

Tomographic Processing on Wireless Ground Sensor Networks

Andrew M. Rittgers

Department of Electrical and Computer Engineering, Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801

Rick L. Morrison and Ronald A. Stack

Distant Focus Corporation, Champaign, IL 61820

David J. Brady

Department of Electrical and Computer Engineering, Fitzpatrick Center for Photonics and Communications Systems, Duke University, Durham, NC 27708

ABSTRACT

New opportunities for battlefield surveillance and modeling are unfolding with the advent of smart sensors linked via digital wireless networks. One exciting prospect is the use of tomographic techniques in order to create real-time three-dimensional modeling and analysis of the environment that is immediately accessible to battlefield forces. We have developed a small-scale ground sensor network for this application. We discuss initial deployment of this network as a tracking system.

Keywords: sensor networks, interferometric sensors, ground sensors, tomographic imaging, wireless networks

1. INTRODUCTION

A set of smart sensors called the Medusa Network was developed by the Photonic Systems group at the University of Illinois for the purpose of developing a platform for tomographic analysis on distributed wireless ground sensor networks. The catalyst for creating the Medusa Network came largely from work done on the Argus Distributed Sensing and Processing Environment at the University of Illinois [1]. Argus is a Beowulf-class parallel computer designed as a test-bed for three-dimensional (3D) imaging using distributed processing. The environment consists of a circular sensor space 14 feet in diameter surrounded by 64 video cameras. Pairs of these cameras are connected to each of the 32 dual-processor Linux machines that make up the Beowulf cluster. The cluster is capable of generating a 3D voxel array, 128 elements on a side, at a rate of two “frames” per second, however, modest improvements in software and system hardware should speed the reconstruction rate up to eight frames per second.

By “tomographic analysis” we mean the reconstruction of multidimensional scenes from projection data. Computed tomography (CT) is used most often in x-ray reconstruction of translucent 3D objects. However, as discussed in Reference 2, algorithms can be applied without modification to 3D reconstruction of opaque visible objects. Digital scene analysis and surface abstraction may be added to CT algorithms to analyze opaque objects. CT algorithms are a related subset of computer vision scene analysis tools. In computer vision one may choose to logically analyze scenes from single perspectives and then to logically fuse scene interpretations over temporally and spatially distributed frames or one may choose to physically integrate frame models to form a multidimensional target model prior to logical analysis. The target model might consist of 3D models of objects distributed across a plane. This paper considers abstraction of object position across the plane from sensor array data. While back projection for target positions is a very weak form of tomography, we refer to this physical space reconstruction as tomography to contrast it with logical frame analysis.

Tomographic analysis allows targets to be analyzed in their native 3D or 4D spatio-spectral spaces and removes many of the ambiguities of conventional two-dimensional (2D) analysis. As the angular range of the captured target data is increased, tomographic analysis becomes increasingly more effective. The angular range can be increased by tracking relative motion between the sensor and the target and by cooperative target analysis across a sensor network. We consider both approaches and focus in particular on distributed tomographic analysis, image analysis and target abstraction algorithms. To date we

have developed and tested the object detection and tracking capabilities of the network. In addition to discussing the motivation behind distributed tomographic sensor networks, this paper also describes the construction of our first functioning sensor network and the field tests performed on this network.

2. TOMOGRAPHIC ALGORITHMS

One of the more difficult tasks in building a networked array of ground sensors is that of gathering information from the distribution of sensors and then integrating that information for the purpose of tracking and target identification. Related efforts have studied network algorithms and data fusion [3,4]. One of the issues found in these studies centers on network granularity. Granularity refers to the sensing and processing capabilities of each node within the network. In a system with fine granularity, sensors communicate all the information they detect to a central processor. In a coarse grain system, target classification is implemented at the sensor level and sensors communicate classification results. Thus the sensor array could be used to combine raw sensor signals into a global model before attempting target analysis (fine granularity data fusion) or the array could be used to combine locally-produced target analyses into a global analysis (coarse granularity data fusion). The fine grain approach requires substantial data transfer between sensor nodes. The coarse grain approach requires substantial processing power and memory at the sensor nodes.

A central design issue when implementing target analysis on a sensor network is determining what level of granularity one should assign to the sensor and processor resources. Tomographic algorithms provide a natural basis for completing this task. To accurately choose the appropriate level of granularity, we must determine how to optimally distribute the computation among the sensors and a central processor. In practice, one could implement a hierarchy of algorithms that form tomographic models on disparate data types (source intensity and target probability densities are example data types). In such a hierarchy, low-level algorithms would form local models based on measured signal intensities. Higher level algorithms would form probability models based on local processors' target identification. Now the design question ultimately reduces to "How should processing and communication be balanced at each level of the processing hierarchy?" The answer to this question depends on several factors. For example, sensor density is critical. On very sparse sensor arrays, the information received by each sensor is likely to be independent of the other sensors. In this scenario, a coarse grain approach is suitable. As one improves array resolution by increasing the sensor density, however, common processing of array data is increasingly attractive. We claim that fine to moderate granularity approaches, which emphasize low-level communication, are more efficient on dense arrays because they can be integrated into array hardware, thus reducing the need for general-purpose central processing.

A second critical design issue of sensor arrays is network structure. Traditionally, a central processor gathers information from sensor nodes and combines the information in an optimal or efficient manner. While it is clear that a network with a star topology (all sensors connected to a central processor) will be able to extract a maximum amount of information from the collected data, this approach has a number of serious drawbacks. For example, the aggregate communication necessary between the sensors and the central processor is a bottleneck due to the large amount of necessary bandwidth and interference in a wireless scenario. In addition to these required communications resources, the central processor must have sufficient computational resources to assimilate all of the collected data. An alternative to such a centralized network is a distributed network, in which integrated sensors and processors communicate as peers. Although the central processor approach is easier to program and conceptualize, it is also less robust against processor failure and requires significantly more power and processing capacity in a single location. Under the distributed approach, local processing may be included in the sensor design and the network topology can be designed to enable tomography and classification through iterative belief propagation of simple, locally computed, information. Eliminating the need for global communication is an appealing potential feature of distributed sensor networks; e.g., in wireless ad-hoc networks, sensors could use power allocation to adjust their transmission power until links to a relatively small number of neighboring sensors are established. Such a scenario would mitigate both power consumption and multiple-access interference.

Considering the points made in the preceding paragraph, choosing the right topology for a particular application may be just as important as how one implements it. One common tomographic algorithm that we explore uses convolution and backprojection [5]. The convolution step weights the output of each sensor based on its orientation to the spatial points of interest. The backprojection step sums the values weighted to produce a source density at each point. This exact approach can be used to combine target probability densities. In addition to the standard convolution and backprojection method, we choose to explore silhouette reconstruction as a less computationally expensive method of tomographic reconstruction. Silhouette reconstruction is a binary method where a sensor contributes a yes or no response as to whether an object is

present at a specific location in the scene. This method can be used as a quick means to determine regions of interest before more costly reconstruction methods are used. Practical implementation of these tomographic algorithms within our sensor module array will use distributed processing and distributed control to achieve model reconstruction. Inter-module data communication will be dramatically reduced through efficient analysis and by limiting exchanges to hypothesis verification and/or resolution enhancement. The initial scenario employs up to four sensor modules, each one equipped with four cameras, to monitor a scene. By limiting the reconstruction volume to a coarse resolution, the system can achieve a throughput of a few models per second. Also, by embedding the control throughout the distributed network rather than within a centralized control station, wireless laptops and personal digital assistants (PDAs) can easily connect with the secure network and immediately request information and displays of the reconstructed environment. Continued enhancements will eventually lead to event triggers where the sensor modules will identify specific activity and request user intervention.

3. SENSOR MODULES

A network consisting of four prototype sensor modules was constructed at the University of Illinois and deployed on a trial basis for evaluating sensor array operation. For this first network, each module, or node, was constructed using off-the-shelf commercially available PC-104 components. Future networks will employ a custom module design and should be expected to conform to higher standards of compactness and power conservation. The PC-104 platform provided a computing standard that conformed to our size requirements and could be rapidly developed with little modifications to software and accessory hardware. A photo of the module without its case is shown in Figure 1.

The core of the module is a PC-104+ 266MHz Pentium processor board with 64M of onboard RAM. The board has power saving capabilities similar to those found in laptops, such as throttling the clock speed of the CPU during idle periods. A PC-104+ frame capture card with four multiplexed input channels was used to acquire images from four CMOS cameras. Stitching together the images from the four cameras, the sensor modules have the added feature of being able to generate 180-degree panoramic views. Interfaces were also designed to allow inputs from the infrared cameras used in the field test. A PCMCIA socket board integrated IEEE 802.11b wireless ethernet cards into the system. Finally, a 96MB CompactFlash module was used for data and system storage. Each module was packaged in a custom designed anodized aluminum housing, which includes the 180-degree array of the four CMOS cameras and the power supply. Using eight rechargeable NiMH “C” sized batteries, each module can operate for more than an hour. By optimizing power management functions, that duration could be increased. Each module can also operate on an AC power supply for long-term development and testing purposes. As mentioned above, the prototype device was not designed for maximum power savings, and future custom designs should improve these values up to ten-fold.

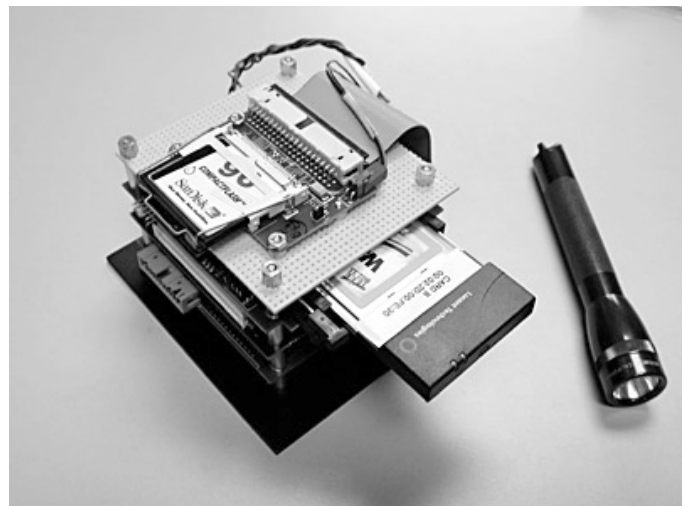


Figure 1. An exposed module is shown next to a 3-inch flashlight.

Each module was driven by a reduced version of the Linux operating system. Based upon common distributions publicly available at the time, we were able to reduce the operating system to a size less than 20MB, running only those functions essential to the operation and stability of the network. Several server-side software applications were custom developed by the group and installed on the modules. Included in this list is an image processing and module control application that provides connectivity through software sockets. Example processes include image acquisition and transmission, background subtraction, panoramic view creation, and image compression. The modules can serve multiple requests by forking each process to a new thread for each connection. On the client side, a graphical user interface (GUI) was written in the Java programming language. Basic Java applications benefit from being easily integrated with web browser applications and relative platform independence, making Java an important factor for integration within a heterogeneous sensor network. While current implementations of the interface operate on desktop or laptop computer environments, we seek to expand the interface to devices such as personal data assistants and other display devices.

4. EXPERIMENTAL SETUP

A primary goal of the project is to demonstrate, using prototype sensor modules, that tomographic data fusion is feasible using existing technology. We defined the scope of the first system field test as using tomographic analysis between two sensor modules to estimate the range and velocity of the specified target. At the time of the test, the network provided both real-time analysis and systematic collection of data for post processing. The latter was used to assist with software enhancements and verify the accuracy of results. Future work on the network will involve automating more of this data collection and processing. The test site had limited space available thus only ranges from 30 to 165 feet were used in the testing. To maximize the accuracy of tracking, we used the IR cameras connected to the sensor modules and a human subject in this test as illustrated in Figure 2. Due to a relatively large contrast between the subject and surrounding environment, the subject easily stands out and eases the process of determining the position of the object in the frame. Due to the limitations for determining precise distances and accurate sensor module orientation from physical measurements, we used a method of digital alignment, whereby we record a “reference” frame that captured the subject at a known and accurately measured location. From the data obtained from this reference frame, we were able to make range estimates of the object as it moved throughout the object space.



Figure 2. Sample IR image from test.

The first test involved two modules separated by about 16 feet. Data was collected for this distance as the subject moved in the sensor space on a preset course. The test was repeated with the separation between the cameras increased to about 82 feet. The movement of the subject was restricted to a single path perpendicular to the base line of the sensor modules, intersecting at a median distance between the sensor modules. This restriction insured that the test was repeatable, allowed us to make accurate measurements of the path, and limited the overall amount of error present in the setup of the test.

5. PROCESSING AND ANALYSIS

We were able to demonstrate object tracking on the 2D plane described by the location of the cameras and the location of the subject. This was accomplished by a triangulation method similar to a restricted tomographic analysis. This method involved the identification of the object, the location of its centroid, and calculations to estimate its position in the field. Calculations were first conducted on the reference images, and subsequently on the data images. Data from subsequent images was then used to calculate an average velocity of the object.

The method of calculating the object location from the camera images is illustrated in Figure 3 and 4. This is accomplished by defining the line equations of the rays that travel between the object and the sensor module. The parameters ϕ and ϕ' are calculated with the following equations,

$$\phi = \tan^{-1} \frac{x}{y}, \phi' = \tan^{-1} \frac{X - x}{y}$$

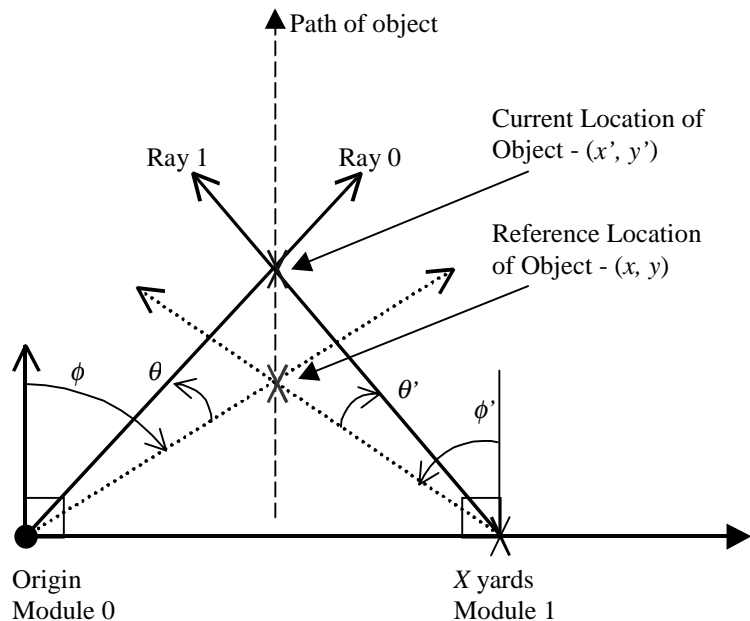


Figure 3. Experimental setup as seen from above

where (x, y) are the coordinates of the reference location, and X is the separation between the cameras. Each subsequent data image yields new values of θ and θ' , thus revealing the location of the object relative to the cameras and reference frame. The values for these parameters are calculated with the following equations,

$$\theta = \frac{Px_0}{Ntotx} \times FOV(^{\circ}),$$

$$\theta' = \frac{Px_1}{Ntotx} \times FOV(^{\circ})$$

where Px represents the difference between the centroids of the current object location and the reference object location, ($Px = C_{obj} - C_{ref}$), $Ntotx$ is the width of the image in pixels, and FOV is the field-of-view of the camera in degrees. The slope of ray 0 (m_0) yields the ratio of the coordinates we are interested in (x', y') , which is dependent upon θ and ϕ . A similar calculation can be made on ray 1 (m_1) using θ' and ϕ' . Both slopes are calculated with the following equations,

$$m_0 = \frac{y'}{x'} = \frac{1}{\tan(\phi + \theta)}, m_1 = \frac{y'}{x' - X} = \frac{1}{\tan(\phi' + \theta')}$$

where values for ϕ and θ were calculated above, and X is still the separation between the cameras. Since the cameras are known to be on the lines describing rays 0 and 1, we use their locations to determine the y-intercepts of these lines (b_0 and b_1) according to the following equations,

$$b_0 = y_{cam0} - m_0 x_{cam0} = 0 \quad b_1 = y_{cam1} - m_1 x_{cam1} = -m_1 X$$

where we define camera 0 to be at the origin, thus b_0 is zero, and the y values for both cameras are zero. We complete the range calculation by solving these simultaneous equations,

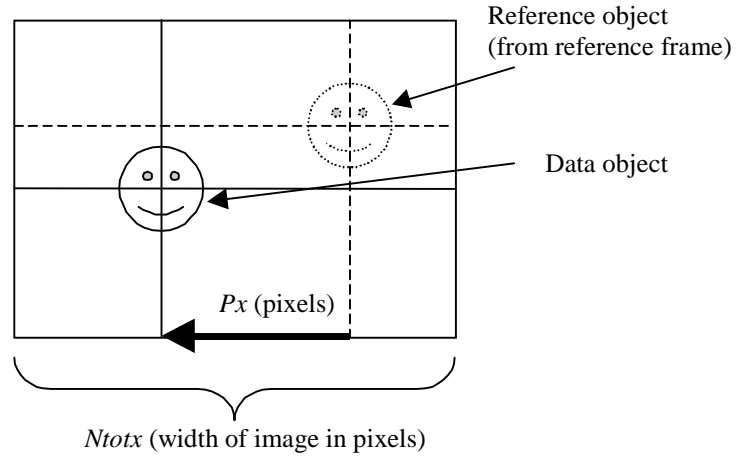


Figure 4. Measurements made from image frame

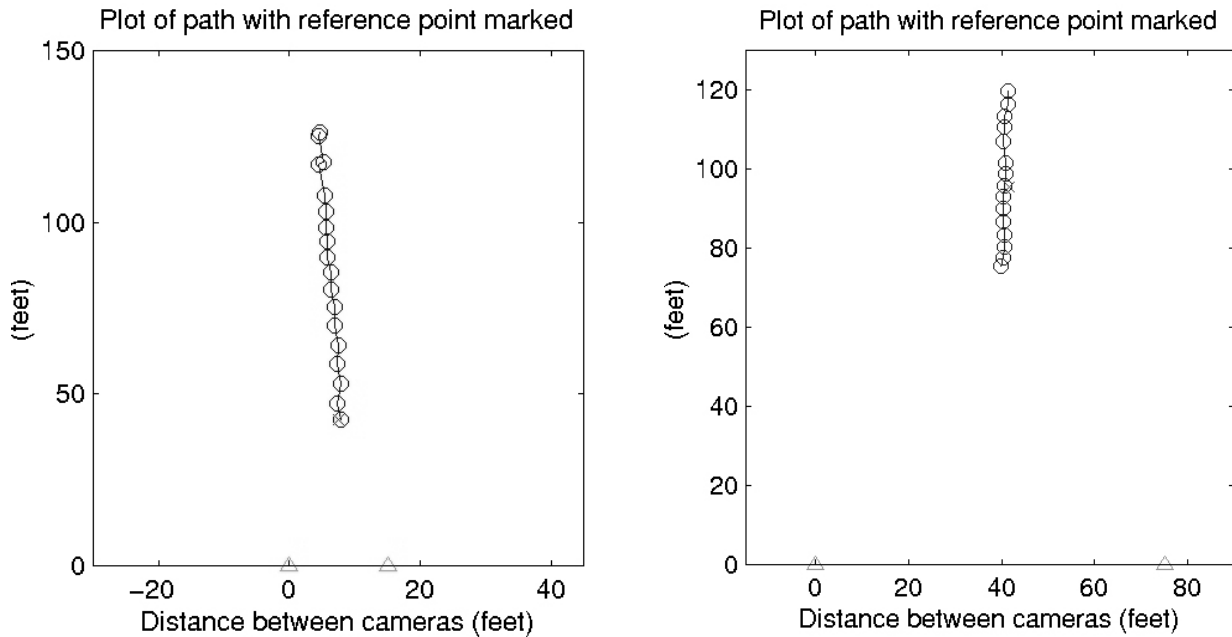


Figure 5. This figure shows plots of the output from the tracking algorithm on tests with camera separations of about 5m(15ft) and 25m(75ft). Missing circles are from frames that were skipped due to inadequate data.

$$\begin{aligned} y' &= m_0x' + b_0 \\ y' &= m_1x' + b_1 \end{aligned} \Rightarrow \begin{bmatrix} y' \\ x' \end{bmatrix} = \begin{bmatrix} 1 & -m_0 \\ 1 & -m_1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

to get (x', y') , the location of the object.

Results of the analysis, shown in Figure 5, are consistent with expectations described in the setup. The plots show the results of the two tests, one at a separation of 15ft, and the other at a separation of 75ft, as the subject walks almost directly away from the cameras at a distance halfway between them. Circles represent the path of the subject, triangles represent the cameras, and the small “x” represents the location where the reference image was captured. The circles appear at regular intervals, which is consistent with the subject walking at a regular pace. The average velocity of the subject was calculated to be 2.71 ft/s and 2.25 ft/s for each of the tests. These values are within the range of average walking speeds.

The third system test was performed using some software enhancements that provided real-time feedback on the tracking status. With a simple input of data collected from a reference measurement, we were able to track a bright object (for ease of object identification) with relatively accurate results. The test was conducted on a short-range basis to accommodate the indoor facility. Again, performing a two-dimensional object tracking, we set up the test with speed and simplicity in mind. The software acquired data by binning columns of pixels together. These bins were compared against one another in a winner-take-all fashion. The winning bin was the one that had the brightest overall value, and it was assumed that the object lay in this column. These coordinates were then entered into calculations similar to the triangulation method described above and range estimates were calculated. The results of this test are shown in Figure 6. It can be seen from this graph that the estimates conform to the ideal curve.

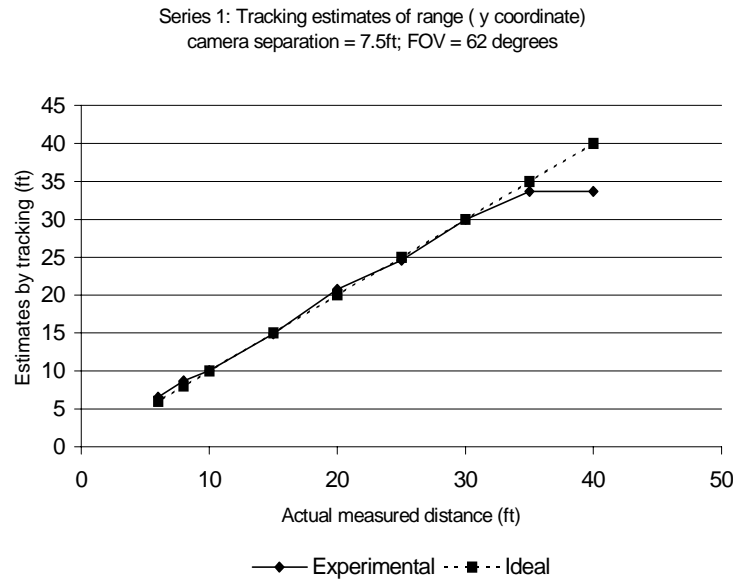


Figure 6. Plot of results from automated tracking algorithm.

6. CONCLUSION

The success of the Medusa Network largely depends on the ability to find the balance between a centralized and a granular approach to distributed tomographic processing. It is the goal of this research project to determine this balance point for object tracking using our ground sensor network. Tomographic analysis has many advantages over conventional imaging for target tracking. By reconstructing targets in their native 3D environments, tomographic analysis yields information not available to conventional 2D systems. With this additional information, ambiguities normally present in a conventional system can be resolved. As such, we will continue to develop more sophisticated tomographic algorithms to explore the

benefits of 3D and 4D analysis over conventional 2D tracking analysis. In addition to improvements in algorithm design, our sensor network will be enhanced as small electronic devices continue to see incremental improvements in computational speed, power consumption, and component cost. Future experiments involve tomographic methods of tracking multiple objects in a 3D volume. At first, this would be at very coarse resolution; however, as algorithms and communication protocols improve tracking resolution will become more detailed with greater accuracy.

7. ACKNOWLEDGEMENT

This work was supported through DARPA's Tactical Sensor Program via Army Research Office Contract DAAD19-00-C-0099. The technical point of contact for this DARPA program is Dr. Edward Carapezza.

8. REFERENCES

1. D. J. Brady, S. Feller, E. Cull, D. Kammeyer, L. Fernandez, R. Stack and R. Brady, "Information flow in streaming 3D video," SPIE Critical Review of Three-dimensional Video and Display, Photonics East 2000.
2. D. L. Marks, R. A. Stack, D. J. Brady, D. Munson, and R. B. Brady, "Visible Cone-beam Tomography with a Lensless Interferometric Camera," *Science*, **284**, pp. 2164-2166, 1999.
3. M. G. Corr and C. Okino. "A Study of Distributed Smart Sensor Networks," Dartmouth College, Thayer School of Engineering Technical Report Preprint, March 2000.
4. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks." In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, pp. 263-270. Seattle, Washington, USA, ACM. August 1999.
5. L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone-beam algorithm," *J. Opt. Soc. Am. A*, pp. 612-620, 1984.