Real-Time High-Performance Attention Focusing in Outdoors Color Video Streams

Laurent Itti

University of Southern California, Hedco Neuroscience Building
3641 Watt Way, Room 30A - Los Angeles, CA 90089-2520 - USA
itti@pollux.usc.edu

ABSTRACT

When confronted with cluttered natural environments, animals still perform orders of magnitude better than artificial vision systems in visual tasks such as orienting, target detection, navigation and scene understanding. To better understand biological visual processing, we have developed a neuromorphic model of how our visual attention is attracted towards conspicuous locations in a visual scene. It replicates processing in the dorsal ("where") visual stream in the primate brain. The model includes a bottom-up (image-based) computation of low-level color, intensity, orientation and flicker features, as well as a non-linear spatial competition that enhances salient locations in each feature channel. All feature channels feed into a unique scalar "saliency map" which controls where to next focus attention onto. In this article, we discuss a parallel implementation of the model which runs at 30 frames/s on a 16-CPU Beowulf cluster, and the role of flicker (temporal derivatives) cues in computing salience. We show how our simple within-feature competition for salience effectively suppresses strong but spatially widespread motion transients resulting from egomotion. The model robustly detects salient targets in live outdoors video streams, despite large variations in illumination, clutter, and rapid egomotion. The success of this approach suggests that neuromorphic vision algorithms may prove unusually robust for outdoors vision applications.

Keywords: Attention, saliency, target detection, robot, video

1. INTRODUCTION

Animals demonstrate unparalleled abilities to interact with their natural visual environment, a task which remains embarrassingly problematic for machines. Obviously, vision is computationally expensive, with an estimated 10^8 bits/s streaming down each optic nerve,[1] and approximately half of the mammalian brain dedicated more or less closely to vision. Thus, for long, the poor real-time performance of machine vision systems could be attributed to limitations in computer processing power. With the recent availability of low-cost supercomputers, such as so-called "Beowulf" clusters of standard interconnected personal computers, however, this excuse is rapidly losing credibility. Seeking to bridge the gap between biological and machine vision, a new discipline has emerged which challenges classical approaches to computer vision research, that of *neuromorphic engineering.* This new research effort promises to develop models of biological systems with unparalleled robustness, on-line adaptability and applicability to real-world challenges.

We recently developed a neuromorphic model of how our visual attention is attracted towards conspicuous locations in a scene.[2,3,1] It replicates processing in posterior parietal cortex and other brain areas along the dorsal ("where") visual stream in the primate brain **(Figure 1)**. Because it includes a detailed low-level vision front-end, the model has been applied not only to laboratory stimuli, but also to a wide variety of natural scenes. In addition to predicting a wealth of psychophysical experiments, the model demonstrated remarkable performance at detecting salient objects in outdoors imagery — sometimes exceeding human performance — despite wide variations in imaging conditions, targets to be detected, and environments. Here, we describe a parallel implementation of the model, which runs at 30 frames/s on a 16-CPU Beowulf cluster. We have previously reported preliminary results on merging this model to a model of object recognition at each attended location, based on the simulation of neurons in inferotemporal cortex and other areas along the ventral ("what") visual stream.[4,5] The combined model, of which a prototype is available, will provide both localization and identification of the few most interesting objects in a scene. Here, we also present a prototype hardware platform that has a small Beowulf cluster on-board and is being developed to test our neuromorphic vision algorithms in real-time, outdoors applications.

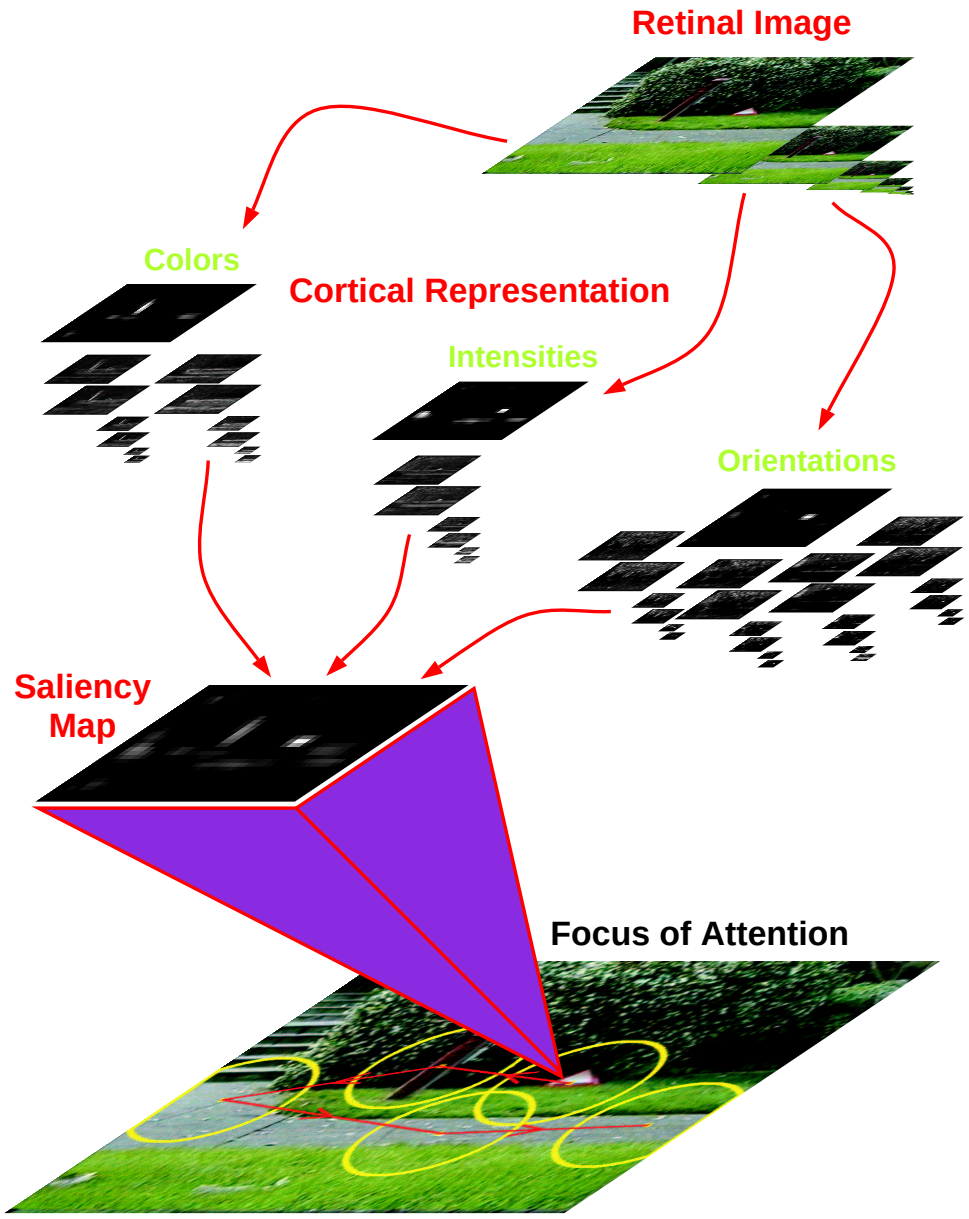

Figure 1. Overview of the bottom-up attentional selection model for single frames (no motion/flicker processing). The input image is decomposed into seven channels (intensity contrast, red/green and blue/yellow double color opponencies, and 0°, 45°, 90° and 135° orientation contrasts) at six spatial scales, yielding 42 feature maps. Here the six intensity maps, 12 color maps, and 24 orientation maps are combined into the three summary maps shown. After iterative spatial competition for salience within each map, only a sparse number of locations remain active and all maps feed into the unique saliency map. The latter is scanned by the focus of attention in order of decreasing saliency, through the interaction between a winner-take-all neural network (which selects the most salient location at any given time) and an inhibition-of-return mechanism (which transiently suppresses the currently attended location from the saliency map, as shown on the four bottom frames). In the example shown, the system first attended to the emergency triangle, mostly because of the strong responses in the orientation and red/green channels.

In this article, we focus on several new components of our approach. First, we describe how saliency for image transients may be computed, yielding an improved version of our model which exploits video streams rather than individual frames. Second, we describe how the architecture of the model allows a fairly easy mapping of the computational algorithm onto a cluster of networked CPUs, for improved real-time performance through distributed processing. Finally, we describe a prototype robotics platform which implements a reduced version of the parallel model for real-time, embarked operation.

2. SALIENCY FOR IMAGE TRANSIENTS

Motion and flicker cues are certainly the most important for primates in attracting attention, but had not been integrated to previous versions of our model, mostly due to computational limitations. Using distributed processing over a Beowulf cluster of computers (see next section), we have been able to increase throughput sufficiently to allow us to start experimenting with dynamic stimuli and video sequences.

Here we focus on the simplest approach, which only considers image transients in the luminance channel. Future implementations will include such transients in all channels, as well as dedicated motion energy detectors that respond to specific motion patterns.[6]

A new multi-scale feature channel is created, whose basic architecture follows that used for the other channels.[2] Starting from the difference between the luminance channels of the current and previous frames, a dyadic Gaussian pyramid[7] is created, with a depth of nine scales. In order to extract regions in the image in which the local transients are different from surrounding transients, a center-surround mechanism is applied, which is implemented in a computationally efficient manner as a simple difference across scales. Thus, with scale 0 being the original luminance difference image and scale 8 being that image down-scaled by a factor of $2^8 = 256$ horizontally and vertically, six center-surround flicker feature maps are computed as the absolute differences between scale pairs 2-5, 2-6, 3-6, 3-7, 4-7 and 4-8.[2,3] A spatial competition for salience is then applied to each feature map, whose role is to suppress those maps in which extended areas of activity are present (e.g., due to movements of the camera) while enhancing maps which contain smaller, isolated transients (e.g., a small moving object). This spatial competition is modeled after the known non-classical surround inhibition processes that have been characterized electrophysiologically in the primary visual cortex of primates.[8–10] After competition, all feature maps are summed to provide input to the saliency map which controls where attention is to be deployed next.

Although extremely simple, this initial implementation of a saliency channel sensitive to image transients has proven very robust to full-field transients (e.g., due to camera motion) which are eliminated by our spatial competition for salience, while efficiently detecting smaller moving objects, such as persons or hands. Thus, despite the noisy and rather unstable input to our flicker channel (simple difference between two frames), our spatial competition for salience scheme appears efficient at selectively enhancing perceptually salient transients.

3. REAL-TIME HIGH-PERFORMANCE IMPLEMENTATION

The architecture of the model as shown in **Figure 1** is well-suited for coarse-grain parallel implementation. Below we describe two very simple extensions of our model for operation on Beowulf clusters of interconnected personal computers: the first fully processes each incoming video frame onto a different processor; throughput (frames processed per second) is multiplied by the number of available processors, but latency (time to fully process one frame) is not improved. The second extension distributes processing of each frame over the computing cluster, with different nodes processing different feature dimensions for a same frame. With this second implementation, both throughput and latency are improved. Both implementations were tested on a 16-node Linux Beowulf cluster with 733 MHz processors and triple 100 Mbps fast Ethernet interconnect (using channel bonding under Linux to group the three separate physical networks into a single logical one).

3.1. Round-Robin Processing

The simplest approach to parallelizing our model consists of processing successive frames on different CPUs, in a round-robin manner **(Figure 2)**. Our implementation of this approach relied on the Parallel Virtual Machine communication library from Oak Ridge National Laboratory (`http://www.csm.ornl.gov/pvm/`). This library provides a simple programming interface that easily allows transmission of images (or other datasets) between nodes in the cluster. The model implementation then simply consists of grabbing a frame, sending it off to the next available

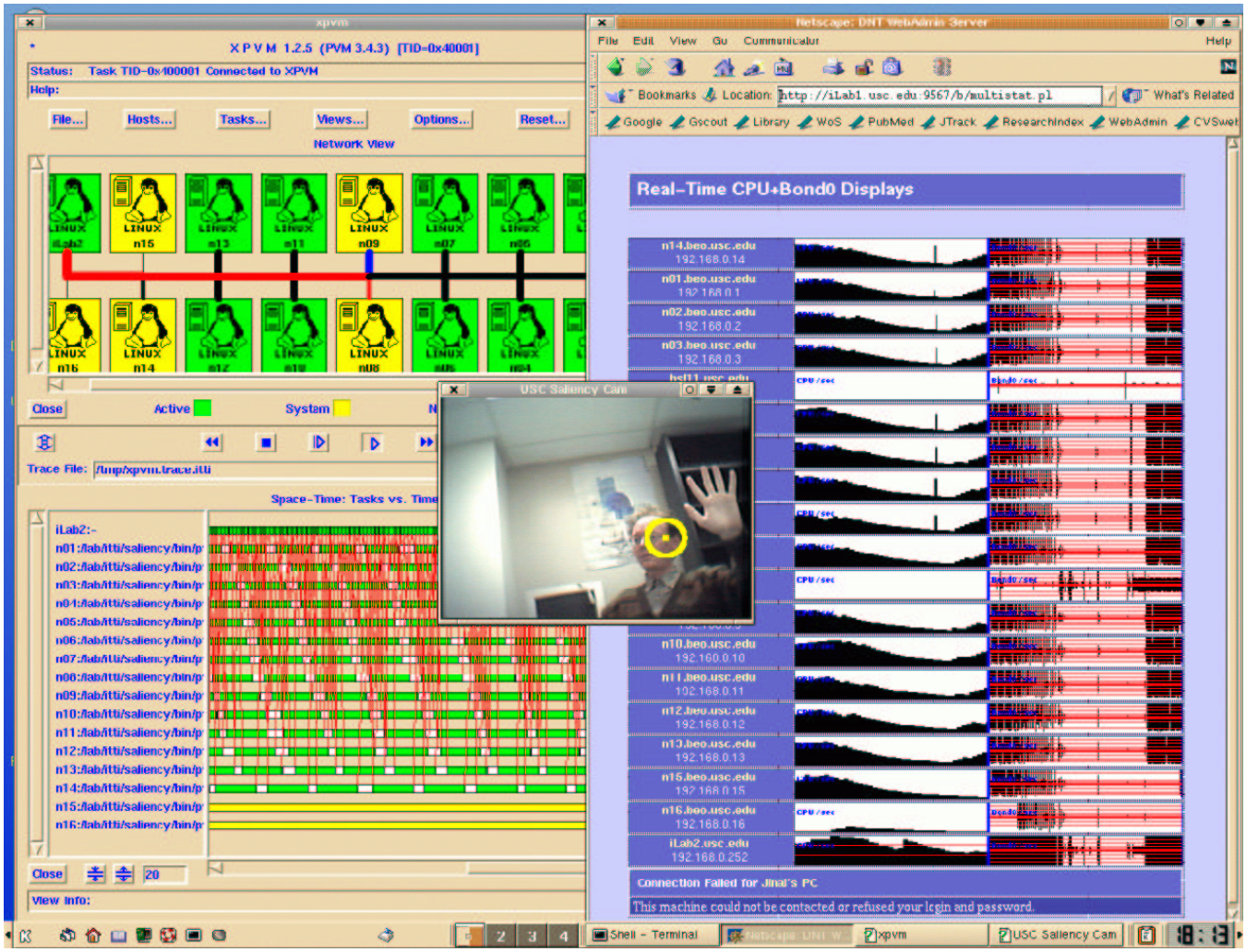

Figure 2. Round-robin processing using PVM. In this example, each 320×240 frame captured on the master node (ilab2) from the video source is processed in $\approx 1.2s$ on a given compute node (n01..n16). The left half of the screen shows how the model uses 14 compute nodes to process the incoming video (at 15 frames/s). The right half of the screen shows CPU and network load on the various nodes which are both high (the blank towards the center of the graph resulted from an interruption in processing).

CPU for full processing, and checking whether results from a previous frame have become available from one of the processing CPUs. Throughput is then limited by the number of available CPUs and approximately improves linearly with that number, while latency is slightly increased compared to single-CPU processing (due to additional network transport latencies).

Although extremely simple, this initial implementation proved interesting in evaluating the performance of PVM for our application. Because whole frames are transferred from the master node (which captures images from a video source) to the compute nodes at a rate of 15 or 30 frames/s, our application is fairly network-intensive. In this context, PVM proved fairly inefficient, with an effective network throughput approximately 8 times lower than the actual raw network capacity (as measured from the round-trip time to send a frame from the master node to a compute node and to receive it back without processing; this time thus includes all overhead introduced by PVM in formatting, packaging, receiving and decoding the message, in addition to network transport time per se). This study thus prompted us to develop a simpler and more direct transport library, based on direct TCP connections among nodes. This allowed us to better approach true network capacity, as the cost of a loss of generality (for example, our library does not support heterogeneous clusters in which different nodes with different hardware architectures may operate with differing floating-point representations or byte orderings). The source code for the library and the complete model is available from our web site at `http://iLab.usc.edu`.

3.2. Distributed Processing

A more sophisticated approach consists of distributing processing of a single frame across the cluster of compute nodes, for example as shown in **Figure 3**. The distributed implementation naturally exploits the fact that the model processes different visual attributes, such as color, orientation, flicker or intensity, in separate channels. Thus, some CPUs are dedicated to processing flicker while others process color, intensity or orientation information.

One master node is responsible for grabbing video frames and displaying the results. Because these operations typically only occupies the processor for 10-15ms of the 33ms available between two successive frames at 30 frames/s, the remaining CPU time is used in our implementation to compute the luminance, red-green, and blue-yellow channels used by our model, from the original grabbed color frame. These three images are sent to three different computation nodes (numbered 0, 1 and 2). Node 0 starts with relaying the received intensity image to nodes 3, 4, 5, 6, and 8, which will compute four orientation channels and the flicker channel. It then computes the intensity *conspicuity map (CMAP),* which is the sum of the six feature maps obtained through center-surround differences in a Gaussian pyramid created from the luminance image, as described in the previous section (also see refs.[2,3]). The resulting CMAP is sent to node 7, which collects all CMAPs for a given frame and sums them to yield the final saliency map for that frame. Nodes 1 and 2 compute color CMAPs in a similar fashion (sum of center-surround differences between six pairs of scales in an image pyramid), while nodes 3, 4, 5 and 6 compute 0°, 45°, 90° and 135° orientation CMAPs and node 8 computes the luminance flicker CMAP. Node 7 maintains a working buffer of saliency maps for up to ten frames, and keeps track of how many of the eight CMAPs have been received for any given frame. Once all eight CMAPs have been received, the location of maximum activity in the saliency map is detected, and its coordinates are send to the master node for display.

Node	Action	Processing time
0	Intensity	3 ms
1	Red/Green	3 ms
2	Blue/Yellow	3 ms
3	0° Orientation	8 ms
4	45° Orientation	8 ms
5	90° Orientation	8 ms
6	135° Orientation	8 ms
7	Collector	< 1 ms
8	Intensity flicker	3 ms

Table 1. Processing time for the various components of the model running on different 733 MHz Pentium-III CPUs, for each frame in a 320×240 video stream in 24-bit color. This table does not account for network transport times.

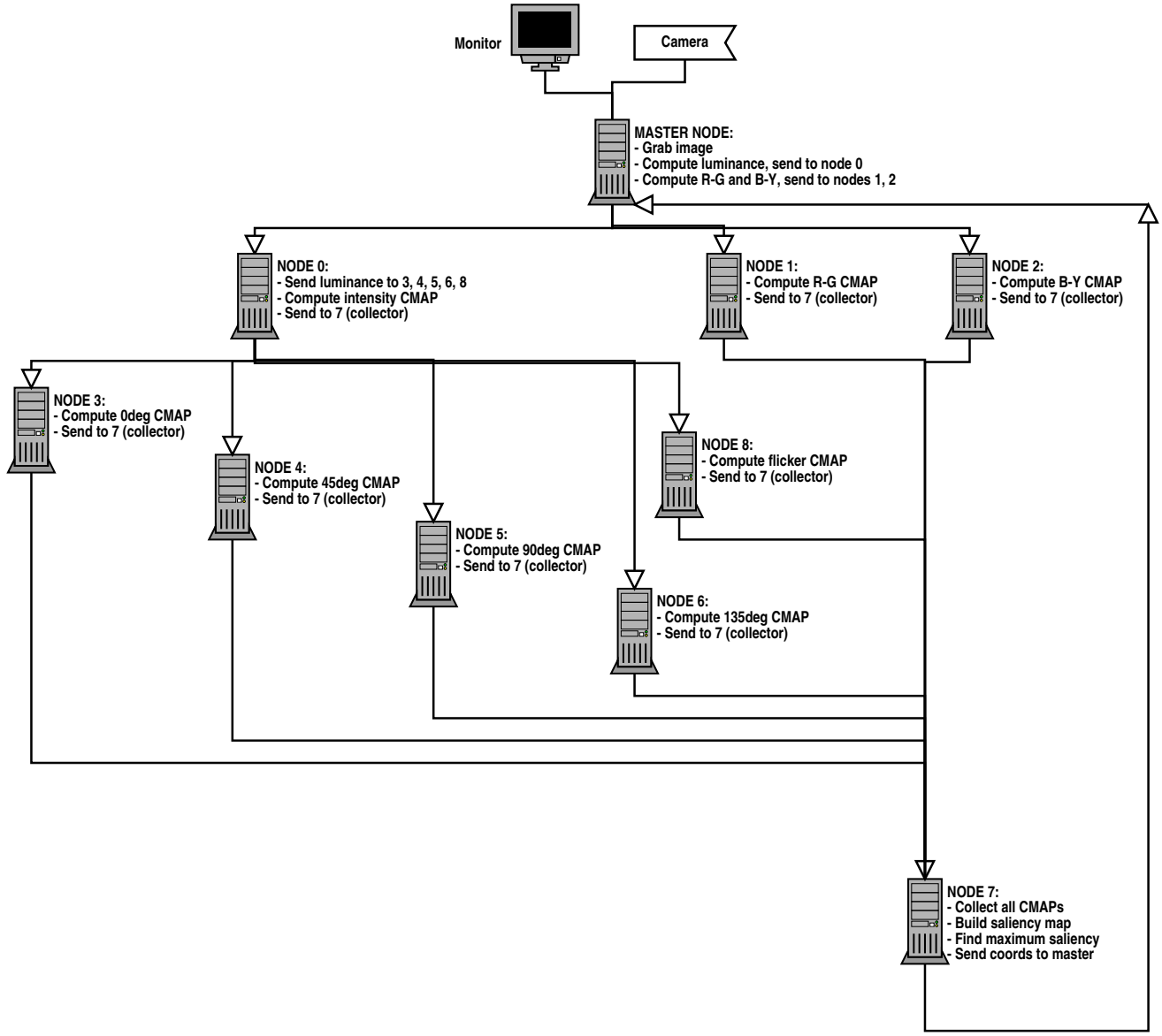

Figure 3. Distributed processing using feature-level parallelism and faster communications. A 320×240 video input is processed by nine compute nodes at 30 frames/s.

With this approach, both throughput and latency are improved compared to the baseline single-CPU implementation. **Table 1** shows the processing times for the different tasks assigned to the nine processing CPUs used for the implementation in **Figure 3**. For 320×240 color images, processing time is sufficiently short to yield an overall latency (including all network transport delays) shorter than two frames (60ms), well suited for interactive operation.

With the two high-performance implementations of our model just described, we have shown how computational neuroscience models may in the near future become much more widely applicable than is currently achieved,

3.3. Towards Embedded Applications

Based on the successful implementation of our model on a Beowulf cluster, we started developing a new autonomous robotics platform, the "Beobots." Beobots consist of small Beowulf clusters mounted on four-wheel-drive mobile platforms **(Figure 4)**. The computational core of the Beobots consists of two 1.26 GHz dual-CPU computer boards, interconnected by a Gigabit Ethernet link.

While claims that machine vision or other engineered systems are based in one way or another on biology have been popular over the last decade, too often these systems are only loosely inspired from broad biological principles. Nevertheless, the idea of biomimetic robots is not new (see, e.g.,[11] for a review). The motivation behind most biorobotics efforts is that a robot which closely resembles biological systems may constitute a good physical model for understanding problems such as locomotion (e.g., the distinct morphologies of the corkroach's three pairs of legs are key in allowing it to both run and climb, and thus has been reproduced in robots[12,11]). This is not our goal here, as our focus is on computational vision. It is hence important to clarify that the main characteristic of Beobots compared to existing robots is that they will use neuromorphic vision algorithms, and that whenever possible these algorithms embody some fundamental aspect of visual behavior (e.g., bottom-up attention to salient targets) rather

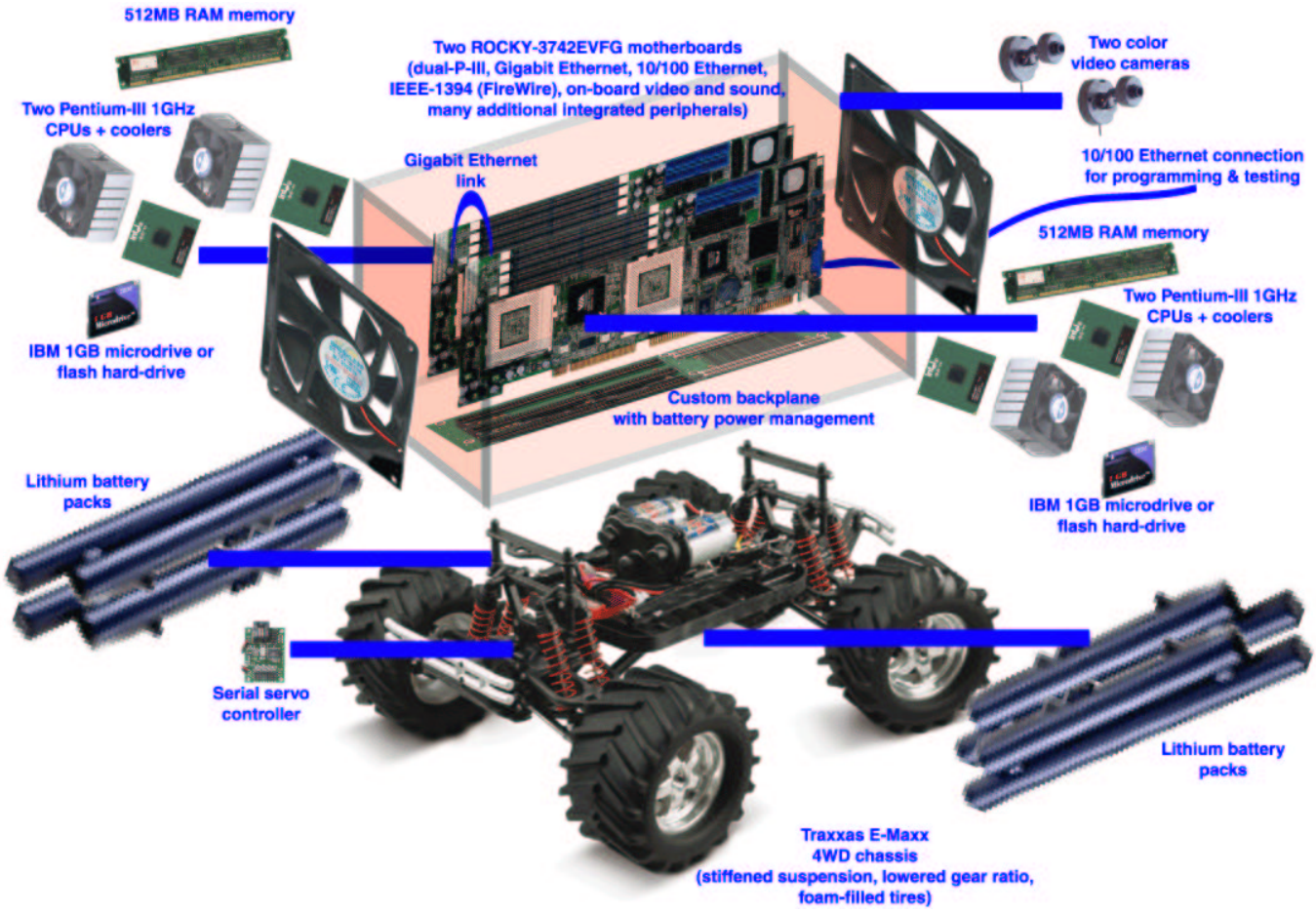

Figure 4. Beobot architecture.

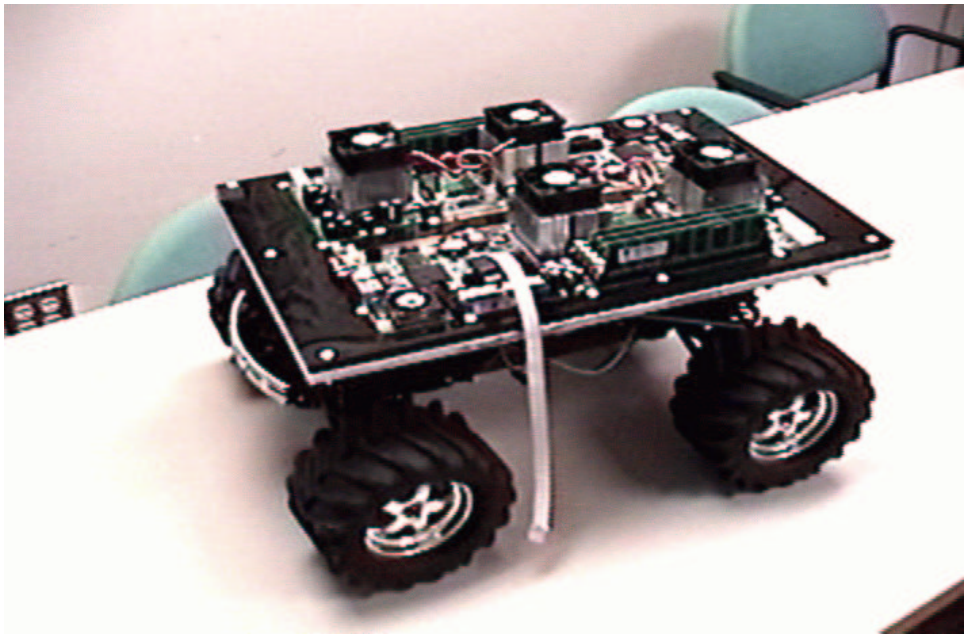

Figure 5. Beobot prototype, which can run our saliency-based visual attention software at 30 frames/s at a resolution of 160 × 120 in 24-bit color.

than being dedicated to a particular task (e.g., an algorithm that matches an internal model of a road to the visual input).

A slightly scaled-down version of the distributed algorithm described in the previous section has been implemented on the Beobot prototype shown in **Figure 5**, which runs at 30 frames/s with 160 × 120 video input.

4. DISCUSSION AND CONCLUSION

Neuromorphic vision algorithms promise greater robustness, generality and adaptability than traditional computer vision approaches, and aim at more closely replicating the so-far unmatched performance of animal vision. Here, we have shown that neural models of biological vision need not be restricted to scientific hypothesis testing using simple laboratory stimuli. Rather, we have extended our neuromorphic model of saliency-based visual attention for real-time operation on clusters of interconnected computers.

Although biological vision is computationally very expensive (with approximately half of the human brain dedicated more or less closely to vision), the recent availability of Beowulf computer clusters has allowed us to fairly quickly implement and test a distributed version of our algorithm. Because Beowulf clusters are general-purpose and allow for quick implementation, compared to, for example, developing custom high-performance processors, we expect that other computationally expensive neural vision algorithms will benefit from similar high-performance implementations.

Our broader goals in demonstrating real-time neuromorphic vision algorithms and developing the Beobots platform are to demonstrate two points: First, that state-of-the-art computational neuroscience models of vision can be extended to yield real, useful, general and widely applicable computer vision systems, and are not restricted to the testing of neuroscience hypotheses based on simple, laboratory stimuli. Second, that a biologically-inspired approach to traditionally hard computer vision problems can yield unusually robust and versatile vision systems, which work with color video streams in real-time, and can efficiently adapt to various environmental conditions, users, and tasks.

What will Beobots be capable of that existing robots cannot already achieve? Again, our main focus is on the vision component. Most robots have underemphasized this component, and relied instead on dedicated sensors including laser range finders or sonars. Much progress has been made in developing very impressive *physically capable* robots, as demonstrated by the Honda humanoid robot or a number of semi-autonomous robots developed for exploration of unfriendly grounds (see, e.g., the Mars exploration robots and "Urbie" the urban robot from

the Jet Propulsion Laboratories at robotics.jpl.nasa.gov). In addition, very sophisticated and powerful algorithms are now available that make robots *intelligent* (e.g., algorithms by which teams of robots collaborate[13] towards a common goal). However, we believe that some improvement still is possible in making robots more *visually capable*, as current systems often rely on simplified, low-computation visual tricks which greatly limit their autonomy (for instance, although the Honda robot can climb stairs, it needs to know the step size in advance, because its vision does not allow it to adapt to various staircases autonomously; Urbie is limited to visual servoing and relies heavily on human operators; the USC interacting robots operate in high-contrast, saturated-color, parallelepipedic block worlds; and most robots that use vision rely on dedicated algorithms developed for a specific task, which may limit their robustness, adaptability, versatility and wider applicability). Our endeavor, thus, is to further our understanding of how neuromorphic vision algorithms may enhance the visual capabilities of robots, thus allowing them to reach greater autonomy than is currently available.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation, the National Eye Institute, the National Imagery and Mapping Agency, the Charles Lee Powell Foundation, and the Zumberge Faculty Research and Innovation Fund.

REFERENCES

1. L. Itti and C. Koch, "Computational modeling of visual attention," *Nature Reviews Neuroscience* **2**, pp. 194–203, Mar 2001.
2. L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, pp. 1254–1259, Nov 1998.
3. L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research* **40**, pp. 1489–1506, May 2000.
4. F. Miau and L. Itti, "A neural model combining attentional orienting to object recognition: Preliminary explorations on the interplay between where and what," in *Proc. IEEE Engineering in Medicine and Biology Society (EMBS)*, in press.
5. F. Miau, C. Papageorgiou, and L. Itti, "Neuromorphic algorithms for computer vision and attention," in *Proc. SPIE 46 Annual International Symposium on Optical Science and Technology*, in press.
6. E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *J Opt Soc Am [A]* **2**, pp. 284–299, Feb 1985.
7. P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Trans on Communications* **31**, pp. 532–540, 1983.
8. M. W. Cannon and S. C. Fullenkamp, "A transducer model for contrast perception," *Vision Res* **31**(6), pp. 983–98, 1991.
9. A. M. Sillito, K. L. Grieve, H. E. Jones, J. Cudeiro, and J. Davis, "Visual cortical mechanisms detecting focal orientation discontinuities.," *Nature* **378**, pp. 492–6, Nov 1995.
10. L. Itti and C. Koch, "Feature combination strategies for saliency-based visual attention systems," *Journal of Electronic Imaging* **10**, pp. 161–169, Jan 2001.
11. B. Webb, "Can robots make good models of biological behaviour?," *Behavioral and Brain Sciences* **24**(6), in press.
12. G. M. Nelson and R. D. Quinn, "Posture control of a cockroach-like robot," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, pp. 157–162, 1998.
13. B. Werger and M. J. Mataric, "From insect to internet: Situated control for networked robot teams," *Annals of Mathematics and Artificial Intelligence* **31**(1-4), pp. 173–198, 2001.