Distributed Systems and Multi-Point Video Recording

Sha Ma <shama@mit.edu>
Supervisor: Glorianna Davenport
AUP Project Report, December 16, 2002

Abstract: This paper describes the application of distributed system techniques to an MIT Media Lab sponsored project on multi-point video recording. The two main improvements to the system are: (1) An algorithm to calculate optimum camera angle based on the locations of other cameras in the system and (2) A method of forwarding information between out-of-range cameras via intermediaries, so that all cameras in the system are aware of each other.

I. Introduction

The Participatory Networked Video Camera Project (PNVC) is an ongoing project in the MIT Media Lab's Interactive Cinema Group. The main goal of the PNVC project is to promote group effort in video production and aid each participant's decision making process through information exchange. Our project focuses on the distributed systems aspect of PNVC. Specifically, we present a design for a protocol that takes advantage of the capabilities of mobile ad hoc networks to send messages between the networked participatory camera units. These messages can relate information such as camera position, camera angle and focus, as well as pictorial information showing what other cameras are filming at the moment. Given these information, the camera unit can then calculate the desired coverage area as well as the best viewing angle, zoom, and location of that camera, and communicate this information to the user or an autonomous controller. When two camera units are beyond each other's broadcast range, intermediary units act as carriers for the message creating a multi-hop wireless ad hoc network. The system offers support for camera units to freely join and leave the network, and can be used for training video-photographers or directing autonomous cameras.

II. Background

A. Existing Equipment and Software Configuration

Two Sony Vaio laptops with built in video cameras and two Garmin Etrex Vista GPS devices are provided to us by the MIT Media Lab. Each GPS unit outputs serial information into the camera unit through the USB port using a serial to USB connector. Wireless communication is made possible by D-Link wireless network cards in the PCMCIA slots on the Sony cameras. Touch screens were installed on the Sony cameras to provide ease of use.

The Participatory Networked Video Camera project is implemented in the Java language, using Java Media Framework (JMF) to process video images and records in .avi format. The camera currently operates in two modes: a camera view that allows the camera operator to see what he is currently filming as well as pictorial information from other camera units (see Figure 1a), and a map view that allows the camera operator to see the location and direction of all camera units (see Figure 1b).

Figure 1a: Camera View with Pictorial Information

Figure 1b: Camera View with Positioning Information

B. Dynamic Decision Making

Dynamic Decisions are characterized by a group of decision makers choosing among various actions at different points in time in order to optimize the performance of a system to achieve a common goal. [1] This project is a practical case of dynamic decision making because of the following reasons:

1. Each camera operator must perform a series of actions over time to achieve the common goal of making the best video production as a group.
2. All actions are independent decisions made by each camera operator given the set of information provided by the networked camera system.
3. The environment of production can change both spontaneously and as a consequence of earlier actions. For example, the best coverage of a soccer game depends on the position of the soccer ball (which is unpredictable) and the position of the camera operator (whether he is in a good position to cover the ball if it comes into view).

We aim to break down the complex dynamic decision making problem into smaller sub-problems that can be handled by individual camera units. Each camera unit will process the information provided to it by the system, and make the best decision given the current state of the situation as a whole.

In this project, each camera unit must not only be able to dynamically calculate the best camera angle, zoom, and location, but also dynamically route the packets to permit multi-hopping as other cameras enter and leave the system.

C. Search Algorithms

There is little public research on the positioning of cameras for sporting events. The camera systems industry is highly competitive, and most companies keep their designs private. Only recently has this area been the source of serious research, in a project out of Carnegie Mellon University to capture 3D images of sporting events [4]. The results of their research, the "eye vision" video system, was unveiled at the 2001 Super Bowl. The eye vision camera system, which is composed of 30 stationary cameras with centralized control, sheds little light on this project however. The realm of distributed autonomous or aided video seems unexplored.

III. Improvements on the System

There are several notable drawbacks and sources for improvement in the camera system. The most obvious problem, upon using the system is how slow it is. The time delay in receiving images from all the cameras can last several seconds. The images, arriving with low quality and large time delays, are not very helpful to the user, and can often be distracting.

Rather than flooding the network with images, it should be possible to use the information available in a more efficient manner, communicating a minimum of information to other users. Ideally, we would like each camera to individually calculate its optimum direction and location dynamically, based on the positions of the other cameras. This way the camera can either operate autonomously, or provide recommendations to live users.

We can quantify the minimum possible information a camera needs to send to notify other users of what it's filming:

- *name*: it is necessary to identify each camera by name, both to ensure no camera is doubly represented in the system, and to allow multi-hop communication, as described in section V
- *2D location*: typically provided in (latitude, longitude) coordinates
- *direction*: the direction the camera is pointing, for example, 10 degrees off of true north
- *zoom angle*: the angular zoom of the camera, indicating its field of vision. This may be a fixed value, constant for all cameras in the

system, in which case it does not need to be sent.

In addition, it would be helpful to human users to send a snapshot of the current camera image every 30 seconds and possibly also after making a big move. This would allow live users to double check where everyone else is, yet does not impede the performance of autonomous cameras, since the image is sent much less often.

IV. Distributed Camera Direction Calculations

A. Concept

Our goal is to create an algorithm that allows each camera unit to decide which direction it should point, based on information about the other cameras in the system, and a predefined area to cover. We can assume that each camera knows the location, direction, and zoom angle of all other cameras in the system.

This algorithm should operate by maximizing the coverage provided to the area by all cameras. That is, the algorithm should recommend a direction to the camera operator, which gives the best total coverage of the area, considering the location of the other cameras. This would also allow, for example, an autonomous camera to respond with "backup coverage" when most cameras in the system are focused on a particular area (see Figure 3).

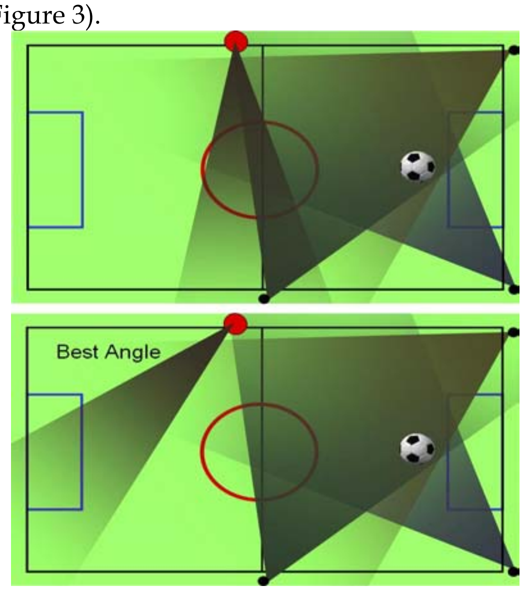

Figure 3: Autonomous camera responding to decisions of live users. The camera moves to provide the best coverage of the total field.

B. Design

The first important concept in designing this system is coverage and quantification. Two factors contribute to the quality of coverage a camera has on a particular spot: distance and zoom angle. That is, the closer a camera is to a spot and the tighter the zoom angle, the better the coverage the camera can provide.

We were unsuccessful at deriving an explicit formula for the coverage of a rectangular area. In theory, the coverage of an area can be expressed by a set of weighted integrals over polygonal shapes, each representing the coverage provided by one camera.
We define an area by a set of points. These may be arranged in a grid, representing a simple rectangular area (see Figure 4a), or they may be in a more complicated formation. This abstraction allows for some useful flexibility in the system. An area need not be any definable geometric shape (for example, the course of the Head of the Charles crew regatta in Boston). Points may also be distributed in an uneven fashion; a higher density of points in a physical area means the area will be deemed more important, and have better coverage (see Figure 4b).

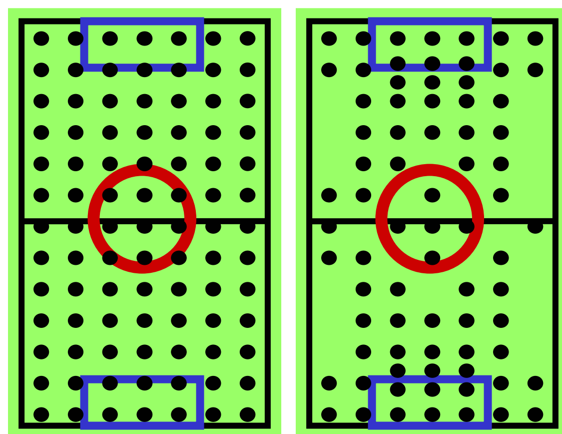

Figure 4: Two breakdowns of a soccer field into a set of points, used for calculating the coverage of the area. The first field (a) has a simple grid-like distribution. The second field (b) is weighted to provide better coverage in areas of interest.

Definition: the *Coverage* of a spot (x,y) by a camera i:

$c_i(x,y) = 1/(r^2\Theta)$ where r = distance to camera, and Θ = zoom angle

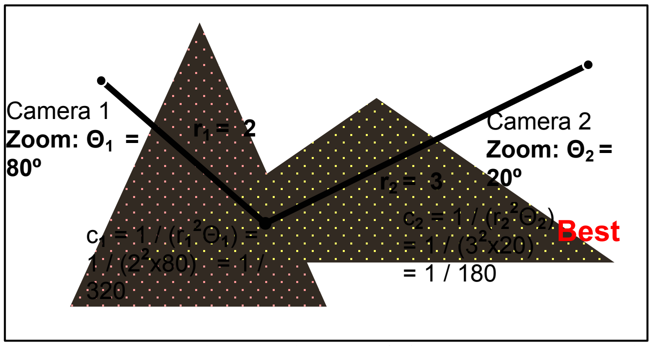

Figure 5: Example calculation of the coverage provided by two cameras. The camera which provides the best coverage of the point is camera 2, with a value of 1/180. This is the total coverage value for that point.

As shown later in section *D*, the choice of weighting methods has little effect. We use $1/(r^2\Theta)$ for intuitive reasons. The distance from the camera is naturally the most important factor, and therefore has the most weight. Coverage is proportional to $1/r^2$, similar to the dispersion of radiation or gravity. The zoom angle is a less important contributor to coverage quality than the distance from the camera; every photographer knows that when comparing two images, one filmed close up, and one filmed at a distance with a powerful zoom, the close-up shot is consistently superior. Thus coverage is proportional to $1/\Theta$, making the actual coverage $1/(r^2\Theta)$.

With multiple cameras, the *coverage* of the spot is $C(x,y) = \max\{c_i(x,y)\}$. This is necessary because in a situation with aggregate coverage, that is, where the coverage of multiple cameras add up, each camera operates independently. In a sense, this would eliminate the penalty of covering an already well covered point. An example calculation of coverage of a point is shown in Figure 5.

C. Implementation

To test the camera angle algorithm, we developed a java-based simulation program. Camera units are represented in the system with the minimum information characteristics defined in Section *A*. Coverage area is defined by a set of points in the (x,y) coordinate system.

The program takes an initial set of cameras and points, and allows adding to or modifying of the camera list at any point in the simulation. It assumes all cameras have a fixed zoom angle of 60 degrees and a common area to cover. Each time-step, each camera in the system calculates the optimum direction it should point based on the positions and directions of the other cameras in the system. This is intended to approximate receiving a message from each camera via wireless communication.

After calculating the best possible direction, each camera changes direction. The user manually advances the system through these time-steps, and a text output indicates the location and direction of each camera, along with the algorithm's recommended direction for each camera. See Figure 6 for an example of the output.

```
...
SkeletalCamera named a at lat: 4.0, long: 2.0, pointing 74.0
SkeletalCamera named b at lat: 2.0, long: 4.0, pointing 320.0
SkeletalCamera named c at lat: 4.0, long: 3.0, pointing 320.0
run 2 times since last camera added.
press 0 to exit or any other key to advance one timestep:
alternately, add a camera to the system with '# [name] [lat] [long]
        [angle]'
<enter to advance timestep>
simulator suggests the direction: 195.0
simulator suggests the direction: 150.0
simulator suggests the direction: 285.0
...
```

Figure 6: Text output from simulation program with 3 cameras.

D. Testing

To test this system, we set up 3 scenarios based on real world examples. Telemetrics Inc, the camera systems company that equipped the stadiums for the 2002 World Cup has a standard design for filming baseball games [3], which we used as a basis for a baseball diamond camera system with 4 cameras. Additionally, we set up two model spectator systems, a single sided system and a double-sided system. The single sided system has 3 cameras lined up against one sideline of a field, and the double-sided system has 2 cameras each on opposite sidelines, for 4 cameras total.

The first phase of testing is centered on the weighting method for coverage. We tested the following possible weighting schemes and compared them:

- $c_i(x,y) = 1/(r^2\Theta)$
- $c_i(x,y) = 1/(r\Theta)$
- $c_i(x,y) = 1/(r\Theta^2)$
- $c_i(x,y) = 1/(r^2 \sin(\Theta/2))$
- $c_i(x,y) = 1/(r \sin(\Theta/2))$

It is immediately clear from our testing that the specifics of the weighting method are largely irrelevant. The imprecision of the camera and human operators (5 degrees rotating accuracy) renders all the weighing schemes effectively equal. There was no significant difference in the recommended angles from the simulator between any of these methods. Thus, we choose our originally conjectured solution, $c_i(x,y) = 1/(r^2\Theta)$, as the definition of coverage.

As a sanity check, we also confirmed that it is necessary that $C(x,y) = \max\{c_i(x,y)\}$, rather than $C(x,y) = \Sigma\, c_i(x,y)$ for all i. A simulation using $C(x,y) = \Sigma\, c_i(x,y)$ for all i confirmed that each camera independently fixes its best angle, ignoring all others in the system.

In our initial testing, we discovered that the simultaneous adjusting of multiple cameras in the system can produce wild oscillations. In fact, 2 out of 3 scenarios failed to achieve steady state. The single sideline spectator scenario, in particular, produced a 70 degree oscillation in one camera, as it alternately took in each side of the field.

As a possible fix to this problem, we tried dampening the movement of the cameras by limiting the distance they could move at a time. This proved largely successful at eliminating the oscillations, however it results in a slower response time, since the cameras have to spend several time-slots to cover large distances. We experimented with several different limits on distance. Figure 7 summarizes our results.

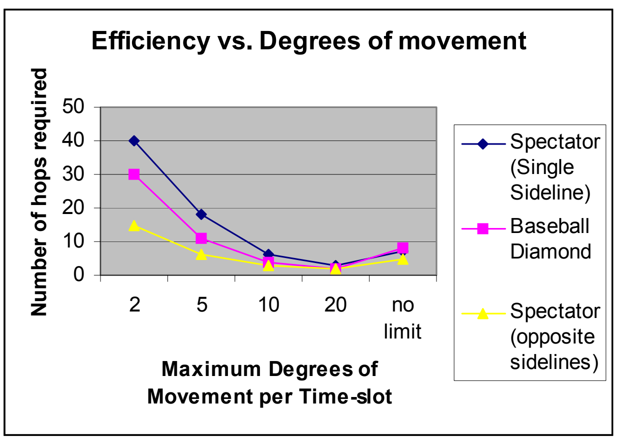

Figure 7: Efficiency vs. Degrees of movement. The number of time-slots required before all productive movement has ceased. This means either the camera's are all stable, or they are oscillating between two positions.

Addition of a camera to the system produces similar response times to those presented above. This is because all cameras reorient themselves immediately to accommodate the new arrival.

As mentioned earlier, with no limit on degrees of movement, two of the scenarios entered an oscillatory state. In both, the case of 20 and 10 degrees of movement, the Spectator scenario (SS) also entered an oscillatory state. Thus, although it slows down reaction time for the cameras, we recommend limiting the movement of the cameras to around 5 degrees at a time.

E. Analysis

This algorithm is linear in number of cameras and in number of points.

Due to limitations in GPS technology, at most 1 meter of location accuracy will be available to the cameras [5]. Thus, it is impractical to design the coverage area with points at intervals closer than around 1 foot. For a football field this could equate to at most 5000 points, and can be easily simulated on the Sony Vaio's.

V. Multi Hop Ad-Hoc Networking

Advances in wireless ad hoc technology have made it possible for mobile wireless users to communicate with each other without the need of a central communications infrastructure. Due to the limited transmission range of wireless network interfaces, multiple network "hops" may be needed for one node to exchange data with another across the network. Therefore, each mobile node, in an ad hoc network, should acts not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may not be within direct wireless transmission range of each other. [2] Each node participates in an ad hoc routing protocol that allows it to discover "multi-hop" paths through the network to another node.

A. Concept

Several routing protocols were researched, and they can be categorized into three main types: distance vector, on-demand, and geographic routing. Distance vector routing protocols such as Destination-Sequenced Distance Vector (DSDV) uses periodic announcement to advertise itself and its neighbors to other nodes, however this routing scheme not only uses up bandwidth, but also loses performance when there is high node mobility as announcements become quickly outdated. On-demand routing protocols such as Dynamic Source Routing (DSR) uses source routing rather than hop-by-hop routing to eliminate the need for periodic route advertisement and neighbor detection packets present in distance vector protocols. However, source routing increases the number of routing overhead bytes as each packet has to carry in its header the complete, ordered list of nodes through which the packet must pass. Therefore, we decided that a geographic routing protocol is best suited for our purpose since our camera systems are designed with an attached GPS device to provide us with location and direction information.

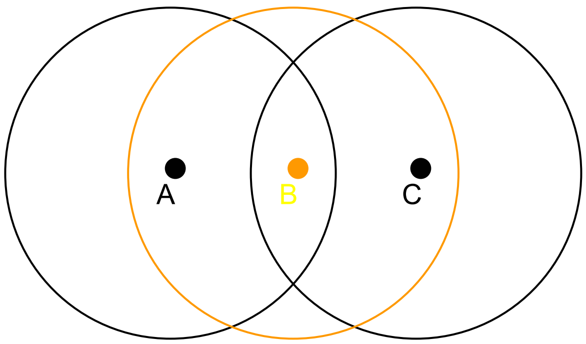

Figure 8: Network topology which allows hops. B rebroadcasts information about A and C.

B. Design

Two designs of geographic routing protocols are proposed in our paper based on two different scenarios.

In open fields with little obstruction, broadcast range can be modeled using a circle of a known radius. We tested our system of two available cameras units on the MIT softball field. Holding one camera stable, the other camera first walked away to a distance of up to 200 feet with reasonable reception, then walked around the stationary camera in a circle of a 200 feet radius, and confirmed our model to be fairly reasonable.

As seen in Figure 8, camera unit B receives information about its neighbors A and C, and can calculate the broadcast ranges of A and C. If B discovers that A and C are not within each other's range, B will relate its most recent information about A and C to each other. Otherwise, B will assume that A and C are within each other's broadcast range.

In an area where lots of obstruction can be found (e.g., cities with tall buildings), it is unreasonable to assume broadcast ranges to be regular circles. In this case, all camera units will broadcast a list of immediate neighbors. If a camera unit detects that two of its neighbors are not on each other's list, it will then act as an intermediary and relate the two out of range cameras' information to each other. For example, Figure 9 shows that intermediary camera unit B notices that A and C are not on each other's neighbors list, and broadcasts its most recent information about A and C to each other.

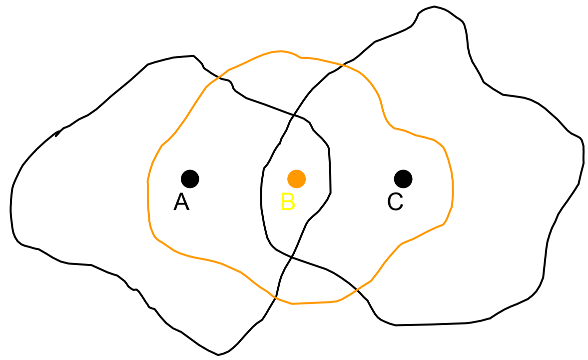

Figure 9: Network topology with uneven broadcast ranges. B cannot predict the range of A or B, and must rely on broadcasts of neighbor lists.

Conflicts in the above two scenarios are resolved according to recency. For example, if both B and D detects that A and C are not within range of each other, A and C will choose as their intermediary the node that provided them with the most recent information according to the time stamp. This is designed while keeping the consideration of node mobility and the freshness of announcements in mind.

C. Alternate Designs considered

Several alternate designs were considered but eliminated due to inefficiency and/or lack of forethought. For example, flooding all camera units with all information about all other camera units that a certain camera unit knows about is highly inefficient and would slow down the network flow of the entire system. Also, we eliminated routing protocols that are based on a source node having the knowledge of the location of the receiver node and choosing the best path accordingly because in our scheme, we assume that a node is unable to detect out-of-range nodes unless this information is provided to it by an intermediary node.

D. Evaluation

We have not yet implemented our designs of geographic based routing protocols due to limited resources. We currently have only two such camera units provided to us by the media lab, and it is infeasible for us to test out our multi-hop scheme without a third camera unit acting as an intermediary. Since the key issue we wish to explore is the frequency of variable radius broadcast ranges, this topology would be difficult to simulate without extensive further testing.

VI. Analysis

The main concern in any distributed systems application is scalability. It is no surprise then that the enhancements to the PNVC project described in this paper propose a few potential scalability problems.

Forwarding information about out-of-range cameras has the potential to flood the forwarding cameras, particularly if the cameras are arranged in a long line. Fortunately this system should never grow to the size where this becomes an issue. Since the system is designed for small groups of video-photographers, there is little chance the system will get out of hand.

Another potential issue is the time required to run the camera angle algorithm. As mentioned above, the algorithm is linear in number of cameras and data points, and the number of data points should never exceed the camera unit's ability to run the program in a short amount of time. The only potential issue is coverage of really large events (the City of Boston on Saint Patrick's Day, for example).

The issue of large coverage areas is fortunately easy to solve. Since most cameras have a limited range and quality, there is only so much area that can be potentially covered at once. There are no requirements in the system that stipulates that all cameras must share the same area. It is fully possible to break down large events into small overlapping areas, which are assigned to cameras in that local.

In addition to the above scalability issues, there is the question of ease of deployment. Since the system is entirely distributed, deployment of the system amounts simply to acquisition and configuration of the camera units. This allows for a highly portable system which can be set up anywhere at any time, with little preparation.

VII. Conclusion

Distributed systems techniques offer solutions to many of the problems facing the Participatory Networked Video Camera Project. The improvements are comprised of two parts.

1. An algorithm to calculate optimum camera angle based on the locations of other cameras in the system.
2. A method of forwarding information between out-of-range cameras via intermediaries, so that all cameras in the system are aware of each other.

These improvements are easily deployable and scalable. Combined with the existing software, they provide a way to use autonomous cameras, to help train video-photographers, or to assist professionals, by providing timely and relevant information.

Acknowledgements

We would like to thank the entire 6.829 staff for their enthusiasm in teaching us this semester. Especially, we would like to thank Professor Hari Balakrishnan for his patience in sitting with us and discussing as well as inspiring us with project ideas. We would also like to thank the MIT Media Lab for their generosity for letting us use their camera equipment.

References

[1] J.R. Busemeyer, "108 Dynamic Decision Making" to appear in International Encyclopedia of the Social and Behavioral Sciences: Methodology. Mathematics and Computer Science. November 15, 1999

[2] J Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols" to appear in Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, October 25-30, 1998, Dallas, Texas

[3] Telemetrics Inc. "Fresno Grizzlies Hit A Home Run With Telemetrics Camera Control Systems." Telemetrics Press Release, Mahwah, NJ, July 18, 2002. http://www.telemetricsinc.com/pages/Fresno%20Grizzlies.html

[4] Kanade, Takeo. "Carnegie Mellon Goes to the Super Bowl." January 2001. http://www.ri.cmu.edu/events/sb35/tksuperbowl.html

[5] United States Coast Guard Navigation Center. "Navstar GPS User Equipment Introduction, Public Release Version." September 1996. http://www.navcen.uscg.gov/pubs/gps/gpsuser/gpsuser.pdf.