MERTZ: A Quest for a Robust and Scalable Active Vision Humanoid Head Robot

Lijin Aryananda and Jeff Weber
MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, Massachusetts, 02139, USA
{*lijin, jaweber*}*@csail.mit.edu*

We present the design and construction of MERTZ, an active-vision humanoid head robot, with the immediate goal of having the robot runs continuously for days without supervision at various locations. We address how the lack of robustness and reliability lead to limitations and scalability issues in research robotic platforms. We propose to attend to these issues in parallel with the course of robot development. Drawing from lessons learned from our previous robots, we incorporated various fault prevention strategies into the electromechanical design. We have implemented a preliminary system, integrating sensorimotor and vision in order to test the full range of all degrees of freedom. We conducted experiments in order to test the robustness of current design and identify failure modes. We collected visual data in order to observe variations across locations and time of day. Lastly, We present the long term research direction for the robot.

Keywords: humanoid robot, social interaction, reliability

1. Introduction

Despite tremendous research progress in humanoid robotics over the past decade, it still remains challenging to develop a robot capable of performing in a robust and reliable manner. Various humanoid robots have been built in research laboratories to explore different capabilities, ranging from navigation to interacting with people. However, given the substantial challenges involved, it is common to employ shortcuts so as to allow for initial progress. For example, we have in the past covered a wooden door with paper because it is frequently detected by the skin-color filter as a person. Such simplifications, while far from reducing the environment to a *blocks world*, naturally raise some concerns. For instance, will the results from a given setting scale to a more complex environment? Could these shortcuts be hampering potential progress?

As accurately described by Bischoff and Graefe [1], robot reliability has not received adequate attention in most research projects on humanoid robotics. One possible cause could be a false belief that when the time comes for robots to be expedited, someone else will eventually address the issue of reliability. The average running periods and total lifetime of research humanoid robots are generally limited. It is not uncommon for robots to require a specific set-up in order to work properly, such as a particular location, background color, or lighting condition. Many project demonstrations, though impressive, provide only a relatively narrow perspective on the robot's capabilities and limitations.

In this paper, we propose to address the issue of robustness and reliability in parallel with the course of robot development. On one hand, it is clear that reliability must be addressed at some point, since any humanoid robot must be robust when ultimately dispatched to the real world. More strongly, we feel that there are important advantages to addressing reliability at an earlier rather than later stage. First, having a more reliable robotic platform suggests important research directions, and also provides a technical foundation on which to study them. As a concrete illustration, a robot that can run for days at a time under various conditions would be able to accommodate long term online learning processes and experiments. Moreover, a robot capable of continuous operation suggests a variety of interesting research directions in social interaction, especially vis a vis longer-term dynamics. Supposing that our pet dogs were awake for only several hours per week, imagine how impoverished our social interactions would be. Second, addressing the issue of reliability is likely to motivate further work on scalable solutions. Consider, for example, the large amount of streaming sensor data that would be collected over several days of experiments; such large volumes of data would require novel software solutions and learning algorithms. Moreover, algorithms are often tuned to the particular setting of the robot laboratory. Obliging the robot to perform at different locations would encourage innovative approaches that are scalable to other settings and environments.

The long-term research direction of this project is to explore scalable robot learning in a social context, inspired by how human infants learn by observing and imitating other people. The idea is to have our robot, MERTZ continuously operate and interact with various people and objects while incrementally developing new skills and learning new knowledge from perceived events online. In this paper, we present the design and development of MERTZ, an active-vision head robot platform, with the immediate goal of having the robot runs continuously for days without supervision. Drawing from lessons learned from our previous robots, we incorporated various fault prevention strategies into the electromechanical design. We have implemented a preliminary system, integrating sensorimotor and vision to produce simple behaviors, i.e. orienting and attending to objects. We conducted a series of experiments where the robot ran for 47 hours within 4 days at four locations, in order to test the robustness of current design and identify failure modes. We also collected visual data in order to observe variations across different times and locations. More testing is under way and thus cannot be included in this report due to time constraint.

In the following section, we describe the robotic platform. Section 3 illustrates our research approach and the scope of the project. Section 4 discusses the mechanical design of the robot, followed by details of the computational hardware and software for the preliminary integrated system in section 5. We report experimen-

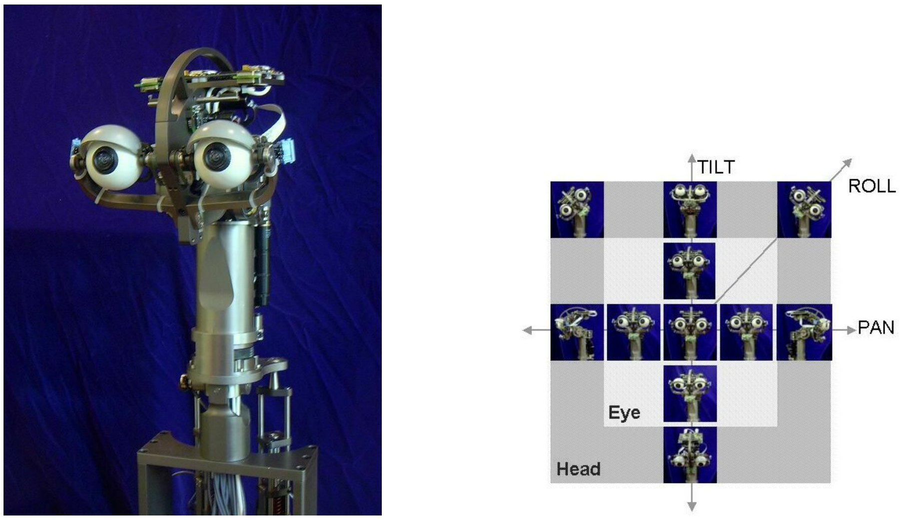

Fig. 1. MERTZ is an active-vision humanoid head robot with 15 degrees of freedom, nine of which have been implemented (left). On the right is the head and eye degrees of freedom.

tal results and observed failure modes in section 6. We sketch out the long term research direction of the project in section 7. Section 8 provides comparisons with related work, followed by conclusion in section 9.

2. Robotic Platform

MERTZ is an active-vision head robot with fifteen degrees of freedom (DOF). Nine DOFs using DC motors have been implemented, as shown in Figure 1. Six additional DOFs for facial features are under way. The head pans and uses a cable-drive differential to tilt and roll. The eyes pan individually, but tilt together. The neck also tilts and rolls using two Series Elastic Actuators [2] configured to form a differential mechanism. The eyelids are coupled as one axis. The lips, eyebrows, and face shell will be added very soon. Figure 2 illustrates the expressive elements of MERTZ's design. As is, MERTZ is limited to using gaze direction to actuate in the world and demonstrate its competence. In the near future, an arm-like component may be necessary in order to expand the repertoire of possible learning tasks, such as poking or pointing to objects.

The robot receives visual input using a Point Grey OEM Dragonfly digital camera per eye. We are also playing with using sonar sensors to assist the vision system in estimating distance to objects in the field of view. The robot receives proprioceptive feedback from both potentiometer and encoder mounted on each axis. In the future, the robot will be additionally equipped to accept audio input and synthesize speech.

A number of primary DOFs are dedicated to emulate each category of human

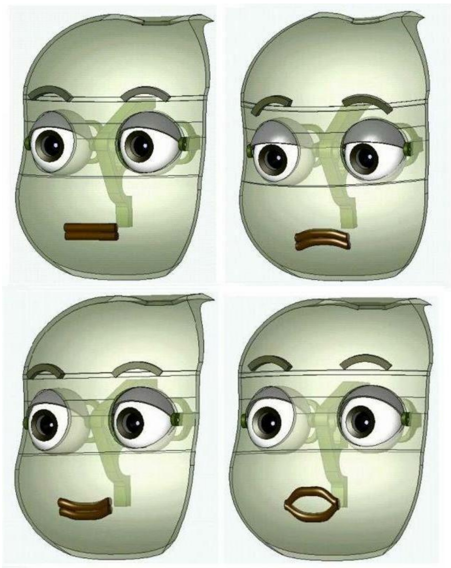

Fig. 2. MERTZ's expressive face shell and features design, showing different configuration to portray emotion expression. The face shell will be made a translucent material.

eye movements, i.e. saccades, smooth pursuit, vergence, vesticular-ocular reflex, and opto-kinetic response [3]. The expressive element of MERTZ's design is essential for the long term project goal, which involves having the robot learn from people through social interaction interface. As a tradeoff between complexity and robustness, we attempted to minimize the total number of DOFs while maintaining sufficient expressivity. Once the robot is fully built, the plan is to mount the entire system on a portable wheeled platform that is easily moved around and can be turned on anywhere by simply plugging into a power outlet.

3. Research Approach and Scope

As human is an incredibly complicated creature, building humanoid robots is a struggle of dealing with a high level of complexity with limited resources and a large set of constraints. Given our concern for robustness, we opted for a bottom-up approach and initially starting with an active-vision head platform with a manageable number of DOFs.

It is important to note that the main focus of this project is not to investigate the best engineering solution for robot reliability. Instead, our focus is to develop the robot with the awareness for the importance of robustness and reliability. We hope that various robustness criteria will encourage us to discover novel approaches that would allow us to move away from the widespread practice of showing project results

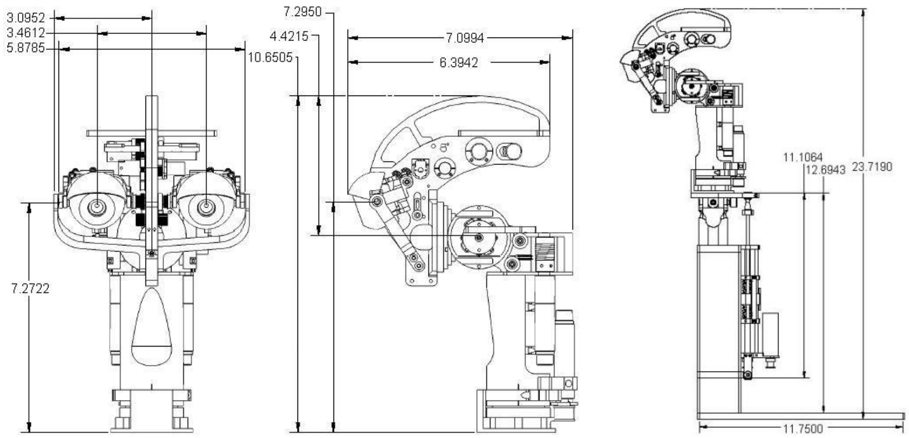

Fig. 3. The overall head dimension is 10.65 x 6.2 x 7.1 inches and weighs 4.25 lbs. The Series Elastic Actuators extend 11.1 inches below the neck and each one weighs 1 lb.

through video clips toward live demonstrations of the robot. We have invested efforts in dealing with robustness issues during the design and construction of the robot. We plan to continue identifying failure modes throughout the rest of the robot development process in order to implement the appropriate fault detection and increase fault tolerance.

4. Mechanical Design

Mertz was mechanically designed with the goal of having the robot be able to run for days at a time without supervision. Drawing from lessons from previous robots, we incorporated various failure prevention and maintenance strategies, as described below.

4.1. *Compact design to minimize total size, weight, and power*

A high-priority constraint was placed during the early phase of the design process to minimize the robot's size and weight. A smaller and lighter robot requires less torque from the motors to reach the same velocity. Also, the robot is less prone to overloading causing overheating and premature wear of the motors. The overall head size is 10.65 x 6.2 x 7.1 inches (see Figure 4) and weighs 4.25 lbs. which is supported by a box frame. The Series Elastic Actuators, which bear the weight of the head at the lower neck universal joint, extend below it 11.1 inches. Mertz's compact design is kept light by incorporating nominal light alloy parts, which retaining stiffness and durability for their small size. Titanium, as an alternative to aluminum, was also used for some parts in order to minimize weight without sacrificing strength.

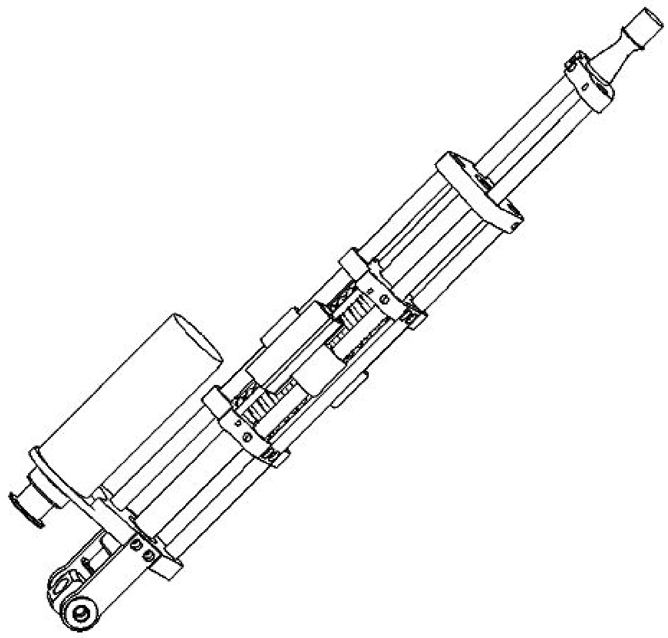

Fig. 4. The Series Elastic Actuator is equipped with a linear spring that is placed in series between the motor and load.

A cable-drive differential was implemented to provide the head roll and tilt in a small and compact design, as one axis provides two degrees of freedom. In contrast to a traditional gear train differential, a cable differential has increased efficiency, has significantly less mass, and eliminates backlash. Additionally, this design shares the load between two motors which reduces the amount of strain on each motor and either doubles the power of the joint as both motors work in unison, or allows for smaller motors to be used.

4.2. *Force sensing for compliancy*

Two linear Series Elastic Actuators (SEA) [2] are used for the differential neck joint, a crucial axis responsible for supporting the entire weight of the head. As shown in Figure **??**, each SEA is equipped with a linear spring that is placed in series with the motor, which act like a low-pass filter reducing the effects of high impact shocks, while protecting the actuator and the robot. These springs also, in isolating the drive nut on the ball screw, provide a simple method for sensing force output, which is proportional to the deflection of the spring (Hooke's Law, F = kx where k is the spring constant). This deflection is measured by a linear potentiometer between the frame of the actuator and the nut on the ball screw. Consequently, force control can be implemented which allows the joint to be compliant and to safely interact with external forces in the environment.

The ball screw that is used in this linear actuator, which is 8mm in diameter with a pitch of 1mm, is equivalent to using a 79 to 1 planetary gear head. Although the linear actuator is longer, it provides necessary advantages in this case. Ball screws are over 90% efficient compared to roughly 80 % for a high quality planetary gear head. They are also back drivable without the risks associated with back driving a planetary or spur gear head, yet they require only a small amount of power to hold static postures. Additionally the SEA's will maintain position of the head when motor power is turned off. Collapsing joints upon power shutdown is a vulnerable point for robots, especially large and heavy ones.

Another concern for allowing the robot to run for long periods of time is heat generated by the motors. With running the robot for periods of 12 hours or more, only the linear actuator motors are generating a significant amount of heat. To deal with this we have been working on software issues to reduce noise in the system, which creates cogging in the motors generating heat, and designed heat sinks to place directly on the motors.

4.3. *Safeguarding position-controlled axes*

The rest of the DOFs rely on position feedback for motion control and thus are entirely dependent on accurate position sensors. Incorrect reading or faulty sensors could lead to a serious damage to the robot, so redundant relative encoder and potentiometer are utilized in each joint. The potentiometer provides absolute position measurement and eliminates the need for calibration routines during startup. Both sensors serve as a comparison point to detect failures in the other. Each joint is also designed to be back drivable and equipped with a physical stop in order to reduce failure impacts.

4.4. *Electrical cables and connectors*

Placement of electrical cables is frequently an afterthought in robot designs, as broken or loose cable is one of the most common failure sources. Routing over thirty cables inside the robot without straining each other or obstructing the joints is not an easy task. MERTZ's head design includes large cable passages through the center of head differential and the neck. This allows cable bundles to be neatly tucked inside the passages from the eyes all the way through to the base of the robot, thus minimizing cable displacement during joint movement. On the controller side, friction or locking connectors are used to ensure solid connections.

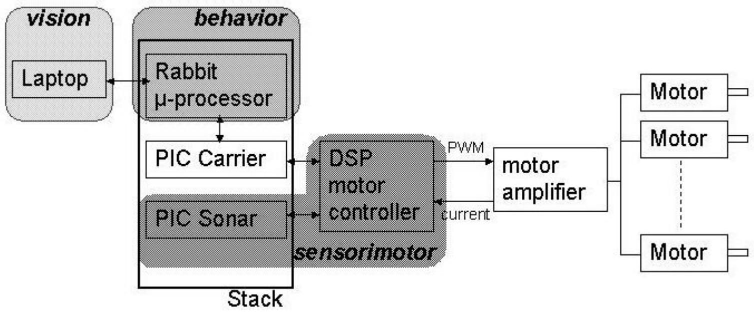

Fig. 5. The robot's preliminary integrated system architecture. Rectangular units represent hardware components. Grey patches with rounded corners represent software modules: vision, behavior, and sensorimotor.

4.5. *Modular design*

The mechanical components of MERTZ were designed in a modular manner, or in a way that allows easy access to all components. This allows for easy maintenance and repair procedures without disassembly of large portions of the robot. One issue in maintenance is the wearing of components such as belts or drive cables. All the belts and drive cables on Mertz have been designed to take at least a 50 % greater load than they are required to, to reduce the possibility of breakage. If this does happen all belts and cables are replaceable with minor disassembly.

5. Preliminary Integrated System Architecture

Figure 5 illustrates the interconnections among the robot's hardware and software modules. The rectangular units represent the hardware components and the grey patches with rounded corners represent the software systems: sensorimotor, vision, and high-level behavior. More details are described below.

5.1. *Hardware Systems*

Whereas our previous work has made use of off-the-shelf motion control products and PC nodes, in this project we implemented custom-made hardware for sensorimotor and behavior control. Off- the-shelf products, though powerful and convenient, are limited to a set of predetermined capabilities and can be unduly complex. Customizing our hardware to more precise specifications gives us greater control and flexibility. One caveat is that it may take some time to develop custom-made hardware to a reliable state.

5.1.1. *Motor Controller and Amplifier*

The motor controller is built using the Motorola DSP56F807. The controller supports PWM generation, encoder support, and A/D conversion for all existing axes. The amplifier uses the LMD18200 dual H-Bridge which accommodates up to 3A continuous output as well as current sensing. The ability to sense current is crucial as it provides a way to detect failures involving stalled motors. Particular attention was given to protect the robot against power cycle or shutdown. The motors and controller use separate power sources. Thus, we added simple circuitry to prevent the motors from running out of control if the controller happens to be off or reset, which could happen depending on the controller's initial state upon power reset.

5.1.2. *STACK*

The STACK [5] is a custom embedded architecture that was built by our group. It is a small footprint and expandable architecture consisting of up to 16 peripheral boards, interconnected through a bus to an 8-bit Rabbit 2000 microprocessor. The peripheral boards are designed to be physically stacked under the main Rabbit processor and communicate using a 9-bit RS-485 protocol through a common bus. [6] An 8 bit PIC microcontroller based carrier board is used to translate between the 9-bit RS-485 protocol to the main processor's 8-bit RS-232 protocol. We have implemented various addressable peripheral boards using PIC microcontrollers to provide a number of functions, such as interfacing to sensors and controlling actuators. In general, however, they can be built using any embedded processors as long as they can interface properly to the communication bus.

In Mertz, the stack consists of the DSP motor controller and a PIC-based board to interface with the sonar. We are currently developing a new version of the STACK which will be communicating using the CAN bus protocol and utilizing an upgraded PIC microcontroller. We are also planning to upgrade the main processor to the much more powerful PowerPC controller.

5.1.3. *Vision Processing*

Mertz receives visual input from two color digital cameras with IEEE-1394 (FireWire) interface, chosen for their superior image quality. They produce 640 x 480 24 bit color images at the rate of 30 frames per second. A laptop running Linux is used to interface with the camera and perform vision processing. The laptop communicates with the STACK's main processor using the RS-232 protocol. In the near future, we will use a cluster of Pentium based processors running Linux for additional vision processing.

5.2. *Software Systems*

5.2.1. *SensoryMotor Control*

Simple PD position and velocity control were implemented on the head and eye axes. Taking advantage of the Series Elastic Actuators, we use force feedback to implement force control for the neck joint. A simple PD position control was then placed on top of the force control. The sensorimotor software control will soon be augmented to include various online fault detection and diagnosis.

In addition, one of the design goals is to eliminate our past robots' elaborate start up sequence that has to be followed in precise order. We aim to be able to turn on the robot instantly. This entails fast and automatic calibration, which is made possible by having a potentiometer at every axis. Each axis uses the potentiometer to find its absolute position and then relies on the digital encoder for more precise position feedback. Upon startup, each axis is programmed to move to a default center location.

5.2.2. *Vision*

The vision system was implemented using YARP, an open source vision software platform developed in our laboratory. [7] YARP is a collection of vision code libraries, providing various functionalities. YARP supports message based interprocess communication distributed across multiple nodes. Processes can be dynamically started and connected to other existing processes.

As part of the preliminary integrated system, we have implemented an visual attention system to compute the most salient target for the robot to orient to. The main goals are to test the full range of all DOFs and create a starting point for the robot's vision system. We initially implemented an attention system previously used in past robots. [8] Inspired by Wolfe's model of attention [9], the system receives parallel input from low-level filters for saturated color, skin, face, and motion. Each filter's output is multiplied by individual weight and summed up to form a total saliency map. The face detector was developed by Paul Viola and Michael Jones. [15]

While this system worked very well in the past, a recent move to a new laboratory building has obliged us to make modifications due to the abundance of bright color interior and wooden walls at the new location. The new setting caused the attention system to be overwhelmed by the bright colored and skin colored background objects. It was simply impossible to attract the robot's attention with toys, moving hands, or faces. Ideally, motion detection should be helpful in distinguishing between foreground and background objects. However, the currently implemented frame difference approach for detecting motion is practically ineffective since the cameras are almost always moving.

In order to bias the attention system toward independently moving object, we implemented an additional mechanism for acquiring and storing color histogram data of independently moving objects. Whenever the robot is not moving and motion is detected, the system tries to segment a uniform color patch and store its color histogram. These stored histograms are then used for detecting other objects w/ similar principal color. By integrating this detector into the attentional system, it is now possible to attract the robot's attention using bright colored toys and other objects.

5.2.3. *Behavior System*

The high level behavior system for Mertz is programmed in Creal (CREAture Language). [10] Given that our aim is to test the full motion range of all the DOFs, we implemented a simple behavior control for the robot to orient its eyes, head, and neck to salient stimuli in the field of view.

Creal is a programming language similar in concept to the earlier *Behavior Language* [11], L [12], and others [13]. These languages were specifically designed to implement behavior based programs in the incremental and concurrent spirit of the subsumption architecture. [14]

Creal allows users to create modules which are collections of threads sharing a local lexical environment. Threads are either scheduled to run periodically, when a condition is true, or on the occurence of some event combinations. Thread scheduler is not preemptive. Threads are scheduled round robin and each one is run whenever appropriate until it suspends. Our benchmark suggests that the Rabbit 2000 processors can support close to 1,000 threads, we have not tested using more than 50 threads on a single processor to date.

Each module is equipped with both input and output ports. Modules are interconnected by virtual wires linking output ports to input ports, carrying either 8 or 16 bit messages. As defined in the original subsumption architecture [14], wires may *suppress* and *inhibit* each other. Moreover, each peripheral board is referenced in Creal as a buffer which can be written to or read from.

As shown in Figure 6, three behavior modules were implemented. Module *orient* and *flee* receives input from both vision and sonar. Module *explore* receives only from vision. Vision input consists of x and y coordinate of the most salient area as determined by the attention system, as well as an approximate depth measured by calculating disparity and relative changes in size of objects within the field of view. Sonar input is a measure of distance to the nearest object in front of the robot.

At the lowest level, module *orient* simply calculates motor commands in order to keep the most salient area in the center of field of view and maintain a minimum distance to objects in front of the robot. The *explore* behavior sends messages to the motor commands inhibiting *orient*'s output whenever the robot has been looking at an area for too long. It simply makes the robot turn a different way to explore. At the highest level, the *flee* behavior subsumes all lower-level behavior's output to protect the robot from objects that are either looming or far too close to the robot. The robot's response is to crank its neck back quickly to avoid the potential hazardous object.

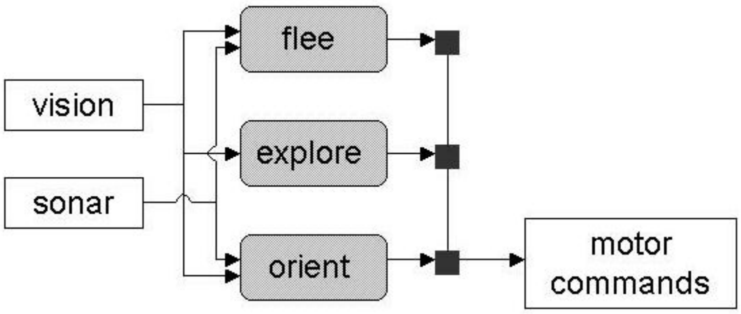

Fig. 6. Simple behavior control for orienting and attending to objects. The goal is to test the full range of all of robot's degrees of freedom.

6. Experiment

We conducted an experiment where we ran the robot continuously for hours at a time in order to test the current design and study failure modes. We setup the robot at four different locations and collected visual data in order to assess how visual input varies across places and time of day.

6.1. *Experimental Setup*

Using the preliminary integrated system described above, the robot is setup to run at the laboratory and three other locations in the building. All existing axes, minus the eyelids, were actuated. Experiment schedule is shown in Table 7. The shortest and longest duration are 6 and 19 hours respectively. Initially, the experiment was conducted with supervision. As the robot showed a reasonable level of reliability (with the exception of the neck joint that has to be rested every couple hours), we started leaving the robot alone and checked on it every hour. During the experiment, people were allowed to approach but not touch the robot. While unsupervised, a sign was placed near the robot to prohibit people from touching it.

6.2. *Results*

The head and eye axes are so far free of failures. Most of the failures originated from the mechanical failures on the neck SEA actuators. Table 8 lists each failure that occurred during the experiment.

All observed failures involve the neck joint and its Series Elastic Actuators. Loose screws seem to be particularly problematic. A probable explanation is that the neck actuators are constantly in motion. The force control loop produces output that is proportional to the linear pot signal plus some noise. A series of filter have been placed in order to minimize noise. In addition, the load of each SEA motor is

	Time	Duration	Location
Day 1	2pm – 10pm	8 hours	Laboratory
Day 2	12pm – 6pm	6 hours	Building Lobby
Day 3	10.30am – 11.30pm	13 hours	Balcony overlooking a student lounge
Day 4	9.30am – 4.30am	19 hours	Laboratory and moved to another area in the lab at 2 am

Fig. 7. Experiment schedule, time, and location.

	Hours after startup	Failure
Day 1	2	The two motors actuating the neck's Series Elastic Actuators started heating up.
Day 2	1	One of the Series Elastic Actuators popped out of the neck joint because of a loose set screw.
Day 3	7	A wire connecting the linear potentiometer signal on the SEA to a signal conditioning board is loose.
	10	A screw was found missing in one of the SEAs, causing the motor to stall and heat up very quickly.
Day 4	0	At startup, we found that the potentiometer placed on the neck's differential tilt joint has been un-calibrated because of a loose screw. Each axis is relying on its potentiometer to calibrate itself to a default initial configuration upon startup.

Fig. 8. List of observed failures during experiment. All failures originate from the neck joint and its Series Elastic Actuators.

very small causing the control output to be very sensitive to even a trivial amount of noise. A dead-band was placed in software to reduce this effects, which eliminates

Fig. 9. Examples of visual input taken from the same location at different times of the day.

Fig. 10. Examples of visual input from different locations in the building.

some but not all of the actuator's jitter. An alternative would be to put additional protection for the screws, i.e. using loctite on as many screws as possible. In addition, a better control mechanism for the SEA is definitely essential.

Collected visual data suggests that visual input varies greatly during the course of a day (see Figure 9) as well as at different locations (see Figure 10), which have different backgrounds, color schemes, and relative scale between objects. Figure 11

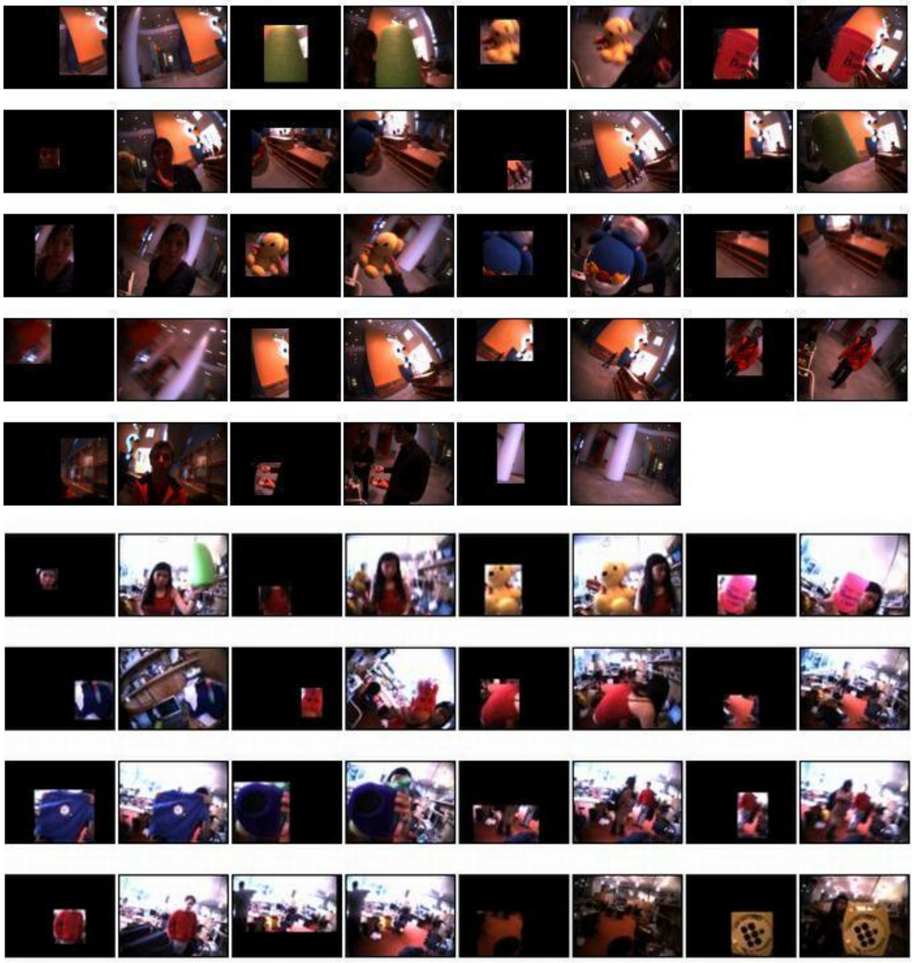

Fig. 11. Examples of the attentional system's output indicating where the robot should orient to.

shows some examples of the attentional system's output. Moreover, the vision system seems to work best in the laboratory where it was developed. At the new experiment locations, unexpected features, such as a large bright orange wall, tend to overwhelm the attention system such that it is very difficult to attract the robot's attention. In the same way that the new laboratory building has forced us to refine our visual attention system, this has motivated the next step in developing our vision system. We will have to incorporate a mechanism for the robot to 'remember' various background objects and not repeatedly re-discover them as the most salient object in the field of view.

7. Research Direction

Our long term research goal is to explore a more scalable learning framework, inspired by how human infants learn. Much of our varying skills and knowledge are acquired from learning from other people in a social setting, for example language and social cues. We plan to investigate an incremental learning approach which is not limited to a particular learning task, such as path planning or object recognition. We intend to create a robot that is always running and continuously developing new skills and gathering new knowledge online. Our vision is to have MERTZ be placed in a public venue for long periods of time, continuously interacting with different people and incrementally learning about patterns and correlations of perceived events. For example, the robot may learn to correlate objects with uttered phoneme sequences, learn to differentiate different people's faces, and learn to produce an initially randomly made motion which consistently lead to desired outcome, etc.

Long periods of continuous running time is particularly important in this context. In order to perform incremental learning online, the robot would need to acquire representative data. As we have seen, visual data varies greatly across time of day. Moreover, longer term patterns can only be captured by continuous observation.

8. Comparison to Related Work

HERMES is an autonomous humanoid service robot, designed specifically for dependability. It has been tested outside the laboratory environment for long periods of time, including an over six-month long exhibition in a museum where the robot is allowed to interact with the public. Although our project is exploring a different research direction, we fully concur with the underlying theme of increasing robot robustness and reliability.

Reliability is also a relevant topic in other museum tour-guide robots. [16,17,18] Deployment to a public venue and the need to operate on a daily basis naturally place reliability demands on these robots. Again, even though our robot is quite different in form and function, we are exploiting a similar demand to have the robot perform on a daily basis and interact with many people in a public venue.

Kismet is an expressive active vision head robot, developed to engage people in natural and expressive face-to-face interaction. [4] The idea is that we can bootstrap from these social competence to allow people to provide scaffolding to teach the robot and facilitate learning. In order for this to happen, the robot would need to be up and running in the midst of people on a daily basis, observing and learning from what it perceives. This would also allow a more dynamic social interaction to occur. These are essentially the long term research goals of our project.

9. Conclusion

We have designed and built MERTZ, an active-vision humanoid head robot for exploring incremental online robot learning in a social context. We have given particular attention to reliability and robustness issues during the design and construction of the robot. We plan to continue identifying failure modes throughout the rest of the robot development process in order to implement the appropriate fault detection and increase fault tolerance. Our immediate goal is to have the robot be up and running at a public venue for long periods of time.

References

1. R. Bischoff, V. Graefe, Design Principles for Dependable Robotic Assistants, International Journal of Humanoid Robotics, vol. 1, no. 1 (2004) 95–125.
2. J. E. Pratt, Virtual model control of a biped walking robot, Tech. Rep. AITR-1581, MIT Artificial Intelligence Laboratory, Cambridge, MA, USA, 1995.
3. R. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati and M. Williamson, The Cog Project: Building a Humanoid Robot, in C. Nehaniv, ed., Computation for Metaphors, Analogy and Agents, Vol. 1562 of Springer Lecture Notes in Artificial Intelligence, Springer-Verlag, 1998.
4. C. Breazeal, Sociable Machines: Expressive Social Exchange Between Humans and Robots, Sc.D. dissertation, Department of EECS, MIT, 2000.
5. R. A. Brooks, et al., ALIVE STACK User Manual, February 2003, unpublished.
6. R. A. Brooks, Proposed CREAL Communication Protocol - V1.0, July 2002.
7. P. Fitzpatrick and G. Metta. http://sourceforge.net/projects/yarp0
8. C. Breazeal and B. Scassellati, A context-dependent attention system for a social robot. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI99). Stockholm, Sweden. 1146–1151, 1999.
9. J.M. Wolfe, Guided Search 2.0: A revised model of visual search. Psychonomic Bulletin and Review,1(2):202-238, 1994.
10. R. A. Brooks, Creature language, Sep. 2003, http://www.ai.mit.edu/people/brooks/creal.pdf.
11. R. A. Brooks, The behavior language user's guide, AI Memo 1227, MIT Artificial Intelligence Lab, Cambridge, Massachusetts, Apr. 1990.
12. R. A. Brooks, C. Rosenberg, L -a common lisp for embedded systems, in Association of Lisp Users Meeting and Workshop, LUV'95 (1995) .
13. I. Horswill, Functional programming of behavior-based systems, *Autonomous Robots* **9** (2000) (1) 83–93.
14. R. A. Brooks, A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation* **2** (1986) (1) 14–23.
15. P. Viola, M. Jones, Robust Real-time Object Detection. Technical Report Series, CRL2001/01. Cambridge Research Laboratory, 2001.
16. W. Burgard, D. Fox, D. Hhnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun and A.B. Cremers, Real Robots for the Real World – The RHINO Museum Tour-Guide Project, Proc. of the AAAI 1998 Spring Symposium on Integrating Robotics Research, Taking the Next Leap, Stanford, CA, 1998.
17. S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Halnel, C. Rosenbert, N. Roy, J. Schultze, and D. Schulz, MINERVA: A second-generation museum tour-guide robot, Proceedings 1999 IEEE International Conference on Robotics and Automation, vol 3, pages 1999-2005, May 1999.

18. I. Nourbakhsh, C. Kunz, T. Willeke, The Mobot Museum Robot Installations: A Five Year Experiment, In Proceedings of IROS 2003, Las Vegas.