of my knowledge, in the mid-term or perhaps even shortly, these tasks will be outsourced to an agency, especially set up for this purpose.

The meta-research part of our work usually boils down to short texts, called "Work Programmes" which loosely specify the content of research projects competing for European monies. This is why I like to refer to this part of our job as "*programming in the very large*".

But ever since Aristotle wrote his famous treatise on Metaphysics[2], so named because it was the book that came after his Physics, there is another customary meaning of the term "*meta*". It relates any subject to which it is prefixed to that which is or may be beyond that subject.

(We note in passing that there is at least one further use of "*meta*", as for instance in Metamathematics[3] and Metadata[4] ..., where it is formally the same thing X that is about X. However, this self-referential meaning of "*meta*" is of less concern in the present context.)

So we may, for the purpose of this talk, coin the term "meta-robotics". It is still missing on the very long list of "*metas*" on the respective Wikipedia page.[5]

Meta-robotics would probably comprise some of the issues that our meta-research addresses. For example the question: Why should the state fund robotics research and development (and not leave it to the market, the mantra of our times)?

[2] `http://en.wikipedia.org/wiki/Metaphysics_(Aristotle)`
[3] `http://en.wikipedia.org/wiki/Metamathematics`
[4] `http://en.wikipedia.org/wiki/Metadata`
[5] `http://en.wikipedia.org/wiki/Special:PrefixIndex/Meta`

But it comprises much more. There are obvious "*meta*"-questions: What are the potential consequences of this research? What impact will it have on our societies? On the economy, in the small and in the large? What impact will it have on us and our children and grand-children as individuals? And what about the dual use problematique as far as autonomous robots are concerned? Are there limits to what robots can or should do? Will there be a case for holding robots responsible for what they are doing? What about liability? And there are questions of a more journalistic and literary flavour, for instance: Will there be a case for treating robots as sentient beings, endowed with rights and to be treated with respect? Apparently taken seriously by many[6]. (Whether Karel Čapek took them seriously is an open question.)

You see, apart from doing (first-order) research with the aim of creating machines that operate – by virtue of their reasoning capabilities - autonomously and sensibly in the "Real World" a lot of human reason and reasoning may be called for in order to cope with the fruits of our joint and individual ingenuity.

Meta-research and meta-robotics demarcate the territories of this lecture. I will first briefly explain the why's and wherefore's of the funding programme you are benefitting from. I will then try to give you some idea of how this programme came about. Unfortunately, I cannot tell you much about its future as I have been, as mentioned before, since more than two years ago out of my office. Fortunately, this is likely to be a better position to speak on meta-robotics, in the final part of this talk.

Why research funding and for what?

Public funding of scientific research and technological development has a long history. With tongue in cheek we may say that it all started with Adam and Eve although they got severely punished as we know, by the higher powers-that-be for accepting funds from the devil. But of course we don't have to go that far back in time. For our purposes it may suffice to link the emergence of the idea of public S&T funding to the English philosopher and politician Francis Bacon who lived around the turn of the 16th to the 17th century. He too wrote a seminal text, entitled "*The New Atlantis*", describing a society that affords a publicly funded research facility called *Salomon's House* (also known as the *College of the Six Days Works*) "*where specially trained teams of investigators collect data, conduct experiments, and (most importantly from Bacon's point of view) apply the knowledge they gain to produce 'things of use and practice for man's life'* "[7].

[6] http://en.wikipedia.org/wiki/Intentional_stance

[7] http://www.iep.utm.edu/bacon/#SH2b

Adam Smith and his modern disciples had not yet been on our planet, so purely economic ends were not on Bacon's horizon. In fact, in the preface to his opus magnum *Instauratio Magna* he wrote: „*Lastly, I would address one general admonition to all; that they consider what are the true ends of knowledge, and that they seek it not either for pleasure of the mind, or for contention, or for superiority to others, or for profit, or fame, or power, or any of these inferior things; but for the benefit and use of life; and that they perfect and govern it in charity.*"[8]

Here Bacon may have wanted to counteract moods prevailing in Renaissance England. Yet he is usually credited with coining the famous adage "*knowledge is power*" ("*scientia potestas est*")[9]. He represents like no other that phase in European (and World) history when the giant wheel with the three spokes *political power – economic power – scientific/technical capacity* was set in motion.

We all know where this wheel led us to. In fact, robots may become the apogee of its path. So let us take a long leap forward to the years right after WW2 and meet a man who could be considered a modern successor of Francis Bacon's: Vannevar Bush[10], polymath, science policy advisor to US presidents (FD) Roosevelt and Truman, and administrator of the Manhattan Project that resulted in the first atomic bombs. In Summer 1945 he authored a report to the President under the heading "*Science the endless frontier*". In the letter of transmittal he wrote: „*Science offers a largely unexplored hinterland for the pioneer who has the tools for his task. The rewards of such exploration both for the Nation and the individual are great. Scientific progress is one essential key to our security as a nation, to our better health, to more jobs, to a higher standard of living, and to our cultural progress.*"[11]

In the core document he went on to suggest to set up a National Research Foundation that "*should develop and promote a national policy for scientific research and scientific education, should support basic research in nonprofit organizations, should develop scientific talent in American youth by means of scholarships and fellowships, and should by contract and otherwise support long-range research on military matters.*" The latter as we know, has in the meantime largely been taken over by DARPA, the funding agency of the US military. Vannevar Bush, by the way, also invented a hypothetical machine, called MEMEX[12], which somehow anticipated the later hypertext systems and thus the Worldwide Web.

From Vannevar Bush's proposal to our European RTD programmes, both national and on a European level, it is but a small step. Their rationale is not too different from what I just quoted. And indeed, the overarching objective of Eu-

[8] http://www.bartleby.com/39/20.html

[9] http://en.wikipedia.org/wiki/Scientia_potentia_est

[10] http://en.wikipedia.org/wiki/Vannevar_Bush

[11] http://www.nsf.gov/od/lpa/nsf50/vbush1945.htm

[12] http://en.wikipedia.org/wiki/Memex

ropean research funding (and presumably of public research funding anywhere in the world) is to boost economic growth through science-based innovation.

It had already been codified in the early treaties of the European Communities, most notably EURATOM. But it took until the early 80's of the 20th century before a full-fledged Europe-wide IT research programme was launched under the name ESPRIT[13]. This was partly in response to similar activities in the US and Japan. Since then we have the well-known successive multiannual "*Framework Programmes on Research and Technology Development (RTD)*" covering many areas of research and development. We are currently in the eighth cycle, somewhat less bureaucratically labeled "*Horizon 2020*"[14].

There are a number of general questions one may ask in relation to spending public money on RTD. First and foremost of course, there is the question "what is worth spending it for". Then: What balance should be kept between basic research, "applied" research, and systems development? What is the role of industry in publicly funded research? (After all, dishing out public monies to private companies could well be perceived as a market-distorting subsidy.)

Different answers to these questions have been given at different times. It would not make sense to go into all of them here and now. Only that much: There is a problem. At least from my perspective these answers – most importantly those given in terms of budget/resource allocation - have somehow emerged from more or less transparent discussions among elected (e.g., committees of the EU parliament) and non-elected bodies (e.g., departments, units of the European Commission). And the closer one gets to the bottom, to defining specific areas and specific issues that ought to be addressed, the less transparent it becomes – at least for non-specialists. The general problem I see is that of legitimacy: of who decides what taxpayers' money should be spent on and according to which criteria.

I believe this is a key problem if we accept that our modern societies are in so many ways shaped by science-based technologies. Given the complex interdependencies between science and society[15] it is a serious problem worth considering if we want to further our democratic ideals.[16] The conclusion may well be that whatever institutions and rules we invent in an attempt to democratise decision-making in complex societies there are limits that cannot be passed. (After all, no referenda have ever been held and no votes have been taken on whether or not we should drive automobiles, fly aeroplanes or use computers.) Given that political and economic players with vested interests can take advan-

[13] `http://cordis.europa.eu/esprit/home.html`

[14] `http://ec.europa.eu/programmes/horizon2020/`

[15] `http://ec.europa.eu/research/science-society`

[16] `http://ro.uow.edu.au/cgi/viewcontent.cgi?article=1793&context=lhapapers`

tage of those limits - a familiar key word in this context is "lobbying" - we ought to be aware of their existence and potential impact.

Fortunately, some awareness of the need for principles guiding the public funding of RTD exists at the highest level of the European Commission. In a 2011 keynote contribution to a special issue (on "*Robotics: War and Peace*") of the journal "*Philosophy and Technology*" Neelie Kroes[17], the Commissioner in charge of the "Digital Agenda" wrote: "*But some questions remain. We cannot and must not curb scientific curiosity but we should ask: are there general principles that might guide public funding of research and the use of its results beyond innovation and competitiveness?*"[18]

In her answer she quotes the famous German playwright Bertolt Brecht and at the same time reminds us of Francis Bacon's *New Atlantis* and *Instauratio Magna*: "*Bertolt Brecht, in 'The Life of Galilei', had the great scientist say: 'I maintain that the only goal of science is to alleviate the drudgery of human life.' Sound advice indeed! We will continue to fund research whose results help create better living conditions for everyone on this planet and research that helps us to better understand ourselves and the world we live in.*"

And she concludes: "*Both go hand in hand — and robots should take their fair share in this ICT landscape*"; prompting me to move closer towards the subject matter of this seminar, at least partly guided by her wisdom.

Why robotics?

Of course, robots have been around for a long time. First and foremost in science fiction stories. (They are still there!) But from the late 1960s onwards also at product assembly lines, in space and on battle fields, to name but a few environments. When I say "robot" I assume that we all have a similar image before our mind's eye: that of an electro-mechanical device, designed and built to help people do jobs that are physically strenuous, potentially dangerous, repetitive and tiring, or simply impossible to do without suitable technical support. To qualify as a robot the device can be stationary or mobile; if stationary it should be able to handle and/or transport physical objects, large or small, heavy or light, depending on the kind of service it is supposed to deliver.

Given the persistent trend in industry to reduce the amount of manual labour in manufacturing goods for example, and keeping in mind the most general objective of research funding, it is easy to see and justify why Robotics was put on the ICT agenda. The specific aims of this research should be equally clear. "Traditional" robots are often nothing but more or less sophisticated

[17] `http://en.wikipedia.org/wiki/Neelie_Kroes`

[18] `http://link.springer.com/journal/13347/24/3/page/1`

machine-tools operating according to preset rules in strictly controlled environments (like an assembly line). To make robots fit for tasks in, say, open environments where remote control is not feasible or desirable, they ought to be endowed with capabilities that we normally find in ourselves but also in animals. In order to sensibly "*perform movements, manipulation, navigation, etc. in a real, noisy and dynamically changing world*" on their own (i.e., autonomously, the *ReaRW task*!) a robot should be able to correctly interpret what is going on in that world (yes, animals can do that). In other words, it should be an exemplar of an artificial "*Cognitive System*"[19] whose reasoning is informed by real-world inputs and results in real-world action.

Enormous sums have been disbursed with the intent to approach this goal. And our European programmes have contributed a substantial share. While topics broadly related to Artificial Intelligence (AI) have been part and parcel of European research programmes ever since they were first launched in the 1980s, Cognitive Systems became prominent as a specific item on the research agenda only in the late 90s when, under the heading Cognitive Vision, a cluster of eight projects was launched in response to a growing demand for more powerful computer vision systems that were able to interpret what they saw and sensibly to act upon it.

From 2002 onwards, this line of funding has been extended to cover both, Cognitive Systems in general and Robotics. It has been firmly established as a key chapter of the 6th and 7th multiannual Framework Programmes (FP6 from 2002-2006 and FP7 from 2007-2013 respectively), and codified in a series of usually biannual Work Programmes that underly the regularly published Calls for Proposals. By the time I left my office the European Commission, under this chapter, had spent more than half a billion Euros on nearly 140 projects and ancillary activities in the areas at issue. [20]

Meta-research on robotics - drafting a robotics research agenda[21]

We had asked the meta-research questions I mentioned at the beginning of my talk, with the understanding that the "first-order" research out there was still far from delivering fully operational systems that would satisfy criteria such as robustness, versatility, reliability, adaptability and last but not least, autonomy

[19] `http://www.vernon.eu/euCognition/definitions.htm`

[20] `http://cordis.europa.eu/fp7/ict/programme/challenge2_en.html`

[21] This section draws on a previous paper of mine: Towards a Scientific Foundation for Engineering Cognitive Systems - A European Research Agenda, its Rationale and Perspectives; in: Biologically Inspired Cognitive Architectures, Volume 1, July 2012, Pages 82–91. (online `http://dx.doi.org/10.1016/j.bica.2012.04.002`, preprint at `http://www.cikon.de/Papers.html`)

(i.e., to be free from outside control). Hence the explicit aim of our programmes became

... to strengthen the scientific foundation for engineering artificial cognitive systems - i.e., artificial systems that perceive and (inter-)act, in accordance with a suitable understanding of their environment;

and, in doing so ...

... to foster technologies that enable a variety of applications involving interaction within "real world" environments pertaining to, for instance, robotics, assistive technologies, and multimodal man-machine interaction.

Among the latter, robotics has undisputedly always been a major focus and most project work is indeed centred on robotic platforms.

A more detailed but sufficiently "liberal" research agenda was developed and at intervals revised after consulting representatives of different disciplines, disciplines that were believed to make relevant contributions to strengthening said scientific foundation. (By the way, this is a case in point illustrating what I alluded to before, regarding "legitimacy".)

For instance, given that "cognition" is first and foremost occuring in the living world one might ask: *What (if anything) do we need to understand about cognition as a biological phenomenon in order to specify, design and build artificial cognitive systems*? In light of the fact that natural cognitive agents (as individuals or species) are (up until now) practically the only entities that are capable of learning through acting on or interacting with complex dynamic environments, it seems evident that the engineering of artificial cognitive systems can be informed by studying natural processes related to cognition and control, including the role of the physical substrates of these processes. So it seemed a good idea to seek input from biologists and in particular, neuroscientists.

On the other hand, aircraft engineers do not draw on ornithology in order to design and build aeroplanes. Ornithology is simply not part of their scientific foundation. Likewise, although mainstream Artificial Intelligence (AI) research was more impressed with man's unique symbolic reasoning and planning capabilities than (for instance) with his gut feelings it managed to yield many interesting and useful results. But little did it contribute to creating the kind of systems aimed at under our programme. (By the same token, modern aeroplanes and even drones do lack some of the most outstanding avian faculties.)

Yet this would certainly not justify excluding traditional and more recent AI disciplines, such as Statistical Learning[22]. So one of the characteristics of our programme was its openness to multi-disciplinarity, inviting computer scientists, engineers, neuroscientists, psychologists, ethologists, mathematicians and possibly more to advise us and to team up in big and not so big projects. It was

[22] `http://en.wikipedia.org/wiki/Statistical_learning_theory`

also entirely agnostic as far as paradigms (e.g., computationalism, connectionism, enactivism) and different approaches to modelling were concerned.

But what about the utilitarian objective? About innovation and new markets? About the famous industry question? No, it has not been neglected. After all, research with a view to supporting engineering must not be confined to an ivory tower. Rather, it should be motivated by and cater to real needs, in line with the strategic goals of (public) European research funding. The FP6 and FP7 Cognitive Systems and Robotics programmes were definitely hospitable to commercial partners providing relevant scenarios in areas such as industrial and service robotics in all sorts of environments, scenarios where methods and solutions could be tested and validated. And if my interpretation is correct of what I hear on the grapevine, industry is given a much bigger part in the current robotics programme (under FP8 = Horizon 2020), hopefully not to the detriment of solving the still unsolved fundamental problems inherent in the "*ReaRW task*".

And hopefully not to the detriment either, of just sheer curiosity, of the desire to understand. Indeed, there is this other side to doing research which has often contributed more to "innovation" than targeted multimillion Euro/Dollar/Yen projects. Moreover, robotics as a science does have the potential of making us better understand our own nature, what "makes us tick" in our worlds, and how we make our worlds. Almost four centuries ago the Italian philosopher (of science) Giambattista Vico[23], regarded by some as one of the early ancestors of modern (radical) constructivism, expressed this in three words: "*verum ipsum factum*", or: "*The criterion and rule of the true is to have made it*."

Commissioner Kroes, in her short note, acknowledges this potential of robotics when she writes (I repeat): "*We will continue to fund research whose results help create better living conditions for everyone on this planet and research that helps us to better understand ourselves and the world we live in.*" So there is hope. You should take her at her word.

Do we need to know how the mind works (to build the ultimate robot) - can we know it?

But we should not get carried away. No, I do not mean with our hope to get more money for feeding our curiosity. I mean: let us not be too optimistic as far as understanding the human condition and the human mind are concerned. Arguing against the Cartesian idea of certain truth as something as clear and distinct as a geometry theorem, Giambattista Vico insists that "*our clear and di-*

[23] http://plato.stanford.edu/entries/vico/

stinct idea of the mind cannot be a criterion of the mind itself, still less of other truths. For while the mind perceives itself, it does not make itself."

Can the human mind make itself? Some people (e.g., those known as "transhumanists"[24] and "extropians"[25]) believe the answer is "yes" and postulate a future when human beings can achieve at least mental immortality (catchword: mind upload[26], a modern form of the dualist belief in an immortal soul). Some dream of phantastic scenarios where robots spread the intelligence evolved on our planet Earth to distant worlds in outer space, thus "conquering the universe". Some people seem to see no limits in what nature (of which we are part) can do. Others may dread a future when humans, as in Karol Čapek's play, become obsolete and are supplanted by their own superior creations. Some of you may remember an article published in Wired in early 2000, by Bill Joy, co-founder of Sun Microsystems, entitled "*The future does not need us*"[27], where he gives words to his concern about a somewhat casual view of some "visionaries" who made up a rather gloomy fate for mankind, apparently based on a very peculiar understanding (some may wish to call it misunderstanding) of what it means to be human.

I find such musings rather amusing. Indeed, if we make a mind it will not be in a human body and hence not be a human mind (unless we do it the traditional way that was invented by nature long before we could have had a say in it). Depending on how narrow or broad we take the concept of "mind" to be we may even say that we have already been creating minds galore; minds in different bodies for sure, but minds that greatly surpass our own, as far as "Algorithmic Intelligence" (another AI!) is concerned - but not more. (For example, just behold the laptop computer in front of you.) The "super-humans" are already there but of course they are not human. In fact they are about as super-human as a tractor is super-equus. Their minds are mere "shadows of our minds" (to recall the title of a 1994 book[28] by Roger Penrose[29], but without endorsing his ideas on quantum consciousness).

In this context it is interesting to note that one of the biggest and most expensive European Projects under "Horizon2020", the "Human Brain Project (HBP)"[30], a so called FET "Flagship"[31], is presently (July 2014) causing a ma-

[24] http://en.wikipedia.org/wiki/Transhumanism

[25] http://en.wikipedia.org/wiki/Extropianism

[26] http://en.wikipedia.org/wiki/Mind_uploading

[27] http://archive.wired.com/wired/archive/8.04/joy.html

[28] http://en.wikipedia.org/wiki/Shadows_of_the_Mind

[29] http://en.wikipedia.org/wiki/Roger_Penrose

[30] https://www.humanbrainproject.eu

[31] http://ec.europa.eu/digital-agenda/en/fet-flagships

jor controversy mainly among neuroscientists[32]. The HBP sets out to simulate the anatomy and physiology of (parts of) the human brain. Its "raison d'être" is (at least) twofold. Firstly, the expectation that this simulation will provide insights into the workings of real brains and thus helps to study brain diseases and to find pertinent remedies. Secondly, the intention to study "neural computation" more closely in order to create effective and more efficient neuromorphic hardware implementations of it.

Many neuroscientists now fear that the second rationale is overly gaining in weight. From my perspective, this is not surprising given that the HBP is funded under a technology and not a biology programme. Designing neuromorphic hardware[33] is certainly a laudable endeavour. But it seems to me that there is a hidden assumption, nurtured by the extropian claims – unproven - that a brain is fully simulable and that its mappings are Turing-realisable, and hence at least in principle replicable through technical artefacts.

One may not have to go as far as Roger Penrose and postulate non-deterministic quantum processes in microstructures of the brain[34], to be more than sceptical about these claims. Whether the mappings effectuated by brains are Turing-realisable is, to the best of my knowledge, simply an open question. (Here we may note in passing that the mappings effectuated by the members of a certain class of artificial neural networks – Analogue Recurrent Neural Networks - are provably super-Turing, a result obtained some 20 years ago by Hava Siegelmann.[35])

And as Anil Seth[36], computational Neuroscientist at the University of Sussex, in a recent (8 July 2014) op-ed article in The Guardian[37], points out: even if more detailed simulations of the brain could be achieved this would "*not inevitably lead to better understanding. Strikingly, we don't fully understand the brain of the tiny worm Caenorhabditis elegans even though it has only 302 neurons and the wiring diagram is known exactly. A perfectly accurate model of the brain may become as difficult to understand as the brain itself, as Jorge Luis Borges long ago noted when describing the tragic uselessness of the perfectly detailed map*." "Understanding", in this context, presumably means being able to falsifiably hypothesise links bet-

[32] http://www.neurofuture.eu/

[33] https://www.humanbrainproject.eu/de/neuromorphic-computing-platform
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3812737/
http://apt.cs.manchester.ac.uk/projects/SpiNNaker/

[34] http://www.quantumconsciousness.org/penrose-hameroff/quantumcomputation.html

[35] http://binds.cs.umass.edu/anna_cp.html
http://binds.cs.umass.edu/papers/1995_Siegelmann_Science.pdf

[36] http://www.sussex.ac.uk/Users/anils/index.html

[37] http://www.theguardian.com/commentisfree/2014/jul/08/human-brain-project-missed-opportunity-simulating-neuron-activity

ween brain structures, functions and processes on the one hand, and observable behaviour on the other hand.

One also has to bear in mind the limitations inherent in models, regardless of whether digital or analogue, of non-manmade natural phenomena. In fact, discovering limits is sometimes more rewarding than assuming there are none and reaching one dead end after the other. Limits have been discovered in Metamathematics a long time ago, for instance to what the most paradigmatic computational model, the Turing Machine, can do[38]. Physics sets hard limits to what we can do given our and the rest of nature's nature. Of course one may ask: can the world be completely specified in formal, mathematical terms?[39] Again, some may believe the answer is "yes" and may even go one step further, to believing that emulating natural phenomena can fully capture the essence of these phenomena. But we know: simulations and emulations are always based on models which at best are homomorphic, but not isomorphic, images of the real thing.

(This may seem trivial but is often forgotten or ignored. It applies, by the way, also to social interaction between people. Which includes economics, a vast field of social interaction where it is perhaps most often forgotten. Instead there may be a tendency there to adapt the real thing to whatever model is *en vogue*.)

Meta-robotics - ethics

Let us get back down to earth, back from the lofty heights of brains, singularities and flagships, to the lowlands of the electro-mechanical devices called robots. Here is another verbatim quote from Mrs. Kroes's keynote commentary:

"Take for instance the concept of an autonomous machine. This could be a self-controlling road vehicle, which may become a reality sooner rather than later given the current speed of technological advancement. There are also various examples of military autonomous vehicles operating on land, at sea or in the air. Who is responsible for their actions? Who is liable in case of damage? Can it be considered that such machines operate on their own accord? The answer is a firm 'no' . Machines are designed, built and programmed so that they can render services. They are always owned and controlled by people. Machines — no matter how sophisticated — are as 'ethical' as the people who design, build, programme and use them. We humans, jointly and individually, have to take full responsibility for what we are doing, good or bad, constructive or destructive, through our own inventions and creations, to each other and our world at large."

[38] http://en.wikipedia.org/wiki/Halting_problem
[39] http://www.idsia.ch/~pape/papers/pape2011agilong.pdf

What our Commissioner addresses here are clearly issues pertaining to meta-robotics. What use should robots be put to, who is responsible for what they are doing, and what implications does using them have for the life of individuals, groups and entire societies? Questions that also underly a whole new scholarly debate on "*Robot-Ethics*". In fact Robot-Ethics has been the dominant theme of an EU supported action called ETHICBOTS[40]. And in October 2013 euCognition, another EU supported network of researchers interested in Cognitive Systems, organised a meeting solely dedicated to "*Social and Ethical Aspects of Cognitive Systems*"[41] including of course, robotics.

The Commissioner strongly denies endowing machines with any kind of responsibility. I can only agree. I would make it even more explicit and submit that man-made machines are categorically different from natural living, feeling, and thinking beings. The more we fancy machines to be human-like, ascribing them intentions, desires and beliefs (c.f., Dennett's intentional stance), the higher the risk of us becoming machine-like ourselves. The more we rely on machines to make decisions that only we can justifiably make, the more we deprive ourselves of our authority, independence and our essential human characteristics. Man-made machines – no matter how sophisticated - have no rights and should not be feared; we can switch them off, take them off line or, ultimately, dismantle them. (Joanna Bryson, University of Bath, in her Essay "*Robots should be slaves*", takes a very similar if not identical view.[42])

By the way, the danger inherent in relying on machines to make decisions in our stead not only concerns robots but technical systems in general. For instance, every bureaucrat knows how easy and convenient it is to hide his or her own incompetence, insecurity or ignorance behind the veil of whatever computerised workflow or transaction systems may have been imposed on him or her.

So what should robots be used for? Of course, they should do all the nice things that proposers of EU research projects like to put forward in order to justify, from a utilitarian perspective, the need for better machine vision, better robot navigation, better object manipulation, more autonomy, et cetera. Again Neelie Kroes: "*The ease of use, safety, and partial autonomy are essential if robotic devices are to leave the shop floor and strictly controlled environments and become truly useful and helpful for people, including those with special needs. This could include steering a wheelchair, driving a car, guiding a blind person, performing precision surgery, operating a leg amputee's prosthesis, or many of our everyday chores.*"

[40] http://ethicbots.na.infn.it/

[41] http://www.eucognition.org/index.php?page=2013-fourth-eucogiii-members-conference-gen-info

[42] http://www.cs.bath.ac.uk/~jjb/ftp/Bryson-Slaves-Book09.html

But she also pointed out that there are people who want your research to inform the engineering of devices that could - for example - replace a soldier on the battlefields of our times and thus make destructive and lethal military action (including full-fledged war) (even) more "acceptable". She did not challenge this kind of use, probably for good reasons of her own (being a member of the "political class"). And there are indeed many who take this kind of use very seriously. So seriously that they devote a considerable amount of effort to researching the possibility of making such battlefield robots "ethical", for instance by having them respect the rules of combat or, what is known as "*ius in bello*"[43]. A good reason for robot reasoning? Perhaps. I for my part believe the prospect of this kind of *dual use* is an even better reason for *human* reasoning, for thinking harder about "*ius ad bellum*", and the reasons for waging war in the first place.[44]

Meta-robotics - economics

Let us return to the traditional mainstay of robotics, to the assembly lines and shop floors. Here it is quite obvious what robots should do: free human workers from hard labour. Headlines such as "*The next generation of robotic assembly lines are emerging*"[45] have appeared only recently, and this is happening not least thanks to the kind of research you are doing or going to do. Foxconn, arguably the world's largest manufacturer of electronic devices, announced only this month (July 2014) to install 10000 "Foxbots" in its new iPhone6 plant, replacing thousands of workers and at the same time greatly increasing the factory output[46]. And Google, apart from its autonomous car venture has embarked on full-fledged robotics through the acquisition of various robot companies[47] including, by the way, Boston Dynamics[48], famous for its BigDog walking robots and its millions of DARPA R&D monies.

Freeing human workers from hard work, well, that is good news. But here again: questions remain. As I mentioned before, industry mechanisation and automation has been going on ever since industrialisation began. It led in some parts of our planet to unprecedented wealth and a growing "service economy"[49]. However, in many traditional industries managers and owners found it

[43] http://www.cc.gatech.edu/ai/robot-lab/online-publications/formalizationv35.pdf

[44] http://www.youtube.com/watch?v=XNpfeLhMT_Q
http://web.stanford.edu/~jacksonm/war-overview.pdf

[45] http://gigaom.com/2014/02/11/the-next-generation-of-robotic-assembly-lines-are-emerging/

[46] http://fortune.com/2014/07/07/apple-foxbot-iphone-6/

[47] http://www.popsci.com/article/technology/why-google-building-robot-army

[48] http://www.bostondynamics.com/

[49] http://en.wikipedia.org/wiki/Service_economy

more advantageous to employ human labour in parts of the world that had not previously benefitted from the blessings of industrialisation. This has over the last decades led to a veritable process of de-industrialisation[50] in Europe and North America.

Can innovation in robotics reverse this trend? Will robot-based re-industrialisation[51] create new jobs for ”the masses”? Can the promise of more jobs through science-based innovation really be kept? Can we keep extrapolating from the past and trust that human labour taken over by robots will find new niches in other domains? Services again? More services? Different services? There will most likely be more jobs for highly qualified people like you.

However, given that service robotics has also become a strong RTD focus[52] there is already a force at work that in a way defeats the assumption that jobs lost in manufacturing will be made up for by expanding services. Recent announcements by Amazon for instance, to experiment with drones for the delivery of parcels to its clients[53], foreshadow future developments. And if we are to believe the proponents of service robotics even jobs in hospitals and old people's homes will be in jeopardy. A frightening, comforting, or amusing prospect, depending on one's point of view.

Would we really want to replace a human carer by a machine, if such an option were available? Or should we not use these machines to complete other tasks or roles and give people more time to care for and help each other?

Perhaps the job question should be rephrased: Can robotics, the current apogee of industry automation (see above), contribute to making our economies more effective and more equitable in terms of providing the means for everyone on this planet to lead a life in dignity and peace? I have grave doubts that this is possible given the present constitution of our economies, their underlying power structures.

What will happen to all those whose money will then be made by robots? After all, robots being owned by few (not by the workers!), are not likely to share the money they make with the workers they are laying off. The money they make belongs to their owners unless we change the law. They do not buy goods either, to keep the economy running[54]. Like all machinery before them they are

[50] http://en.wikipedia.org/wiki/Deindustrialization

[51] http://ec.europa.eu/commission_2010-2014/tajani/priorities/reindustrialisation/index_en.htm

http://ec.europa.eu/enterprise/policies/industrial-competitiveness/competitiveness-analysis/european-competitiveness-report/files/eu-2013-eur-comp-rep_en.pdf

[52] http://www.springer.com/engineering/robotics/journal/11370

[53] http://www.amazon.com/b?node=8037720011

[54] http://www.vqronline.org/essay/machines-dont-buy-goods

likely to increase the gap between the haves and the have-nots (as documented in the recently published bestseller "Capital in the 21st Century"[55] by Thomas Piketty[56])?

So, do we have to rephrase our question yet again? For instance: must our economies be restructured in such a way as to harness the enormous increase in productivity that is likely to be brought about by robots and robotic devices, for the common good and the benefit of all?[57] A big question and not an easy one to answer given the well-known doomed (for whatever reason) approaches to socio-economic reform that in various parts of the world have been taken since the days of the European Enlightenment.

The philosophers of the Enlightenment taught us that we can be the masters of our own fate, individually and collectively. And scientists and engineers know that they can provide the means to change the world to our advantage. It is up to all of us but in particular to our elected representatives and rulers, sometimes referred to as the "political class", appropriately to respond to the challenges - positive and negative - posed by new technologies and insights, and to adapt the political and socio-economic structures accordingly. The law is made by law makers and law makers can change the law. A great challenge to HUMAN REASON in a world changing through human reasoning - called science.

Be aware of the bigger picture

Well, today all these issues are not of your immediate concern. So I am not going to keep you much longer from doing what you came here to do: to learn how to make robots reason in the real world. But I do believe that everyone, but in particular scientists and engineers, should be aware of the bigger picture of what they are working on professionally, its contexts and ramifications. There are many more or less prominent role-models in this regard.

One of them is Joseph Weizenbaum[58]. When he was 13 years old he escaped with his parents from Nazi-Germany to the United States of America. There he eventually became a professor of Computer Science at MIT. Later in his life he returned to Berlin where he died in 2008, aged 85. Many computer scientists

---

http://www.theatlantic.com/business/archive/2011/10/why-workers-are-losing-the-war-against-machines/247278/

[55]http://www.lrb.co.uk/v36/n13/benjamin-kunkel/paupers-and-richlings

[56]http://piketty.pse.ens.fr/en

[57]http://www.cepr.net/index.php/blogs/beat-the-press/robots-dont-cost-jobs-bad-economic-policy-does

[58]http://en.wikipedia.org/wiki/Joseph_Weizenbaum

remember his ELIZA[59] program, written in the early sixties, that could mimick inter alia the conversational patterns of a so called Rogerian psychiatrist. What mainly qualifies him to be named in the present context is a book he published some forty years ago: *Computer Power and Human Reason - from Judgement to Calculation*[60]. What motivated him to write it was the fact that many people took his ELIZA program seriously, suggesting for instance that it could make up for the shortage of psychiatrists. (Today we can observe something similar around IBM's "Jeopardy"-winning WATSON[61], a greatly beefed-up version perhaps of ELIZA.) Much of what people are discussing today under the heading "*Robot ethics*" can be found in this book. There are of course new issues some of which I tried to point out. So maybe one day someone authors a sequel to Weizenbaum's treatise which might then bear the title "*Robot Power and Human Reason*". The subtitle may even remain unchanged.

The second name I would like to mention is Noam Chomsky's[62]: he is five years Weizenbaum's junior and hardly needs an introduction. Given the scope of publications[63], ranging from "*Syntactic Structures*" to "*Profit over People*" and more recently, "*On Western Terrorism: From Hiroshima to Drone Warfare*", I need not explain either why I am shortlisting him. He has become the proverbial public intellectual.

I conclude this list with Nikola Tesla[64], a name you may have heard before (perhaps in your physics class in high-school). Apart from his many inventions, Tesla's main claim to fame is perhaps his part in designing the electric power grid in the United States of America, in the late 19th century. In a way he did for the electricity networks then what Vinton Cerf[65] and his colleagues did for the information networks in the second half of the 20th century.

Tesla was a somewhat colourful character with a tendency to bragging and presenting himself as a kind of star. But he is also widely considered a father (if not the father) of the robot as a technical device. Thus he provides the ideal closing bracket to this lecture where Karel Čapek provided the opening one. In his own way Tesla was a polymath greatly interested in explaining to the general public the potential of the technologies of his time, a bit like a futurologist and science fiction writer. In 1935 he published a short article entitled "*A Machine*

---

[59] http://en.wikipedia.org/wiki/ELIZA

[60] http://en.wikipedia.org/wiki/Computer_Power_and_Human_Reason

[61] http://en.wikipedia.org/wiki/Watson_(computer)

[62] http://en.wikipedia.org/wiki/Noam_Chomsky
http://www.chomsky.info/

[63] http://en.wikipedia.org/wiki/Noam_Chomsky_bibliography

[64] http://en.wikipedia.org/wiki/Nikola_Tesla

[65] http://internethalloffame.org/inductees/vint-cerf

to End War"[66] in a popular magazine. Unfortunately that machine has not yet been realised. And he made other predictions which nowadays appear rather outlandish. But some of his predictions may be more realistic. It is still upon you to make them come true. One of them is right at the end of his article:

"Today the robot is an accepted fact, but the principle has not been pushed far enough. In the twenty-first century the robot will take the place which slave labor occupied in ancient civilization. There is no reason at all why most of this should not come to pass in less than a century, freeing mankind to pursue its higher aspirations."

What aspirations? Whose aspirations? Chomsky's? Weizenbaum's? Or whose? Given that this text was written in 1935 you still have 20 years to find out. Good luck and thanks for listening.

[66] `http://www.tfcbooks.com/tesla/1935-02-00.htm`

Meta-Research and Meta-Robotics

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Hans-Georg Stork

(ex-European Commission,
DG INFSO/CNECT)

Avant-propos

Introduction

Why research funding and for what?

Why robotics?

Meta-research on robotics -
drafting a robotics research agenda

Do we need to know how the mind works (to build the ultimate robot) - can we know it?

Meta-robotics – ethics

Meta-robotics – economics

Be aware of the bigger picture

Karel Čapek (1890-1938)

Avant-propos

Introduction

Why research funding and for what?

Why robotics?

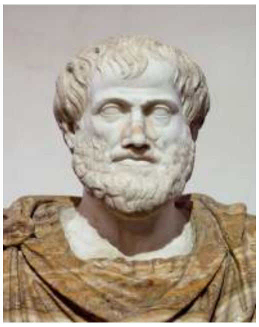
Meta-research on robotics -
drafting a robotics research agenda

Do we need to know how the mind works (to build the ultimate robot) - can we know it?

Meta-robotics – ethics

Meta-robotics – economics

Be aware of the bigger picture



Aristotle ((-384) - (-322))
Author of Metaphysics

Avant-propos

Introduction

Why research funding and for what?

Why robotics?

Meta-research on robotics -
drafting a robotics research agenda

Do we need to know how the mind works (to build the ultimate robot) - can we know it?

Meta-robotics – ethics

Meta-robotics – economics

Be aware of the bigger picture

Adam&Eve (~4000 BC)

Francis Bacon (1561-1626)

Adam Smith (1723-1790)

Vannevar Bush (1890-1974)

Neelie Kroes (1941-)

Bertolt Brecht
(1998-1956)

Galileo Galilei (1564-1642)

Avant-propos

Introduction

Why research funding and for what?

Why robotics?

Meta-research on robotics -
drafting a robotics research agenda

Do we need to know how the mind works (to build the ultimate robot) - can we know it?

Meta-robotics – ethics

Meta-robotics – economics

Be aware of the bigger picture

Avant-propos

Introduction

Why research funding and for what?

Why robotics?

Meta-research on robotics - drafting a robotics research agenda

Do we need to know how the mind works (to build the ultimate robot) - can we know it?

Meta-robotics – ethics

Meta-robotics – economics

Be aware of the bigger picture

Giambattista Vico (1668-1744)

Avant-propos

Introduction

Why research funding and for what?

Why robotics?

Meta-research on robotics -
drafting a robotics research agenda

Do we need to know how the mind works (to build the ultimate robot) - can we know it?

Meta-robotics – ethics

Meta-robotics – economics

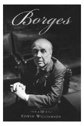
Be aware of the bigger picture

Bill Joy (1954-)

Roger Penrose (1931-)

Hava Siegelmann(??-)

Anil Seth (1972-)



Jorge Luis Borges
(1899-1986)

Avant-propos

Introduction

Why research funding and for what?

Why robotics?

Meta-research on robotics -
drafting a robotics research agenda

Do we need to know how the mind works (to build the ultimate robot) - can we know it?

Meta-robotics – ethics

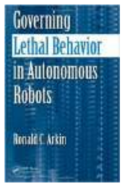
Meta-robotics – economics

Be aware of the bigger picture

Neelie Kroes (1941-)

Daniel Dennett (1942-)

Joanna Bryson (??-)



Avant-propos

Introduction

Why research funding and for what?

Why robotics?

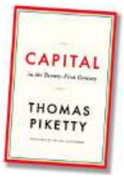
Meta-research on robotics -
drafting a robotics research agenda

Do we need to know how the mind works (to build the ultimate robot) - can we know it?

Meta-robotics – ethics

Meta-robotics – economics

Be aware of the bigger picture



Avant-propos

Introduction

Why research funding and for what?

Why robotics?

Meta-research on robotics -
drafting a robotics research agenda

Do we need to know how the mind works (to build the ultimate robot) - can we know it?

Meta-robotics – ethics

Meta-robotics – economics

Be aware of the bigger picture

Joseph Weizenbaum (1923-2008)

Noam Chomsky (1928-)

Nikola Tesla (1856-1943)

Thanks for listening

(Transcript: http://www.cikon.de/Text_EN/PragTalk.pdf)

Cognitive Robotics

David Vernon
University of Skövde

www.vernon.eu

Reasoning in the Robot World Summer School 2014
Czech Technical University in Prague
Center for Machine Perception
29 – 31 July 2014

Reasoning in the Robot World Summer School 2014
Czech Technical University in Prague
Center for Machine Perception
29 – 31 July 2014

Cognitive Robotics

Talk 1

- The link between Robotics and Cognitive Systems
- Capabilities of a cognitive system
- Different approaches to modelling cognition
- Characterization of Cognitive Robotics

Cognitive Robotics

Talk 2

- Cognitive architectures
  - Soar
  - ISAC
  - Clarion
  - …

Cognitive Robotics

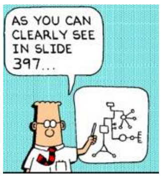
Talk 3

- Aspects of a cognitive architecture
  - Memory & Prospection
  - Knowledge & Representation

Talk 1 ➔ Talk 2 ➔ Talk 3
Progressive Deepening

"Extreme Tutorial"

HEALTH WARNING



HEALTH WARNING

***No** videos,*
Few images,
What are we waiting for!

***No equations either**!*

COGNITIVE ROBOTICS

WHAT WHY HOW

Non-Cognitive Robotics

http://9gag.com/gag/ajrowGR?ref=9g.m

Non-Cognitive Robotics

http://9gag.com/gag/ajrowGR?ref=9g.m

Cognitive Robotics

Talk 1

– The link between Robotics and Cognitive Systems

– Capabilities of a cognitive system

– Different approaches to modelling cognition

– Characterization of Cognitive Robotics

Industrial Robots

Service Robots

Robotics

ISO 8373 definition of service robot

A robot is an actuated mechanism programmable in two or more axes with **a degree of autonomy**, moving within its environment, to perform intended tasks

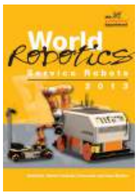
– Autonomy in this context means the ability to perform intended tasks based on current state and sensing, without human intervention

Robotics

- A **service robot** is a robot that performs useful tasks for humans or equipment excluding industrial automation application
- A **personal** service robot or a service robot for personal use is a service *robot* used for a non-commercial task, usually by lay persons
  - Examples are domestic servant robot, automated wheelchair, and personal mobility assist robot
- A **professional** service robot or a service robot for professional use is a service robot used for a commercial task, usually operated by a properly trained operator
  - Examples are cleaning robot for public places, delivery robot in offices or hospitals, fire-fighting robot, rehabilitation robot and surgery robot in hospitals

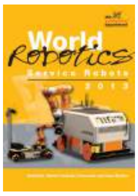
Robotics

- Service robots for personal/domestic use
  - Robots for domestic tasks
    - Robot companions/assistants/humanoids
    - Vacuuming, floor cleaning
    - Lawn mowing
    - Pool cleaning
    - Window cleaning
    - Others
  - Entertainment robots
    - Toy/hobby robots
    - Multimedia/remote presence
    - Education and research
    - Others
  - Elderly and handicap assistance
    - Robotized wheelchairs
    - Personal aids and assistive devices
    - Other assistance functions
  - Personal transportation (AGV for persons)
  - Home security & surveillance
  - Other Personal / domestic robots



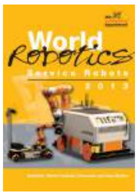
Robotics

- Service robots for professional use
  - Field robotics Agriculture
    - Milking robots
    - other robots for livestock farming* Forestry and silviculture
    - Mining robots
    - Space robots
    - Others
  - Professional cleaning Floor cleaning
    - Window and wall cleaning (including wall climbing robots)
    - Tank, tube and pipe cleaning
    - Hull cleaning (aircraft, vehicles, etc.)
    - Other cleaning tasks
  - Inspection and maintenance systems
    - Facilities, plants
    - Tank, tubes, pipes and sewers*
    - Other inspection and maintenance systems
  - Construction and demolition
    - Nuclear demolition & dismantling
    - Building construction
    - Robots for heavy/civil construction
    - Other construction and demolition systems



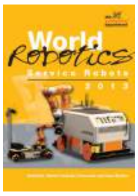
Robotics

- Service robots for professional use
  - Logistic systems
    - Automated guided (AGV) vehicles in manufacturing environments
    - AGVs in non-manufacturing environments (indoor)
    - Cargo handling, outdoor logistics
    - Other logistics
  - Medical robotics
    - Diagnostic systems
    - Robot assisted surgery or therapy
    - Rehabilitation systems
    - Other medical robots
  - Rescue & security applications
    - Fire and disaster fighting robots
    - Surveillance/security robots
    - other rescue and security robots

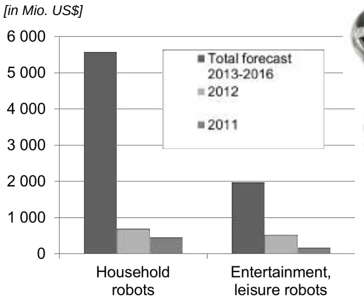


Robotics

- Service robots for professional use
  - Defense applications
    - Demining robots
    - Unmanned aerial vehicles
    - Unmanned ground based vehicles (e.g. bomb fighting)
    - other defense applications
  - Underwater systems
  - Powered Human Exoskeletons
  - Mobile Platforms in general use
  - Public relation robots and joy rides
    - Hotel and restaurant robots
    - Mobile guidance, information robots Robots in marketing
    - Robot joy rides



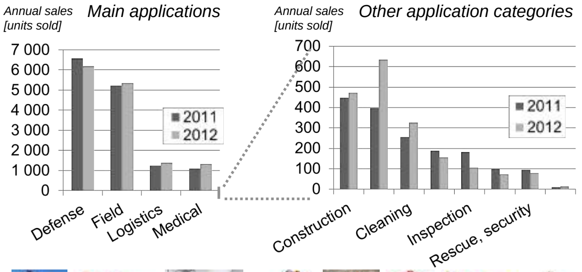
Sales: Personal/Domestic Robots



Photos:
Samsung
neato xv
Aldebaran NAO
Festo Robotino
Friendly Robotics

Martin Hägele, Fraunhofer IPA & RockEU Robotics Coordination Action

Sales: Service Robots for Professional Use



Courtesy: Lely, Siemens, Bluebotics, Husqvarna, Cleanfix, Schilling, Vecna Robotics
World Robotics Report; www.worldrobotics.org

Martin Hägele, Fraunhofer IPA & RockEU Robotics Coordination Action

Robotics

- According to the definition, “*a degree of autonomy*” is required for service robots
  - ranging from partial autonomy (including human robot interaction)
  - to full autonomy (without active human robot intervention)

Motivation for Studying Cognitive Systems

Motivation

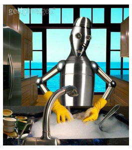
- Controlled environment

  We know what to expect,
  and can program the robot
  to do what we want

- Complex environment

  We don’t know what to expect,
  and the robot has to be
  flexible and adaptable

  Uncertainty, incomplete knowledge, change



Motivation

- Controlled environment

  We know what to expect,
  and can program the robot
  to do what we want

- Complex environment

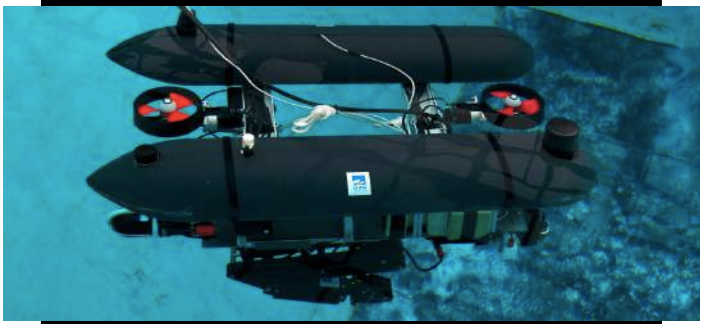
  We don’t know what to expect,
  and the robot has to be
  flexible and adaptable

  Uncertainty, incomplete knowledge, change

c.f. Maria Petrou’s Ironing challenge; see http://www.commsp.ee.ic.ac.uk/~mcpetrou/iron.html

http://sparc-robotics.eu/

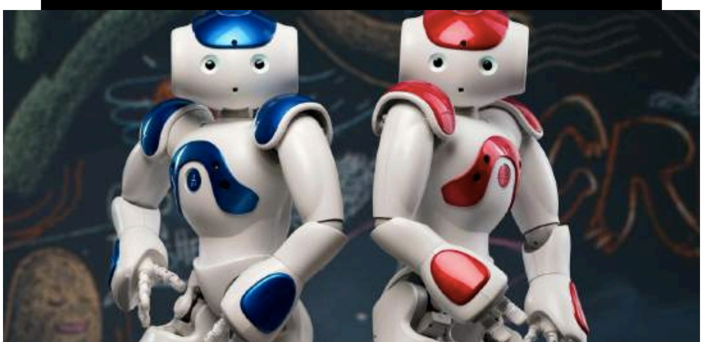
http://sparc-robotics.eu/



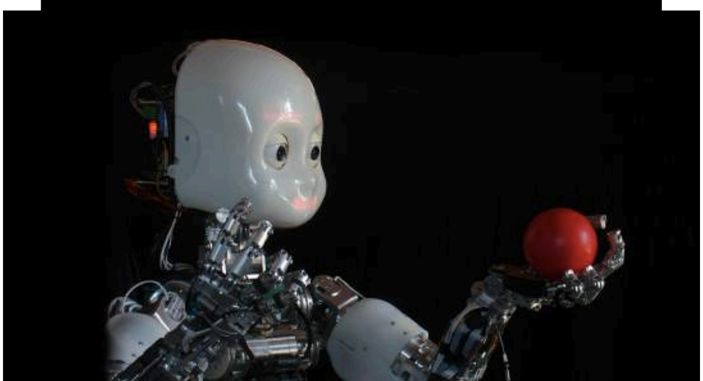
http://sparc-robotics.eu/

http://sparc-robotics.eu/

http://sparc-robotics.eu/

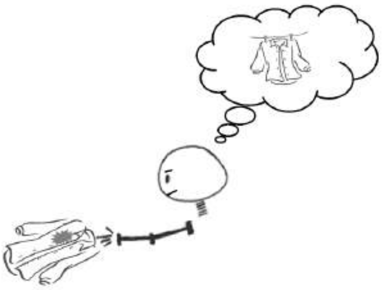


http://sparc-robotics.eu/

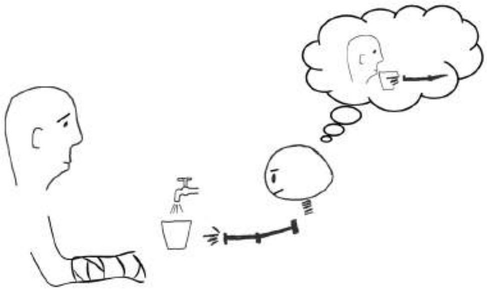


http://sparc-robotics.eu/

Motivation



Motivation

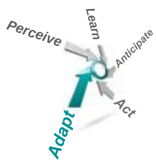


Definitions:

What is cognition, anyway?

Cognitive System

"A **cognitive system** is
an **autonomous** system that can
perceive its environment,
learn from experience,
anticipate the outcome of events,
act to pursue goals, and
adapt to changing circumstances."



Vernon, D. "Cognitive System", in Computer Vision: A Reference Guide, Springer, pp. 100-106, (2014).

Something that understands what's going on around it
and acts in an appropriate way

Understanding: Perceiving and Acting…

Incomplete information
Changing circumstances

How do you get this information?
How do you fill in the gaps?
How will it change?

What do you do with it?
How do you act?

A system that constantly predicts the need for actions

EFFECTIVE ACTION

Breaking free of the present
(and the limitations of perception)

Cognition is Prospective

Breaking free from pre-programmed knowledge

Drinking from the fire-hose of experience

Cognition: guide actions

- Missing information
- Uncertain information
- Changing information
- LATE information

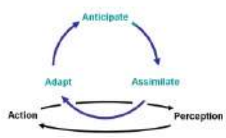
- Adapt (make sense of the world)
- Anticipate (predict what might happen)

Cognitive systems

- Anticipate
- Assimilate
- Adapt

Learn & develop

- Predict future events when deciding how to act
- Learn from what actually happens
- Modify subsequent predictions
- Autonomously



1. Cognition is the process by which an autonomous self-governing agent acts effectively in the world in which it is embedded

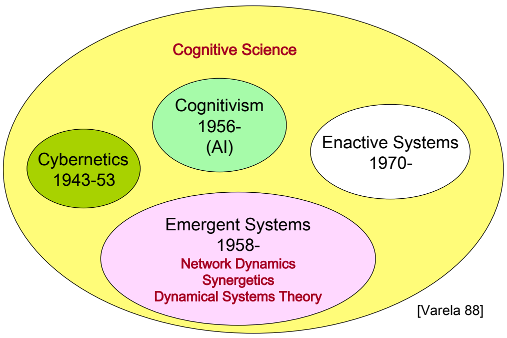
2. Dual purpose:

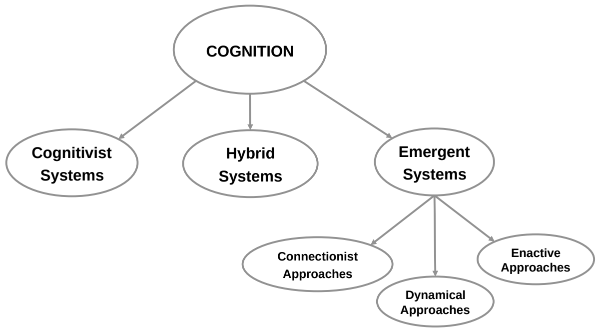
(a) Increase agent's repertoire of effective actions

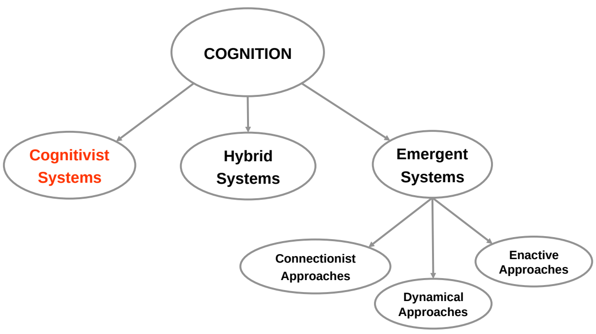
(b) Increase agent's power to anticipate the need for future actions and their outcomes

3. Development plays an essential role in the realization of cognitive capabilities

- What makes an action the right one to choose?
- What type of behaviour does cognition enable?
- What motivates cognition?
- How is perception guided?
- How are actions selected?
- What makes cognition possible?
- Cognitive skills can improve, but what do you need to get started?
- What drives the developmental process?







'The world we perceive is isomorphic with our perceptions of it as a geometric environment'

[Shepard & Hurwitz '84]

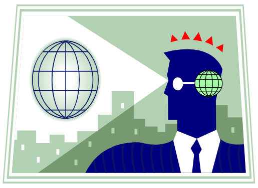
'Cognition *is* a type of computation'

'People "instantiate" … representations physically as cognitive codes and that their behaviour is a causal consequence of operations carried out on these codes'

[Pylyshyn '84]

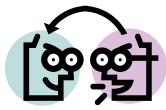
Cognitivist Systems

- Strong representations
  - Explicit & symbolic
  - Representations *denote* external objects
  - Isomorphic
  - Implies an absolute and accessible ontology
  - That is consistent with human expression …



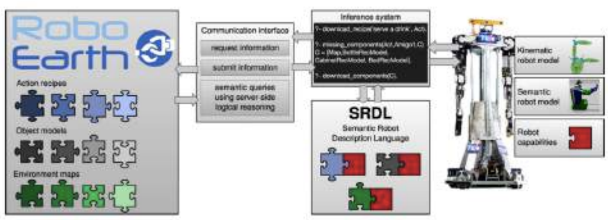
Cognitivist Systems

- Representations
  - Descriptive product of a human designer
  - Can be directly accessed & understood by humans
  - Human knowledge can be directly injected into an artificial cognitive system

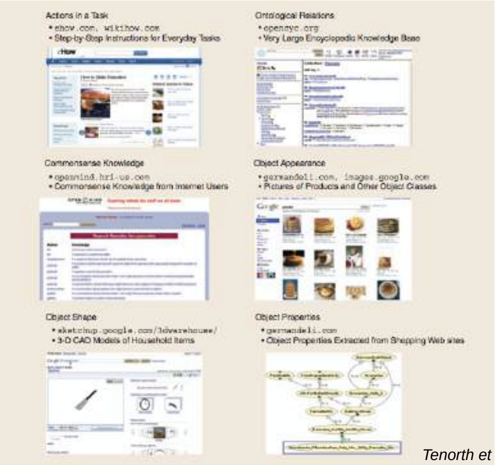


Cognitivist Systems

- Representations
  - Descriptive product of a human designer
  - Can be directly accessed & understood by humans
  - Human knowledge can be directly injected into an artificial cognitive system



Tenorth et al. 2013



Tenorth et al. 2011

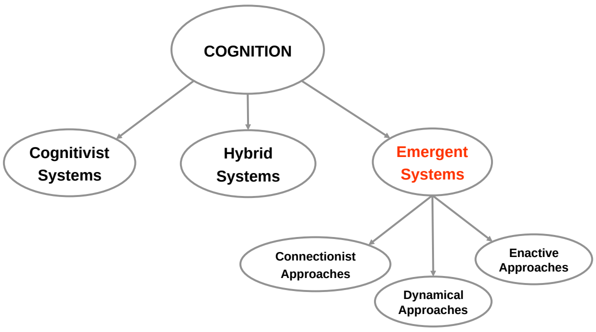
Cognitivist Systems

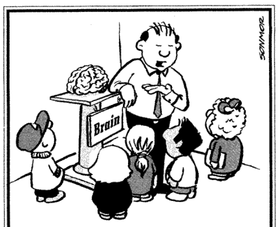
- But … Programmer-dependent representations bias the system

  ‘blind’ the system (cf. Winograd & Flores)

- Can you anticipate every eventuality in your design?

- The Symbol Grounding problem: how do symbols acquire meaning?





“...the most complex arrangement of matter in the universe...produced without the least bit of intelligence.”

Self-generated AI:
AI by orchestrating the processes that generate it

Luc Steels

Emergent Approaches

- Cognition is the process whereby an autonomous system becomes viable and effective in its environment

- It does so through a process of self-organization
  - System is continually re-constituting itself
  - In real-time
  - To maintain it operational identity
  - Through moderation of mutual system-environment interaction and co-determination

Maturana & Varela 87

Emergent Approaches

- Co-determination
  - Cognitive agent is specified by its environment
  - Cognitive process determines what is real or meaningful for the agent
  - The system constructs its reality (world) as a result of its operation in that world
  - Perception provides sensory data to enable effective action, but as a consequence of the system's actions
  - Cognition and perception are functionally-dependent on the richness of the action interface [Granlund]

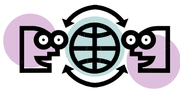
Emergent Approaches

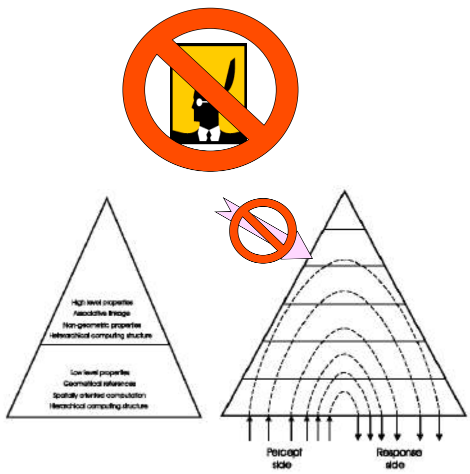
- Cognition is the complement of perception [Sandini]
  - Perception deal with the immediate
  - Cognition deals with longer time frames
- Primary model of cognitive learning is anticipative skill construction (not knowledge acquisition)
- The root of intelligence is to act effectively, anticipate the need to act, and increase the repertoire of actions
- Embodied as physical systems capable of physical interaction with the world

Emergent Approaches

'Cognitive systems need to acquire information about the external world through learning or association'

[Granlund'02]





Emergent Approaches

- Self-organization
  - "Self-organizing systems are physical and biological systems in which pattern and structure at the global level arises solely from interactions among the lower-level components of the system."
  - "The rules specifying interactions among the system's components are executed only using local information, without reference to the global pattern."
- Emergence
  - A process by which a system of interacting elements acquires qualitatively new pattern and structure that cannot be understood simply as the superposition of the individual contributions.

Camazine 2006

Connectionist Systems

- Rely on
  - Parallel processing
  - Non-symbolic distributed activation patterns in networks
  - Not logical rules
- Neural networks are the most common instantiations
  - Dynamical systems that capture statistical regularities or associations

Dynamical Systems

- Dynamical Systems
  - A dynamical system is an open dissipative non-linear non-equilibrium system
  - System: large number of interacting components & large number of degrees of freedom
  - Dissipative: diffuse energy – phase space decreased in volume with time (⇨ preferential sub-spaces)
  - Non-equilibrium: unable to maintain structure or function without external sources of energy, material, information (hence, open)
  - Non-linearity: dissipation is not uniform – small number of system's degrees of freedom contribute to behaviour
    - Order parameters / collective variables

Kelso '95: Dynamic Pattern – The Self-Organization of Brain and Behaviour

Dynamical Systems

$$\dot{\boldsymbol{q}} = N(\boldsymbol{q}, \boldsymbol{p}, n)$$

time derivative

noise

control parameters

state vector

From [Shoner Kelso 1988]

Enaction

- Orthodoxy (cognitivist)
  - World as the system experiences it is independent of the cognitive system (knower)
- Enactive view
  - Known and knower 'stand in relation to each other as mutual specification: they arise together'

Enaction

- Five key elements to enactive systems
  - Autonomy
  - Embodiment
  - Emergence
  - Experience
  - Sense-making

Enaction

- Autonomy
  - Self-maintenance
  - Homeostasis
  - Not controlled by outside agencies
  - Stands apart from its environment

Enaction

- Embodiment
  - Exists as a physical entity
  - Directly interacts with its environment
    - Structural coupling
    - Mutual perturbation
  - Constitutive part of the cognitive process

Enaction

- Emergence
  - Cognitive behaviour arises from dynamic interplay between component parts
  - Internal dynamics
    - Maintains autonomy
    - Condition the system's experiences through their embodiment in a specific structure

Enaction

- Experience
  - History of interaction with the world
  - Interactions don't control
  - Interaction do trigger changes in system state
  - Changes are structurally-determined
    - Phylogeny
    - Structural coupling

Enaction

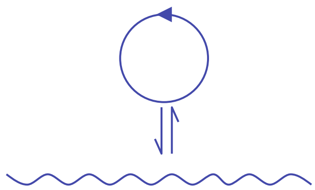
- Sense-making
  - Knowledge is generated by the system itself
  - Captures some regularity or lawfulness in the interactions
  - The 'sense' is dependent on the way interaction can take place
    - Perception & Action
  - Modify its own state (CNS) to enhance
    - Predictive capacity
    - Action capabilities
  - Development: generative autonomous self-modification

Development

Progressive ontogenetic acquisition of anticipatory capabilities

- Cognition cannot short-circuit ontogeny
- Necessarily the product of a process of embodied development
- Initially dealing with immediate events
- Increasingly acquiring a predictive capability

Cognition and perception are functionally-dependent on the richness of the action interface

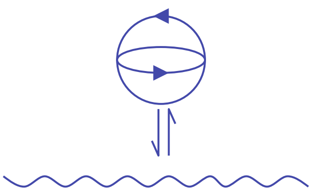


Structural Coupling

Autonomy-preserving mutual interaction

Perturbation of the system is only effected by the environment

[Note: this ideogram and similar ones to follow were introduced in Maturana and Varela 1987]



Cognitive system: operationally-closed system with a nervous system

Nervous system facilitates a highly-plastic mapping between sensor and motor surfaces

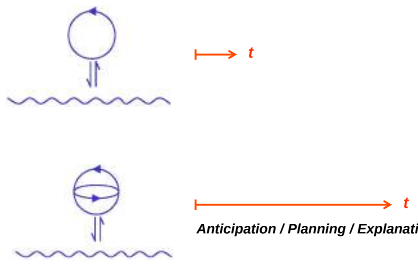
Perturbation by both environment and system (of receptors & NS)

Enaction

NERVOUS SYSTEM

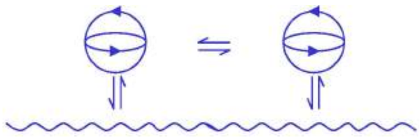
(a)Facilitates huge increase in the number of possible sensor-motor patterns (that result from structural coupling with the environment)

(b)Creates new dimensions (DoF) of structural coupling by facilitating association of internal states with the interactions in which the system is engaged



Anticipation / Planning / Explanation / Prediction

INTERACTION



A shared activity in which the actions of each agent
Influence the actions of the other agents in the same interaction
Resulting in a <u>mutually-constructed pattern of shared behaviour</u>
[Ogden et al.]

Meaning emerges through shared consensual experience
mediated by interaction

Bond of Union
M. C. Escher, 1956

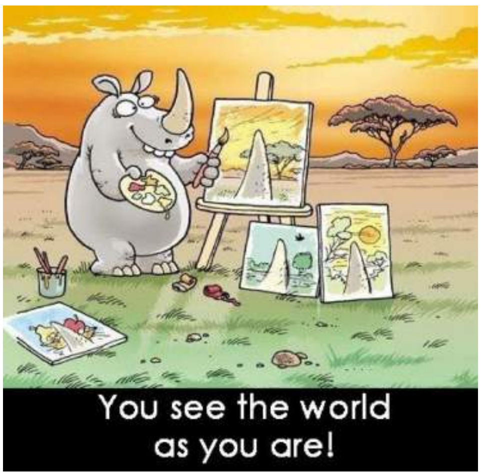
Interaction is a shared activity in which the actions of each agent influence the actions of the other agents engaged in the same interaction, resulting in a mutually-constructed patterns of shared behaviour

Ogden, Dautenhahn, Stribling 2002

THE SPACE OF PERCEPTUAL POSSIBILITIES

Is predicated not on an objective environment,
but on the space of possible actions
that the system can engage in whilst still
maintaining the consistency of its coupling with the environment

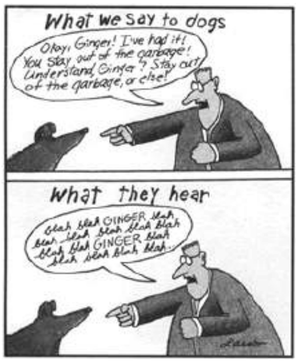
… Cognition involves seeing as <u>if</u>, rather than seeing as <u>is</u>



Enactive Cognition

MEANING

Emerges through shared consensual experience
mediated by interaction



Development

Progressive ontogenetic acquisition of anticipatory capabilities

- Cognition cannot short-circuit ontogeny
- Necessarily the product of a process of embodied development
- Initially dealing with immediate events ⟼ *t*
- Increasingly acquiring a predictive capability ⟼ *t*

Cognition and perception are functionally-dependent on the richness of the action interface

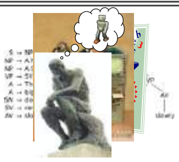
Differences between Cognitivist & Emergent Paradigms

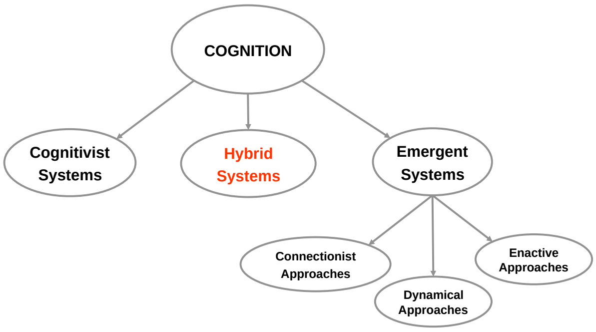
1. *Computational operation*
2. *Representational framework*
3. *Semantic grounding*
4. *Temporal constraints*
5. *Inter-agent epistemology*
6. *Embodiment*
7. *Perception*
8. *Action*
9. *Anticipation*
10. *Adaptation*
11. *Motivation*
12. *Autonomy*
13. *Cognition*
14. *Philosophical foundation*

[Vernon, Von Hofsten, Fadiga 2010]

The Cognitivist Paradigm vs. the Emergent Paradigm

Characteristic	Cognitivist	Emergent
Computational Operation	Syntactic manipulation of symbols	Concurrent self-organization of a network
Representational Framework	Patterns of symbol tokens	Global system states
Semantic Grounding	Percept-symbol association	Skill construction
Temporal Constraints	Atemporal	Synchronous real-time entrainment
Inter-agent epistemology	Agent-independent	Agent-dependent
Embodiment	No role implied: functionalist	Direct constitutive role: non-functionalist
Perception	Abstract symbolic representations	Perturbation by the environment
Action	Causal consequence of symbol manipulation	Perturbation by the system
Anticipation	Procedural or probabilistic reasoning	Traverse of perception-action state space
Adaptation	Learn new knowledge	Develop new dynamics
Motivation	Criteria for goal selection	Increase space of interaction
Autonomy	Not entailed	Cognition entails autonomy
Cognition	Rational goal-achievement	Self-maintenance and self-development
Philosophical Foundation	Positivism	Phenomenology

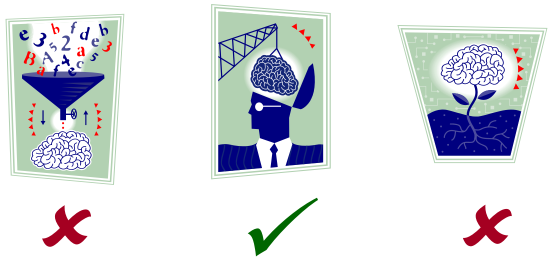




Hybrid Models

- Combination of cognitivist & emergent approaches
- Don't use explicit programmer based knowledge
- Can use symbolic representations (& representational invariances) but these are populated by the system itself as it learns
- Still emphasize perception-action coupling
- Often with action-dependent perception

Hybrid Models



Cognitive Robotics

- There is an increasing need for robots that are able to interact safely with humans in everyday situations
- These robot must be able to
  - Understand its environment and
  - Anticipate the effects of its own and other's actions

Cognitive Robotics

- Merging of two streams of research:
  - How to implement physical systems specifically designed for interaction with unconstrained environments … robotics research
  - How to design control architectures taking explicitly into account the need to acquire and use experience … cognitive science and artificial intelligence
- The result is a field of "Cognitive Robotics"

Cognitive Robotics

- The field has developed important tools in both areas
  - New actuators and sensors
  - Multimodal perception and action representation
- Merging of these tools has significant advantages
  - E.g. a robot acting safely in a human populated environment has to rely both on compliant actuation providing a "reactive" protection as well as on cognitive abilities providing the "predictive" contribution to safety
- Unlikely that there is a viable solution to safe human-robot interaction without this synergy

Cognitive Robotics

- Cognitive robotics: adaptive robots that can learn from experience
- Cognitive robots achieve their goals by
  - perceiving their environment
  - paying attention to the events that matter
  - planning what to do
  - anticipating the outcome of their actions and the actions of other agents
  - learning from the resultant interaction

Cognitive Robotics

- Cognitive robots deal with the inherent uncertainty of natural environments by continually learning, reasoning, and sharing their knowledge

Cognitive Robotics

- Focus on predictive capabilities to augment immediate sensory-motor experience and to allow a robot to anticipate future events relevant to the task being executed
- Cognitive robotics facilitates interaction between a robot and other cognitive agents
- Consequently, a cognitive robot can anticipate people's intended actions
  - During direct interaction (e.g. a robot assisting a surgeon in theatre)
  - Indirect interaction (e.g. a robot stacking shelves in a busy supermarket)

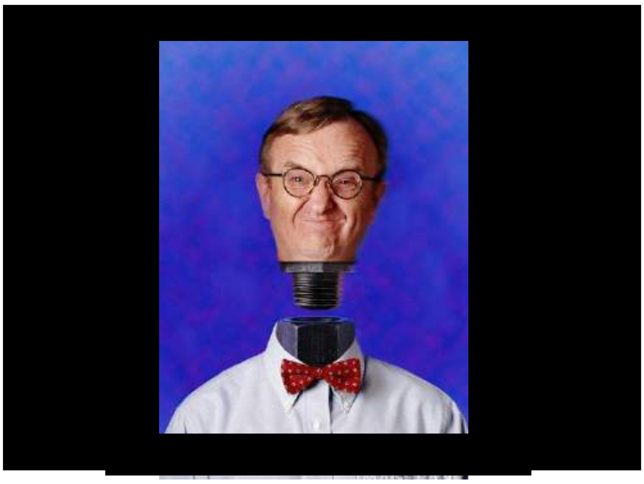
.

Cognitive Robotics

- The robot body is more than just a vehicle for physical manipulation or locomotion:

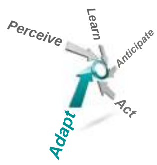
  it is a component of the cognitive process

- Embodied cognition: exploiting the agent's physical morphology, kinematics, and dynamics, as well as the environment in which it is operating



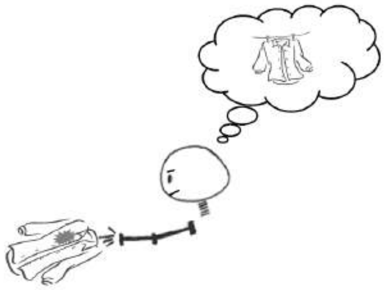
Cognitive ~~System~~ Robot

"A **cognitive system** is
an **autonomous** system that can
perceive its environment,
learn from experience,
anticipate the outcome of events,
act to pursue goals, and
adapt to changing circumstances."

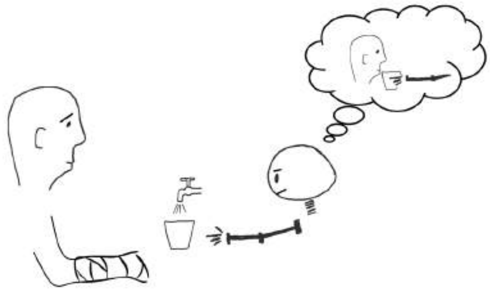


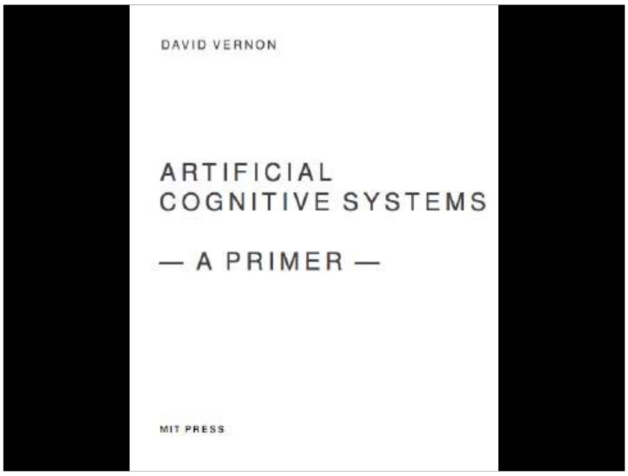
Vernon, D. "Cognitive System", in Computer Vision: A Reference Guide, Springer, pp. 100-106, (2014).

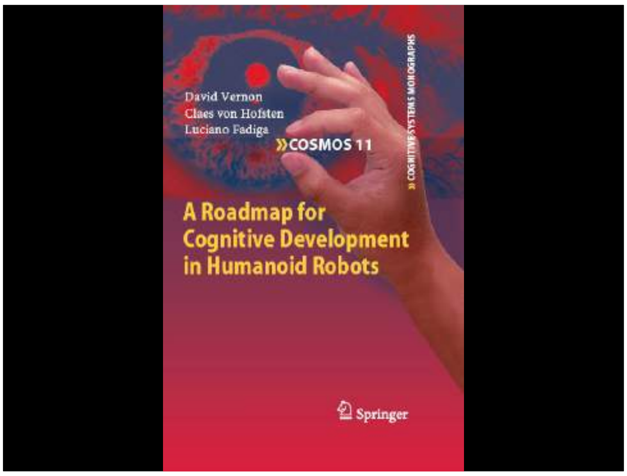
Cognitive Robot



Cognitive Robot







Development of robot-enhanced therapy for children with autism spectrum disorders

EU FP7 Project 611391

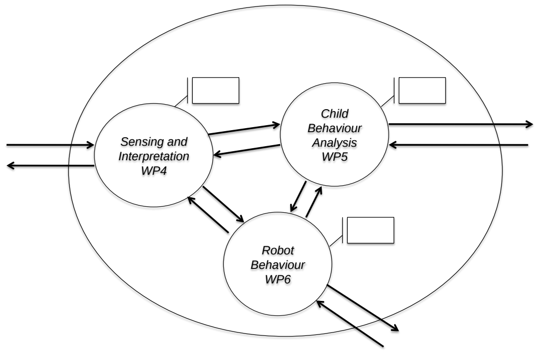
NAO *Probo*

Development of robot-enhanced therapy for children with autism spectrum disorders

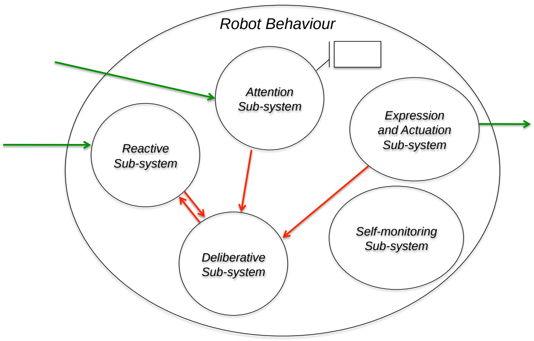
EU FP7 Project 611391

Imitation: sand tray   Imitation: arm movements

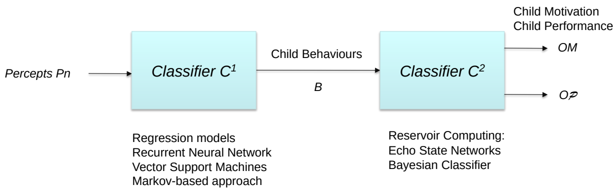
The DREAM System Architecture



The DREAM System Architecture



Child Behaviour Analysis



Cognitive Robotics

Talk 2

- Cognitive architectures
  - Soar
  - ISAC
  - Clarion
  - …

Cognitive Robotics

Talk 3

- Aspects of a cognitive architecture
  - Memory & Prospection
  - Knowledge & Representation

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Cognitive Robotics

David Vernon
University of Skövde

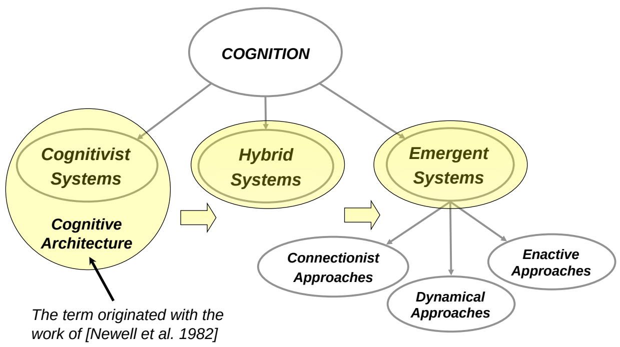
www.vernon.eu

Reasoning in the Robot World Summer School 2014
Czech Technical University in Prague
Center for Machine Perception
29 – 31 July 2014

Cognitive Robotics

Talk 2

– Cognitive architectures
  - Overview
  - Soar
  - ISAC
  - Clarion



[Vernon, Metta, Sandini 2007]

Cognitivist Cognitive Architectures

Attempts to create *Unified Theories of Cognition (UTC)*:

UTCs cover a broad range of cognitive issues:

- Attention
- Memory
- Problem solving
- Decision making
- Learning
- …

from several aspects

- Psychology
- Neuroscience
- Computer Science
- …

[Byrne 2003]

Cognitivist Cognitive Architecture

An embodiment of a **scientific hypothesis** about those aspects of **human cognition** that are

relatively **constant over time** and

relatively **independent of task**

[Ritter & Young 2001]

Cognitivist Cognitive Architecture

Generic computational model:

- Not domain-specific
- Not task-specific

Knowledge provides the required specificity

Cognitive Architecture + Knowledge = *Cognitive Model*

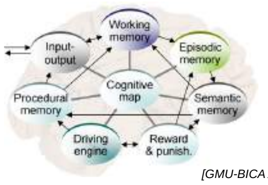
[Lehman et al 97, also Anderson & Labiere 98, Newell 90]

Cognitivist Cognitive Architecture

Overall structure and organization of a cognitive system

– **Modules**
– **Relations** between these modules
– **Algorithmic** and **representational details** in each module

[Sun 2007]



[GMU-BICA Architecture: Samsonovich 2005]

Cognitivist Cognitive Architecture

Commitment to formalisms for

– Short-term & long-term **memories** that store the agent's beliefs, goals, and knowledge
– **Representation** & organization of structures embedded in memory
– Functional **processes** that operate on these structures
  • Performance / utilization
  • Learning
– **Programming** language to construct systems embodying the architectures assumptions

[Langley 05, Langley 06, Langley et al. 09]

Emergent Cognitive Architecture

Emergent approaches focus on **development**

– From a primitive state
– To fully cognitive state, over the system's lifetime

Jean Piaget

Lev Vygotsky

Emergent Cognitive Architecture

The cognitive architecture is the system's
phylogenetic configuration

– The basis for ontogenesis: growth and development
  • Innate skills
  • Core knowledge (cf. Spelke)
– A structure in which to embed mechanisms for
  • Perception
  • Action
  • Adaptation
  • Anticipation
  • Motivation
  • *… Development of all these*

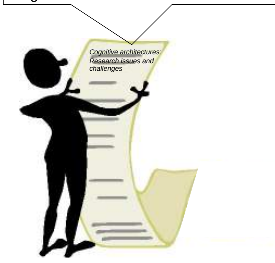
Emergent Cognitive Architecture

Focus on

– **Autonomy-preserving anticipatory and adaptive skill construction**
– The **morphology** of the physical body in which the architecture is embedded

Desirable Characteristics

Cognitive architectures: Research issues and challenges



1. *Recognition & categorization*
2. *Decision-making & choice*
3. *Perception & situation assessment*
4. *Prediction & monitoring*
5. *Problem solving & planning*
6. *Reasoning & belief maintenance*
7. *Execution & action*
8. *Interaction & communication*
9. *Remembering, reflection, & learning*

[Langley et al. 2009]

Desirable Characteristics

The importance of cognitive architectures …

1. *Perception*
2. *Categorization*
3. *Multiple representations*
4. *Multiple types of memory*
5. *Decision making*
6. *Reasoning*
7. *Planning*
8. *Problem solving*
9. *Meta-cognition*
10. *Communication*
11. *Action control and execution*
12. *Several types of learning*

[Sun 2007]

The importance of the interconnectivity between these processes

Desirable Characteristics

Cognitive Architectures of Developmental Systems
[Krichmar & Edelman 2005, 2006]

1. Address connectivity and interaction between circuits/regions in the brain
2. Effect perceptual categorization, without a priori knowledge
(a model generator, rather than a model fitter, cf [Weng 2004])
3. Embodied & capable of exploration
4. Minimal set of innate behaviours
5. Value system (set of motivations) to govern development

Facets of a Cognitive Architecture

- Component **functionality**
- Component **interconnectivity**
- System **dynamics**

Facets of a Cognitive Architecture

- System dynamics
  - Cognitivist cognitive architectures
    - Add knowledge to determine the dynamics and flow of information
  - Emergent cognitive architectures
    - Not so straightforward … can't just add knowledge

Facets of a Cognitive Architecture

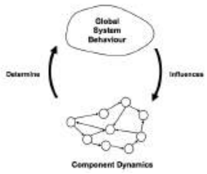
- System dynamics
  - Emergent cognitive architectures
    - Dynamics result from interaction between the components
      - Driven by an **embedded value system** that governs the developmental process
      - **Not by explicit rules** that encapsulate **prior declarative and procedural knowledge**

Facets of a Cognitive Architecture

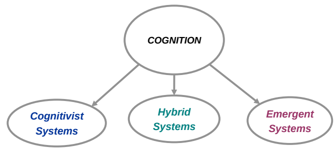
- System dynamics
  - Emergent cognitive architectures
    - Need to specify the **interactions** between components
      - Small ensembles (at least)
      - Whole system (ideally)

Facets of a Cognitive Architecture

- System dynamics
  - Emergent cognitive architectures
    - This is a tough challenge
      - Assemblies of **loosely-coupled concurrent processes**
      - Operating **asynchronously**
      - **Without a central control unit**
      - Dynamics depend on **circular causality**



- *Organizational decomposition*
- *Explicit inter-connectivity*
- *Representational formalism*
- *Algorithmic formalism*



- *Framework in which to embed knowledge*
- *Memories*
- *Formalisms for learning*
- *Programming mechanism*

Phylogeny - basis for development:

- *Innate skills*
- *Core knowledge*
- *Formalism for autonomy*
- *Formalism for development*

Example Architectures

Surveys:

Biologically Inspired Cognitive Architectures Society,
Comparative Repository of Cognitive Architectures ,
http://bicasociety.org/cogarch/architectures.htm
(25 cognitive architectures)

A Survey of Cognitive and Agent Architectures,
University of Michigan, http://ai.eecs.umich.edu/cogarch0/
(12 cognitive architectures)

Example Architectures

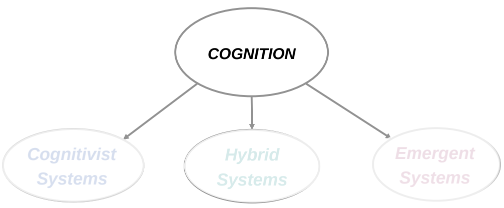
Surveys:

W. Duch, R. J. Oentaryo, and M. Pasquier.
"Cognitive Architectures: Where do we go from here?",
Proc. Conf. Artificial General Intelligence, 122-136, 2008.
(17 cognitive architectures)

D. Vernon, G. Metta, and G. Sandini,
"A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents",
IEEE Transactions on Evolutionary Computation, Vol. 11, No. 2, pp. 151-180, 2007.
(14 cognitive architectures)

D. Vernon, C. von Hofsten, and L. Fadiga.
"A Roadmap for Cognitive Development in Humanoid Robots",
Cognitive Systems Monographs (COSMOS), Vol. 11, Springer
Chapter 5 and Appendix I
(20 cognitive architectures)

Example Architectures

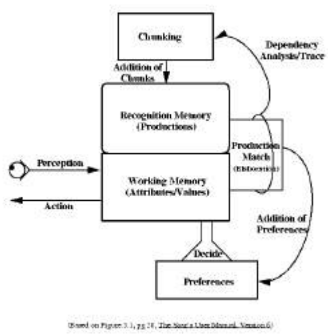


Soar [Newell 1996]
EPIC [Kieras & Meyer 1997]
ICARUS [Langley 05, Langley 2006]
GLAIR [Shapiro & Bona 2009]
CoSy [Hawes & Wyatt 2008]

CLARION [Sun 2007]
ISAC [Kawamura 2007]
ACT-R [Anderson et al. 2004]
KHR [Burghart et al. 2005]
LIDA [Franklin et al. 2007, Baars & Franklin 2009]
PACO-PLUS [Kraft et al. 2008]

iCub [Vernon et al. 2010]
Global Workspace [Shanahan 2006]
SASE [Weng 2004]
Darwin [Krichmar et al. 2005]
Cognitive-Affective [Morse et al. 2008]

Soar



Soar

Laird, J.E., Newell, A., Rosenbloom, P.S.: Soar: an architecture for general intelligence. Artificial Intelligence 33, 1–64 (1987)

Rosenbloom, P., Laird, J., Newell, A. (eds.): The Soar Papers: Research on Integrated Intelligence. MIT Press, Cambridge (1993)

Lehman, J.F., Laird, J.E., Rosenbloom, P.S.: A gentle introduction to soar, an architecture for human cognition. In: Sternberg, S., Scarborough, D. (eds.) Invitation to Cognitive Science. Methods, Models, and Conceptual Issues, vol. 4. MIT Press, Cambridge (1998)

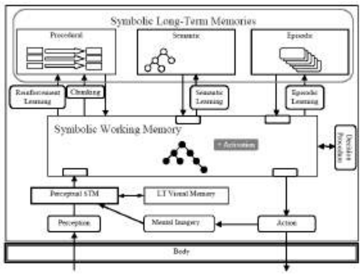
Lewis, R.L.: Cognitive theory, Soar. In: International Encyclopedia of the Social and Behavioural Sciences. Pergamon, Elsevier Science, Amsterdam (2001)

Laird, J. E. Towards Cognitive Robotics, Unmanned Systems Technology XI. Edited by Gerhart, G. R., Gage, D. W., Shoemaker, C. M., Proceedings of the SPIE, Volume 7332, pp. 73320Z-73320Z-11 (2009).

Soar

Soar continues to evolve … [Laird 2009]

- Perception
- Action
- Mental imagery
  (internal simulation)
- Procedural memory
  & reinforcement learning
- Semantic memory
  & learning
- Episodic memory
  & learning



Soar

- Newell's candidate for a Unified Theory of Cognition
- Archetypal and iconic cognitivist cognitive architecture
- Production (or rule-based) system
  - Production: effectively an IF-THEN condition-action pair
- Rules in Soar are called associations

Soar

- The core of Soar comprises two memories
  - Long-term memory
    - Sometimes referred to as recognition memory
    - holds the productions rules
  - Working memory
    - Also called declarative memory
    - Holds the attribute values that reflect Soar's perceptions and actions

Soar

- Several processes
  - *Elaboration:*
    - Matches the productions and the attribute values
    - Decides which productions can fire
  - Determining the preferences to use in the *decision* process
  - *Chunking* which effectively learns new production rules (called *chunks*)

Soar

- Operates in a cyclic manner
  - Production cycle
    - All productions that match the contents of declarative (working) memory fire
      - A production that fires may alter the state of declarative memory
      - and cause other productions to fire
    - This continues until no more productions fire.
  - Decision cycle
    - a single action from several possible actions is selected
    - The selection is based on stored action preferences

Soar

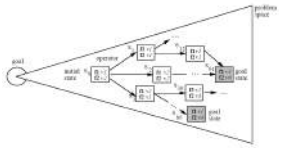
- No guarantee that the action preferences will lead to a unique action or any action
- In this case, the decision cycle may lead to an '*impasse*'
  - Soar sets up an new state in a new problem space — *universal sub-goaling* — with the goal of resolving the impasse
  - Resolving one impasse may cause other impasses and the sub-goaling process continues
- Eventually, all impasses should be resolved
  - In the case where the situation degenerates with Soar having insufficient knowledge to resolve the impasse, it chooses randomly between possible actions.

Soar

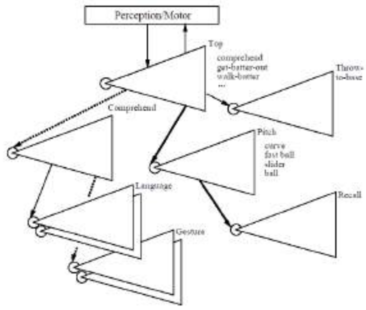
- Whenever an impasse is resolved, Soar creates a new production rule, i.e. a new association
  - Summarizes the processing that occurred in the sub-state in solving the sub-goal
  - As we noted above, this new learned association is called a *chunk* and the Soar learning process is referred to as *chunking*

Soar

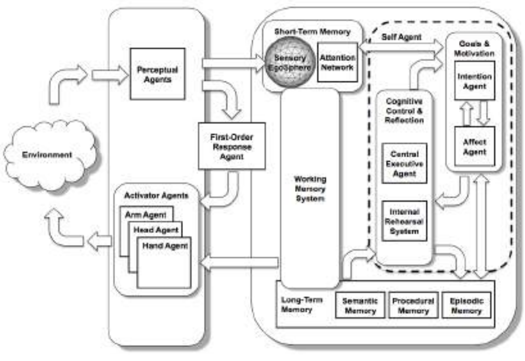
- Behaviour as movement through problem spaces
- Soar's architectural mechanisms process the four types of objects
  - Goal (circle)
  - Problem space: expanding set of possibilities that can unfold over time (triangle)
  - States (rectangles)
    - Vocabulary of features (bold)
    - Their possible values (italics) ... values can also be a set of features
  - State transition (arrows) ... operators reflecting internal or external behaviour



Soar



ISAC



ISAC

K. Kawamura, S. M. Gordon, P. Ratanaswasd, E. Erdemir, and J. F. Hall. Implementation of cognitive control for a humanoid robot. International Journal of Humanoid Robotics, 5(4):547–586, 2008.

ISAC

- ISAC — Intelligent Soft Arm Control
  - Hybrid cognitive architecture for an upper torso humanoid robot (also called ISAC)
  - Constructed from an integrated collection of software agents and associated memories
  - Agents encapsulate all aspects of a component of the architecture
  - Agents operate asynchonously and communicate with each other by passing messages

ISAC

- Comprises activator agents for
  - motion control
  - perceptual agents
  - a First-order Response Agent (FRA) to effect reactive perception-action control
- Three memory systems
  - Short-term memory (STM)
  - Long-term memory (LTM)
  - Working memory system (WMS)

ISAC

- STM
  - Robot-centred spatio-temporal memory of the perceptual events currently being experienced
  - This is called a Sensory EgoSphere (SES)
  - Discrete representation of what is happening around the robot, represented by a geodesic sphere indexed by two angles: horizontal (azimuth) and vertical (elevation).
  - STM also has an attentional network that determines the perceptual events that are most relevant and then directs the robot's attention to them

ISAC

- LTM
  - Stores information about the robot's learned skills and past experiences
  - Semantic memory
  - Episodic memory

    robot's declarative memory of the facts it knows
  - Procedural memory — *representations of the motions it can perform*

ISAC

- Episodic memory
  - Abstracts past experiences & creates links or associations between them
  - Information about
    - External situation (i.e. task-relevant percepts from the SES)
    - Goals
    - Emotions (internal evaluation of the perceived situation)
    - Actions
    - Outcomes that arise from actions
    - Valuations of these outcomes (e.g. how close they are to the desired goal state and any reward received at a result)

ISAC

- Episodic memory
  - Episodes are **connected by links** that encapsulate behaviours
    - Transitions from one episode to another
  - Multi-layered

ISAC

- WMS
  - Inspired by neuroscience models of brain function
  - Temporarily stores information that is related to the task currently being executed
  - A type of cache memory for STM and the information it stores, called chunks
  - Encapsulates expectations of future reward (learned using a neural network)

ISAC

- Cognitive behaviour is achieved through **the interaction of several agents**
  - ***Central Executive Agent*** **(CEA)**
  - ***Internal Rehearsal System*** (simulates the effects of possible actions)
  - ***Goals & Motivation sub-system***
    - ***Intention Agent***
    - ***Affect Agent***
  - the CEA and Internal Rehearsal System form a compound agent called the ***Self Agent***

ISAC

- Cognitive behaviour is achieved through the interaction of several agents
  - The CEA is responsible for cognitive control
  - Invokes the skills required to perform some given task on the basis of the current focus of attention and past experiences
  - The goals are provided by the Intention Agent
  - Decision-making is modulated by the Affect Agent

ISAC

- ISAC works the following way
  - Normally, the First-order Response Agent (FRA) produces reactive responses to sensory triggers

ISAC

- ISAC works the following way
  - Normally, the First-order Response Agent (FRA) produces reactive responses to sensory triggers
  - However, it is also responsible for executing tasks

ISAC

- ISAC works the following way
  - Normally, the First-order Response Agent (FRA) produces reactive responses to sensory triggers
  - However, it is also responsible for executing tasks
  - When a task is assigned by a human, the FRA retrieves the skill from procedural memory in LTM that corresponds to the skill described in the task information

ISAC

- ISAC works the following way
  - Normally, the First-order Response Agent (FRA) produces reactive responses to sensory triggers
  - However, it is also responsible for executing tasks
  - When a task is assigned by a human, the FRA retrieves the skill from procedural memory in LTM that corresponds to the skill described in the task information
  - It then places it in the WMS as chunks along with the current percept

ISAC

- ISAC works the following way
  - Normally, the First-order Response Agent (FRA) produces reactive responses to sensory triggers
  - However, it is also responsible for executing tasks
  - When a task is assigned by a human, the FRA retrieves the skill from procedural memory in LTM that corresponds to the skill described in the task information
  - It then places it in the WMS as chunks along with the current percept
  - The Activator Agent then executes it, suspending execution whenever a reactive response is required

ISAC

- ISAC works the following way
  - If the FRA finds no matching skill for the task, the Central Executive Agent takes over

ISAC

- ISAC works the following way
  - If the FRA finds no matching skill for the task, the Central Executive Agent takes over
  - Recalls from episodic memory past experiences and behaviours that contain information similar to the current task

ISAC

- ISAC works the following way
  - If the FRA finds no matching skill for the task, the Central Executive Agent takes over
  - Recalls from episodic memory past experiences and behaviours that contain information similar to the current task
  - One behaviour-percept pair is selected, based on the current percept in the SES, its relevance, and the likelihood of successful execution as determined by internal simulation in the IRS

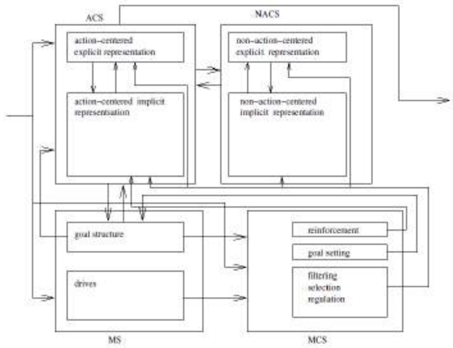
ISAC

- ISAC works the following way
  - If the FRA finds no matching skill for the task, the Central Executive Agent takes over
  - Recalls from episodic memory past experiences and behaviours that contain information similar to the current task
  - One behaviour-percept pair is selected, based on the current percept in the SES, its relevance, and the likelihood of successful execution as determined by internal simulation in the IRS
  - This is then placed in working memory and the Activator Agent executes the action

CLARION

- Hybrid cognitive architecture
  - Symbolic representations
  - Connectionist representations
- Four sub-systems
  - ACS – Action-centred subsystem
  - NACS – Non-action-centred subsystem
  - MS – Motivational subsystem
  - MCS – meta-cognitive subsystem

CLARION



CLARION

Sun, R.: A tutorial on CLARION. In: Cognitive Science Department. Rensselaer Polytechnic Institute (2003), http://www.cogsci.rpi.edu/rsun/sun.tutorial.pdf

Sun, R.: The importance of cognitive architectures: an analysis based on CLARION. Journal of Experimental & Theoretical Artificial Intelligence 19(2), 159–193 (2007)

CLARION

- All four subsystems have two levels of knowledge representation
  - Implicit connectionist bottom level
  - Explicit symbolic top level
  - Implicit and explicit levels interact and cooperate both in action selection and in learning
- Able to learn with or without a priori domain-specific knowledge
- Able to learn continuously from on-going experience

CLARION

- Action-centred Subsystem (ACS)
  - Controls actions
    - External physical movements
    - Internal mental operations

CLARION

- Action-centred Subsystem (ACS)
  - Given some observational state, i.e. a set of sensory features x
    - The bottom level evaluates the desirability ("quality") of all possible actions

      $Q(x, a_1), Q(x, a_2), \dots, Q(x, a_n)$
    - The top level identifies possible actions from a rule network based on the input x sent up from the bottom level

      $(b_1, b_2, \dots, b_m)$

CLARION

- Action-centred Subsystem (ACS)
  - The bottom-level actions a_i and top-level actions b_j are compared and the most appropriate top-level action b is selected
  - Action b is performed and the outcome is observed
    - The next state y and (possibly) a reinforcement r are determined
    - The Q values at the bottom level are updated using the *Q-Learning-Backpropagation* algorithm
    - The top-level rules are also updated using the *Rule-Extraction-Refinement* algorithm
  - This process continues indefinitely

CLARION

- Action-centred Subsystem (ACS)
  - The bottom level comprises several modules of small neural networks
    - Each adapted to a distinct sensory modality or task
    - These modules can be developed by the system
      - based on experience (i.e. though **ontogenesis**) through trial-and-error exploration
      - or they can be specified a priori and hard-wired into the cognitive architecture (i.e. as the system **phylogeny**).

CLARION

- Action-centred Subsystem (ACS)
  - In the top level, explicit symbolic conceptual knowledge is captured in the form of symbolic rules
  - Explicit knowledge can be learned in several ways
    - Independent experiential hypothesis-testing learning
    - Mediation of implicit knowledge: bottom-up learning … Autonomous Generation of Explicit Conceptual Structures

CLARION

- Action-centred Subsystem (ACS)
  - The implicit bottom level & the explicit top level representations interact to effect bottom-up learning
  - If an action selected by the bottom level is successful
    - the system extracts an explicit rule that corresponds to the sensory features and the selected action
    - adds the rule to its top level rule network

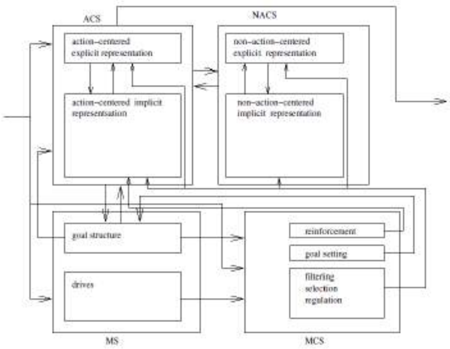
CLARION

- Action-centred Subsystem (ACS)
  - The system subsequently verifies the extracted rule by considering the outcome of applying the rule
    - If the outcome is successful, the rule is generalized (made more universal and applicable to other situations)
    - If the outcome is unsuccessful, the rule is refined (made more specific and exclusive of the current situation)
  - i.e. autonomous generation of explicit conceptual structures by exploiting implicit knowledge acquired by trial-and-error learning

CLARION

- Action-centred Subsystem (ACS)
  - Assimilation of externally-given conceptual structures
    - ***Internalizing*** externally-provided knowledge in the form of explicit rule-based conceptual structures with existing conceptual structures at the top-level
    - ***Assimilating*** these into the bottom level implicit representation … top-down learning

CLARION



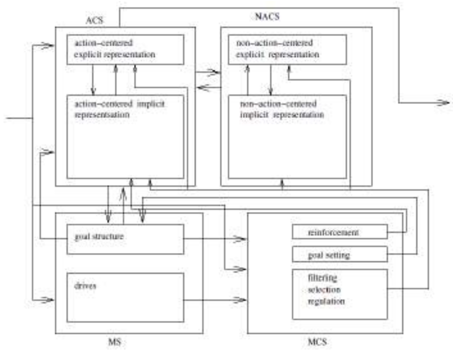
CLARION

- Non-Action-centred Subsystem (NACS)
  - Maintains the system's general knowledge
    - Implicit knowledge in connectionist form
      - Associative memory networks (mapping input to output)
    - Explicit knowledge in symbolic form
      - A network of nodes
      - Each node corresponds to an entity-specific chunk comprising
        - » an entity identifier (e.g. table_1)
        - » a vector of feature dimensions / feature value pairs (e.g. (size, large) ... (colour, white), (number_of_legs, 4))

CLARION

- Non-Action-centred Subsystem (NACS)
  - Maintains the system's general knowledge
    - The feature values are represented by nodes in the bottom level associative memory
    - Chunks are linked through association rules
  - **Both bottom-up and top-down learning can take place**
    - Extract explicit knowledge in the top level from the implicit knowledge in the bottom level
    - Assimilate explicit knowledge of the top level into implicit knowledge in the bottom level

CLARION



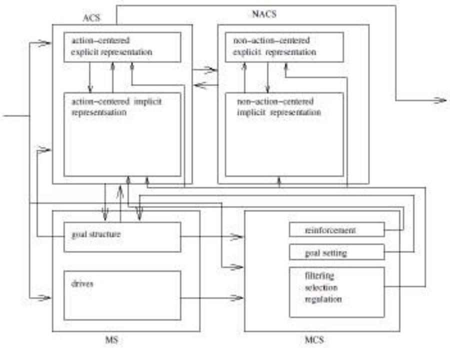
CLARION

- Motivational Subsystems (MS)
  - Provides
    - The drives that determines what the agent does
    - Evaluates the feedback
      (were the outcomes of an action satisfactory or not)

CLARION

- Motivational Subsystems (MS)
  - Provides the ACS with goals derived from
    - Low-level drives concerning physiological needs (e.g. need for food, need for water, need to avoid danger, need to avoid boredom, …)
    - High-level drives (e.g. desire for social approval, desire for following social norms, desire for reciprocation, desire for imitation of other people, …)
      - Primary hard-wired drives (cf. Maslow's hierarchy of needs)
      - Secondary derived drives (changeable, acquired mostly in the process of satisfying primary drives)

CLARION



CLARION

- Meta-cognitive Subsystem (MCS)
  - Monitors, regulates, and modify the overall behaviour of the cognitive system to improve cognitive performance
    - By setting goals for the action-centred subsystem
    - By setting essential parameter values the action-centred and non-action-centred subsystems
    - For example, setting reinforcement functions
    - Can be achieved by setting drive states in the motivational subsystem
  - Also comprises a top level (explicit) and bottom level (implicit)

Caveat

"In the stream of cognitive processes one can conceptually isolate certain components, for instance

(i) the faculty to perceive,

(ii) the faculty to remember, and

(iii) the faculty to infer.

But if one wishes to isolate these faculties functionally or locally, **one is doomed to fail.**

Consequently, if the mechanisms that are responsible for any of these faculties are to be discovered, then the totality of cognitive processes must be considered."

Heinz von Foerster

Cognitive Robotics

Talk 3

- Aspects of a cognitive architecture
  - Memory & Prospection
  - Knowledge & Representation

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Cognitive Robotics

David Vernon
University of Skövde

www.vernon.eu

Reasoning in the Robot World Summer School 2014
Czech Technical University in Prague
Center for Machine Perception
29 – 31 July 2014

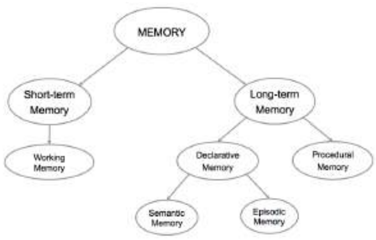
Cognitive Robotics

Talk 3

- Aspects of a cognitive architecture
  - Memory & Prospection
  - Knowledge & Representation

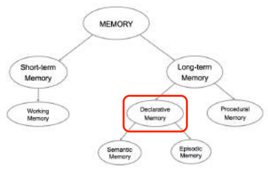
Types of Memory

- Declarative
- Procedural
- Semantic
- Episodic
- Long-term
- Short-term
- Working
- Modal
- Amodal
- Symbolic
- Sub-symbolic
- Hetero-associative
- Auto-associative



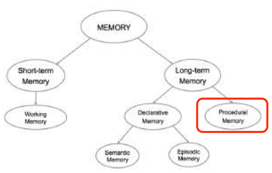
Types of Memory

- Declarative
  - **Knowledge of things / facts**
  - “Knowing that”
  - ***Propositional memory*** (true or false)
  - Can be communicated from one agent to another through language
  - Can be acquired in a single act of perception or cognition
  - Accessible to conscious recall
  - ***Explicit memory***



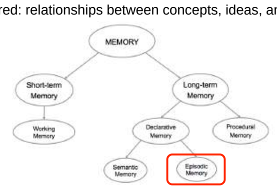
Types of Memory

- Procedural
  - **Skill-oriented memory of actions**
  - “Knowing how”
  - Can only be demonstrated
  - Acquired progressively and may require an element of practice
  - Not accessible to conscious recall
  - ***Implicit memory***
  - ***Non-declarative memory***



Types of Memory

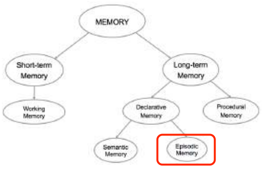
- Episodic
  - Specific instances in the agent’s experience: **autobiographical**
  - **Explicit spatial and temporal context**
    - what happened, where it happened, and when it happened
    - This temporal sequencing is the only element of structure in episodic memory
  - Highly structured: relationships between concepts, ideas, and facts
  - Sub-symbolic



Types of Memory

– Episodic

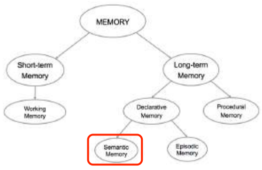
- Episodic memory is a **constructive process**
  - Each time an event is assimilated into episodic memory, past episodes are re-constructed a little differently each time
  - Related to the role that episodic memory plays in the process of internal simulation that forms the basis of prospection, the key anticipatory function of cognition



Types of Memory

– Semantic

- General knowledge about the agent's world: facts, ideas, and concepts
- May be independent of the agent's specific experiences
- Memory necessary for the use of language
- Derived from episodic memory through a process of generalization and consolidation
- Symbolic



Types of Memory

– Modal memory

- Tied directly to a particular sensory modality such as vision, audition, or touch
- Episodic memory though is more likely to be modal since it is closely tied to an agents's specific experiences

– Amodal memory

- Amodal memory has no necessary association with the sensorimotor experiences
- Semantic declarative facts, represented symbolically, are typically amodal

Types of Memory

– Associative memory

- An element of information or some pattern is linked to another
- The first element or pattern is used to recall the second, by association

– Hetero-associative memory

- Recalls a memory that is different in character from the input
- A particular smell or sound, for example, might evoke a visual memory of some past event

– Auto-associative memory

- Recalls a memory of the same modality as the one that evoked it
- A picture of a favourite object might evoke a mental image of that object in vivid detail

Role of Memory

- The role of memory – why do we remember things?
  - To recognize objects, events, and people we've encountered before
  - To act towards them in some appropriate way (attraction/avoidance)
  - Memory is what makes it possible for the changes that occur as a result of learning and development to persist
  - **Memory makes it possible to project forwards into the future**

Role of Memory

"It's a poor sort of memory that only works backwards"

Remarks of the White Queen to Alice
in Lewis Carroll's *Through the Looking Glass*

Memory is Prospective

Role of Memory

- One of the central pillars of cognitive capacity:
  - the ability to **simulate internally the outcomes of possible actions** and select the most appropriate one for the current situation
  - Memory can be seen as a mechanism that allows a cognitive agent **to prepare to act,** overcoming through anticipation the inherent "here-and-now" limitations of its perceptual capabilities
  - **a cognitive system** doesn't operate just on the basis of its current sensory data bu**t readies itself for what it expects and adjusts to the unexpected**

Role of Memory

- Memory is an **active** & **constructive** process, and it is fundamentally **associative**
  - Memories are recalled by associated triggers, possibly other memories
  - If you have a network of associative memories, you can run through this network **backwards** or **forwards**
  - Running through it **forwards** provides the **anticipatory predictive** element of memory suggesting possible sequence of events leading to a desired goal
  - Running through it **backwards** provides a way of **explaining** how some event or other might have occurred

Self-Projection, Prospection, & Internal Simulation

- Memory plays at least four roles in cognition
  1. Remember past events
  2. Anticipate future ones
  3. Imagine the viewpoint of other people
  4. Navigate around our world
- All four involve **self-projection**
  - Ability of an agent to shift perspective from itself in the here-and-now
  - Take an alternative perspective
  - It does this by **internal simulation**, i.e. the mental construction of an imagined alternative perspective

Self-Projection, Prospection, & Internal Simulation

- There are four forms of internal simulation
  1. **Episodic memory** (remembering the past)
  2. **Navigation** (orienting yourself topographically, i.e. in relation to your surroundings)
  3. **Theory of mind** (taking someone else's perspective on matters)
  4. **Prospection** (anticipating possible future events)
- Each form of simulation has a different orientation (past, present, or future)
- Each refers to the perspective of either the agent itself or another person

Self-Projection, Prospection, & Internal Simulation

- Recent evidence suggests that all four kinds of internal simulation involve a single core brain network
  - This network overlaps what is known as the default-mode network
  - A set of interconnected regions in the brain that is active when the agent is **not** occupied with some attentional task

Self-Projection, Prospection, & Internal Simulation

- Episodic memory
  - **Re-experience** your past
  - **Pre-experience** your future
- Projecting yourself forward in time is important when you form a goal
  - Creating a mental image of yourself **acting out the event**
  - **Episodically pre-experiencing** the unfolding of a plan to achieve that goal
  - **Episodic Future Thinking** [Atance and O'Neill 2001]

Self-Projection, Prospection, & Internal Simulation

- **Episodic memory is inherently constructive**
  - Old episodic memories are reconstructed slightly differently every time a new episodic memory is assimilated or remembered
  - The **constructive episodic simulation hypothesis** [Schacter and Addis 2007]
    - Episodic memory allows the **simulation of multiple possible futures**
    - This imposes an **even greater** need for a constructive capacity because of the need to extrapolate **beyond past experiences**
  - Episodic memory is not an exact and perfect record of experience but one that conveys the essence of an event and is open to re-combination

Internal Simulation and Action

- So far, internal simulation considered entirely in terms of memory-based self-projection
  - Using re-assembled combinations of episodic memory to
    - Pre-experience possible futures
    - Re-experience (and possibly adjust past experiences)
    - Project ourselves into the experiences of others
- However, action plays a significant role in our perceptions so does action play a role in internal simulation?
- **YES**

Internal Simulation and Action

- Internal simulation
  - extends beyond episodic memory
  - includes simulated interaction, particularly embodied interaction
- Terms
  - Simulation
  - Internal simulation
  - Mental simulation
  - *Emulation*

Internal Simulation and Action

- Several simulation theories, but perhaps the most influential is known as the **Simulation Hypothesis** [Hesslow 2002, Hesslow 2012]
- Three core assumptions

Covert action / covert behaviour

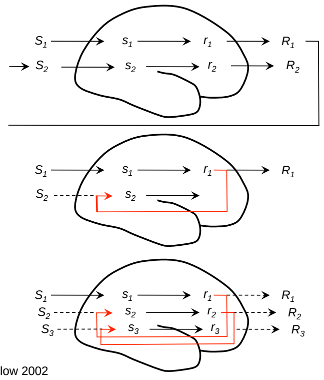
1. The regions in the brain that are responsible for motor control can be activated ***without*** causing bodily movement
2. Perceptions can be caused by internal brain activity and not just by external stimuli

Simulation of perceptions

3. The brain has associative mechanisms that allow motor behaviour or perceptual activity to evoke other perceptual activity

Simulated action elicit perceptions

Internal Simulation and Action



No internal simulation

A motor response to an input stimulus causes the internal simulation of an associated perception …

This elicits a covert action which in turn elicits a simulated perception and a consequent covert action

Hesslow 2002

Internal Simulation and Action

- There is an increasing amount of neurophysiological evidence in support of all three assumptions

For example, see:

H. Svensson, S. Thill, and T. Ziemke. Dreaming of electric sheep? Exploring the functions of dream-like mechanisms in the development of mental imagery simulations. *Adaptive Behavior*, 21:222–238, 2013.

Internal Simulation and Action

- Action-directed internal simulation involves three different types of anticipation:
  - **Implicit anticipation**
    - Prediction of motor commands from (possibly simulated) perceptions
  - **Internal anticipation**
    - Prediction of the proprioceptive consequences of carrying out an action, i.e. the effect of an action on the agent's own body
  - **External anticipation**
    - Prediction of the consequences for external objects and other agents of carrying out an action

Internal Simulation and Action

- **Implicit anticipation** selects some motor activity (possibly covert, i.e. simulated) to be carried out based on an association between stimulus and actions
- Internal and external anticipation then **predict** the consequences of that action
- Collectively, they simulate actions and the effects of actions
- Covert action involves ***motor imagery***
- Simulation of perception is referred to as ***visual imagery (perceptual imagery)***

Internal Simulation and Action

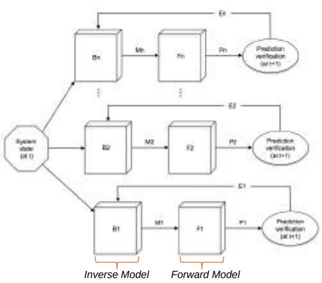
- Motor imagery is also a form of perceptual imagery
  - It involves the proprioceptive and kinesthetic sensations associated with bodily movement
- Covert action often has elements of both motor and visual imagery
- *Vice versa*, the simulation of perception often has elements of motor imagery
- Visual and motor imagery are sometimes referred to collectively as ***mental imagery***
- Mental imagery can be viewed as a synonym for internal simulation

Internal Simulation and Action

- HAMMER accomplishes internal simulation using forward and inverse models [Demiris and Khadhouri 2006, Demiris et al. 2014]
  - The inverse model
    - Takes as input the current state of the system and the desired goal, and it outputs the motor commands necessary to achieve that goal
  - The forward model
    - Acts as a predictor
    - Takes as input the motor commands and simulates the perception that would arise if this motor command were to be executed

      (just as the simulation hypothesis envisages)

Internal Simulation and Action



HAMMER Architecture

Demiris and Khadhouri. 2006

Multiple inverse models (B1 to Bn) take as input the current system state, which includes a desired goal, suggesting motor commands (M1 to Mn), with which the corresponding forward models (F1 to Fn) form predictions of the system's next state (P1 to Pn). These predictions are verified at the next time state, resulting in a set of error signals (E1 to En).

Internal Simulation and Action

- HAMMER it provides for the hierarchical composition of primitive actions into more complex sequences
- The action is selected using an internal attention process
- It has been implemented both in robot simulations and on physical robotic platforms

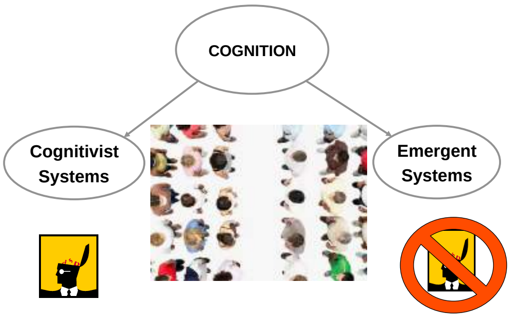
Internal Simulation and Action

- HAMMER goes beyond the scope of episodic memory in effecting internal simulation by invoking actions and behaviours
- The sensorimotor associations involved in internal simulations, for forward and inverse models, requires both **episodic memory** and **procedural memory**
- **Episodic memory** is needed for **visual imagery**, including proprioceptive imagery
- **Procedural memory** is needed **for motor imagery**

Internal Simulation and Action

- Classical treatments of memory (above) usually maintain a clear distinction between
  - Declarative memory and procedural memory, in general,
  - Episodic memory and procedural memory, in particular
- Contemporary research takes a slightly different perspective
  - **Joint perceptuo-motor representations**
  - E.g. Marco Iacoboni's instantiation of *Ideo-motor Theory*
  - *Theory of Event Coding* by Bernhard Hommel and colleagues

Knowledge and Representation



Joint Perceptuo-motor Representations

- Mental imagery – internal simulation – comprises both visual imagery (or, better still, perceptual imagery) and motor imagery
- These two forms of imagery are tightly entwined
  - the **simulation of perception and covert action** both involve elements of visual and motor imagery
- Consider two different approaches:

  1. Theory of Event Coding

  2. Object-Action Complexes

Joint Perceptuo-motor Representations

- **Sensory-motor Theory** and **Ideo-motor Theory** [Stock & Stock 2004]
  - ***Sensory-motor action planning***
    - Treats actions as reactive responses to sensory stimuli
    - **Assumes that perception and action use distinct and separate representational frameworks**
    - Builds on the classic uni-directional approach to perception, proceeding stage by stage from stimulus to percept and then to response
    - **Doesn't allow the resultant (or intended) action to impact on the related sensory perception**

Joint Perceptuo-motor Representations

- **Sensory-motor Theory** and **Ideo-motor Theory** [Stock & Stock 2004]
  - ***Ideo-motor action planning***
    - treats action as the result of internally-generated goals
    - **The selection and control of a particular goal-directed movement depends on the anticipation of the sensory consequence of accomplishing the intended action**
    - The agent images (e.g. through **internal simulation**) the desired outcome and selects the appropriate actions in order to achieve it

Joint Perceptuo-motor Representations

- **Sensory-motor Theory** and **Ideo-motor Theory** [Stock & Stock 2004]
  - *Ideo-motor action planning*
    - There is an important difference between the concrete movements comprising an action and the higher-order goals of an action
    - Actors do not voluntarily pre-select the exact movements required to achieve a desired goal
    - **Instead, they select prospectively-guided intention-directed goal-focussed action, with the specific movements being adaptively controlled as the action is executed**
    - Anticipatory idea-centred way of selecting actions and as a way of bridging the higher-order conceptual representations of intentions and goals with the concrete adaptive control of movements when executing that action

Joint Perceptuo-motor Representations

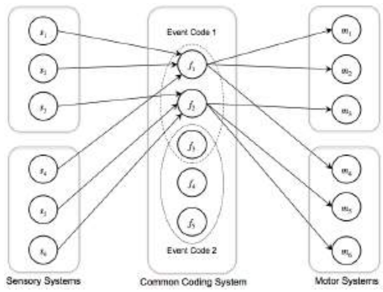
- **Sensory-motor Theory** and **Ideo-motor Theory** [Stock & Stock 2004]
  - **Ideo-motor action planning**
    - How can the goal, achieved through action, cause the action in the first place?
    - How can the later outcome affect the earlier action?
    - **Prospection**! It is the anticipated goal state, not the achieved goal state, that impacts on the associated planned action
    - **Goal-directed action is a centre-piece of ideo-motor theory**
    - Also referred to as the **goal trigger hypothesis** [Hommel et al. 2001]

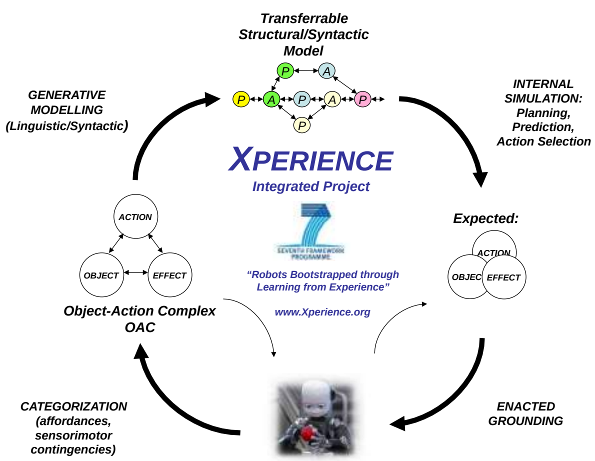
Joint Perceptuo-motor Representations

- **Sensory-motor Theory** and **Ideo-motor Theory** [Stock & Stock 2004]
  - **Ideo-motor action planning**
    - Anticipatory idea-centred way of selecting actions and as a way of bridging the higher-order conceptual representations of intentions and goals with the concrete adaptive control of movements when executing that action
    - **Perception and action share a common representational framework**

Joint Perceptuo-motor Representations

- The Theory of Event Coding (TEC) [Hommel et al. 2001]





Joint Perceptuo-motor Representations

- Object-Action Complex, or OAC [Kruger 2011]
  - An OAC is a triple, i.e. a unit with three components: (E, T, M)
    - E is an "execution specification"; think of it as an action
    - T is a function that predicts how the attributes that characterize the current state of the agent's world will change if the execution specification is executed
      - Think of T as a prediction of how the agent's perceptions will change as a result of carrying out the actions given by E. S is just the space of all possible perceptions of the agent
    - M is a statistical measure of the success of the OAC's past predictions

Joint Perceptuo-motor Representations

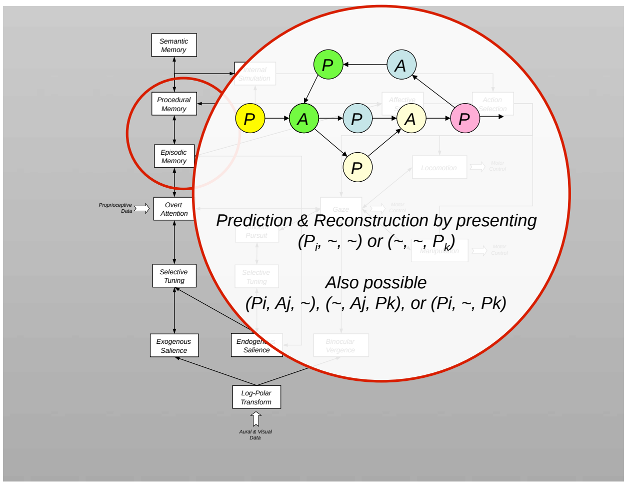
- Object-Action Complex, or OAC [Kruger 2011]
  - **An OAC combines the essential elements of a joint representation with a predictor** that links current perceived states and future predicted perceived states that would result from carrying out that action

Joint Perceptuo-motor Representations

- Object-Action Complex, or OAC [Kruger 2011]
  - **An OAC combines the essential elements of a joint representation with a predictor** that links current perceived states and future predicted perceived states that would result from carrying out that action
  - For example, an OAC might encode how to grasp a object or push an object into a given position and orientation

Joint Perceptuo-motor Representations

- Object-Action Complex, or OAC [Kruger 2011]
  - **An OAC combines the essential elements of a joint representation with a predictor** that links current perceived states and future predicted perceived states that would result from carrying out that action
  - For example, an OAC might encode how to grasp a object or push an object into a given position and orientation
  - OACs can be learned and executed, and they can be combined into more complex representations of actions and their perceptual consequences.



Summer School Reasoning in the Robot World Knowing and acting in an uncertain world a practical view - part 1

Michael Zillich
zillich@acin.tuwien.ac.at

Vision for Robotics Group
Institute of Automation and Control
Vienna University of Technology

Prague, July 30, 2014

Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Introduction

About me

- Primarily vision
- Robotics (indoor)
- ... is a "science of integration" (Tamim Asfour)

About this lecture

- Real world examples, anecdotes
- Do's and don'ts, best practices
- Integration wisdom
- Philosophy

Overview

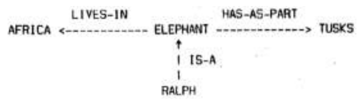
- Lecture 1: Robot knowledge
- Lecture 2: Reasoning and acting
- Lecture 3: Use cases, lessons learned, best practices

A caveat ..

Drew McDermott: "Artificial intelligence meets natural stupidity", SIGART Bull. 57, p. 4-9, 1976

Wishful mnemonics

- UNDERSTAND vs. NODE-NET-INTERESECTION-FINDER GOAL vs. FETCH & TRY-NEXT
- "semantic" nets, ontologies



A caveat ..

Unnatural language

- Natural language says almost nothing
- One term can denote different things "The barn burned to the ground five years ago and was completely rebuilt", or different term denote one thing "morning star" - "evenig star"
- Cognition is terms of language is equivalent to vision in terms of stored images

A caveat ..

L. Wittgenstein, Tractato Logico-Philosophicus, 1921

1 The world is everything that is the case.

1.1 The world is the totality of facts, not of things.

1.11 The world is determined by the facts, and by these being all the facts.

1.12 For the totality of facts determines both what is the case, and also all that is not the case.

5.47321 ... 80 pages

$\Rightarrow$ language defines the world

L. Wittgenstein, Philosophische Bemerkungen, 1940's

"Brick" ..

$\Rightarrow$ language games

A caveat ..

**Only a preliminary version of the program was actually implemented*

- Having identified the shortcoming of version 1 of a program is not equivalent to having written version 2.
- Demo-ism: systems don't generalise from the one case where they worked

Overview: Lecture 1

- Task related knowledge: domain knowledge, process knowledge
- Dealing with real world percepts, uncertainty
- Types of representations: signals vs. symbols

Overview: Lecture 1

- **Task related knowledge: domain knowledge, process knowledge**
- Dealing with real world percepts, uncertainty
- Types of representations: signals vs. symbols

Explicit domain knowledge

What the robot knows and can reason about

- Object instances, classes, locations
- Maps (metric, topological)
- Grasp points
- Colour models
- Labels
- What else?

Implicit domain knowledge

What the developer knows

- Objects are on tables
- Flat floors ...
- No direct sunlight
- Cooperative people
- No pets
- What else?

Implicit domain knowledge

What the developer knows

- Objects are on tables
- Flat floors ...
- No direct sunlight
- Cooperative people
- No pets
- What else?

... assumption invalid

Implicit domain knowledge

What the developer knows

- Objects are on tables
- Flat floors ...
- No direct sunlight
- Cooperative people
- No pets
- What else?

Operating conditions

What the robot is made for

- No humidity, dust
- Traversable terrain
- Reeaching height
- Maximum load
- ...

Process knowledge

How the robot can act

- Low level: path planning (arm, base), speech acts, ...
  = steps of a task
- High level: behaviour
  = how the task is executed as a sequence of steps

Behaviour

- Implicit

```
...
if(numberOfObjects > 1)
  selectObject(objList)
else
  pick(objList[0])
```

- Explicit (e.g. PDDL)

```
(:action move
:parameters (?a - robot ?to ?from - place)
:precondition (and
(= (is-in ?a) ?from)
(connected ?from ?to)
(not (occupied ?to)))
:effect (and
(assign (is-in ?a) ?to)))
```


Where does knowledge come from?

- Human tutor

Learning object models

Human tutor

- Pairwise registration: Estimate relative transformation between views, tracking scene features
- Multi-view refinement: Optimise all camera poses simultaneously, noise filtering [Aldoma ea ICRA 2013]

Learning object models

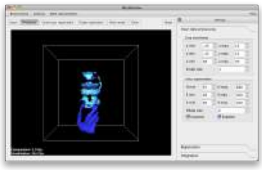
50+ 3D models trained, various texture, shape

Where does knowledge come from?

- Human tutor
- Let the robot find for itself

In hand learning

- Manipulator and Object Tracking for In Hand Model Acquisition [Krainin ea ICRA 2010]
- Point Cloud Library (PCL) support for in-hand learning `http://pointclouds.org/documentation/tutorials/in_hand_scanner.php`



Identifying novelty

- Meta rooms: The static structure of cluttered office environments [Ambrus ea IROS 2014]
- Created through an iterative process where dynamic elements are eliminated from the scene and previously occluded elements are added.
- Once a meta-room has been created, dynamic elements can easily be extracted from new observations
- Based on point cloud differencing

Identifying novelty

(a) Metaroom and dynamical structure

(b) Detected objects

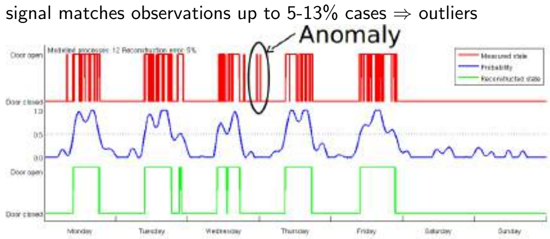
Periodic changes

FREquency Map ENhancement (FREMEN) [Krajnik ea ICRA 2014]

- Lots of environment changes are caused by humans
- Often follow periodic schedules
- Fourier analysis on binary variables: approximate observations with n harmonic functions
- E.g. open/closed office door, 30Hz x 7 days 18 000 000

Periodic changes

- signal matches observations up to 5-13% cases ⇒ outliers



Periodic changes

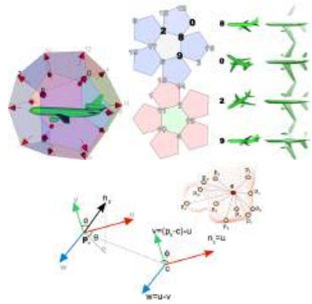
- Predict visual appearance of the environment (in terms of visual features) and use the predicted model for topological localization

Where does knowledge come from?

- Human tutor
- Let the robot find for itself
- Learning from the web

Learning from the web - object classes

- RGBD Object categorisation [Wohlkinger ea 2012]
- Train from 3D CAD models, web database
- Extract 3D shape descriptors per view VFH [Rusu ea 2010], CVFH [Aldoma ea 2011], SDVS [Wohlkinger ea 2011], ESF [Wohlkinger ea 2011]), SHOT [Tombari ea 2010]
- k-NN classifier (k=10), confidence = ratio of majority vote
- Up to 200 classes



Learning from the web - object classes

Real time object classification (3x)

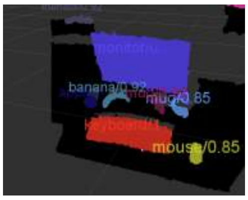
Learning from the web - object classes

Classification rate around 80% (200 classes) [Wohlkinger ea 2012]

class name	1-NN	10-NN	confusing class
per scenes OVERALL	58.22 %	78.23 %	
per class OVERALL	49.10 %	71.39 %	
apple	81.40 %	98.45 %	pumpkin
banana	54.79 %	69.86 %	pistol
bottle	48.77 %	79.01 %	suv
bowl	50.00 %	76.47 %	hat
car	11.52 %	43.64 %	suv
donut	20.00 %	62.00 %	cap
hammer	83.41 %	96.10 %	axe
mug	91.96 %	99.46 %	watch
tetra pak	47.09 %	72.09 %	mug
toilet paper	2.11 %	16.84 %	armchair

Learning from the web - object classes

Classifiying a table top



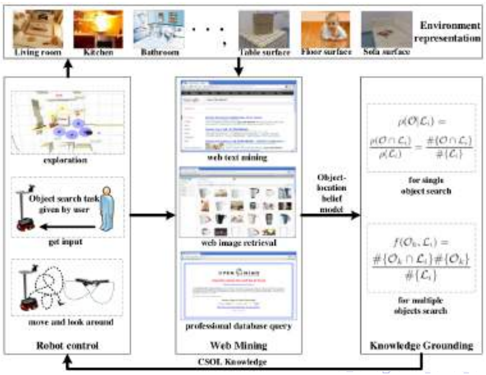
Learning from the web - object classes

Classification in the wild

- Segmentation
- Object size / distance
- Degenerate views
- Database with relevant objects

Learning from the web - object locality

Common sense knowledge about object locality (CSOL) [Zhou ea 2012]



Learning from the web - object locality

Most and second most likely location

Object	Living room	Kitchen	bedroom	bathroom	Dining room	Office	Corridor
suitcase	34,41 %	0,24 %	23,59 %	1,43 %	0,48 %	0,60 %	39,26 %
soap	0,00 %	2,95 %	0,33 %	96,48 %	0,00 %	0,23 %	0,00 %
snack	6,98 %	18,72 %	0,85 %	2,02 %	18,44 %	9,00 %	0,00 %
radio	39,00 %	18,34 %	24,06 %	9,97 %	5,62 %	3,01 %	0,00 %
lamp	24,42 %	2,01 %	50,33 %	5,81 %	8,97 %	2,63 %	5,82 %
cushion	23,05 %	3,50 %	11,49 %	0,00 %	1,75 %	8,21 %	0,00 %
jacket	0,92%	3,56%	10,51%	14,80%	51,36%	18,85%	0,00%
cereal	1,22 %	52,93 %	0,00 %	4,88 %	9,76 %	1,22 %	0,00 %
candle	6,13 %	2,56 %	14,48 %	34,34 %	1,85 %	0,65 %	0,00 %
pillow	16,79 %	0,78 %	19,22 %	1,38 %	9,35 %	1,98 %	0,00 %
handbag	0,00 %	0,20 %	1,36 %	0,00 %	0,00 %	7,13 %	41,30 %
magazine	11,60 %	8,17 %	6,04 %	33,20 %	0,00 %	41,09 %	0,00 %
dish	3,83%	67,53%	2,94%	12,28%	12,83%	0,59%	0,00%
bra	4,54%	0,45%	18,68%	19,71%	0,00%	6,61%	0,00%
keyboard	15,15%	1,86%	3,14%	6,46%	3,26%	22,12%	0,00%
pot	5,93%	54,99%	2,90%	25,83%	1,88%	1,46%	7,07%
printer	21,12%	12,11%	11,21%	1,00%	0,43%	54,13%	0,00%
toy	63,32%	2,64%	21,28%	4,18%	0,00%	8,58%	0,00%
underwear	2,97%	3,40%	15,16%	66,52%	6,44%	5,51%	0,00%
guitar	16,73%	3,90%	10,62%	7,20%	4,48%	3,54%	3,51%
laptop	17,89%	7,53%	50,23%	4,44%	2,95%	16,97%	0,00%
wine	11,25%	62,74%	0,93%	2,50%	20,50%	2,08%	0,00%
briefcase	0,16%	0,16%	0,00%	1,74%	0,00%	3,48%	44,46%
bag	31,59%	20,62%	12,36%	12,43%	1,56%	8,43%	13,01%
alarm clock	1,57%	30,21%	45,20%	2,51%	0,00%	0,52%	0,00%
cherry	15,52%	16,26%	3,24%	13,38%	3,60%	0,00%	0,00%
cockroach	0,91%	2,73%	0,91%	38,18%	1,82%	0,00%	5,45%
card	4,98%	6,62%	6,39%	33,50%	2,07%	42,11%	4,33%
book	22,30%	9,62%	47,58%	5,93%	4,57%	9,95%	0,00%
ipad	8,67%	27,71%	49,94%	3,74%	0,63%	9,31%	0,00%
box	10,34%	4,80%	8,91%	6,22%	48,34%	7,15%	14,24%
kid	29,54%	23,32%	6,99%	15,39%	1,66%	19,23%	3,47%
baby	22,22%	10,07%	41,13%	17,30%	0,21%	5,04%	4,02%

Overview: Lecture 1

- Task related knowledge: domain knowledge, process knowledge
- **Dealing with real world percepts, uncertainty**
- Types of representations: signals vs. symbols

Icoming percepts - Uncertainty

- Probabilitiy theory
  - Discrete distributions: histograms
  - Continuous distributions: parametric (typ. Gaussian) and parameter free (particle filter)

Icoming percepts - Uncertainty

- Probabilitiy theory
  - Discrete distributions: histograms
  - Continuous distributions: parametric (typ. Gaussian) and parameter free (particle filter)
- Dempster Shafer theory
  - Can be used to collect evidence from different sources in sensor fusion
  - Degrees of belief as masses, that are spread over the power set of states
  - Belief (sum of evidence in favour) $\leq$ plausibility (sum of evidence against)

Hypothesis	Mass	Belief	Plausibility
Null (neither alive nor dead)	0	0	0
Alive	0.2	0.2	0.5
Dead	0.5	0.5	0.8
Either (alive or dead)	0.3	1.0	1.0

  - Inference computationally expensive

Improbable probabilities ...

- Not each number $\in [0, 1]$ is a probability
- Confidences, residuals, fitness, . . .
- Precise meaning (prob. that object is present, prob. that object is known, prob. of object position)

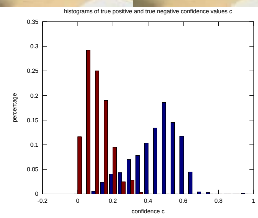
Learning meaningful probabilities

- When have we learned the model well enough to guarantee detection success with a given probability?
- Problem: confidence c is not a probability
- What is a meaningful threshold?
- Generative detector model [Meger ea 2010]:
  probability of observing a specific confidence given successful or failed detection

$$p(c \mid o = true)$$
$$p(c \mid o = false)$$

Learning meaningful probabilities

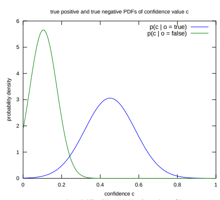
- Virtual training examples (1000 rotations, 252 scales, noise and blur)
- Create true positives and true negatives
- Histograms for true positives (blue) and true negatives (red)
- Estimated Gaussian PDFs of confidence for true positives (blue) and true negatives (green)

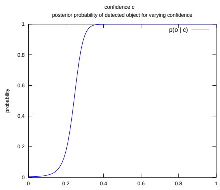


Learning meaningful probabilities

- Estimated Gaussian PDFs of confidence for true positives (blue) and true negatives (green)
- Observed Detection Success = probability of having object detected given observed confidence [Zillich ea IROS 2012]

$$p(o \mid c) = \frac{p(c \mid o)\ p(o)}{p(c)} = \frac{p(c \mid o)\ p(o)}{\sum_{k \in \{t,f\}} p(c \mid o = k)} \quad (1)$$





Learning meaningful probabilities

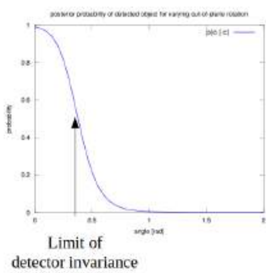
Predicted detection success

- Same as ODS, but conditioned on view angle: probability of observing a specific angle given successful or failed detection

$$p(\theta \mid o_j = true)$$
$$p(\theta \mid o_j = false)$$

- Probability of detecting an object view j for given angle

$$p(o_j \mid \theta) = \frac{p(\theta \mid o_j)\ p(o_j)}{p(\theta)} = \frac{p(\theta \mid o_j)\ p(o_j)}{\sum_{k \in \{t,f\}} p(\theta \mid o_j = k)}$$

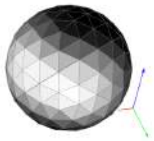


Learning meaningful probabilities

Predicted detection success

- Having learned a set of views, what is the probability of detecting the object from any angle?
- Sum over sphere

$$\hat{p}(o) = \sum_{\theta,j} p(o_j \,|\, \theta) p(\theta)$$



Overview: Lecture 1

- Task related knowledge: domain knowledge, process knowledge
- Dealing with real world percepts, uncertainty
- **Types of representations: signals vs. symbols**

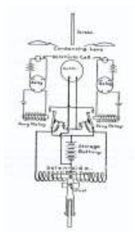
Signals vs. symbols

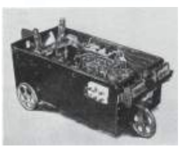
- Symbols = chunks of the world we cut out
  - Where to cut?
  - Uncertainties
- Just stay in signal level?
- . . . historic debate

First true robot in history – guesses anyone?

Seleno the Electric Dog

- John Hammond, Benjamin Miessner (1912)
- Phototaxis (move to or avoid light)
- Technical realisation of theories of phototaxis (heliotropism) in animals by Jacques Loeb
- "... inherit almost superhuman intelligence"
- ".. the dog promptly, almost fearfully, backed away."
- "Mechanical creature that will fight burglars, sweep and dust and be generally useful."





[http://cyberneticzoo.com]

Turtle Bots

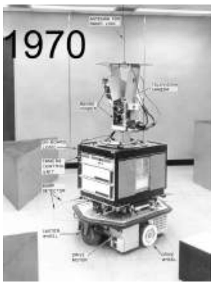
- William Grey Walter (ca. 1950)
- Turtles Elmer and Elsie
- Phototaxis to find recharging station
- "social interaction" between 2 robots

[http://cyberneticzoo.com]

… to symbols

- Stanford Cart, SRI Shakey
- Computer controlled
- From simple behaviours towards symbolic AI
- Sense-think-act

[Brooks 1986]



[Nilsson 1984]

… and back again

- Rodney Brooks (1986): decomposition by activity
- Subsumption architecture: higher layers subsume lower layers, always complete control system
- Intelligence without representation (1991): The world is its own best model

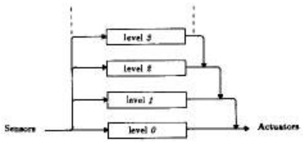
Ghengis

[Brooks 1986]

… and back again

- Rodney Brooks (1986): decomposition by activity
- Subsumption architecture: higher layers subsume lower layers, always complete control system
- Intelligence without representation (1991): The world is its own best model

Ghengis



[Brooks 1986]

… and again …

- Intelligence with representation (Steels 2003)
- Behaviours don’t scale up – higher level tasks require some form of representation / conceptualisation
- Where do representations / concepts come from? How do they acquire meaning (for a robot(s) and its user)?
- External representations instead of purely abstract entities
- “Representations …organisers of activity rather than abstract models of some aspects of reality” (example of trodden path on lawn)

Learning object models

The talking heads experiment (Steels ea 2002)

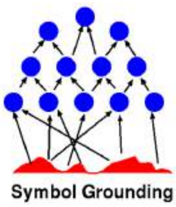
… and again …

The Human Brain Project, H2020 Flagship, 1 billion EUR

- “A key goal of the Human Brain Project is to construct realistic simulations of the human brain this will require molecular and cellular information and from that we will be able to model and understand biological and medical processes. In addition, we will be able to use that information to design and implement new kinds of computers and *robotics*.” Prof. Seth Grant, University of Edinburgh

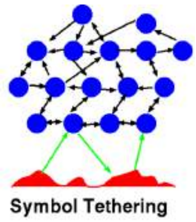
Signals and symbols

- Symbol grounding (Stevan Harnad, 1990)
  - "... every symbol gets its meaning from below (i.e. sensory-motor, information)."
  - Simple concepts abstracted directly from experience of instances.
  - Complex concepts constructed from simple concepts using logical and mathematical methods of composition.



Signals and symbols

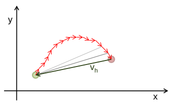
- Symbol tethering (Aaron Sloman, 2007)
  "... much of the meaning of a symbol comes from its role in a rich theory, expressed in a formalism that allows conclusions to be drawn in a formal way. Tethering merely adds further constraints that help to pin the meaning down."

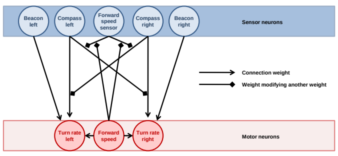


[Sloman 2007]

As low as it gets: Ant navigation

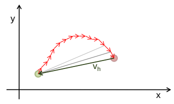
- Biologically plausible path integration model [Vickerstaff 2005]
- Evolved CTRNN neural network model
- Reach goal (food) using camera, and return to the starting location (nest) without the use of landmarks, while avoiding obstacles [Papauschek ea ROBIO 2010]

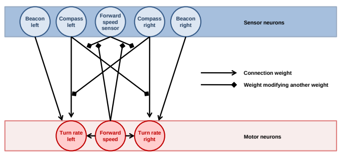




As low as it gets: Ant navigation

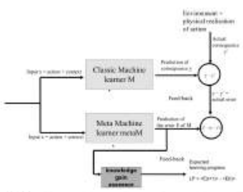
- Biologically plausible path integration model [Vickerstaff 2005]
- Evolved CTRNN neural network model
- Reach goal (food) using camera, and return to the starting location (nest) without the use of landmarks, while avoiding obstacles [Papauschek ea ROBIO 2010]

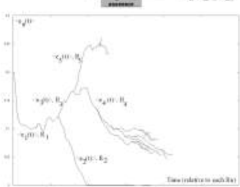




Curiosity driven learning

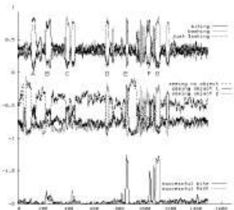
- Infants don't learn randomly, they seek out interesting situations and repeat until bored
- Curiosity as intrinsic motivation – Intelligent Adaptive Curiosity [Oudeyer et al. 2007]
- Evolution of prediction error rate drives exploration: where error decreases over time





Curiosity driven learning

- Play pen experiment with Aibo dog: looking at objects, bashing, biting
- Developmental stages with predominant behaviours
- Increasing complexity
- Stages of development

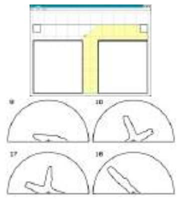


... to symbols: Bootstrapping representations

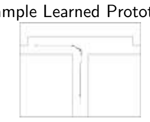
- Intentionality Problem: Where does meaning come from?
- Firehose of Experience: "... extremely high bandwidth stream of sensor data that the agent must cope with, continually." [Kuipers 2008]
- Trackers point from spatio-temporal region of the input stream to stable symbolic representation
- Kuipers (2000, 2008): Spatial Semantic Hierarchy (SSH)
- Distinctive states defined by behaviours of the dynamical system given by the agent, its environment and control laws
- From raw uninterpreted sensor data to navigation in topological maps via several hierarchies of abstraction

Bootstrapping representations: abstracting

- Provost et al (2006): Self-Organizing Distinctive State Abstraction (SODA)
- Reduce "problem diameter"
- Learn prototypical situations via vector-quantization, define perceptual neighborhood
- Hill climbing to distinctive state
- Trajectory following from one dist. state into neighborhood of another



Example Learned Prototypes



Navigation using learned abstraction

Overview

- Lecture 1: Robot knowledge
- Lecture 2: Reasoning and acting
- Lecture 3: Use cases, lessons learned, best practices

Summer School Reasoning in the Robot World Knowing and acting in an uncertain world a practical view - part 2

Michael Zillich
zillich@acin.tuwien.ac.at

Vision for Robotics Group
Institute of Automation and Control
Vienna University of Technology

Prague, July 30, 2014

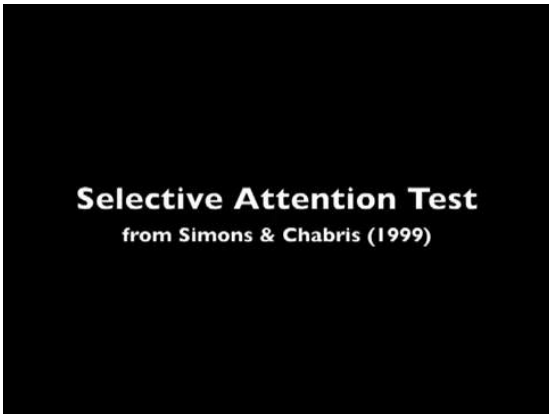
Presented at ReaRW 2014 summer school in Prague, July 29-31, 2014
http://summerschool2014.ciirc.cvut.cz/

Attention!

- Kuiper's "firehose of experience"
- Acting in the real, dynamic, confusing world is about selecting what to attend
- Biological systems do
  - In "hardware" - foveated vision
  - Bottom up saliency
  - Top down attention

A test for you

- Test showing the necessity and effectiveness of attention for the human visual system
- In the following video, count how many times the players wearing white pass the basketball
- Just observe and count silently, don't distract the other participants
- Ready ...?



Around 50% of people don't see it ...

Attention!

- RGBD object segmentation ... up to several seconds
- How fast is fast enough?

From coluoured point clouds .. .. to separated object hypotheses

[Richtsfeld ea JVCI 2014]

Another test

Look at the scene ...

Another test

- How many boxes?
- How many objects had red in them?
- Was the laptop turned on?
- How many books?
- Speed of processing in the human visual system [Thorpe ea 1996]: ca. 150 ms to get scene gist

Attention!

- How fast is fast enough?
- $\Rightarrow$ As long as you have
- Anytime algorithms: return some result fast, better results with more time
- Graceful degradation
- “Attentive robot systems”: shift attention and processing very dynamically

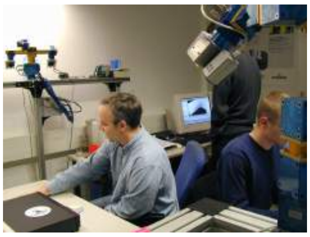
Cognitive system architecures in practice

- ActiPret (EU project, TUW coord.)
- CoSy (EU project, KTH coord.)
- CogX (EU project, Birmingham coord.)

ActiPret

Project goals

- Observe human manipulating objects
- High level description: activity plan
- Feedback on correct plan execution



ActiPret

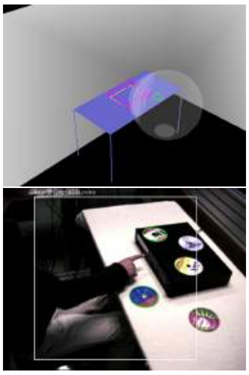
Requirements

- System is on-line with the observed scene: react in the temporal order of activities observed (soft real time) with restricted resources
- Solve several tasks concurrently
- Tasks are complex in terms of the methods and combinations applied and require control and integration of methods
- Pro-activity: task-driven processes take initiative in focusing resources
- Scalability: avoid central control component, control on per-component basis
- Modularity: for re-use and dynamic resource allocation

ActiPret

Approach

- Distributed processing
- Focus processing on relevant scene parts
- Top down control
  - Higher level components request (specific) information of lower level components
  - Proceeds to lower levels
  - Spaces Of Interest
- Active vision: robot mounted cameras



ActiPret

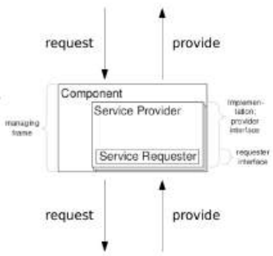
zwork (middleware)

- Hierarchy principle
  - Top down control
  - Pro-activity
- Service Principle
  - Modularity
  - Re-use
  - Resource management

ActiPret

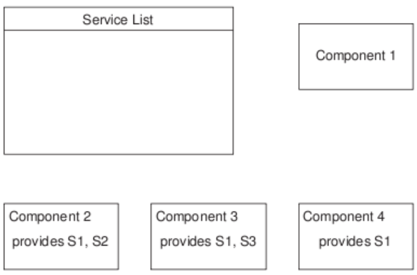
Hierarchy principle

- Control flows from top to bottom
- Data flows from bottom to top
- Data consistency: Requester and provider know about their data, requester responsible for fusing data from multiple providers



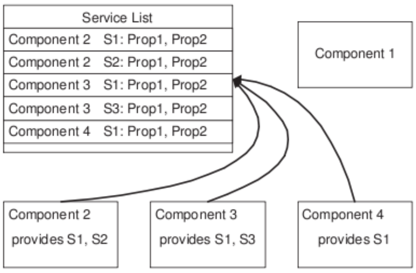
ActiPret

System state before any services are offered



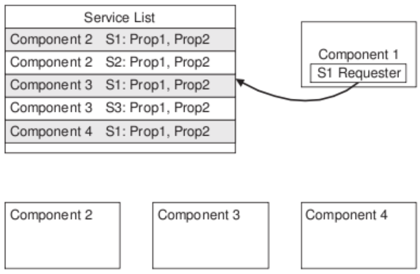
ActiPret

Components offer their services with current properties



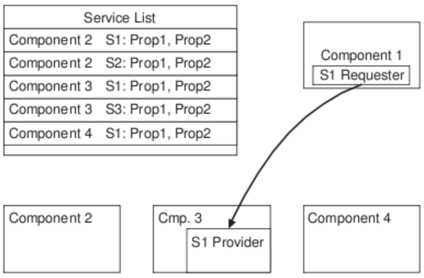
ActiPret

Service requester asks for services with specific properties (e.g. slow and accurate vs. fast)



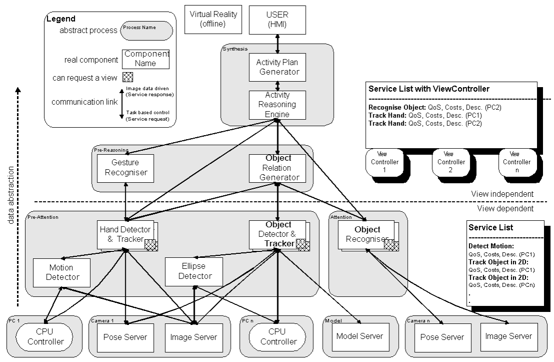
ActiPret

Service requester establishes a link to the selected service provider



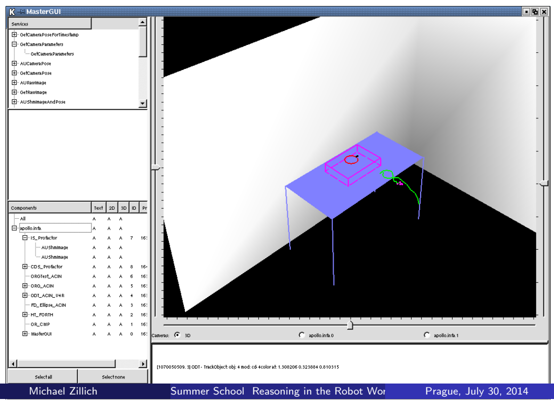
ActiPret

Final system overview



ActiPret

Final system overview



ActiPret

Pros

- Clear flow of control
- Data consistency
- Modularity, reuse

Cons

- Control is implicit, initiated inside modules
- Difficult to inspect current knowledge of the system (only GUI)

CoSy Architecture Schema (CAS)

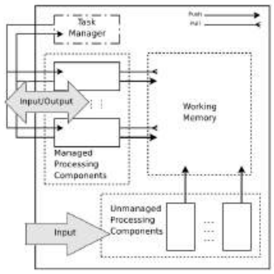
CAST, the CAS Toolkit

- Parallel and incremental information processing
- Distributed data and processing
- Maintain coherent view of changing world
- Binding information from distinct representations

CoSy Architecture Schema (CAS)

Subarchitecture (SA)

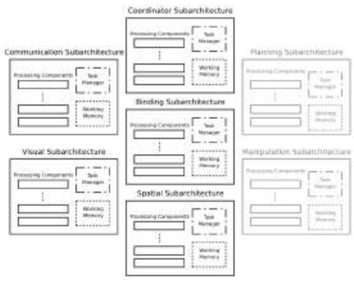
- Modules that are “close” (type of data, temporal granularity of data, machine . . .)
- Working Memory (WM) as (semi) central storage
- Frames in WMs
- Distinction between “dumb” data movers and “smart” managed components
- Distributed blackbaord-ish control



CoSy Architecture Schema (CAS)

Complete system instantiation

- WM to WM communicaton of selected data items
- Central binding subarchitecture



CoSy Architecture Schema (CAS)

Cross-subarchitecture Binding

- Binding WM mirrors all other WMs
- *Binder* receives *candidates* form other SAs
- Abstraction and filtering of candidates from their source domain: binding *monitor* in each SA
- Binder binds candidates into *instance bindings* = the system’s representation of objects

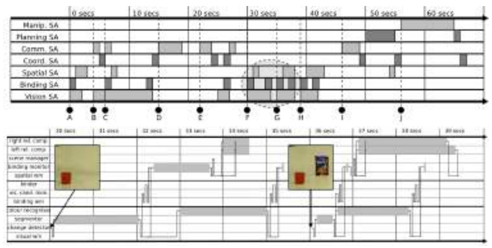
CoSy Architecture Schema (CAS)

Cross-subarchitecture Binding

- Common language between binder and SAs: *features* and *relations*
- Features: shape, colour, position
- Relations: e.g. spatial relations between objects
- Features in common with features of other modalities ⇒ common ground
- Binder compares features
  - No match ⇒ new instance binding
  - One match ⇒ bind to that existing instance binding
  - Several matches ⇒ disambiguation problem
- Each object in symbolic plan represented as reference to instance binding
- *lazy binding*: “the **blue things** near the green thing” ⇒ bound during execution

Attention!

- A: put down red, B: (T) “This is a red thing”
- C: put down blue, D: (R) “Is that red?”
- E: (T) “No, this is a blue thing”
- G, H: put blue, I: (T) “Put the blue things to the left of the red thing.”



CoSy PlayMate

More human robot interaction (Explorer scenario)

CoSy Architecture Schema (CAS)

Pros

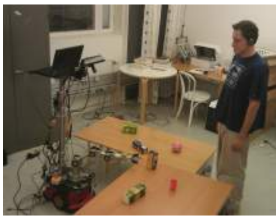
- Balance between local and global storage and control
- Blackbaord nicely supports concurrency
- Two scenarios (PlayMate and Explorer) with the same system

Cons

- Keeping data consistent (write locks . . .)
- Division into subarchitectures ad-hoc

CogX (CAST)

- Cognitive Systems that Self-Understand and Self-Extend [Hawes ea TAMD 2010]
- Modelling knowledge and knowledge gaps
- Reuse of CAST
  - + Explicit modelling of beliefs
  - + Motivation
  - + Markov Logic Network (MLN) for binding
- 3 scenarios: Dora the explorer, Curious George, Dexter
- George: interactive learning in mixed initiative dialogue



CogX (CAST)

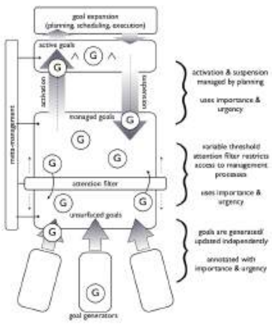
Beliefs

- Form a-modal representation of multi-model multi-agent (tutor and robot) facts of the world
- Belief = multivariate probability distributions over feature-value pairs (e.g. color = blue)
  - *Private beliefs*: robot perceptions of the environment
  - *Attributed beliefs*: information that robot attributes (but not necessecarily agrees) to another agent, after communication
  - *Shared beliefs*: between human and robot
- Drive the unfolding of dialogue

CogX (CAST)

Goal management

- Multiple, possibly interleaved, goal-directed activities
- *Goal generators*: concurrently active processes which monitor the systems and produce goals to satisfy the systems drives (fill knowledge gaps)
- *Filters* surface goals: quick, coarse selection
- *Managers* activate goals: selection of surfaced goals
- Filters and managers guided by goal *importance* and *urgency*



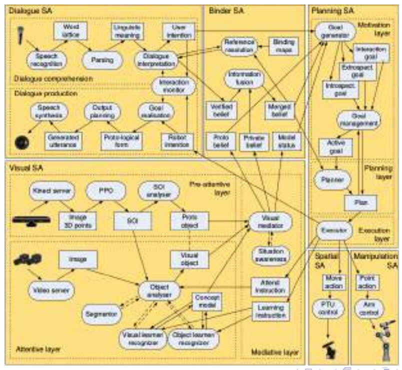
CogX (CAST)

Goal priority levels (filter heuristic)

- Interaction drive:
  - Answering the tutor's requests
  - Situated tutor-driven learning: (T) "This is a red object"
- Extrospection drive
  - Attention mechanism: a new object pops up
  - Situated autonomous learning: (R) "I know this is red, I will update my model"
  - Situated tutor-assisted learning: "Is the color of this object red?"
  - Exploring the scene: Pan-tilting around the table
- Introspection drive
  - Non-situated tutor-assisted learning: (R) "Could you show me something red?"

CogX (CAST)

George system overview



CogX (CAST)

CogX (CAST)

Pros

- Explicit reasoning over epistemic states
- Solid grounding of language
- Robust, revovery from failures

Cons

- Still problem of keeping data consistent
- Precise flow of control difficult to follow

STRANDS

STRANDS (EU project, Univ. of Birmingham, 2013 - 2017)

- Long term robot autonomy: weeks and months auf autonomous data gathering
- ROS middleware, smach state machine
- Using MongoDB as central data storage
- industry grade object oriented database
- Ensures data consistency, mirroring, caching od parts of the database

The killer of many a demo ...

Never sleep!

- Rate r(10);
- while(ok())
  - ... do some work, publish some messages, etc. ...
  - spinOnce();
  - r.sleep();

Only use callbacks!

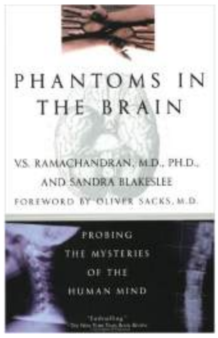
- Don't use sleeps to overcome racing conditions ("just wait a bit until I am sure the data has arrived, like 10 ms")
- Waits pile up.
- In the full load system (i.e. the review demo), the timing will be all off.
- Process switching is expensive, around 30 microseconds.
- Busy waiting is sin.

So much for robots . . .

A Human Example

V. S. Ramachandran: Phantoms in the Brain, 1998

- “The sound of one hand clapping”
- Anosognosia: denial of desease (lit.)



Thank you!