Pervasive Pose-Awareness for People, Objects, and Robots

or ...

Empowering Robots to Clear Your Dining-Room Table

Seth Teller
LCS/AI

April 30, 2003

Overview

This idea may or may not be dangerous, but...

I propose a new infrastructure, with the potential to change:
- how humans interact with (information about) the world
- how robots plan and execute their actions
- how ordinary objects behave

An analogy about human navigation

People navigate conventionally using:

- Experience (possibly collective, e.g. maps)
- Continuity (starting points, dead reckoning, landmarks)
- Environment and device infrastructure
  - Sun, moon, stars
  - Compass, sextant, clock

A powerful, existing navigation infrastructure

In the 1970's and 1980's, the US military designed and deployed the Global Positioning System infrastructure:

- Dozens of satellites in non-geosynchronous orbits
  - Each with an atomic clock and radio transceiver
  - Each broadcasting its time and position values
- Dozens of fixed ground tracking stations
  - estimating satellite positions to centimeter accuracy
- GPS receiver hearing four or more satellites can solve for
  - the time t and its location (x, y, z) in Earth coordinates

Today, an inexpensive GPS receiver can report its coarse position (accurate to tens of meters) in open outdoor areas

Non-military applications of GPS

After GPS deployment in 1989, non-military applications emerged:

- Civilian navigation (land, marine, air)
- Precision surveying and aerial photography
- Shipping and supply-chain management
- Resource exploration
- Precision agriculture
- Environmental and wildlife monitoring
- ... and many others

Robotic uses of GPS include:

- Drone aircraft
- Satellite orbital maintenance
- Aircraft auto-pilots, hands-free landings

Things (e.g., shipping containers, rental cars, backpacks) can log or report their GPS location for supply-chain management, contract enforcement, tracking prisoners, finding children etc.

We propose an analogous infrastructure:

	GPS	Proposed infrastructure
Coverage	Outdoors, far from buildings	Indoors, and outdoors close to buildings
Receiver reports	3-DOF position in Earth coordinates	6-DOF position & orientation in locally defined coordinates
Position accuracy	Coarse (tens of meters)	Fine ($\approx$ 1 centimeter)
Orientation accuracy	n/a	Fine ($\approx$ 1 degree)
Receiver technology	Battery, ASIC	Passive, MEMS+ASIC

(More details about how to actually build this, later in the talk)

Functional geometric models

Collection of space descriptors, associated with:

- Geometric delineation of space boundaries
- Space type info & metadata (office, cave, flooring type, etc.)
- Space adjacency info (for human, robotic traversal)
- Space's contents (resources of interest, app-specific)

Compelling applications

Putting it all together, what are the compelling applications?

- Navigation, resource location
- Direct (in situ) environment annotation
- Direct model population, correction
- Direct Information Overlay (Augmented Reality)

Navigation, resource location

Resource location, route-finding

Hand-held display of map, your position and orientation, and how you should move next to get to where you want

Similar to car-navigation systems now in use outdoors

Direct (in situ) environment annotation

User points to (fixed) location, adds metadata to database

These "virtual" annotations are

- Persistently stored, searchable in a spatial, relational database
- Readable only with permission (example: security audit)

Contrast with conventional annotations (e.g. dig-safe), which are:

- Not persistent (erosion, new construction, new annotations)
- Publically readable by default
- Can only be read if user is physically present

Direct model population, correction

Imagine adding a laser range-finder to hand-held device
- Can populate objects to model simply by pointing at them
- Can correct model in situ, producing as-built CAD documents

Contrast with conventional model construction and population
- Someone sits at a desk, with a CAD program and mouse-based GUI,
  - far from the structures and objects of interest,
  - unsure of the relationship between model and reality

Direct Information Overlay (Augmented Reality)

Imagine adding a digital projector to hand-held device

Enables direct (in situ) information overlay:

- Text metadata and instructions
- Assembly/disassembly diagrams
- Hidden infrastructure (wiring, piping, etc.)
- With two devices, can “draw” directly onto the environment

Contrast with conventional “augmented reality” using HMD

How to Implement this Infrastructure?

Pose-awareness: Cricket beacons and listeners

Fixed beacons, emitting RF, ultrasound (US) pulses

- RF encodes beacon location, optional space information

Mobile listeners, receive RF & US, on-board CPU

- Compute distance to beacon from RF, US arrival times
- From 4 or more beacons, compute (x, y, z) and sound speed

Extended listener computes attitude from multiple ultrasound receivers, difference in arrival phase

Pose-Aware Radio-Frequency Identity (RFID) Tags

Conventional RFID tags report identity bits when queried

- Application retrieves object metadata from separate database

Imagine augmenting RFID tag with passive MEMS device

- Reports 6-DOF transform from (factory-assigned) object coordinates to (infrastructure-assigned) world coordinates

How is this "pose-aware" object newly useful?

- From object's ID, we can fetch its metric CAD description
- Pose-awareness enables us to situate it in world coordinates

Robotic Application: Scenario

Deploy an autonomous robot with articulated body and graspers
  Direct it to clear your dining room table and put dishes in dishwasher

Conventional approach:
  Use sonar & vision to simult. localize robot, map environment
    and obstacle avoidance alg'ms to get from home position to table
  Once successfully at the table, use computer vision to:

1. Segment objects from background and each other
2. Recognize objects from catalog, using stereo or active ranging
3. Use motion-planning to move articulated arm into position
4. Use grasp control, force feedback to grasp/remove object

Of those four sub-problems, only motion-planning is well-solved!
  Most would agree that the others are farther from solution.

Alternative: Pose-aware scenario

Now imagine that your house has the proposed infrastructure,
and that *every object* in the house is pose-aware

1. Robot, as it moves, repeatedly queries environment:
Where am I? Where is the dining room table?
What objects are close to me and might obstruct me?

2. Implicated objects report back appropriately
enabling robot to fetch each object's CAD descriptions, default pose

3. Robot uses motion planning algorithm to approach the table

4. Once at the table, robot queries environment again:
All objects above (known) table surface, report yourselves

5. Robot then computes:
Support relationships among objects
A feasible order in which to remove them
Appropriate reaching and grasping plan for each object,
from the object's (known) material properties (size, shape,
mass distribution (moments), stiffness, coefficient of friction)

Observation

This doesn't make the problem easy, or even necessarily solvable

But it does eliminate the hardest aspects,
and so should make the problem significantly easier!

Potential Objections to this Approach

Technology doesn't yet exist to make suitable pose tags

This is an engineering issue that can be pursued over time

Even if pose tags existed, and were pervasively deployed,
you would also need tags on your kids, your dog, etc.

These are non-rigid objects, so the problem gets harder

Could enforce a "safety zone" around anything alive

The infrastructure deployment effort would be substantial

So it was for rural electrification, interstate highways etc.

This is a practical, not a theoretical, objection

Scalable Infrastructure Deployment

Incorporate pose tags into every object during manufacture

How to deploy pervasive networks of cricket beacons?

- Incorporate beacons into building materials (surfaces, fixtures)
- Have human installers sprinkle beacons throughout environment

After deployment, each beacon must be configured with its own pose

- Manual configuration won't scale; self-configuration alg'ms needed
- Beacons form ad hoc network, range to neighbors, iterate coords
- LCS TR 825, April 2003 (w/ Priyantha, Demaine, & Balakrishnan)

Autonomous Robotic Deployment

Imagine beacons as small, mobile devices able to attach to surfaces (Examples: McLurkin's swarm robots, Rus's reconfigurable robots)

Think of beacon network as a single, self-configuring entity, extending pseudopods to "coordinatize" new spaces

Three nice side effects of this model:

1. Robots deployed densely enough *are* a map of environment
2. Robots can maintain a "live" model of environment (& changes) by swarming to areas of construction, demolition, or renovation
3. Adding sensors (temperature, light, occupancy, etc.) to bots produces a real-time distributed, location-aware sensor network

Conclusions

Infrastructure for pervasive "pose-awareness"

Sketched devices, applications, robot scenarios

Proposed one plausible infrastructure implementation

Autonomous deployment strategies

Barriers to realization

Discussion ?