In V. Mittal, H. Yanco, J. Aronis and R. Simpson (Eds.),
Assistive Technology and Artificial Intelligence.
Lecture Notes in Computer Science, Springer Verlag, New York, 1998.

Integrating vision and spatial reasoning for assistive navigation*

William S. Gribble[1], Robert L. Browning[2], Micheal Hewett[2], Emilio Remolina[2]
Benjamin J. Kuipers[2]

[1] Dept. of Electrical and Computer Engineering, The University of Texas at Austin
[2] Dept. of Computer Sciences, The University of Texas at Austin

Abstract. This paper describes the goals and research directions of the University of Texas Artificial Intelligence Lab's Intelligent Wheelchair Project (IWP). The IWP is a work in progress. The authors are part of a collaborative effort to bring expertise from knowledge representation, control, planning, and machine vision to bear on this difficult and interesting problem domain. Our strategy uses knowledge about the semantic structure of space to focus processing power and sensing resources. The *semi-autonomous* assistive control of a wheelchair shares many subproblems with mobile robotics, including those of sensor interpretation, spatial knowledge representation, and real-time control. By enabling the wheelchair with *active vision* and other sensing modes, and by application of our theories of spatial knowledge representation and reasoning, we hope to provide substantial assistance to people with severe mobility impairments.

1 Introduction

The Intelligent Wheelchair Project is working to build an assistive agent capable of autonomous and semi-autonomous navigation in an initially unknown, dynamic world. We address this general goal by focusing on the specific problem of providing intelligent navigation assistance to profoundly mobility-impaired people. People with moderate motor control can operate a standard powered wheelchair via a variety of mechanical and electronic interfaces. However, many people do not have enough motor control to operate such interfaces reliably or at all, and can be left dependent on others to make even the most basic movements from one place to another. Those with enough motor control to operate mechanical or joystick interfaces, but who are spastic or have some perceptual problems, may be unable to navigate in difficult environments without putting themselves in danger. The community of wheelchair users that will benefit most from our

* This work has taken place in the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Qualitative Reasoning Group is supported in part by NSF grants IRI-9504138 and CDA 9617327, by NASA grant NAG 9-898, and by the Texas Advanced Research Program under grants no. 003658-242 and 003658-347.

approach are those who do not have enough motor control to steer a wheelchair using traditional interfaces, or who can control a wheelchair to a limited extent but wish to have some assistance with difficult, tedious, or high-precision navigation tasks. The problem of assisting a user who has high cognitive function but severe motor and communication problems can be characterized as a case of taking noisy, error-prone, low-bandwidth control information from the human driver and providing smooth closed-loop control of the wheelchair's motion in service of specific navigation tasks.

1.1 Augmented wheelchair capabilities

We have built a mobile experimental platform with real-time vision from off-the-shelf computing hardware and a customized electric wheelchair. It has a stereo vision system with independent pan and tilt controls for each camera, an on-board computer, a laptop for user feedback and control, and a small embedded system to manage the wheelchair's sensor and motor hardware.

The wheelchair base is a Tinman II from the KISS Institute for Practical Robotics.[3] This is a Vector Velocity wheelchair, retrofitted with twelve infrared proximity sensors, seven sonars, and a small embedded computer which manages the drive systems and collects input from the sensors.

The principal on-board computer is a dual processor Pentium Pro machine running Debian Linux. Two frame grabber cards allow us to acquire images for dual-monocular or stereo image processing. The available computing power is sufficient to do substantial image processing on-board. User interaction (other than joystick commands) is handled through a laptop, also running Linux, which is connected to the main computer via an on board Ethernet network. Two CCD cameras provide our system's visual input, with each camera mounted on a Directed Perceptions pan-tilt head.

1.2 Tasks addressed by this work

The assistive tasks addressed by the Intelligent Wheelchair Project span a range of spatial scales and degrees of autonomy:

- **Hazard avoidance.** Obstacles, fast-moving people, and other hazards can cause problems for users who cannot react quickly to avoid them. An important task of the intelligent wheelchair is to detect and avoid such hazards using sonar, vision, and other available sensors.
- **Semi-autonomous routines to assist in difficult maneuvers.** Many doorways, elevators, and wheelchair ramps have narrow entrances and tight quarters, requiring precise control of the wheelchair. A user who is spastic or has no control of his or her hands may find such precision navigation frustrating or impossible. For common situations such as the ones listed above, it may be appropriate to have *semi-autonomous routines*, triggered by

[3] `http://www.kipr.org`

the wheelchair driver. A semi-autonomous routine identifies correspondences between the local environment and a plan for handling the situation, and then follows the plan under executive supervision of the driver. Such routines depend on the wheelchair's driver to identify an appropriate context in which to execute them. A typical semi-autonomous routine might turn around a wheelchair and back it into an elevator under real-time visual control.

– **Providing "hands-free" control over medium distance travel.** Continuous manual control of a powered wheelchair can be taxing for people with limited motor control. Where a control strategy can be well-defined, such as along a sidewalk, down a hallway, or following a person, the driver could simply select from a set of available control routines. The wheelchair can use vision and other sensors to serve this type of simple autonomous control. The success of CMU's Navlab [13] and other automatic driving systems show that this is a feasible approach for environments with regular structure, especially when a human user is available to handle transitions and indicate when the environmental context is correct.
– **Navigating over large-scale distances using a spatial knowledge base.** Campus-scale environments can provide significant barriers to travel in a wheelchair. Automatic assistance in route-finding, combined with control strategies for medium-scale (following streets or sidewalks) and small-scale (traversing doors and wheelchair ramps) motion, can make it possible to semi-autonomously navigate a path from one point on campus to another.

Each of these problem domains is difficult in itself, and solving them all is a daunting challenge. However, assistive navigation has one important difference from the similar problem of autonomous mobile robot navigation which makes assistive navigation simpler in some respects and more difficult in others. That difference, of course, is that the wheelchair has a driver who is, at all times, in control (direct or executive) of the wheelchair's behavior.

1.3 The wheelchair as assistive agent

We can conceive of the intelligent wheelchair as an *assistive agent* whose purpose is to serve tasks given to it by the driver. Having a human controller is a boon to the agent when it is confronted with environments or situations that are too complex for it to understand. Rather than fail gracelessly (as autonomous mobile robots do all too often) the agent can depend on manual control by the driver (if the driver has sufficient motor abilities) or engage in an interactive dialog to clarify the situation.

The driver also places requirements on the agent's behavior that makes some traditional mobile robotics techniques inappropriate. Many spatial-knowledge representation strategies rely on autonomous exploration of the environment to construct a map or model that is grounded in the agent's sensorimotor experience. For the most part, autonomous exploration is simply out of the question for an assistive agent. It must build its knowledge base through passive observation, or active strategies that do not interfere with the user's goals. Certain types of control strategies may also be inappropriate, particularly those that cause the wheelchair to make sudden changes in direction or rely on repetitive motion to build estimates of the location of particular features. Smooth motion and unobtrusive data collection are important for a user community whose differences already bring too much unwanted attention.

A prototypical assistive task of moderate complexity is the problem of helping to drive the wheelchair from one's office to fetch a cup of coffee on another floor. To assist a user having a near-complete lack of mobility, the agent must have enough knowledge to perform the task almost completely autonomously. In particular, it must be able to do the following things:

- **Represent knowledge about the world.** In order to plan effectively, the agent must have knowledge about the structure of large-scale space, such as the layout of buildings, campuses, and so on. Spatial knowledge can be *metrical*, *qualitative*, or some combination of the two. Other important knowledge is non-spatial, for instance, knowledge about the agent's state of being or the progress of its current activities, and must also be represented.
- **Receive and interpret user commands.** Persons with severe mobility impairments may not have enough motor control to reliably or completely specify their desires to the assistive agent. Interpreting noisy, imprecise command inputs requires knowledge of the agent's current situation and status, the user's likely action choices, and so on.
- **Build a plan from a library of primitive actions.** The user's command indicates a goal or desired state. The agent must use its knowledge of the world's state and its own abilities to construct a plan for achieving the specified goal. This plan will contain references to physical objects, landmarks, and other perceivable features of the environment embedded as direct and indirect objects of action. A significant part of the planning task is *attentional selection* of these objects (plan nouns).
- **Establish grounding connections between plan nouns and sensory input.** For each primitive action to be taken, the agent must find parts of the physical world which match the nouns referred to by that action. For example, a command to go through a door requires that a suitable door be located in the local area. All such relevant objects must be collected into representations of the local *visual space* and monitored with sensors to keep representations up-to-date.

These skills allow the intelligent wheelchair to accept a goal indicated by the user, construct a plan to achieve that goal, and connect the free nouns in the plan to physical objects in the world in order to perform the locomotor control needed to achieve the goal. In the remainder of this paper, we describe our approach to each subproblem.

2 Representing knowledge about large-scale space

Our approach for representing large-scale space is the Spatial Semantic Hierarchy (SSH) [8, 9]. The SSH abstracts the structure of an agent's spatial knowledge in a way that is relatively independent of its sensorimotor apparatus and the environment within which it moves. In addition, the SSH integrates reactive behaviors, different spatial scales, and lends itself to a simple design for a human-wheelchair interface (see Section 3).

The Spatial Semantic Hierarchy (SSH) is an *ontological hierarchy* of representations for knowledge of large-scale space.[4] An ontological hierarchy allows multiple representations for the same kind of knowledge to coexist. Each level of the hierarchy has its own *ontology* (the set of objects and relations it uses for describing the world) and its own set of inference and problem-solving methods. The objects, relations, and assumptions required by each level are provided by those below it. The SSH is composed of 5 levels:

- **Sensorimotor.** The *sensorimotor level* of the agent provides continuous sensors and effectors, but not direct access to the global structure of the environment, or the wheelchair's position or orientation within it.
- **Control.** At the *control level* of the hierarchy, the ontology is an egocentric sensorimotor one, without knowledge of geographically fixed places in an external environment. A *distinctive state* is defined as the local maximum found by a hill-climbing control strategy, climbing the gradient of a selected feature, or *distinctiveness measure.* Trajectory-following control laws take the wheelchair from one distinctive state to the neighborhood of the next, where hill-climbing can find a local maximum, reducing position error and preventing its accumulation.
- **Causal.** The ontology at the SSH *causal level* consists of views, distinctive states, actions and schemas. A *view* is a description of the sensory input obtained at a locally distinctive state. An *action* denotes a sequence of one or more control laws which can be initiated at a locally distinctive state, and terminates after a hill climbing control law with the wheelchair at another distinctive state. A *schema* is a tuple $((V, dp), A, (V', dq))$ representing the (temporally extended) event in which the wheelchair takes a particular action A, starting with view V at the distinctive state dp, and terminating with view V' at distinctive state dq.[5]
- **Topological.** At the *topological level* of the hierarchy, the ontology consists of *places, paths* and *regions*, with connectivity and containment relations. Relations among the distinctive states and trajectories defined by the control level, and among their summaries as schemas at the causal level, are effectively described by the topological network. This network can be used to guide exploration of new environments and to solve new route-finding problems. Using the network representation, navigation among places is not dependent on the accuracy, or even the existence, of metrical knowledge of the environment.

[4] In large-scale space the structure of the environment is revealed by integrating local observations over time, rather than being perceived from a single vantage point.

[5] Given a schema $S = ((V, dp), A, (V', dq))$, the *context*, *action* and *result* of S are V, A, and V', respectively. We intuitively read the schema S as, in context V, after executing A, expect V'.

– **Metrical.** At the *metrical level* of the hierarchy, the ontology for places, paths, and sensory features is extended to include metrical properties such as distance, direction, shape, etc. Geometrical features are extracted from sensory input, and represented as annotations on the places and paths of the topological network.

As an example, consider how the different SSH levels allow the wheelchair to represent spatial information about a university campus. The topological map associated with the campus is a graph where nodes (places) represent street intersections and edges represent the streets. This graph is useful for establishing a high level route from one place to another as well as to interact with the wheelchair's driver. Information from the control and causal levels is used to associate with each edge of the map the semi-autonomous routine needed for traveling a particular street. The local geometry of places in the map (i.e. street intersections) is represented by a detailed local metrical map (for example, an occupancy grid) which allows the wheelchair to find a plan for actually crossing a street.

3 Human Interface

The driver-wheelchair interface must allow the driver to communicate commands in a relatively natural language of spatial structure and action.[6] The wheelchair's spatial representation must facilitate the understanding of such commands and must present a sensible set of command options to the driver when required. For our purposes, the relevant properties of an interface device are the number of alternative commands the human driver may select from at any given time and the frequency with which selections must be made to guide the wheelchair. We abstract away the medium used to present the choices (for example, computer display, scanning menu, generated speech) and the device used to capture the choice (for example, joystick, pushbutton, "sip and puff" tube).

The different levels of the SSH lend themselves to a simple design for a useful interface:

– **Topological.** At the SSH topological level, the environment is represented as a graph (the topological map) of places and paths. Assuming the wheelchair has a sufficiently complete map, the driver can instruct the wheelchair where to go by inspecting the map, selecting a place, and saying, "Go there!". To display a topological map clearly, it is helpful to have a single frame of reference with a familiar orientation and a relative location for each place, which can be used to map the graph onto the coordinates of the display medium.[7] Names alone could be used to specify destinations, but not all important destinations have concise names.

[6] These ideas developed through discussion with Sengul Vurgun and the other members of the Intelligent Robotics seminar, Fall 1997.

[7] A qualitative description of the SSH metrical information is used to create a visualization of the topological map.

- **Causal.** At the SSH causal level, the environment is represented as a set of routes, each consisting of causal relations among context, action, and resulting situation. During the time while one action is executing, the wheelchair can display the possible actions at the next decision point (very likely a small set such as Turn Right, Turn Left, Go Straight, and Stop). Any time until the decision point is reached, the user can select one of these actions. Where a manual control law would require a continuous stream of (for example) joystick commands to adjust the wheelchair's course, the causal interface requires approximately two bits of information at each decision point.
- **Control.** At the SSH control level, the interface allows the driver to select the control law to follow from the current state. For example, the driver might wish to rotate clockwise to the next distinctive orientation or move along a corridor. The environment itself will provide the constraints that determine whether, for example, following a corridor requires a midline-following or wall-following control strategy. The SSH control level also includes a second interface: a simple interrupt signal that can be interpreted as "Stop!" or "Watch out!" or "Here it is!", depending on context. This signal is issued by the driver who may notice that a destination point has been reached, or a hazard is being approached, while the wheelchair may not notice it. Within the SSH framework, this signal can be thought of as signaling the end of a qualitatively uniform segment of the environment.

In this analysis, we have related the information-acquisition needs of the wheelchair user interface to the different representations for spatial knowledge in the SSH. In most cases, control of the wheelchair requires the selection of actions (control laws) from among small finite sets, without substantial time pressure. This reduces the communication bandwidth required between driver and wheelchair, when compared with direct control devices such as joysticks.

4 Constructing a plan and directing attention

The intelligent wheelchair has limited processing power. High-bandwidth sensors, such as color cameras, provide enough information to make brute-force computing approaches impractical. Fortunately, human perception demonstrates that it is often unnecessary to process all of the data incident on sensors. Humans use *perceptual attention* to focus on a subset of the relevant stimuli, greatly reducing the amount of required sensory processing [14]. In order to similarly reduce the processing needs of the wheelchair, we use a perceptual attention strategy modeled on that of humans.

Attention is directed to (i.e. processing power is allocated to) various portions of the perceptual input stream based on the current set of goals. For example, if the wheelchair is in the process of locating the elevator buttons, visual input will dominate the perceptual processing. Specific features (such as color, shape and orientation) and recognizable objects (such as hallways, doors, people and signs) receive preferential processing. Using these preferences, percepts can be filtered and prioritized before arriving at the cognitive level. This reduces the cognitive load by pre-identifying percepts that are the most interesting and relevant to the current goals of the system.

4.1 Plan generation

Plans for movement in large-scale space can be constructed from the topological, causal, and control knowledge embodied in the SSH representation of space. Steps in the plan are the *actions* from each relevant schema at the SSH causal level, such as: *Move through the door, Turn left, Move down the hallway, Stop in front of the elevator*. Nouns in each step (*door, hallway, elevator*) indicate objects that are important to the task. The attention of the perceptual system can be focused mainly on these objects, allowing the plan to work as a top-down attentional selection mechanism. The system may use one or more of the available sensors to locate the important objects.

4.2 Coordinating the perceptual subsystems

We are designing a *sensor coordinator* to handle the problem of assigning perceptual tasks to multiple sensors. The process of allocating *sensor tasks* to sensors starts by collecting the objects mentioned in the current goals into the SACK (Set of Activated Concept Knowledge). Using a form of spreading activation [1], objects in the knowledge base that are related to the goal objects are added to the SACK. The level of activation needed to enter the SACK can be adjusted based on the current cognitive load of the system and the importance of the current goals. The contents of the SACK constitute the set of objects that are perceptually interesting to the system at the current time. The SACK is updated when the system's goals change.

Once the SACK has been updated, the sensor coordinator retrieves the *perceptual descriptions* (PD) of the objects in the SACK. Each PD is matched to sensor descriptions to find the appropriate set of sensors for perceiving that object. Sensor tasks are then allocated to each relevant sensor. Each task contains a sensor-specific description of the object.

To illustrate the interplay between cognition and perception, consider an intelligent wheelchair whose current goal is to enter an elevator. The `elevator` concept and its associated concepts `elevator-door`, `elevator-controls` and `elevator-display` will be members of the SACK. An `elevator-control` corresponds to an up or down button, whose perceptual description might be a "high-contrast circular region". The description is used by the vision system to locate the elevator button in the visual scene. At the same time, the description "obstacle within one foot" might be used by the sonar system to detect the wall near the elevator.

Current research problems in sensor coordination. The sensor coordinator described above is currently under development. Some of the questions we are addressing during this phase of the research include:

– **What is the perceptual description language?** The PD language is crucial to correct sensor allocation. The Sensor Coordinator uses it to match sensors and construct sensor tasks. It must be general enough to encompass all sensors, yet precise enough to encompass all objects.
– **Does the sensor coordinator interfere with the need for fine control of perceptual sensors?** Control laws for robot movement often require fine control of effectors and sensors, each of which can influence sensor data. How do we allow this while using the Sensor Coordinator, which adds a layer of abstraction to the perception process?
– **How do we prioritize sensor tasks?** There may be many sensor tasks allocated to each sensor. How do we prioritize them and how is the priority used?

5 Grounding plan nouns with real-time vision

The SSH approach to control is built upon *trajectory following* control laws which cause the agent to follow a particular path defined by a feature or a set of features in the environment. Typical trajectory-following control laws might be "maintain a constant distance from the wall on your right," or "pass through the center of the door." A plan for moving through the environment consists of a sequence of such trajectory-following laws with *transition conditions* specifying when to switch from one to another. Once such a plan has been constructed, the first step in its execution is to connect the *plan nouns* needed by control laws (people, objects, features of the environment such as walls or doorways, etc.) to perceptions, in some sensor data stream, of corresponding objects in the physical world.

At the highest level, these nouns are placed into the SACK (see Section 4) and control and spatial-reasoning routines begin to reason about the position and properties of the corresponding physical objects. It is the responsibility of sensor subsystems to ensure that the correspondence between an activated concept and a physical object is made and maintained. The most demanding (and most powerful) sensor subsystem available to the wheelchair is the real-time vision system.

Our approach to vision has its lower levels based in the *active vision* paradigm [2, 7] but significantly extends this model to handle a visual world that extends past a single camera's field of view. In this section, we discuss the ARGUS system [3, 4] which provides the low-level foundations of our vision system, the *unified dynamic object memory* which addresses the *field of view problem*, and our ideas about integrating visual observations into coherent structural representations of *visual space* over time.

5.1 Focusing visual attention

A real-time, stereo, color image feed is more of a firehose of data than a manageable stream. Short of massive brute-force computation, processing all of the data incident on the sensors is impossible. Fortunately, the goals of a real-time vision system can be adequately served without processing all of the incoming data, or indeed processing much of it at all. By focusing processing on regions of the image likely to contain important objects, we can maintain grounding connections for objects in the SACK without being drowned by the firehose of visual data.

We identify two modes of attentive processing for the vision system: *top-down* or *task-directed attention*, which converts sensing goals into specific processing directives (for instance, "Find a fire extinguisher in the current field of view") and *bottom-up* or *pop-out attention*, which identifies and classifies unexpected, suddenly-appearing, or hazardous objects in the visual stream.

Our work to date has focused principally on *top-down* direction of attention, since the principal purpose of vision in our work is to ground symbols from the SACK which represent physical objects.[8] Restricting ourselves to tracking physical objects opens up a variety of problem constraints and optimizations.

Tracking primitive image features. The images of particular physical objects occupy restricted areas of the agent's field of view, meaning that they can be identified in a small subwindow of the whole frame. They move, scale, and deform continuously over time, meaning that they can be tracked by image-processing routines of moderate complexity (applied only to a small portion of the whole frame) coupled with well-understood tracking algorithms. Objects with resolution-invariant properties, such as color or straight edges, can be viewed and tracked through low-resolution subsampled images (which we call *attention buffers*) to reduce computation. When there are multiple objects of interest in the visual scene, they can be tracked independently through separate subwindows. Rather than applying image analysis routines to an entire video frame, a subwindow is extracted and analysis is performed in its restricted frame of reference. The inspiration for our development of window-based processing came from Pierce's "roving eye" experiments [12] but the same techniques have been successfully pursued elsewhere and are generally credited to Hager[6].

Tracking objects. Multiple *attention buffers* with different image processing algorithms can be used together to track an object with multiple features. We have developed a simple language to describe the features making up an object and their geometric relationships. We use a probabilistic technique to detect and correct tracking errors in individual features.

Stereoscopic tracking. Tracking features in a pair of cameras provides additional constraints which actually makes tracking easier. Constraints from epipolar geometry and from knowledge of the physical size of objects of interest can minimize correspondence problems and can help in detecting tracking errors (once an estimate of the 3-d size of an object has been made, that estimate can be

[8] Hazard avoidance and other tasks which detect pop-out phenomena can be recast as top-down tasks ("Find any object which is moving quickly") when necessary.

used to make a sanity check on the results of image-space tracking). Since we are concerned with the qualitative spatial relations between objects rather than with precise reconstruction of the 3-d scene, camera calibration is not critical and dense depth-map generation is unnecessary. We use a simple pinhole projection model, a mechanically imprecise mounting rig, and point-feature triangulation to compute 3-d position estimates which are fine for our purposes.

5.2 The field of view problem

The primary goal of the wheelchair's vision system is to ground symbols representing relevant objects from the SACK. These objects may be in front of the wheelchair, behind it, to the side, or even above the wheelchair, and with several objects in the current SACK, perhaps several or all of these simultaneously. However, most sensors (and, in particular, vision sensors) can view only a limited portion of the environment at any instant.

Standard cameras can only see the full 360 degree spatial surround by physically shifting their field of view.[9] Typical lenses for machine vision applications have fields of view which range from 10 or less degrees horizontally to 120 degrees or more for "fisheye" lenses. Lenses with greater than approximately 65 degrees horizontal field of view suffer both from low spatial resolution (normal video capture hardware generally has on the order of 256K pixels per captured frame) and from image distortion that can be troublesome to remove in a real-time system. A camera with a narrow enough field of view to have low distortion and high resolution must direct its field of view intelligently if it is to have a chance of seeing all the relevant parts of the environment.

We want to use *closed-loop* control of the wheelchair to avoid the problems associated with out-of-date world models and unexpected changes in the local situation. Our experiments with *visual servoing* control of a hand-eye robot [5] and real-time vision have demonstrated that this is tractable on the available hardware [3, 4]. However, with finite sensing hardware, we cannot guarantee that all relevant objects will be in view at any given time.

5.3 Unified dynamic object memory

To circumvent the field of view problem for closed loop control, we use a *unified dynamic object memory system* to mediate between sensor inputs and control-signal calculations. The unified dynamic object memory integrates real-time visual perception of visible objects with short-term active memory traces of objects that are not currently in the field of view. When information about an object's position or motion is needed to compute a control signal, the unified object memory provides that information. The unified object memory connects directly to visual perceptions when objects are visible and to estimates otherwise. This is

[9] Some researchers have used special lenses or panoramic imaging techniques with mirrors, but the problem of focusing perceptual attention remains.

a direct generalization of the *visual servoing* approach, which uses direct observation of the image stream to control motion but cannot handle objects outside the field of view.

We use a Kalman filter (KF) estimate of object position and motion. The KF provides both a dynamic estimate of the object's state, which evolves even when no direct observations are being made, and a measure of the uncertainty in the state estimate, which grows over time if no direct observations are made. The agent's field of view is controlled by an *investigatory action scheduler* which examines position uncertainty estimates in the unified object memory and selects an object for direct observation. The scheduler tries to maximize a metric which takes into account both the reduction of uncertainty that would result from observing an object and the importance of that object to the current task. Selection of an appropriate attention policy for each significant object is a function of the current context, the nature of the task, and the recent behavior of the object in question.

Unified object memory and an independent investigatory action scheduler allow plans for locomotion to be carried out without regard to the limitations of the agent's field of view. Any objects or features of the environment that must be perceived to control the wheelchair will be represented in the persistent dynamic object memory. Control laws for the wheelchair can ignore the actions needed to locate or track objects as the chair moves, or to switch cameras between one object and the next. These actions are generic to any task needing to know the position or properties of an object, and can be handled by a single investigatory action scheduler for all tasks.

5.4 Representing visual landmarks over larger spatial scales

The unified object memory determines and maintains object position estimates over short time scales and within local spaces. It does not address the issue of maintaining representations of these perceptions over longer time scales and greater distances. Sets of observations of the same regions from different vantages can be integrated into larger scale representations of an area, and these larger scale, longer term representations of the spatial layout of observed visual landmarks can support more sophisticated reasoning, including the planning of novel routes through untraveled regions, the determination of appropriate landmarks for piloting through these regions [10], the improved estimation self-location, and, as the representations become more descriptive with the assimilation of more information, the support of more critical evaluation of future input.

Building broader representations of visual landmarks. The positions of objects as measured by a sensor system are naturally encoded in coordinates relative to the observer (an *egocentric* frame of reference). While egocentric, metrical representations of landmark positions are useful, world-centered (*allocentric*) frames of reference are more appropriate for many applications, and *qualitative* information about position (such as left-right and front-back orderings) is often more robust than metrical information, and can be more reliably extracted from sensor data.

Allocentric representations can be constructed from collections of egocentric representations when those collections share enough landmarks to align their coordinate frames. An allocentric representation describes the positions of perceived objects within a local coordinate system from a "third person" perspective. Allocentric representations become useful when large spatial scales and observer motion make reasoning about accumulated egocentric data impractical. Allocentric maps can be related to one another in a loose hierarchy when inter-map relations are known; one related approach to this kind of arrangement can be found in the work of McDermott and Davis [11].

Planning with landmark information. Information about the position of a series of landmarks allows the agent to devise completely novel travel plans. If some chain of known visual landmarks covers regions between the current location and a goal location, the agent may be able to generate a successful plan using these landmarks as piloting beacons. This topic is discussed extensively in the context of outdoor navigation in [10].

Knowledge of the arrangement of local landmarks can also improve efficiency when planning on much smaller scales. For example, when approaching a known area, even if from a novel direction, recalled maps of a given environment can allow the agent to determine where it should expect to find any landmarks relevant to its next intended actions. This knowledge can substantially speed up visual search operations.

Determining the agent's location. By measuring its position with respect to some number of known visual landmarks, the agent can evaluate its current location in a global frame of reference. Depending on the nature of the cognitive representations of the area, this may result in an accurate quantitative estimate or a more general qualitative one. Even when the agent is hopelessly lost it can use its landmark knowledge combined with local sightings to generate a list of its most likely current locations. From this list, using knowledge of the associated areas, it can generate a plan of travels and observations which will identify the correct one.

Evaluating new input in context. When a new observation must be incorporated into existing models, determining the appropriate meaning of the observation is key. Consider the observation of a trash can in a substantially different location from that of another trash can recorded in one of the agent's cognitive maps of the same general area. What does this mean: "Did the trash can move?", "Is there another trash can?", or "Is the map in error?" If the latter, then "In what way?"

Coming up with a correct answer to these questions is a difficult task, but there are a number of ways to address the issue. To determine if the trash can moved, a visual query can be made asking if a trash can can be spotted in the old, predicted location. The accuracy of this sighting can even be double checked by issuing queries for other objects which the map indicates should be nearby. If another trash can is spotted, then it is quite likely that there are in fact two trash cans. If not, then the possibility that either the trash can moved, or that the map is in error can be evaluated by a sequence of visual queries, expanding outward from the expected observation angle of the trash can in order to determine the accuracy of the rest of the map. Furthermore, the possibility that the object in question just moved can be influenced by annotations in the agent's knowledge base indicating the relative expected permanence of various objects. For example, it might be recorded that the probability that a trash can has moved is much higher than that of a building, or an elevator. Furthermore, nodes in the egocentric and allocentric maps must record "support" information so that newer, more "vivid" observations (or lack thereof) that contradict the existing regional representations can override stale information.

6 Summary

For an agent to be a useful assistant in navigation, route finding, or other mobility-related tasks, it needs several things: an understanding of space, tools of representation and inference for reasoning about space and action, sensors to perceive objects and spatial relations, and effectors to move through the world. The Intelligent Wheelchair Project is working to bring together the relevant theory and practice from the various related fields of AI, vision, and robotics to build such an agent. In this paper, we have outlined our approach, which uses focused perceptual attention to allocate available sensors most effectively for vision-guided navigation.

As we complete assembly of the software and hardware portions of our experimental platform, we will incrementally bring our agent's capabilities on-line. We have demonstrated the basic technologies of real-time active visual control, path-following control of a mobile robot, and construction of a spatial semantic map from controlled exploration. Integration of these modules will provide significant challenges in the months ahead.

References

1. J. R. Anderson. Spreading activation. In J. R Anderson and S. M. Kosslyn, editors, *Tutorials in Learning and memory: Essays in honor of Gordon Bower*. W. H. Freeman, 1984.
2. Dana H. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.
3. William S. Gribble. Slow visual search in a fast-changing world. In *Proceedings of the 1995 IEEE Intl. Symposium on Computer Vision (ISCV-95)*, 1995.
4. William S. Gribble. `ARGUS`: a distributed environment for real-time vision. Master's thesis, University of Texas at Austin AI Lab, Austin, Texas, 1995.
5. William S. Gribble, Robert L. Browning, and Benjamin J. Kuipers. Dynamic binding of visual percepts for robot control. Unpublished manuscript, 1997.
6. Gregory D. Hager. *Task-directed sensor fusion and planning: a computational approach.* Kluwer Academic Publishers, 1990.

7. A. Bandopadhay J. Aloimonos and I. Weiss. Active vision. *Int. J. Computer Vision*, 1:333–356, 1988.
8. B. Kuipers. A hierarchy of qualitative representations for space. In *Working papers of the Tenth International Workshop on Qualitative Reasoning about Physical Systems (QR-96)*. AAAI Press, 1996.
9. B. Kuipers and Y. T. Byun. A robot exploration and mapping strategy based on semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 8:47–63, 1991.
10. B. J. Kuipers and Tod Levitt. Navigation and mapping in large scale space. *AI Magazine*, 9(2):25–43, 1988. Reprinted in Advances in Spatial Reasoning, Volume 2, Su-shing Chen (Ed.), Norwood NJ: Ablex Publishing, 1990.
11. Drew McDermott and Ernest Davis. Planning routes through uncertain territory. *Artificial Intelligence*, 22:107–156, 1984.
12. David M. Pierce and Benjamin Kuipers. Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92:169–227, 1997.
13. Charles E. Thorpe, editor. *Vision and Navigation: the Carnegie-Mellon Navlab*. Kluwer Academic Publishers, 1996.
14. Anne Treisman. Features and objects. In A. Baddeley and L. Weiskrantz, editors, *Attention: Selection, Awareness and Control*. Clarendon Press, 1993.

For further information, see `http://www.cs.utexas.edu/users/robot`.
The authors can be contacted at {grib,rlb,hewett,eremolin,kuipers}@cs.utexas.edu.