7. Acknowledgments

The authors would like to thank Dr. George Bekey, for providing the facilities and environment in which this research was conducted. We would also like to thank Jim Montgomery for writing the display software used in generating the random sampled images. Discussions with Andrew Fagg, Gaurav Sukhatme and Michael McHenry were valuable in the development of the concept of dynamic foveation and during the debugging of the hardware.

8. References

Bederson, B.B, *A Miniature Space-Variant Active Vision System*, Ph.D. dissertation, NYU department of Computer Science, 1992.

Fagg, A. H., Lewis, M. A., Montgomery, J. F., Bekey, G. A., "The USC Autonomous Flying Vehicle: an Experiment in Real-Time Behavior-Based Control". *Proceedings of the 1993 IEEE Conference on Intelligent Robots and Systems*, July, Yokohama, Japan, pp. 1173-1180.

Horswill, I., Yamamoto, M., *A $1000 Active Stereo Vision System*, submitted to CVPR 94.

Lewis, M.A, "AFV-II: Robotic Aerial Platform for Autonomous Robot Research," submitted to AUVS '94.

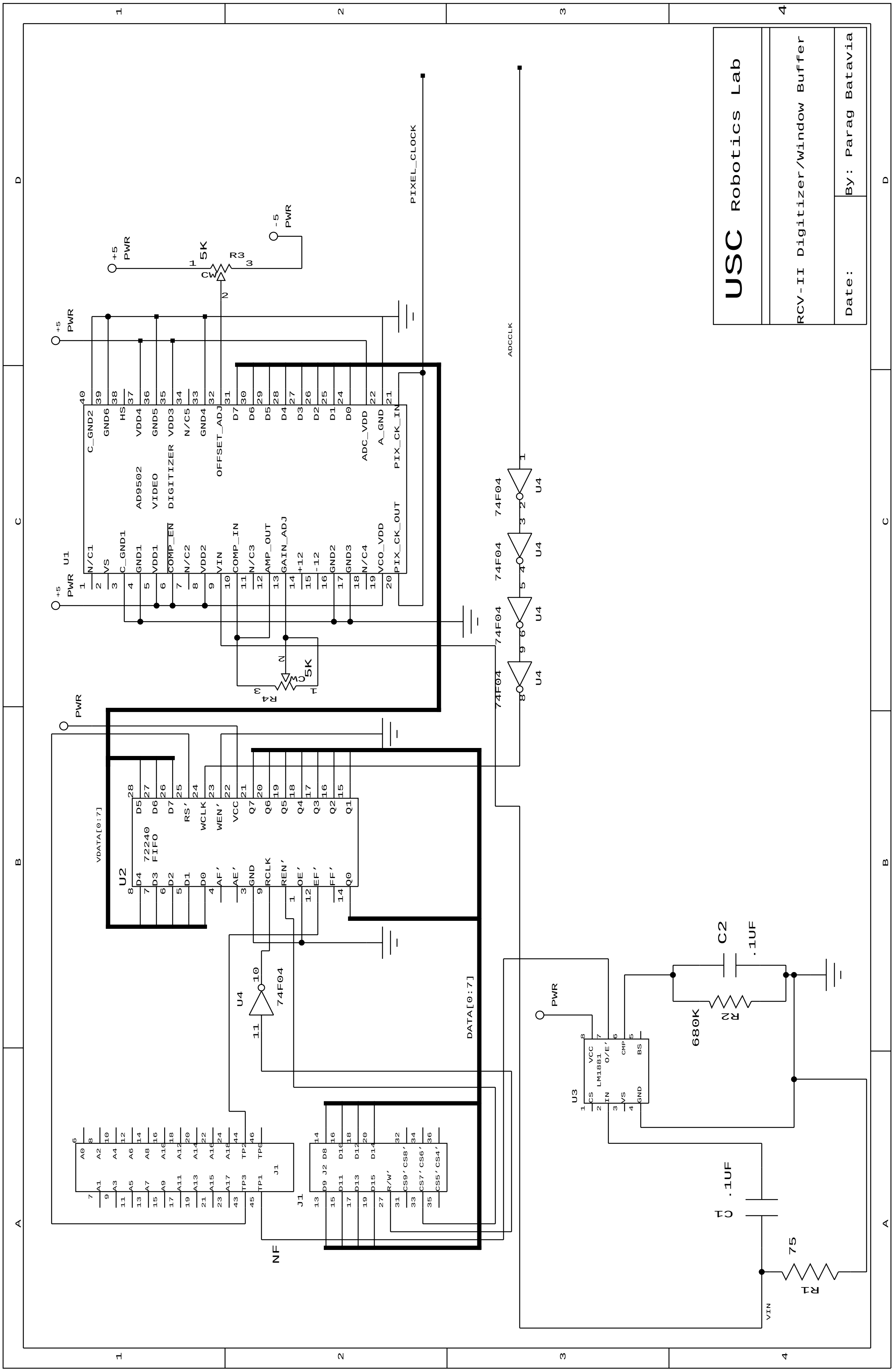

Figure 8. The Window Buffer section of the RCV II.

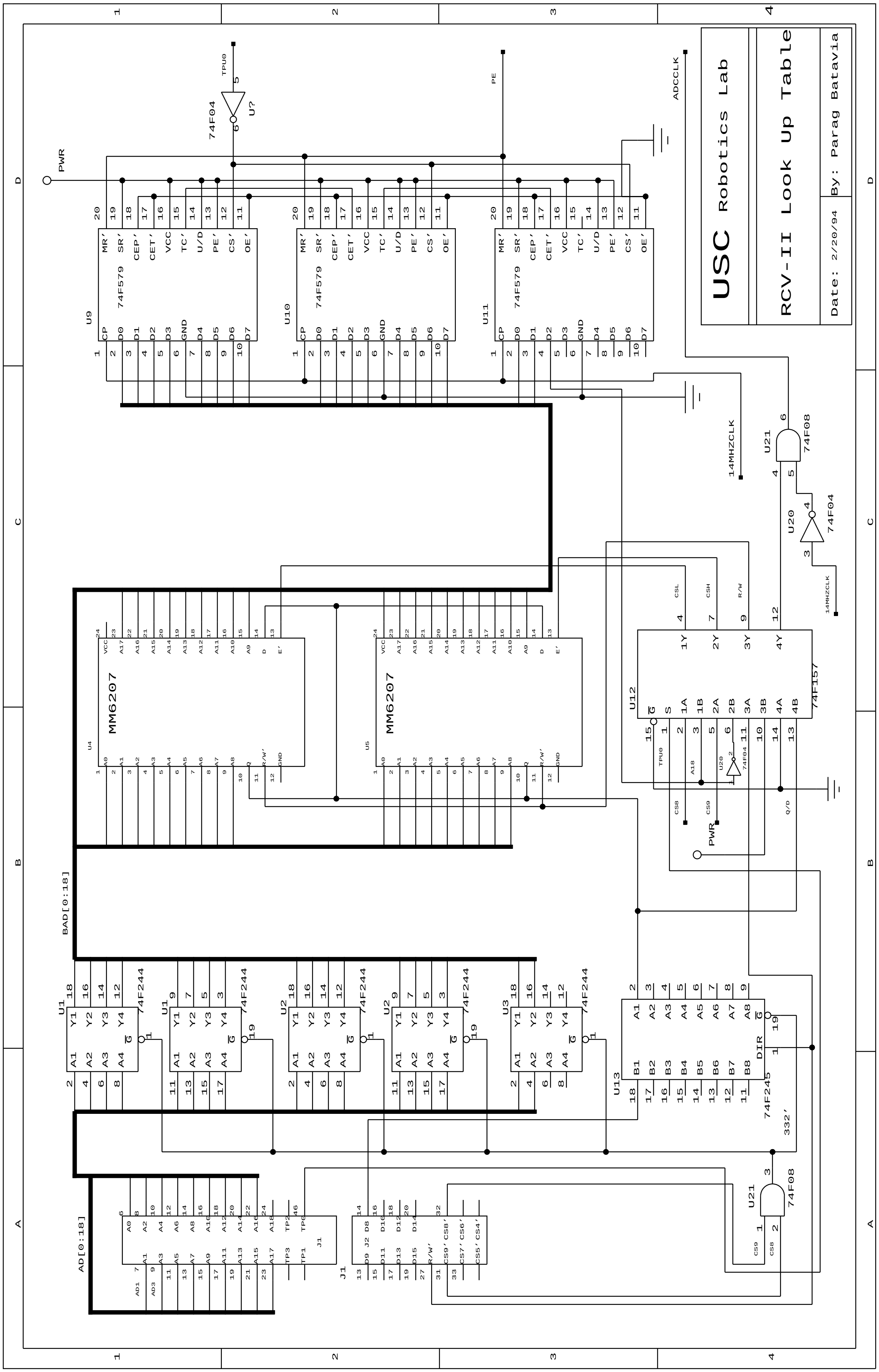

Figure 7. The Look Up Table portion of the RCV II

Figure 5. A normal sparse sampling of a portion of our lab.

Figure 6. The same scene, as it appears when the sampled points are chosen randomly.

Figure 1. A sparse sampling of the entire frame at an effective 384x384 resolution.

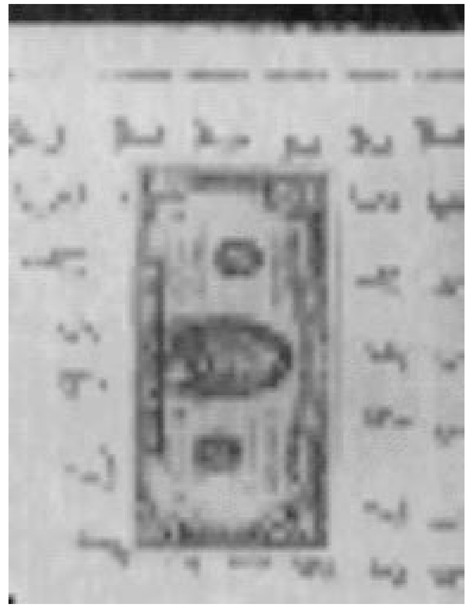

Figure 2. A denser sample, taken when an "interesting" object was found. Effective resolution is 256x256.

Figure 3. By zooming in even further, more details are available, even though there is still only 4K of data. The resolution is 128x128.

Figure 4. At the densest possible sample of 64x64, the maximum detail is available, but this covers only 1/36th of the full frame.

5. Future Work

5.1. Applications

We plan to start using the RCV in a number of our mobile robot platforms. First and foremost will be its use in the USC Autonomous Flying Vehicle (Fagg, 1993 & Lewis, 1994). The AFV was designed to compete in a yearly competition at the Georgia Institute of Technology. The goal of the competition is to develop a flying vehicle which is capable of autonomously finding and picking up target disks, and returning them to a predetermined drop off location. The RCV will be used to locate the disks and to identify landmarks on the playing field which will be used for navigation.

The RCV will also be used in Meno II, which is our second generation 4-legged walking machine. Meno II will be used in landmark based navigation and gaze stabilization experiments.

5.2. RCV III

The next generation of the RCV will make use of newer chipsets designed for video digitization, such as the Samsung KS0116 series which is a 3-chip set capable of decoding, digitizing and encoding color NTSC, SVHS, and PAL video. This will allow us to generate processed images in real time and display them on color monitors. The use of these chipsets will also eliminate a one pixel roll that we sometimes encounter due to our not using a phased locked loop to synchronize to the start of frame.

We will also be moving away from the 68332BCC as the main controlling processor to Texas Instruments DSP chips, such as the 320C40, which is capable of up to 50 MFlops, as opposed to the'332's 2 MIPS. The recent reduction in price of these VLSI chips will keep the cost of the RCV near the current level, while greatly enhancing its capabilities.

6. Conclusion

The USC RCV II is an experiment in low cost/low complexity vision processing that will have a large impact in the use of vision in the USC Robotics Lab.

We hope to make the RCV available to interested universities. The schematics are included (Figures 7-8) and permission is granted to construct and perform experiments with this board.

As we are not currently interested in sampling greater than 4K worth of data, we grab the data as fast as it comes in. As the data comes in, it is stored in onboard RAM. Once the required data is all available, the window buffer is reset, the counters are shut down, and control of the LUT is returned back to the CPU. Note that this "context switch" consists of merely switching the status of two TPU lines, and therefore takes very little time.

4. Results

4.1. Tracking

Using the USC RCV, we were able to demonstrate tracking of a object - a piece of paper - at a frame rate of 15 frames per second. We would grab an image and place it through an adaptive threshold. After that, the image was shrunk and grown twice to remove noise, and then a center of mass was found.

We used the center of mass information as input to a position/velocity tracking algorithm, which was able to track the object over a large range of speeds. The tracking was only one dimensional, however we will soon be using a saccade device which is capable of performing saccadic movements at near human velocities in two dimensions.

4.2. Region Extraction

We have also almost finished developing a "zooming" algorithm which will be able to find the salient features of a scene, as was simulated in Figures 1-4. We have already demonstrated the identification of blobs using a connected region extraction routine. Using this we hope to be able to automatically take a sparse sample, and find the important features in it.

4.3. Random Sampling

To demonstrate the versatility of spatially variant sampling, we took digitized an image based on a purely random sampling strategy. Sampling points were generated at random, and then a display was generated using an algorithm which "grew" pixels around the sampled points until collisions occurred. The resulting images are in Figures 5-6. Because the information content of the image is of arbitrary density, it cannot be reliably used to guide a tracking system or identify objects. It does, however, demonstrate that the RCV is capable of any form of sampling desired.

3.5. Window Buffer

The window buffer is the second major alteration to the design of the RCV. Previously, a 4Kbyte dual port memory was used for storage. This had the advantage of random access capability, but there was no built in synchronization mechanism. Therefore, frames were continuously grabbed, and the CPU never knew when an entire new frame was in memory. This lead to skewing when the system was in rapid motion, as one frame would actually consist of part of two frames. Another problem was that of concurrent access between the two sides of the dual port. When the CPU would try to read a location in the dual port that was simultaneously being written to, garbage data would be returned.

Using a bidirectional FIFO eliminates these problems. This is because the FIFO, an Integrated Device Technology IDT72240, contains built in synchronization signals. Empty, full, almost empty, and almost full flags are present. Therefore, when data begins to arrive, the CPU is notified by the FIFO itself indicating that it has new data. Once the data is read by the CPU, it is discarded by the FIFO. The FIFO is 4K deep, allowing 4K of data to arrive before the CPU has to pay any attention to it. As we normally only grab 4Kbytes of data, the entire image can be buffered without CPU intervention.

3.6. Operations

To set up operation, then, the first thing that is done is the CPU takes control of the LUT and generates a fill pattern for the virtual frame based on the number and location of the pixels to be sampled. After this is done, the system is ready to begin capturing frames.

When the RCV determines that it wants a new frame, it unmasks the new frame interrupt. When the interrupt arrives, the first thing that happens in the interrupt routine is control of the LUT is turned over to the window buffer, and the free running address counters are activated.

The counters, which are clocked by a 7.23 MHz pixel clock provided by the AD9502, sequentially accesses the lookup table, starting at the beginning of the LUT's virtual frame. As the virtual frame is clocked through, a sample is taken whenever a 'high' is output by the LUT. This signals the window buffer FIFO to take a sample from the AD9502. To reduce the AD9502's conversion time, samples are continuously taken, and stored only when the LUT indicates a sample to be taken.

3.3. Analog Section

One of the first changes was the incorporation of a hybrid NTSC video digitizer chip, the Analog Devices AD9502. This greatly simplified the analog front end of the system by incorporating everything needed to digitize a raw video signal on one chip and a few external passive components. Previously, a standard 8 bit flash A/D converter had been used. This required pre-processing of the video signal in the form of amplification, plus a greater amount of filtering. Even with that circuitry, image quality was grainy and noisy, even at the best resolution available. The images generated by the AD9502 are crisper, have a lower signal to noise ratio, and have better contrast than the previous version.

The analog section also includes a National Semiconductor LM1881 sync stripper. This IC takes an NTSC signal as input, and provides vertical and horizontal sync signals, and an odd/even frame signal. The odd/even signal is used to determine the start of a new frame. When the signal transitions from low to thieving to odd), a new frame is beginning. This is tied to a TPU line on the 332 set up as an interrupt transition counter. Therefore, whenever the CPU is looking for a new frame, it unmasks the interrupt tied to the TPU line, and when the new frame begins, the CPU is interrupted, and can begin gathering the data.

3.4. Look up Table

The most critical section of circuitry is the look up table (LUT). It is here that the sampling strategy is set up. The look up table is a bitmap of the entire possible video frame. When a pixel is to be sampled, it is set to a high state by the CPU. This is done for all the pixels (up to 4096) which need to be sampled.

The LUT consists of 2 motorola MM6206 128K*1bit static RAMS. This allows for 256K pixels, which corresponds to a 512*512 image, and enables a virtual frame to be represented. The memory is converted to dual port memory by multiplexing it between the CPU and the window buffer subsection. This is done by buffering the address bus between the memory and the LUT, and using tri-state counters to generate the window buffer address which will be discussed below. The one data pin is switched between the CPU and the frame buffer with a TTL multiplexer. The CPU uses a TPU pin to determine whether the LUT is connected to the CPU or the window buffer.

3.1. Description of the 68332

The Motorola 68332 is a SISD processor based on the 68000 series family, commonly used in the Macintosh. However, as it is a microcontroller, rather than simply a microprocessor, it has a number of features which make it suitable for the control of external hardware, such as ten programmable chip select (CS) lines, 15 programmable time-processor unit (TPU) lines, and a queued serial module (QSM) for inter-processor communications.

The time processor unit is an independent module of the 332 which is capable of capturing signals and generating waveforms without using CPU time. The 332 merely has to set up the initial conditions, and then, in the case of an input capture, it is notified via an interrupt when the capture has occurred. In the case of an output, the CPU only has to specify the parameters of the waveform to be generated, and it is then continuously generated with no CPU overhead.

The 332 also allows for a "protected-mode" operating system to be installed by implementing supervisor and user modes. When the CPU is in supervisor mode, all instructions and areas of memory may be accessed. When the CPU is in user mode, however, certain areas of memory may be placed off-limits to the currently running process, preventing it from affecting and being affected by other processes, allowing for a degree of robustness without heavy operating system overhead.

3.2. RCV Rev. 2 circuit overview

The first version of the RCV system consisted of a motorola 68332 BCC, and 22 MSI and LSI chips. This version consumed approximately 8 watts of power without the camera. With the camera, total power consumption was nearly 10 watts. A further limitation of this design was that it was never able to process more than 8 frames per second.

Our current version consists of the 68332 BCC and only 15 MSI and LSI chips. Power consumption was also reduced to 5 watts. Additionally, we are also able to perform simple operations such as adaptive thresholding at near frame rate and object tracking at half frame rate.

2. Related Work

Because the RCV's primary goal was to provide a low cost and low energy consuming vision platform which could be easily interfaced to mobile robots, it does not have all the functionality of more expensive and complicated systems.

One such system is the *Cortex-I* developed at NYU by Bederson and Schwartz (1992) which uses a log-polar camera to generate spatially variant images and two DSP chips to process them. It is this work which has inspired our efforts in active vision. Their sensor provides a dense foveated center with high resolution, which falls off exponentially as distance from the center increases. The RCV is capable of log-polar mapping, but it has the advantage that it is not limited to any one form of scanning. In one instance a log-polar map may be the most suitable option, but later a gaussian pyramid may be the optimal choice. The RCV can dynamically alter it's sampling to accommodate this. However, unlike *Cortex-I*, the RCV does not make use of DSP's and its ability run sophisticated algorithms is limited by the processing power and lack of floating point on the 68332.

Another system is MIT's Cheap Vision project (Horswill, 1993). Their systems currently consists of a 68332 interfaced directly to a CCD chip. This allows for an even greater reduction in cost and power consumption than the RCV, however, the CCD they use provides only a 192 x 165 resolution, and they are limited to a fixed sampling scheme. They can, and do, subsample an image, but they cannot use arbitrary sampling schemes.

3. System Description

There are three key components to the RCV, the CPU submodule, digitizer module, and window buffer. This report describes the second version that we have built. Except for the CPU submodule, there have been substantial changes in the design of the RCV between the two revisions.

1 Introduction

The USC Reduced Complexity Vision board was designed as an alternative to high priced, high power vision engines. Rather than spending tens of thousands of dollars on dedicated vision workstations, we designed a system which can perform low level vision tasks at frame rates up to real time, using less than $500 in parts. This was primarily accomplished by limiting the amount of data that can be processed per frame. This results in two savings - less memory is required to store an image, and less CPU power is required to process the image.

These savings allow the vision system to be used in rapid prototype situations, where an autonomous robot needs to be developed and tested within a couple of months. Because a PCB is being developed for the RCV, adding vision to a mobile robot will become a matter of purchasing the proper components and assembling them on the PCB.

To retain some of the functionality of higher end vision systems, a concept called *dynamic foveation* was used. Dynamic Foveation allows the user of the system to change the sampling scheme of the device on the fly. This way, when initially surveying a scene, a sparse sampling of the entire field can be taken. This image, which at most can be 4Kbytes of data (generally a 64 by 64 pixel array), can be quickly analyzed for any points of interest.

When such a region is found, the system can reconfigure itself to more closely examine that portion. Therefore, instead of taking a sparse sampling of a large area, a denser sample of a smaller area is processed, yielding more information.

The system can also support *spatially variant* digitizing. In this form of active vision, the resolution of the image changes across the frame. This allows for the system to maintain its awareness of the entire scene while focusing a greater proportion of its attention and resources on a particular region. If the region of interest were to shift suddenly, the shift would still be detectable because of the sparse sampling along the outlying regions of the frame. To demonstrate the generality of our spatially variant digitizing capability, we were able to take a random sample, in which the pixels to be digitized were randomly generated.

Table of Contents