REPORT DOCUMENTATION PAGE

AFRL-SR-AR-TR-02-

0391

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
		4/1/99 TO 3/31/02 Final

4. TITLE AND SUBTITLE
Distributed Optoelectronic Processing of Multidimensional Digital Imaging

5. FUNDING NUMBERS
61102F
2305/DX

6. AUTHOR(S)
Professor Brady

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Univeristy of Illinois
109 Coble Hall
801 S Wright St
Champaign IL 61820-6242

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of the Air Force
Air Force Office of Sdentific Research
801 N. Randolph St Rm 732
Arlington. VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F49620-99-1-0229

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Radical improvements in electronic sensor and processor capabilities in the past decade have destabilized basic definitions of imaging in general and three-dimensional imaging in particular. Conventionally, imaging refers to analog focal or holographic systems that integrate information acquisition and processing. Increasingly aggressive digital processing, however, diminishes the processing role in the sensor head. Particularly in sensor arrays, there is often no need for a well-formed "image" in analog space.

The divide between digital and analog systems is particularly pronounced for multidimensional imaging. Holographic and stereoscopic sensors record the illusion of 3D scenes, but do not in fact construct 3D models. Tomographic and other 3D scene analysis schemes create true 3D digital models from sensor array data. In most cases, however, users do not demand and cannot process full 3D models.

3D video systems may be categorized in the following hierarchy:

20020909 126

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

Argus

Distributed Optoelectronic Processing for Multidimensional Digital Imaging

Final Report

AFOSR Grant Number
F49620-99-1-0229

5 August 2002

Submitted by

David J. Brady
Department of Electrical and Computer Engineering
Fitzpatrick Center for Photonics and Communication Systems
Duke University, Durham North Carolina

Background

This program was part of a larger effort exploring the integration of digital and analog processing in optical sensor systems. This work explores the nature of imaging as imaging systems continue to evolve to highly digital processing from dense sensor arrays. These systems are particularly relevant to Air Force target tracking and analysis applications.

This program was conducted for two and a half years at the University of Illinois, after which the team moved to Duke University. Duke has provided substantial support in extending the Argus array developed under this program to more heterogeneous tracking applications consistent with AFOSR interests. Currently, the algorithms, optical designs and data management tools developed under this program are being applied under AFOSR support to interferometric telescope tracking applications on the three College observatory in Greensboro, North Carolina. We are coordinating this project with Professor Bob Plemmons at Wake Forest University. Professor Plemmons is involved in the Air Force Space Awareness program in Maui, Hawaii. We hope to ultimately transfer these technologies to space awareness and flying object tracking.

Objectives

This project began with our presentation "Computed Tomography in the Visible Spectral Range" at the Air Force Science Advisory Board meeting in Dayton, Ohio on November 18, 1998. At that meeting we described progress on computational imaging systems growing out of interferometric imaging efforts under previous AFOSR support. We discussed how larger testbeds would allow us to build computational imaging systems capable of capturing and analyzing complex environments on a distributed processing network.

Our goal was to develop embedded optoelectronic processing components and algorithms for efficient tracking and analysis of Air Force targets. In pursuit of this goal we constructed a flexible sensing and processing testbed for sensor array development and sensor data fusion. We began construction of the testbed with the start of this program in the spring of 1999 and completed the first phase construction in September 1999. As discussed in the original proposal, components for the testbed were obtained using internal support from the Beckman Institute to match the AFOSR commitment. As a result of Beckman support, the testbed was somewhat larger than originally proposed.

The testbed consists of an array of microcomputers. The array is interconnected to implement distributed parallel processing of sensor data. Each computer supports several data acquisition ports. A variety of sensors and sources, including CCD and CMOS cameras, microphones, specialty CMOS smart sensors, and laser diode arrays have been integrated into the array acquisition ports for system development and testing.

The first goal of the testbed was to demonstrate real-time 3D video acquisition. A 3D video is a sequence of 3D images obtained at video rates. We used computational inversion of tomographic projections to construct 3D video. The testbed was arrayed as a frame of cameras surrounding a room. The 3D scene in the room was captured and transmitted at video rate.

Data management on distributed sensor arrays involves many computational tasks, including data acquisition, abstraction, analysis and communications. Exploration of novel distributions of these tasks was the primary theme of this project. In particular, the Argus testbed has illuminated the imbalance between scene analysis and information distribution in conventional systems. Most sensor arrays rely on hierarchical trees for data fusion and assume that all sensor information feeds into a single user. With the Argus project, our goal was to embed as much processing as possible at the lowest possible levels and to gather and process data for multiple simultaneous uses. With this in mind, the analysis of the digital data flow limits and processing range of the array was a primary goal of the project. This report covers the development and design of the Argus imaging testbed and discusses the application of this system for 3D video streaming as well as distributed sensing and computation for multi-user stereo imaging applications. We also discuss the inclusion of ad hoc wireless sensors developed under this initiative.

RESULTS OF EFFORT

Radical improvements in electronic sensor and processor capabilities in the past decade have destabilized basic definitions of imaging in general and three-dimensional imaging in particular. Conventionally, imaging refers to analog focal or holographic systems that integrate information acquisition and processing. Increasingly aggressive digital processing, however, diminishes the processing role in the sensor head. Particularly in sensor arrays, there is often no need for a well-formed "image" in analog space.

The divide between digital and analog systems is particularly pronounced for multidimensional imaging. Holographic and stereoscopic sensors record the illusion of 3D scenes, but do not in fact construct 3D models. Tomographic and other 3D scene analysis schemes create true 3D digital models from sensor array data. In most cases, however, users do not demand and cannot process full 3D models.

3D video systems may be categorized in the following hierarchy:

1. **Integrated capture and display systems**. These systems capture a stereo view and project that view for the user. Such systems include holographic and autosteroscopic recording and display solutions. These systems may be represented by this block diagram:
2. **Discrete capture and display systems.** These systems include stereo camera recording/holographic or stereo display pairs. A block diagram for this approach is shown above.
3. **Capture, inversion and display systems.** As exemplified by tomographic imagers, these systems form complete 3D models of the scene and reproject these models for viewers. A block diagram for this approach is shown below.

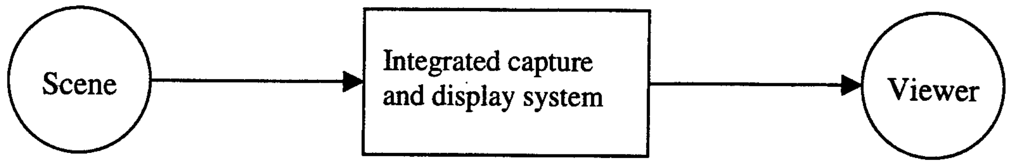


Figure 1. An integrated stereoscopic capture and display system.

Approach 1 developed in the age of analog processing. Ubiquitous digital processing power makes 2 and 3 increasingly attractive. The fundamental attraction of these two approaches is that they support multiple users. 2 and 3 are also more adaptive and allow separate optimization of capture and display parameters.

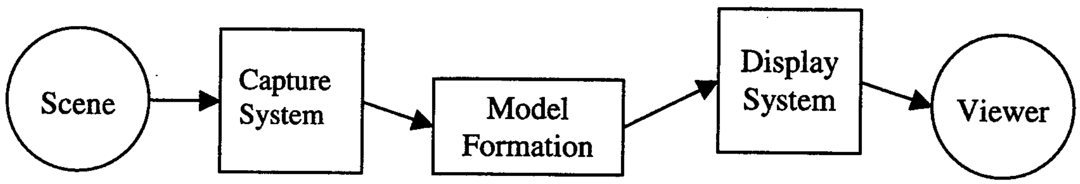
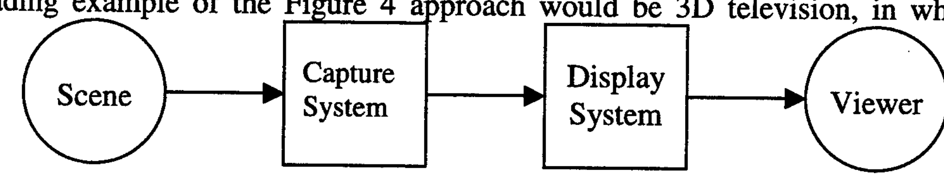


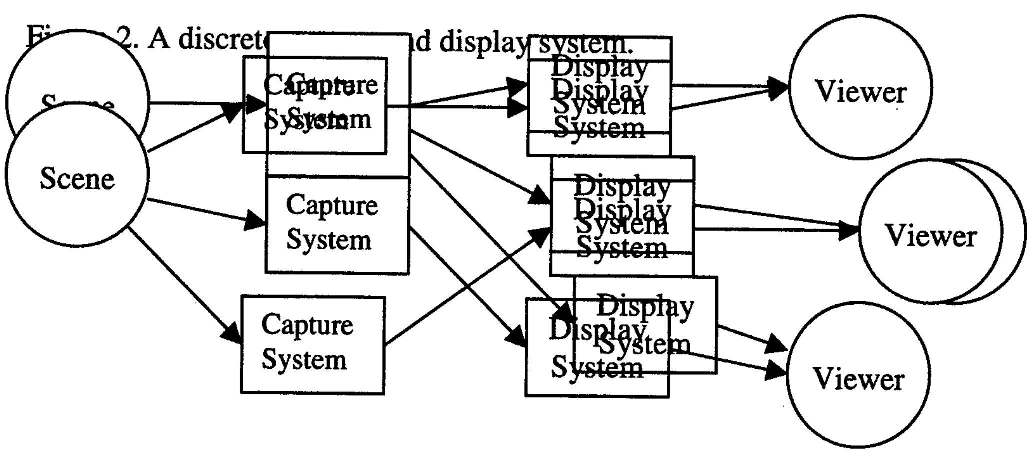
Figure 3. A capture, inversion and display system.

Our focus in this paper is multi-user functionality. We are interested in "streaming" 3D video, meaning video that is mapped in real-time across networks. In this context "video" refers to real-world images rather than computational graphics. (Although we allow for the possibility that computed views may be used to augment, approximate or otherwise improve the visual illusion of displayed real-world scenes. Just as digital imaging is blurring the distinction at the analog/digital interface between image formation and image processing, digital display is blurring the distinction at the digital to analog interface between image projection and image creation.)

Approaches 2 and 3 become considerably more complex in multiuser environments. The primary issue is *at what level should branching into multiuser pathways begin?* Approach 2, for example, might be constructed based on the single capture system block diagram shown in Figure 4 or the multiple users and multiple capture system shown in Figure 5. A leading example of the Figure 4 approach would be 3D television, in which a single



F[illegible]2. A discret[illegible]d display system.



[illegible]

stereo view is shown to many users. The Figure 5 approach has been less commonly implemented, but has been developed on Argus, as described below.

Image inversion systems also correspond to a variety of potential information flow maps. One can imagine complex interconnectivities between distributed capture systems, full and partial inversion systems and distributed displays.

While considerable analysis has been applied to localized capture of stereo and tomographic data, to the inversion of this data for scene models and to the reprojection and display of partially and fully inverted data, relatively little attention has been devoted to how networks of sensors and display systems might be integrated in interactive 3D video. To address the need for analysis of data flow on distributed sensor networks, we have constructed a distributed sensing and display testbed at the University of Illinois. This report describes our 3D video testbed, which we have named Argus. According to the Greek Mythology link http://homepage.mac.com/cparada/GML/Argus1.html

> *Argus has been called "the all-seeing because he had eyes in the whole of his body. Other say he had one hundred eyes in his head and that they slept two at a time in turn while the rest remained on guard.*

Our Argus is a distributed supercomputer with 64 distributed digital cameras. This paper describes the hardware architecture of this system and presents results of 3D image streaming. Section 2 describes the hardware, section 3 describes the algorithms used in preliminary testing, section 4 describes the results of tests implemented locally in Illinois and streamed over the network from Illinois to Japan and North Carolina.

Argus is composed of two major parts. The first is the digital hardware. The second is the sensor space, which is essentially a studio surrounded by cameras. Argus computer system consists of 32 Dual Pentium-II slave computers, a master computer, and a file server. All 34 computers run version 7 of the Mandrake Linux operating system release. The operator uses the master node to initiate data acquisition and computation.

The master node consists of the following components:

- Dual Pentium-II 450 Mhz with 512 MB ECC RAM
- Supermicro P6DBE dual Pentium-II BX based motherboard
- 6.4 GB Western Digital EIDE hard disk
- Diamond Viper V550 AGP RIVA TNT based video adapter
- One 3C905B ethernet card
- One 3C16925 gigabit ethernet card (fiber optic)
- 21 inch Mitsubishi monitor
- Keyboard, mouse, and other necessary utensils

The file server consists of the following components:

- Dual Pentium-II 400 Mhz with 256 MB ECC RAM

- Supermicro P6DBE dual Pentium-II BX based motherboard
- Adaptec 2940-U2W SCSI controller
- 17.4 GB Barracuda Seagate Ultra-2 Fast/Wide SCSI hard disk
- 12/24 GB Scorpion DDS-3 Seagate tape backup drive
- S3 Virge 4 MB video card (no monitor)
- One 3com 3C905B ethernet card

The 32 slave nodes each consist of

- Dual Pentium-II 400 Mhz with 256 MB ECC RAM
- Supermicro P6DBE dual Pentium-II BX based motherboard
- 6.4 GB Western Digital EIDE hard disk
- S3 Virge 4 MB video card (no monitor)
- One 3com 3C905B ethernet card
- Two Haüppauge Win/TV Brooktree BT878-based video capture cards (which are supported by Video4Linux)
- Two Omnivision OVT 5016AB black and white CMOS cameras
- No monitor, keyboard, or mouse

In addition, there are two linked 24-port 3Com Superstack 3300 switches connecting the computers together. One of the Superstack 3300 switches contains a gigabit module that connects to the master node. The system interconnectivity is illustrated in Figure 6.

Argus uses the latest advances in distributed processing and real-time computing to generate models efficiently. The computational work done by the system is performed on a Beowulf class computer cluster as described by[1, 2]. The Message Passing Interface (MPI) [3] is a common tool used for communicating on clusters

As illustrated in Figure 7, the sensor space consists of a fourteen-foot diameter camera framework. The framework is constructed from 2-inch pipe in an octagonal shape. The cameras are equally spaced within a wooden circular frame that circumscribes the octagonal pipe construction. Arranging the cameras along the wooden circle instead of

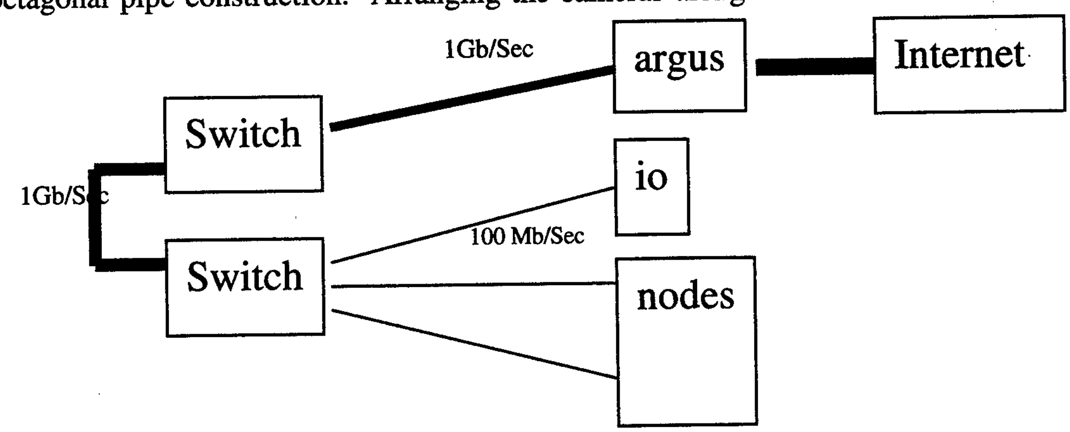


Figure 6. Argus connectivity. Io is the file server. Argus is the master node.

along the main octagonal frame simplifies data gathering since the position of the

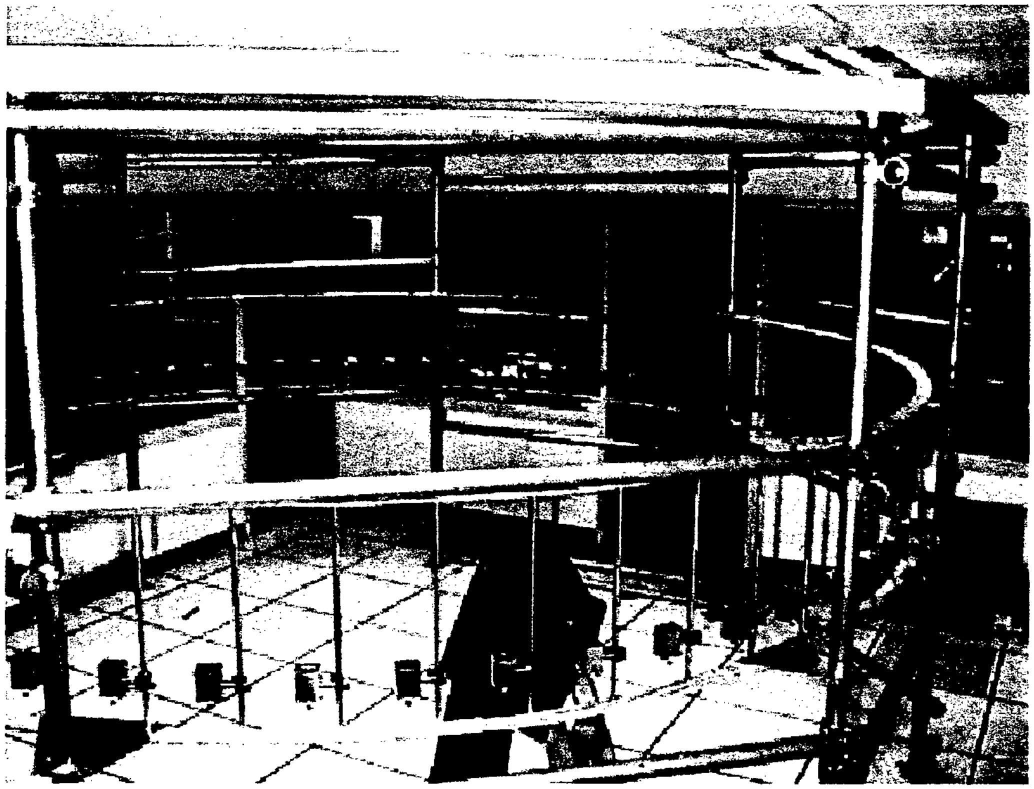


Figure 7. The Argus sensor space. The small silver boxes are the cameras. In use, the image cage is draped with black curtains and the floor is covered with black carpet.

cameras is known. In order to keep light sources and objects outside the sensor space from interfering with the imaging process, an eight foot tall black curtain is installed on the inside of the framework with holes cut for the lenses of the cameras. For similar reasons, the floor was covered with black carpet. Ten 500 W halogen lamps placed across the top of the frame provide even lighting within the sensor space.

The initial sensors used in Argus were grayscale CMOS focal planes with 2 mm lenses. The output from the cameras is an NTSC signal with 320 x 240 resolution. In the final stages of the project, we included 64 color Firewire cameras with a YUV resolution of 640 x 480. The cameras as evenly spaced along the circumference of the sensor space, pointing inward. A common power supply/controller for the cameras was constructed to provide the capacity for frame synchronization on the analog CMOS cameras. For the Firewire cameras, we rely on the computer clock for frame synchronization.

Due to the imprecise method of mounting the cameras, alignment is a significant challenge. Initially, the cameras were all placed at the same height with a laser leveling system. From there, the cameras were aimed by hand until they reported a point source at the center of the imaging volume to be at the center of the captured image. The imprecise camera mounting hardware caused this method of alignment to be accurate within two pixels. Digital alignment is required to align the cameras further.

To digitally calculate position and alignment vectors for all cameras a reference object is placed in the imaging volume. Figure 8 shows a sensor space view of a calibrated tetrahedron developed for this process. Point light sources are mounted on the vertices of the tetrahedron for alignment. Figure 9 shows a camera view of these point sources. Automated triangulation on the point sources for each camera creates a pixel level position correction for each camera. This correction is used at the first step of the cone beam algorithm to register projection data.

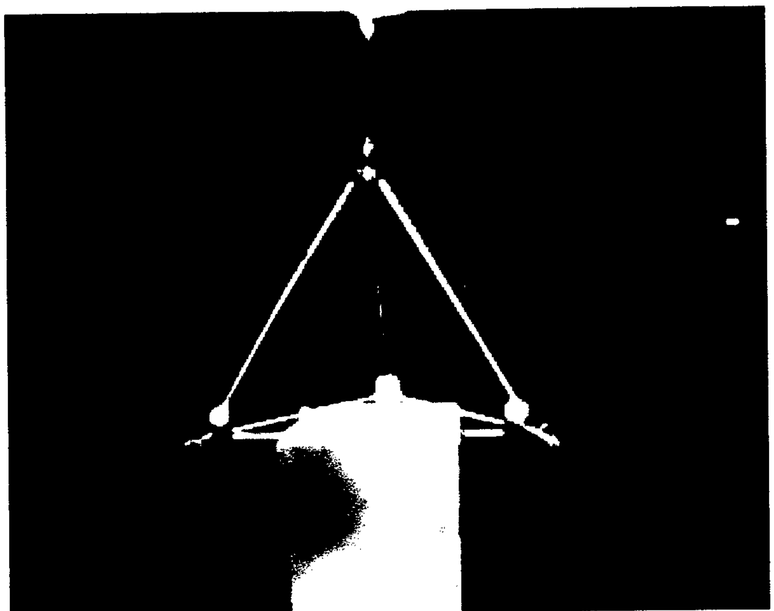


Figure 8. Image of the tetrahedron used for the alignment process.

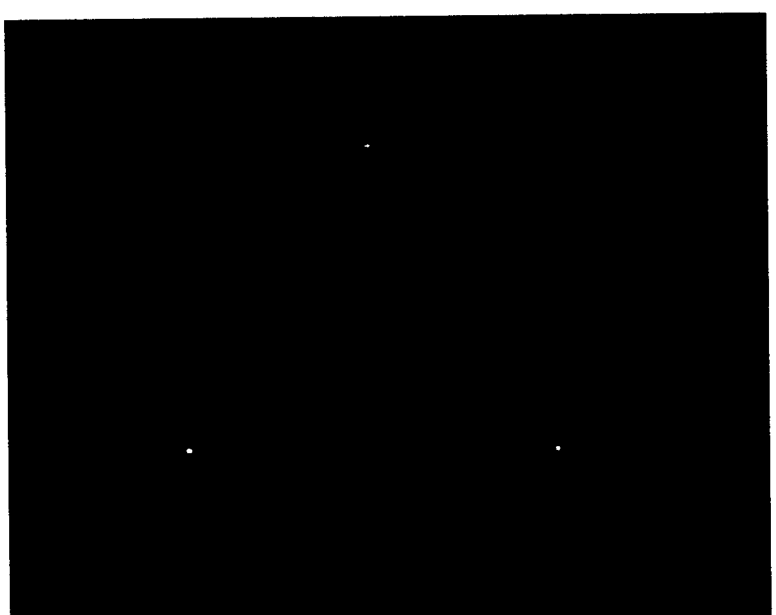


Figure 9. Image of the lights located at the vertices of the tetrahedron with no external lighting.

Argus is capable of generating both stereo pair views of the image space (treating the sensor array as a discrete mesh as in Figure 5) and complete three-dimensional models of the space (as depicted in Figure 3). Each type of data has different characteristics and performance capabilities in distributed display applications. This section will describe both data types and discuss the advantages and disadvantages of each.

Showing a user a separate image for each eye with slightly shifted perspectives produces the illusion of three-dimensionality. This slight shift of viewpoint mimics the way view the real world where the shift is determined by the separation of the eyes. The end result is a very convincing three-dimensional experience. The images shown to the left and right eye are considered a stereo pair. Stereoscopes have used this technique for years to give viewer a sense of depth.

In the Argus project, stereo pairs are produced using images captured from two adjacent cameras. These images are then digitally aligned to make up for any errors in the physical camera placement. A shift is also necessary to prevent the images from converging to produce a cross-eyed effect. Since the system consists of a large number of cameras in a circle, a stereo pair can be produced at all points of interest around the exterior of the space. The limited amount of processing required in generating stereo pair data results in high-speed data streams. An example of a stereo pair streamed from Argus is shown in Figure 10.

As the stereo pair data is being streamed, the system is able to store images from the cameras locally on the cluster. This allows a user to control the timeliness of the system. At the display end, time can be stopped, slowed down, sped up, or even reversed, much like the controls on a VCR. The ability to control time allows a flexible way to view the space and makes this type of data an attractive method for analyzing time dependent object motion.

Alternatively, cone-beam tomography can be used to generate a complete three-dimensional model of the imaging space. Each point in the voxel array corresponds to a

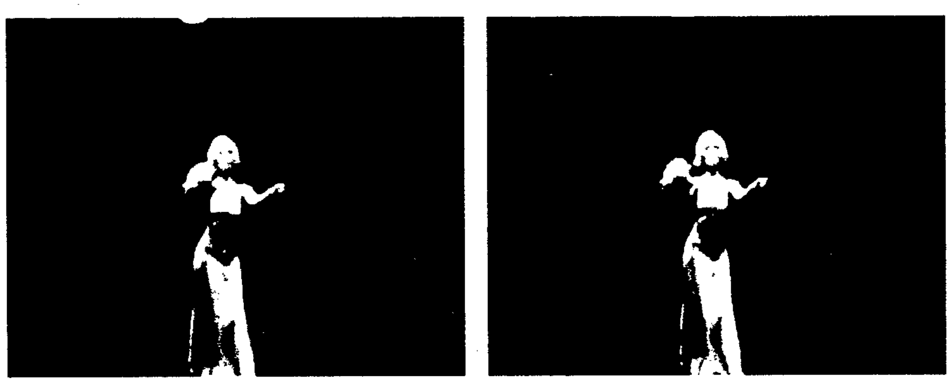


Figure 10. Stereo Pair images of a dancer in Argus. Notice how the perspective is slightly different between the left and right views.

matching point in space. The intensity of each voxel is an indication of the probability that the space it represents is filled. High intensity values are likely to indicate an object is located in the space while voxels of zero intensity represent free space. The voxel space can then be viewed using CRUMBS[4] or similar three-dimensional imaging tools.

For this project, we use a tomographic algorithm to integrate the data from the disparate cameras. We use a variation of tomography algorithms that were originally used for medical and industrial x-ray imaging to reconstruct models of objects in the visible regime. Tomographic algorithms treat a space as a collection of voxels, or volume pixels. Thus, each voxel represents the calculated intensity at a point in space, and these samples are arranged in a three-dimensional grid pattern. In this case, we use a 128x128x128 array of voxels to represent the space.

In an ideal pinhole imaging system, each point on the imaging plane corresponds to the intensity along a ray formed by the line segment between the point and the pinhole. All of the rays originate at the pinhole and the locus of the rays forms a cone, which is the bases for the cone-beam tomography. This is contrasted to fan-beam imaging in the two-dimensional case, and cylinder-beam imaging in which the rays are parallel. In our system, we use a fixed-focus lens instead of a pinhole. However, since the aperture of the lens is modest, all objects within Argus are within the depth of field of the camera. This means that the rays imaged through each lens can be treated as projections through the sensor space and tomographically backprojected. Tomographic systems normally have simply matrix solutions, although the number of elements can be in the millions, making the matrices difficult to invert quickly. Feldkamp's algorithm [5] provides a simplified method of computation that, although an inexact solution, provides a good quality approximation with high computational efficiency.

Feldkamp's algorithm is of the convolution-backprojection type. That is, the two major steps involved in solving for the volume are filtering the images, and then projecting each image through the reconstruction volume. From a computational standpoint, the filtering can be accomplished quickly by means of the Fast Fourier Transform. However, backprojection is still computationally expensive.

To approximate real-time performance, it is necessary to harness the processing power of all of the computers in the computational cluster simultaneously. We use the MPICH [3] implementation of the MPI parallel computer interface to coordinate all of the computers. For a frame, each node synchronizes with the network as a whole, grabs an image from the camera, filters it, and begins backprojecting. According to Feldkamp's algorithm, the final volume is the sum of the contributions of each camera. To achieve this, each node receives part of the volume from the previous node in the ring, contributes its information, and sends on the chunk to the next node. When the last node adds its contribution, the process is complete and the volume can be displayed.

In order for the tomographic algorithm to work, the internal model of the space and cameras must precisely match the physical situation. In particular, it is important for information about which rays intersect and where to be correct. In order to provide this

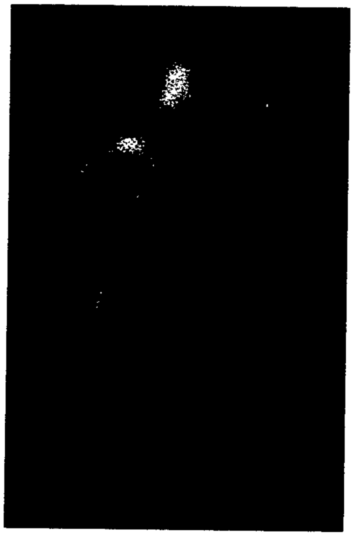


Figure 11. Reconstructed view from a 3D tomographic model of the

Figure 12. Raw camera view of a dancer in the Argus imaging system.

level of accuracy, the cameras must be precisely aligned. To overcome physical alignment limitations, software techniques can be used to correct for the alignment errors to a limited degree. For initial alignment, we used a laser pointed at each camera, and rotated the camera until the laser source and incident beam were centered on the image plane. For registration, two methods were used. First, a light was placed in the center of the imaging volume, and its location recorded. The image was skewed to correct for this variation. Second, a level line was placed in the center of the volume, and its angle in each camera's image was recorded, and corrected for in the ray geometry calculations. Figure 11 shows a ray projection of a reconstructed 3D volume in Argus. For reference, Figure 12 shows a single camera view of the same scene. One of the challenges of streaming volumes, as opposed to stereo, is that the display hardware must be much more sophisticated. Figure 11 was constructed using SGI hardware at the National Center for Supercomputing Applications. The level and cost of this hardware (a multiple processor Onyx system with specialty 3D projection hardware) is well beyond the typical capabilities of viewers. Of course, the 3D projection capacity of desktop displays is developing rapidly. In any case, it is difficult to visualize the utility of this data from projections like Figure 11. The real value is only understood when interactively rotating and exploring the reconstructed volume in a CAVE virtual reality space or a similar facility.

The effective bandwidth at which data can be transmitted out of the Argus is a major factor in the utility of the system. Although we are constrained by the network bandwidth between Argus and a remote site, some work can be done locally to help improve performance, both in terms of overall throughput and timeliness at the receiving end.

The throughput of the system can be drastically improved if a compression algorithm is applied to the data. A simple compression algorithm such as run length encoding (RLE) can reduce the size of a volumetric dataset by a factor of four. The stereo pair data was only reduced moderately with this type of compression. Since the computational resources affect the speed data frames can be generated, the amount of compression particular Argus camera contains quite a lot of redundant information when compared to neighboring cameras.

When viewed as a whole, the set of 64 images that Argus captures is a set with incremental differences between adjacent images. A normal video sequence is also a set of images with very small differences between images. The close similarity between video data and Argus data allows standard video compression algorithms to effectively compress data from Argus. The MPEG-2 video coding standard was chosen for Argus image compression because of its versatility and compression performance. The basis of MPEG-2 compression is inter-frame motion estimation that is applied to an image separated into discrete cosine transformed blocks. Individual images within a video sequence compressed via motion estimation can depend on prior images, or both prior and following images. Normally, a video sequence in time is compressed as the images are recorded, requiring only a small cache of images on which to base the motion estimation. The major challenge in adapting the MPEG-2 standard to work on the Argus array was parallelizing the MPEG-2 algorithm [7].

To satisfy data dependencies between the inter-compressed frames, an encoding sequence as shown in Figure 13 must be used. This chart only shows the compression sequence for a quarter of the Argus processor / camera pairs—the other three sections are compressed in the same manner. The frames are labeled *I*, *P*, and *B*. The *I* frames are not compressed with inter-frame compression. *P* frames are only backward dependant to previous *P* or *I* frames for compression. *B* frames are compressed based on data from both previous and later *I* or *P* frames. The blank spaces on the chart show when a processor is idle.

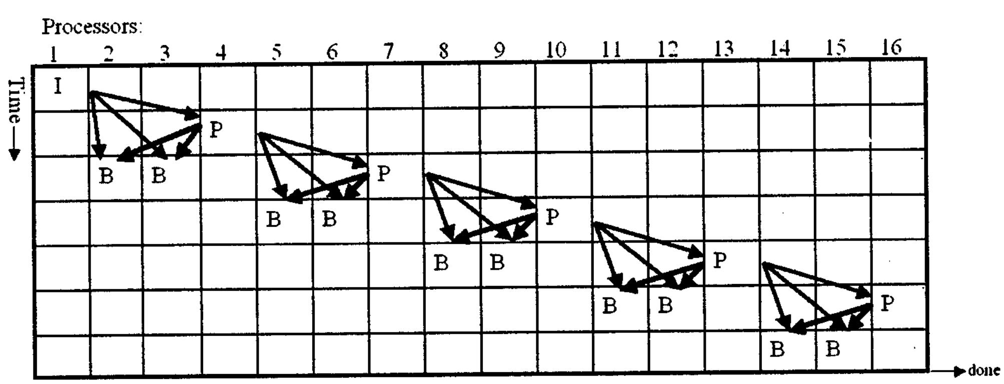


Figure 13. MPEG-2 encoding sequence. Each horizontal segment is approximately two seconds. Arrows denote data dependencies.

optimization of the backprojection. With concerted optimization, speeds of up to eight volumes per second seem feasible.

At 1.2 volumes per second, the data rate required for transmission of the volumes is 40 Mb/sec. Simple RLE compression reduces the size of the volumes by a factor of four, so ten megabits is enough to provide this level of performance. Internally, however, RLE is not used during the computation of the volumes, so the aggregate bandwidth required inside the cluster to perform the computation is 1.3Gb/sec. Our network can handle greater data rates than this. However, as the network load increases, the processors are forced to spend more time handling network traffic, and therefore, are able to do less computation.

In a reconstructed model, features become discernable in the volumes when they are 3-4 pixels in size on a camera. This does not compare favorably to other optical tomography experiments [6]. There are several reasons for this. The consistency in intensity response is poor between cameras, and therefore contributions from each camera are not weighted equally. In other experiments, a single camera was used and the object was rotated, so this was not an issue. This problem could be corrected by measuring the response of each camera to a known test object. The registration is not perfect. Using more sophisticated test objects and using anti-aliasing techniques to obtain sub-pixel registration could correct this. Also, the lenses suffer from manufacturing inconsistencies, and have curvature of field. Furthermore, the lenses form a non-orthoscopic projection that is not corrected for in our software. This causes some images to be quite blurry, and introduces distortions on all of the images. Using a grid test object to measure and subsequently correct for the various distortions in each lens could correct these distortions.

The advantage of transmitting a volume reconstruction is that remote users can interact with the 3D space in real-time on their own display facilities, rather than waiting for Argus to respond to requests for updated views. While a stereo view system is limited by the network performance, real-time volume display tends to be limited by the model generation rate. Each volume frame is a data cube containing two million bytes. Transmission at the generation rate requires ten megabits per second of bandwidth. Currently, our software only handles 128x128 images, however this is not a fundamental limitation of the algorithm.

With the grayscale CMOS cameras, each camera within the Argus array is capable of producing a 600 kilobits per second video stream, yielding a 37.4 megabits per second bandwidth for the entire array. The upgrade to color IEEE 1394 cameras pushes the array's bandwidth to 6.6 gigabits per second (Gbps). In practice, the full bandwidth is limited by the Ethernet switch backplane bandwidth of 2 Gbps. Transmitting gigabits of information per second between the sensor space and a remote user currently infeasible except on private networks. Storage also becomes a problem with data generation rates as large as these. While a standard video compression technique may be applied to each Argus camera individually, the Argus camera array has the additional advantage that any performed needs to be balanced with the impact the compression algorithm will have on the computational resources of the system.

The timeliness of the system can be enhanced by generating only as many frames as can be sent over the network. A performance hit occurs when too many frames are generated by the system. For example, if the system is able to generate frames 1.5 times faster than they can be transmitted, it is possible that the network will have to wait on a frame to complete if it is overwriting the latest frame. Although this overlap is typically not excessive, it can affect system performance.

We have implemented stereo pair display and cone-beam tomography algorithms on Argus. The relative advantages of the two systems depend on the viewer display network. For a network of users in the vicinity of the array, it often makes more sense to directly calculate views and to map individual views across the network to users. For a cluster of users at a remote site, it is probably more efficient to calculate, compress and transmit a global world model. This qualitative view was confirmed in demonstrations of real-time interactive 3D display to sites on the Illinois campus, to the Internet 2000 global summit at the Pacifico Convention Center in Yokohama, Japan and to the Duke University campus in North Carolina.

The rate that stereo pair data can be streamed is determined by the network bandwidth and the size of the frames. Each stereo pair frame is a combination of two 320x240 images, or 153,600 bytes. A network bandwidth of 4.6 Mb/sec is necessary to transfer the entire data stream without dropping frames. A 10 Mb/sec network is able to sustain the maximum data rates produced by this system. In typical situations, the stereo pair reconstruction is slowed to transmit 6-8 stereo view pairs/sec over the network. Each pair is a 153.6 Kb data set. A sustained effective network bandwidth of just over 1 Mb/sec is thus required for this application.

For relatively local demonstrations, real-time operation is attractive. Stereo views can be calculated at video rate without inversion, allowing real-time interactive stereo visualization of the Argus studio. At remote sites, in contrast, the latency of the network makes real-time interactivity difficult. In Yokohama, for example, the latency was considerable with variable delays of up to six seconds. This latency is primarily due to packet switching across the network rather than network transmission time and is determined by logical network distance rather than physical distance. In North Carolina, the experimental latency was about 2 seconds. Latency is typically well under a second on the Illinois campus.

Our cone beam inversion implementation is able to reconstruct only 1.2 volumes per second. This is quite favorable compared to other implementations of Feldkamp's algorithm. This is achieved by dividing up the computation between 64 processors, which makes much more computing power is available than is typical. Further performance gains could be achieved by making the application multi-threaded so that nodes would not be waiting for video-capture, improved network performance, and further

We have transferred technologies developed for Argus to more ad hoc sensor networking systems. Sensor/processor and communication modules form the core of these networks, such as the 4-inch high box shown in figure 13. Generally, most desktop/laptop processing systems use a hard disk for OS, application, and data storage. Hard disks consume considerable power, are subject to mechanical and environmental factors, and are far larger in capacity than needed by many of our applications. Instead, we use Compact Flash memory cards for storage. These cards, originally developed for digital camera image storage, are widely available in sizes up to several hundred megabytes and provide a low power, nonvolatile storage unit. As an additional feature of using Compact Flash cards, we can quickly change the functionality of a module by quickly interchanging preprogrammed cards.

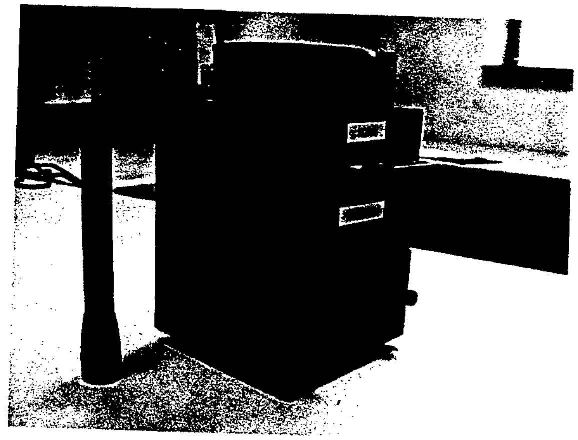


Figure 13. Packaged sensor module. 4 CMOS cameras are distributed across the top of the module.

Our ad hoc sensors systems use a video capture board to acquire images. Four video sources are connected to the acquisition board and an onboard multiplexer allows rapid selection between the input channels. Several of these sensor modules have been configured with four monochrome CMOS cameras, mounted in a half circle, and attached to the video inputs. This configuration provided the capability of creating 180-degree panoramic images. Sensor modules in general can also be outfitted with infrared cameras, specialized interferometric sensors, and other thermal, audio, biochemical sensors.

Each module was designed with a PCMCIA adapter that provided the means of integrating an 802.11b standard wireless network card. With these cards, a 10 Mbps encrypted data stream was available for transmitting control, data, and images between sensor modules and display devices.

Either an AC adapter or a set of NiMH batteries can be used to power the sensor modules. The batteries provide a means for rapid sensor deployment, although for only a short duration. Since the current set of modules was designed for processing flexibility as opposed to power efficiency, the current battery lifetime is approximately one to two hours. However, advances in microprocessor technology (e.g. the Crusoe chip from Transmeta) and modern software applications that throttle module operation until there is sufficient demand for its resources will improve battery lifetime significantly.

System functionality is distributed over several sensor modules and control and display platforms. The software architecture is also divided into several components operating across the network. The sensor module runs a reduced set of the Linux operating system. Linux was selected because it provides maximum flexibility in developing sensor drivers and code and is relatively simple to reconfigure. A web server operates on each module providing a means of serving data and images and provides a portal for receiving control information and queries through the cgi-bin interface. By using Internet web protocols, the module takes advantage of many of the security features that restrict access on the internet.

The primary image acquisition and processing code is a C application that provides connectivity via software sockets. Using standard TCP/IP networking protocols, the application carries out commands to acquire and transmit images, perform background subtractions, stitch together camera images to create a panoramic view, compress images, and perform other assorted image processing activities. Each new data connection forks a new process thread, thereby providing the module with the ability to serve multiple requests.

Data and network security are fundamental issues for restricting access to this intelligence gathering network. The wireless networking cards selected for this application provide 128-bit encryption to the digital data stream. In addition, module access is restricted in the application by posting a password request/challenge whenever a new module socket is initialized. While the configuration shown uses Argus CMOS video cameras, we have also constructed a module with a real-time coherence imager.

ACCOMPLISHMENTS

We demonstrated of two of the many possible topologies for streaming 3D video from a distributed sensing and processing array to distributed users. We found advantages for source side stereo projection included:

- Lower computational loads at both the data capture and display interfaces allow higher quality images to be streamed for stereo pair systems.
- Stereo projection requires fewer assumptions on scene interpretation and introduces fewer system artifacts.
- Stereo processing is much faster at the source end, allowing faster interactivity and real-time display for local users.

Advantages of tomographic reconstruction and volume transmission included:

- High latency inhibits interactivity with source-side stereo projection. This problem is resolved with display side projections.
- Aggregate bandwidth to multiple remote users is reduced while allowing independent view selection for each user with volume transmission.
- While tomographic backprojection does not automatically reduce the data volume of the joint set of images taken by the camera array, compression of the reconstructed volume is more computationally straightforward than joint compression of a complete set of images. Thus, transmission of the reconstructed volume tends to be more efficient than transmission of the complete raw data set.

While the two approaches do not represent a complete spectrum of possible schemes for streaming 3D video from a sensor space, they do represent benchmarks for what can be done. We are most pleased by how well processing is integrated into the physical structure of the sensor space on this system.

To our knowledge, Argus was the first demonstration of a distributed supercomputer with embedded sensor resources. Many additional possible streaming topologies can be implemented on this system. In practice, we expect neither of the benchmarks we have described here will represent an ideal solution. Various approaches to joint real-time compression across the array and embedded processing on the network to the maximize data flow and the visual experience of users might be attempted. Since our experience has shown that the ideal approach for one network topology differs from the ideal for other topologies, we expect that adaptive algorithms will be particularly attractive for this application. Algorithms might particularly focus on the user topology and on user display resources. Since no user can process the full sensor data set, the ultimate problem should be viewed as a mapping between distributed capture and display systems, as in Figure 5. In contrast with Figure 5, however, the ideal system will include joint processing nodes that integrate data from multiple capture heads.

Ultimately the problem of streaming 3D video comes down to trade-offs between the cost of sensing and the cost of computation. Our stereo pair transmission approach emphasizes sensing over computation. The tomographic approach emphasizes computation. The ideal system will adaptively exploit available sensor and processing resources.

PERSONNEL SUPPORTED

David J. Brady was the principal investigator on this program and worked on its development as specified in the proposal. Professor Brady's summer 2000 salary was partially supported by this grant. Graduate students Steve Feller and Evan Cull were also partially supported by this program over the 1999-2000 academic year and summer 2000. An undergraduate programming assistant, David Kammeyer, worked on the project as an hourly assistant. As an open testbed, many additional people have used and contributed to Argus's development without direct AFOSR support.

PUBLICATIONS

This report is based in part on "Information flow in streaming 3D video," by David J. Brady, Steve Feller, David Kammeyer, Evan Cull, Lilian Fernandes, Ronald Stack and Rachael Brady which is to appear in SPIE critical review of 3D video, CR 76. Two further reviews of the Argus testbed, one focusing on tomographic algorithms and one focusing on data compression and distribution, are in preparation.

The following publications are based in whole or in part on results from this program:

- Brady, D. J., S. Feller, D. Kammeyer, E. Cull, L. Fernandez, R. Stack and R. Brady (2001). Information flow in streaming 3D video. SPIE Proceedings. B. Javidi and F. Okano. Bellingham, SPIE Press. **CR-76:** 306-321.
- Brady, D. J., A. Rittgers, J. Gallachio, R. A. Stack and R. L. Morrison (2000). Sensing, communications and processing budgets for tomographic distributed ground sensor arrays. Proceedings of SPIE - The International Society for Optical Engineering, Orlando FL Bellingham WA, Society of Photo-Optical Instrumentation Engineers.
- Johnson, A. J., D. L. Marks, R. A. Stack, D. J. Brady and D. C. Munson, Jr. (1999). "Three-dimensional surface reconstruction of optical Lambertian objects using cone-beam tomography." IEEE International Conference on Image Processing **2**: 663-667.
- Marks, D. L., R. Stack, A. J. Johnson, D. J. Brady and D. C. Munson (2001). "Cone-beam tomography with a digital camera." Applied Optics **40**(11): 1795-1805.
- Morrison, R., D. J. Brady, A. Rittgers and R. Stack (2001). Wireless integrated sensing, processing and display networks for site security. Proceedings of SPIE - The International Society for Optical Engineering, Boston, MA.

NEW DISCOVERIES, INVENTIONS OR PATENT DISCLOSURES

None.

References

1. Reschke, C., T. Sterling, D. Ridge, D. Savarese, D. Becker, P. Merkey, *A Design Study of Alternative Network Topologies for the Beowulf Parallel Workstation.* Proceedings of the 5th IEEE Symposium on High Performance Distributed Computing (HPDC), 1996.
2. Ridge, D., D. Becker, P. Merkey, and T. Sterling, *Beowulf: Harnessing the power of parallelism in a pile-of-pcs.* Proceedings, IEEE Aerospace, 1997.
3. Gropp, W., E. Lusk, N. Doss, A. Skjellum, *A High Performance, Portable Implementation of the MPI Message Passing Interface Standard.* Parallel Computing, 1996. **22**(6): p. 789-828.
4. Brady, R., J. Pixton, G. Baxter, P. Moran, C.S. Potter, B. Carragher, A. Belmont, *Crumbs: a virtual environment tracking tools for biological imaging.* Proceedings of the IEEE Symposium on Frontiers in Biomedical Visualization, 1995: p. 18-25.
5. Feldkamp, L.D., L. and Kress, J., *Practical cone-beam algorithm.* Journal of the Optical Society of America A, 1984. **1**: p. 612-620.
6. Marks, D.L., et al., *Visible cone-beam tomography with a lensless interferometric camera.* Science, 1999. **284**(5423): p. 2164-2166.
7. Haskell, B., A. Puri, A. Netravali, *Digital Video: An Introduction to MPEG-2.* 1997: Chapman & Hall.