Techniques for Addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support Systems

Scott E. Hudson and Ian Smith
Graphics, Visualization, and Usability Center, and College of Computing
Georgia Institute of Technology, Atlanta, Georgia, 30332-0280
{hudson, iansmith}@cc.gatech.edu

ABSTRACT

This paper describes a fundamental dual tradeoff that occurs in systems supporting awareness for distributed work groups, and presents several specific new techniques which illustrate good compromise points within this tradeoff space. This dual tradeoff is between privacy and awareness, and between awareness and disturbance. Simply stated, the more information about oneself that leaves your work area, the more potential for awareness of you exists for your colleagues. Unfortunately, this also represents the greatest potential for intrusion on your privacy. Similarly, the more information that is received about the activities of colleagues, the more potential awareness we have of them. However, at the same time, the more information we receive, the greater the chance that the information will become a disturbance to our normal work.

This dual tradeoff seems to be a fundamental one. However, by carefully examining awareness problems in the light of this tradeoff, it is possible to devise techniques which transmit just the right type and quantity of information, so that awareness can be achieved without invading the privacy of the sender, nor creating a disturbance (or consuming too many resources) for the receiver. This paper presents four such techniques, each based on careful selection of the information transmitted.

Keywords: Distributed Work Groups, Awareness Support, Privacy, Audio, Video, Visualization, Media Spaces.

BACKGROUND AND MOTIVATION

Recent technological advances have made the transmission of audio, video, and other media across digital networks quite economical. For example, one can now buy inexpensive systems for personal computers which can communicate with audio, video, and shared objects across even relatively slow networks. This has made it possible to envision the widespread use of this technology to support distributed work groups.

This work was supported in part by a grant from the Intel Corporation, and in part by the National Science Foundation under grants IRI-9500942 and CDA-9501637.

While rich communications media, such as live video, can allow distributed work groups to operate more smoothly, they are still typically not nearly as natural as working co-located [12].

There are several reasons for this (see for example [2,3,6,11,13,19] for more detailed discussions). One reason is that much of co-located interaction tends to be implicit, informal, and serendipitous. Many interactions occur apparently by chance, and certainly with little effort. For example important interactions may occur simply on the basis of people "bumping into each other" in the hall, or because interested participants overhear the conversation of colleagues and join in with additional details or knowledge.

To support informal serendipitous interactions, it is important to operate in a continuous fashion (typically between whole groups), rather than strictly on the basis of explicit connections between individuals. In general, one needs to support interactions in modes more like sharing a space than like making a call on a telephone. Although older systems using analog technology (see for example [1,2,6]) were often connection-oriented because of the limitations of the technology, the goals behind many *media space* systems have been to use audio, video, and other media to create these kinds of virtual spaces which afford the opportunity for serendipitous interactions.

In addition, to serendipity, co-located interactions also operate within the context of a high degree of awareness of one's colleagues. Awareness comes in may forms and degrees. At the simplest level, we are merely reminded of the existence of our colleagues on a regular basis. We also are aware of the location, activities, and actions of our co-workers. We for example, might know, or be able to easily find out, whether a person is in their office, currently busy, in the middle of a rush project, or simply in a bad mood. These forms of immediate awareness help serve as a catalyst for communications, and are used in various social protocols that drive our interactions.

(a) Working at a Workstation. b) Entering the Room and Sitting Down

Figure 1 The Privacy Preserving Shadow-View Technique Applied to a Home Media Space.

Over time we also come to know of our co-workers typical schedules, habits, skills and interests, and even their personalities. All these aspects of awareness contribute to our "knowing" our colleagues, and this awareness forms a crucial background for our interactions. Without such a shared background, interactions tend to be more distant, formal, and less fluid — specifically, more like interacting with strangers, and less like interacting with team mates.

The cues that drive our awareness come in various forms. Many pieces of awareness information are visual (e.g., seeing our co-workers presence, their expressions, their actions, etc.). However, in a shared space we also use auditory cues (e.g., overhearing conversations in the hall), and even spatial or environmental cues (e.g., noting that an office door is open or closed, or even the presence of a car in a parking lot).

Because of its importance as a backdrop and catalyst for communications, one important goal of most media space systems has been to support awareness in various forms by using various media — most notably video and audio transmissions.

THE DUAL TRADEOFF

Systems which attempt to support awareness in distributed work groups immediately face several important challenges. First among these is the widely recognized issue of privacy. In fact, we believe there is a fundamental tradeoff between providing awareness information and preserving privacy. In general, the more information transmitted about one's actions, the more potential for awareness exists among those receiving the information. At the same time, however, the more information transmitted, the more potential for violation of one's privacy exists. There is also a dual to this tradeoff: the more information one receives about others, the greater awareness of them is possible. However, at the same time, the more information one receives, the more likely it is to disrupt normal activities or consume too many resources.

Characterizing and understanding these tradeoffs is central to the work presented in this paper. These issues are not entirely new (they are discussed in a somewhat different form in, for example [9,10]). However, using these tradeoffs as a lens for viewing awareness system problems can point the way to new techniques which both meet awareness goals, and preserve important privacy and non-disruption properties. Four of these techniques are discussed here.

Privacy

Privacy has been widely recognized as an important issue for media spaces. In a shared physical space we have a well established set of social protocols for dealing with issues of privacy. For example, the distinction between a public and a private space is normally immediately clear, and most adults know how to adjust their behavior for each with little effort. However, in a virtual space, it is often the case that the normal cues of public versus private spaces are absent. For example, when one walks into a small private office containing a video camera, all the physical space social cues may indicate a private or semi-private space, despite the fact that the office might also be contained in a large public media space. Because of the confusion and uncertainty that this entails, people are often (at least initially) uncomfortable with the idea of working in front of a video camera. This is understandable, since it presents the same situation as working in front of a one-way mirror. One never knows when someone might be watching, or in general, who might be watching. This effect is amplified by the technology since, on typical networks, this information can normally be received (or intercepted) by any user of the network who has the proper software (see [16] for a cryptographic approach to overcoming this problem).

Particularly challenging privacy issues arise if we attempt to support awareness for work at home. Consequently, to test our approach, the first of our new techniques attempts to address this difficult domain.

The home is often thought of as a protected and private space and part of the advantage of working at home is being able to operate in that more relaxed, and informal setting. For example, the first author frequently works at home at odd hours, and has been known to get out of bed to write down a thought, or fix a bug. In addition, home work spaces are often shared by family members who are not part of the work group, and who have important expectations of privacy in their home. In both these cases, turning an otherwise private physical space into part of a very public virtual space (e.g., with a live video feed) is really not acceptable. On the other hand, working at home can easily cut one off from the rest of a (distributed or co-located) work group if no awareness support is provided.

This situation presents a primary example of our fundamental tradeoff. At first glance providing awareness comparable to a live video feed without changing the private nature of the home would seem to be very difficult, if not impossible. However, viewing the problem in the light of this tradeoff can lead to interesting new solutions. In particular, in order to overcome what seems like a fundamental limitation, it is necessary to carefully examine what information is, can be, or should be transmitted in terms both of its awareness support content, and in terms of its effect on privacy.

Figure 1 contains screen dumps of displays a user would see if they were using our first technique (described in detail below). This technique is very carefully crafted to provide just the right information so that some awareness can be provided, while retaining the basic privacy of the space. In particular, it shows information about the location and movement of people in various parts of a room (hence indirectly about activities) without actually transmitting any live images. As described below, the technique works by modifying a static image of the scene (previously captured when the room was empty), then darkening small squares within that image to indicate recent movements (as detected by frame-to-frame differencing from a live video image). In Figure 1a, we can see that the user is working on the machine which faces away from the camera, and in Figure 1b, we can see that the user has entered the room and sat down in the chair in the center of the work area.

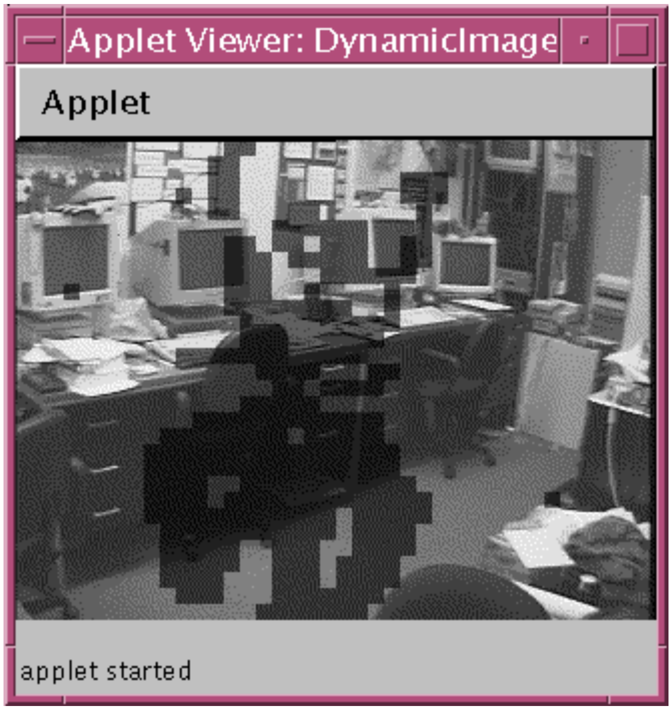
Another widely used approach to privacy problems is to enforce *reciprocity* [5,6], that is to ensure that whenever someone can see or hear you, you can also see or hear them. This is normally a property of physical spaces and can allow many conventional social protocols to apply.

However, reciprocity has several drawbacks that limit its use. First, reciprocity forces all spaces to be public in nature. This clearly would not work for our home media space example. In addition, even if the highly public nature of reciprocity is acceptable, enforcing reciprocity really works smoothly only in connection-oriented systems. In continuously operating systems, everyone is normally "connected" to everyone else sharing the same space, and so, although reciprocity may be technically enforced, it is much less useful. However, even in connection-oriented systems, reciprocity can produce additional undesirable effects because it can easily cause disruptions. For example, the equivalent of looking around a large room, or walking down a hall [15], might cause changes to the user interfaces appearing on a whole series of workstations. While it is possible to try to reduce the attention demanding effects of these changes (see for example [18]), even small interruptions can change the social effects of an action (for example, from the analog of quietly walking down a hall, to the analog of running down the hall talking loudly). As a result, while reciprocity is a worthy goal, and can be effective in some situations, it is not always appropriate and additional measures to address privacy issues are typically needed.

Disruption and High Resource Utilization

In the area of disruption, continuously operating systems provide a unique challenge. Here, because everyone sharing a space is always "connected" to everyone else, resource demands can be high, and the opportunities for unwanted interruptions of "normal work" can go up dramatically. For systems of this sort to work well, it is important to place at least partial control of overt interruptions in the hands of the receiver of information [9]. In addition, because we would like these systems to scale to moderate or large work groups, it is also important that, in general, they do not consume too many resources from the receiver. These recourses include both cognitive (e.g., attention) resources, and machine resources (e.g., screen space and CPU cycles).

These dual tradeoffs between sending awareness information and privacy, and between receiving



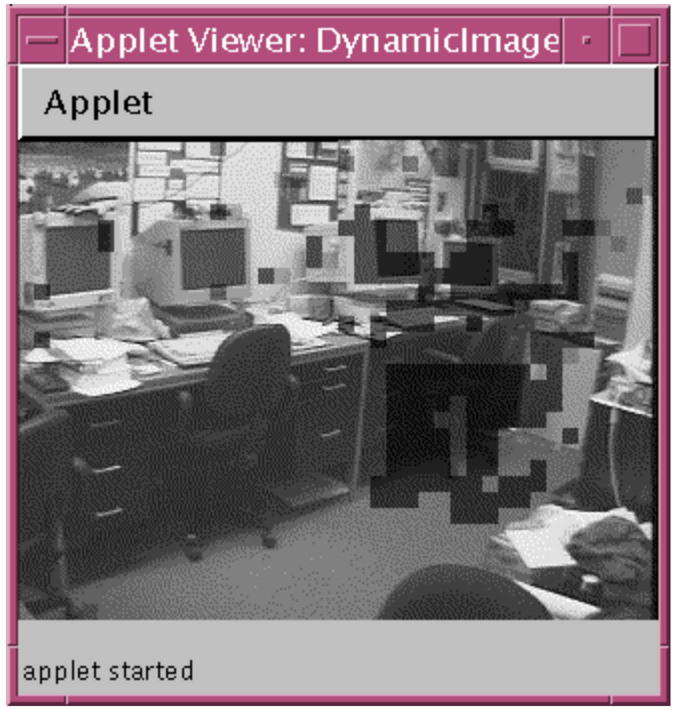


Figure 2 Views of Activity by Two Different Workers in an Office Setting .

awareness information and disruption or resource consumption, seem to be fundamental at some level. However, like any tradeoff, different points in the tradeoff scale can have different properties, and there may be techniques which make very good compromises with regard to these tradeoffs. Further, by explicitly examining problems with regard to these tradeoffs, it may be possible to devise new techniques which transmit just the right type and quantity of information so that they have more desirable overall properties than existing techniques.

In the remainder of this paper we consider four such techniques. Each of these techniques is designed to explore some part of the tradeoff spectrum and to produce a design solution that provides awareness information while still preserving privacy or reducing resource utilization for the receiver. The first of these techniques, the "shadow-view" technique, is designed to explore issues of privacy.

THE SHADOW-VIEW TECHNIQUE

As described above, the problem of providing awareness from the home — particularly something comparable to a live video feed — presents considerable challenges with respect to privacy. However, by considering the problem in the light of our tradeoff dimensions it has been possible to construct a technique which sends just the right information so that we can provide a significant amount of awareness information, but not make a large impact on privacy. We call this technique *shadow-views*.

Figures 1 and 2 show screen dumps of the display a user would see if they were using our shadow-views system. Here, a static *reference image* is used to provide a spatial context for interpreting a visualization of movement data. For example in Figure 1 we can see work at a particular workstation, and a person entering the room, while in Figure 2, we can see that work is being done at two particular chairs in an office environment. This reference image is a single still shot taken from the video camera when the work area was empty. This image is then broken into a grid of 8x8 pixel regions. These regions are dynamically made lighter and darker based on movement data derived from *live* video input from the same camera (in the same location, pointed in the same direction).

An area of the base image is made darker — to about one quarter of its original brightness — when that same region of the live image has “activity” in it (as measured by frame to frame differences above a small threshold). Thus, the static image is darkened in areas where people are *currently* active. Keep in mind, that the image presented to the user is still based on the original static image — no pixels from the current video feed are displayed. Over time, inactive regions are gradually lightened back to their original intensity. In particular, periodically — presently about every 20 seconds — a pass is made over the displayed image and all 8x8 regions currently not at their normal brightness are brightened by some amount (currently 25%). This allows activity in the scene to persist for some period of time leaving a "ghost image" of a person's movements within the space.

The system that supports the shadow-view display above consists of two parts: the ShadowServer and a client applet. The ShadowServer runs on the machine transmitting information, and computes difference areas which are sent to one or more applets which display the information for receiving users. In general, the ShadowServer sends *only* the coordinates of the regions with changes to its applet clients. The client applets are responsible for doing the display of the static image and darkening and lightening the regions accordingly. This property is important since it insures against surreptitious capture of the video (since it never leaves the local machine) and because it dramatically reduces bandwidth requirements.

The ShadowServer

The ShadowServer is written in Java, with some native methods (foreign functions) written in C for interfacing to the frame digitizing hardware. The interface to the digitizing hardware is a modification of the portable NV video system [7] to work with Java. Our current implementation of the ShadowServer samples (digitizes) a new image about once every 10 seconds. This delay is to minimize the load on the machine doing the digitizing and this is a specific effect of the (somewhat slow) digitization hardware in use at our site. However, because slow update of the final client image seems acceptable (or even preferable, in order to keep resource utilization down), more frequent digitization may not be necessary.

After capturing a video frame (in greyscale), the ShadowServer compares each grid square with the previous frame. We currently use a very simple algorithm for making the determination of whether or not there is activity in a given 8x8 grid square of the image. Each pixel in an 8x8 region of the current image is compared to the corresponding pixel in the previous image. If the difference in the values is greater than a threshold (currently about 8% of the dynamic range, or 20 out of 256 greyscale units) a counter is incremented. If at any point in the region the counter reaches a threshold value (currently 25% of samples), the region is considered *active* and client is informed of this region's activity. We have also experimented with allowing a "short circuit" of the above algorithm in which if *any* two pixels of the compared images differ by a large amount the region is considered to be active. This is useful if the threshold difference in the normal part of the algorithm is set to a large value to avoid noise in the digitization process causing false positives in the results.

Applets

Each user who wishes to receive awareness information, runs a visualization applet inside a World Wide Web browser which supports Java applets (such as Netscape 2.0). The display can be surrounded by a Web page which has explanatory text, links to contact information, other views, etc. The applet gets the static reference image to use as the base via the normal HTTP protocol. It then creates a network connection to the ShadowServer to request and receive change information. The applet only receives the coordinates of the region to be updated, so it is responsible for actually modifying the image. The process of periodically updating the image and lightening regions which are not at their normal intensity is accomplished via a background thread.

After some experience with the system, a couple of anecdotal results may be interesting. First, a user sitting at a workstation — even if engaged in solitary computer work — almost always moves enough to cause the algorithm to be triggered and the resulting display to have a dark patch in the area where the user is working. Given the context provided by the static image this is generally enough to determine, for example, if the user is working at the computer, or engaged in some other task. Second, sticking out one's arm (or similar gesture) in the region covered by the camera will generate a dark, vaguely arm-shaped region in the resulting image on the client workstation. This may indicate that our 8x8 pixel regions are too small (at least for some camera distances); larger regions would give out a less defined image in such a circumstance. Finally, we have observed that the ShadowServer can be fairly computationally expensive. In general, it will be forced to process the data corresponding to every pixel of a 320x240 image several times (at least once during capture and twice for comparisons). We are currently exploring difference calculations that look at a subset of the pixels (e.g. every other pixel or 1/4 total) in order to reduce this load.

A SHARED AUDIO TECHNIQUE

In addition, to the shadow-view technique, an audio technique with both privacy preserving and low-disturbance properties has also been developed on the basis of the dual tradeoff principles outlined above. (This technique is fully described in [17] and we will only give an outline of it here.)

For awareness purposes, it would be useful to maintain a shared audio space where co-workers could hear each other. However, such an "open-microphone" situation would clearly be unacceptable in most situations. While it is in reality rather difficult to do anything terribly embarrassing in front of a live video feed from an office (at least with low frame rates, and small images), we constantly say things that are intended only for a limited set of "ears". Further, constant conversation between members of a large group can be disturbing for those currently engaged in solitary work. None the less, eliminating all but

Figure 3. The Synthetic Group Photo Applied to an Artificially Inflated Group

explicit audio contact between distributed workers also eliminates opportunities for awareness and serendipitous interactions.

To provide some awareness information, while overcoming these difficulties, a new audio technique was developed which is designed to again transmit just the right type of information. This technique processes a speech signal into a non-speech audio signal that has several critical properties. First, all intelligible words are removed. This removes privacy concerns, and also significantly reduces the attention demanding properties of the sound. Second, the attention demanding properties of the signal are further reduced by techniques such as muffling, and volume reduction. Despite these transformations of the signal, enough information — in particular, both typical frequency distribution of the speaker and cadence information — are preserved to allow speaker identification. The result is a sound which allows one to determine who is speaking, but not what they are saying, and which is not demanding of attention and hence can fall into background noise.

Briefly, this technique works by taking a fixed sample of speech from the participant. Gaps of silence are removed from this signal and then it is repeatedly mixed with itself at random offsets. This creates a sound analogous to crowd noise, but from a crowd of one. This signal is further muffled, a small amount of white noise is added, and its volume reduced in order to reduce its attention demanding properties. Finally this signal is normalized to create a *characteristic signal* for the participant. This signal retains the typical overall frequency distribution of the participant, but contains no words. This characteristic signal essentially serves as an audio icon [8] for the person.

The overall technique works by providing a modified open microphone. The signal from the microphone is used to produce a coarse resolution envelope representing the volume of current sound. The receiver of the signal hears, not the actual live audio, but instead the characteristic signal of the sender, modulated by the volume envelope of the live signal. This provides live cadence information. When combined with the frequency distribution information from the characteristic signal, this is typically sufficient for speaker identification.

This technique, like the shadow-view technique, was designed specifically around an analysis of information with respect to the dual awareness tradeoff. In this case, the specific information isolated for transmission is speaker identity. By devising a technique which transmits only that information (while being carefully designed not to demand attention), it is again possible to provide awareness information while not violating privacy, nor causing undue disruption.

THE SYNTHETIC GROUP-PHOTO

For our third technique — the *synthetic group photo* — we consider aspects of disturbance and resource utilization. Live video or periodically updated still images [4] are very useful for providing awareness of co-worker's presence and more generally their comings and goings. Our own experience with a local media space system has provided anecdotal evidence of the benefit of simply being able to determine when someone is in their work area in order to coordinate more explicit communication such as a phone call. However, even half size (320x240) video images will quickly fill the screen if there is one for each member of even a relatively small work group of, say, 10 people (not to mention the CPU utilization typically necessary for maintaining many simultaneous images). Moderate sized work groups of 30 or more, clearly cannot make use of these techniques.

The synthetic group photo technique, focuses on information about the presence or absence of colleagues (both as individuals, and aggregated as a group) and is designed to overcome this problem by providing a very compact, but still visually rich, visualization of this specific information. Because it is compact and driven from very low bandwidth information, it is suitable for continuous "background" use. Further the display itself can be used as a simple framework for invoking tools for explicit communication, or more detailed awareness tools.

This simple technique leverages off of the fact that people have a high degree of skill in recognition of faces. We can recognize people we know at great distances, or in our case, on the basis of small images. Because of this, people in group photos are typically easily identifiable, even through the photos often involve pressing many people into a small space, using multiple rows with significant overlap, etc.

The technique described here creates a synthetically constructed group photograph by packing together static "head and shoulder" images from participants into fairly tight, and in fact overlapping, configurations analogous to a group photo. In addition to packing images together, this technique also uses a simulation of depth which displays smaller images for people "in the back" and larger images for those "in the front" (similar to what would be seen looking into the audience of a theater). This allows differential use of the scarce resource of space. For example, more resources can be devoted to close collaborators by "seating" them in the front rows. Infrequent collaborators can be "seated" in the middle rows, and other members of an organization can be "seated" in the back row to help provide a gestalt awareness of the overall group. Figure 3 shows the layout of such a group photo for an artificially constructed group (since our actual work group is not this large). This image shows over 100 participants in a relatively small space.

Once a group photo has been constructed, we can use an estimation of the presence or absence of each worker to drive the dynamic inclusion or elision of their image. Presence estimation information can come from a number of possible sources including the video change detection algorithms of the shadow-view technique, mouse and keyboard activity, or even instrumentation of the work environment with technology such as motion sensors or active badges [14].

Layout Algorithm

Although it would be possible to construct group photo layouts "by hand" using an image editing program (and in fact we did several versions of this in preparing to build our display), this is a rather tedious task and would be difficult to keep up to date with frequent personnel changes. This is particularly true since it involves not only constructing a new layout, but also measurement of where each image is placed. Further, it is desirable to allow custom layouts for each user so that one's friends and closest collaborators can be "seated" first. Consequently, an automatic layout algorithm for constructing synthetic group photos has been developed. Although the layouts produced by this algorithm are not quite as good as a manual layout, they are generally comparable in density to the layouts we produced by hand, and have the general appearance we were seeking.

Based on our experiments with manual group-photo layouts we were able to conclude that just the ability to see *most* of a person's face was sufficient for recognition. Specifically, overlapping of the shoulders and even parts of the heads of people in these simulated photos allowed the images to still be quite recognizable, while achieving fairly tight packing. In addition, we found that the theater metaphor of (approximate) rows working from large to small images works well, and allows packing of substantial number of participants into a relatively small space.

The first step of the algorithm is preprocessing the images to get them into a canonical form. In normal use each participant might provide an image of themselves already in this form. For our initial prototype we prepared several photos from those available on existing web pages within our center. These were canonicalized to a size (preserving aspect ratios) which made the bounding box of the person's head between roughly 90 to 110 pixels high and about 50 pixels across (exact sizing is not critical to the algorithm). After canonicalizing the images, the bounding box of the person's head was recorded. Next the average vertical position of their eyes was measured so eye lines of the images could be lined up in the layout algorithm. Finally, a background removal was performed leaving only the head and shoulder images, along with a mask for indicating foreground versus background pixels.

The actual layout of images is performed in a priority order that can be established by the user of the system. Pictures ranked as most important are positioned first (but drawn last) using images that are 100% of the original canonical size. Lower priority images will fall into later rows and be of smaller size (down to 20% of the original).

The first row of images is portrayed at 100% of their normal size and placed in a fixed pattern. Currently these images are placed so that there is a gap equal to 90% of the average head width between the bounding box of each head image. Once the first row has been placed, successive images are placed in available gaps. Images are placed in groups designed to approximate rows, with each group successively reduced by an additional 20% from the original size.

The overall algorithm does placement only in terms of the bounding box for the head portion of the image. Shoulder images are allowed to fall

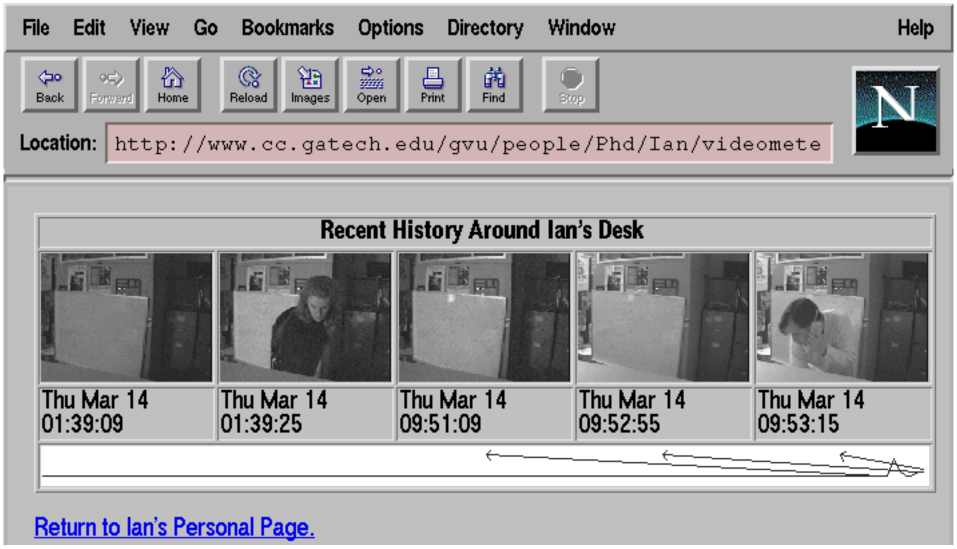


Figure 5. A View of Recent History Around the Author's Desk

wherever they may, and often overlap other shoulders (but only occasionally other heads).

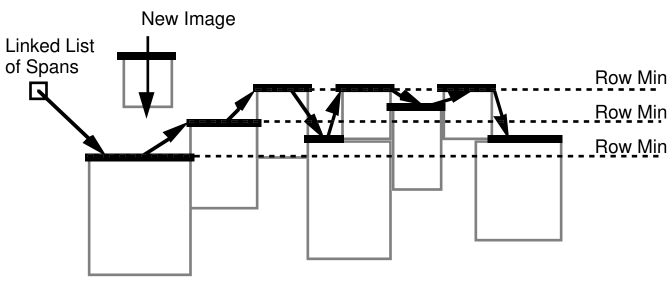


Figure 4. Layout Profile Data Structure

As illustrated in Figure 4, the algorithm maintains a data structure which represents the top profile of the head boxes of the existing layout. To place a new image, the spans of the data structure are searched looking for the deepest gap wide enough to hold the head box of the image and at least deep enough that the chin position of this new image is at or below the lowest top of a head from a previous row. To produce approximate row alignment, images are positioned vertically within this gap such that their eye lines line up with the minimum top of the head from the previous row (but never overlapping the current head box with head boxes directly below it). If no suitable gaps are found, the row is considered completed and the next row, at the next smallest size, is started. If rows reach a minimum size (currently 20%) and more images remain, remaining rows are all done at the minimum size.

The layout algorithm and display software is implemented in approximately 1000 lines of Java code. The resulting displays can hence be embedded in web pages to provide additional information and functionality.

Because the layout algorithm works only with rectangles, it can also be used to compose objects other than head and shoulder images. In particular, although it breaks the metaphor to some extent, it might be desirable to provide one or two quarter size live video displays for the highest priority participants. Space for these can be integrated into the display using the same algorithm.

WHEN DID KEITH LEAVE

Our final technique also addresses resource consumption and focuses on presence information, although in this case providing more specific detail about an individual. (This technique is not, however, privacy preserving and hence must be used with care, and may not be suitable for all circumstances.)

A live video feed or periodically updated still image is very useful for being aware of the comings and goings of colleagues. However, when this type of awareness information is available in conventional form, we only get a view of what is currently happening, and unless we keep the information displayed on the screen and pay attention to it, we do not develop awareness of larger patterns that indicate current levels of busyness, typical schedules, etc. Further, in a live feed, one will often see the empty chair of a co-worker you need to interact with. The fact that they are not currently in their work area is valuable information. However, this typically leads directly to additional questions such as "when did they leave?", "when they left, did they take their coat?", and "did they take their briefcase or backpack?". To be able to answer these questions and to provide a more general idea of recent patterns of activity without requiring the constant attention of the receiving user, we have developed a technique for visualizing a recent history of activities.

One form of recent history could be provided by recording a live video feed and playing it back with a VCR-like interface. However, the actual video images from a typical office environment are very boring, and searching through them is probably not a good use of one's time. Further the resources necessary to store useful amounts of video can be prohibitive (although we are currently considering time-lapse techniques which may be more useful).

The technique presented here (which we call locally "*When did Keith leave*?") attempts to take a more targeted approach. It collects selected still images which are designed to express the flow of activity in an area. An example view created by this technique is shown in Figure 5. Here we see the activity around the one of the authors' desks overnight.

The five selected frames show the work area at different points in time. We can see the following activity: At 1:39am the area is empty, then a few seconds later a co-worker stops by (to change the CD currently playing on the shared stereo). The area then remains empty (and although difficult to see here, it is somewhat dim because some of the lights have been turned off) until the author arrives at about 9:53 in the morning, turns on the overhead lights, and enters his work area. Notice that the long period of inactivity from 1:39am to 9:51am is not shown and does not consume either user or system resources.

The line graph at the bottom of the visualization shows total measured change (see below) over a shorter period of about two hours, and provides lines indicating where each captured image lies in this time line. By this we can see that the last three images are very recent, while the first two are much older — somewhere past the left end. Since the sequence of saved images can be spaced very non-uniformly in time, this provides a way to quickly see when the images were taken, and when there are long periods of inactivity (as happens in most of the left-hand portion of this particular visualization). Also, if there is a large amount of recent activity, and this dominates the display, the line graph can provide an indication of how long this activity has gone on.

The technique works by capturing a video frame periodically (currently approximately twice per minute, although this varies considerably depending on system load). Selection of stills is driven off of a frame-to-frame difference change indicator similar to the one used for the shadow-view technique. Here a global metric over the entire image is used. Each pixel is considered to have changed if it is more than 10% of the dynamic range (25 out of 256 greyscale values) different from the previous image. The total percentage of changed pixels is used as our metric.

A very simple (but effective) frame selection algorithm is used. The most recent frame is always displayed in the fifth position. When a new frame is captured, the algorithm decides either to shift the images to the left by one frame, retaining the previous frame, or to discard the previous image. The visualization is shifted if either the display is not yet full, or if more than about 20% of the new image (16000 out of 76800 pixels) has changed. Although a more sophisticated frame selection algorithm could be used, we have found this simple one very effective in practice. It tends to capture both the beginning and end of periods of high activity (or a single frame for activity occurring within only one frame).

This technique is implemented in Python, C, shell scripts, and HTML. The main driver (in Python) invokes frame capture routines in C (again based on the NV system [7]), frame differences are also calculated in C for performance reasons, then several shell scripts are used for manipulating images with standard tools. Finally a web page is provided to display the results. This web page uses the "client pull" facilities of Netscape to cause the page to be automatically updated every few minutes. (This update is more visually disturbing than we would like, so a new embedded implementation in Java is being considered.)

An earlier version of this system was used by the authors in early 1995 when we were separated by several thousand miles and connected only by the web due to restrictions imposed by a firewall. Although only employed in one direction because of the firewall, we still found the system effective in maintaining increased general awareness, and also in avoiding "phone tag". For example, in one instance, Hudson (the receiver of the information) was reminded to check his voice mail after returning to his desk and seeing an image of Smith having tried to call him. Similarly, rounds of "phone tag" were avoided because Hudson could easily determine whether Smith could currently be reached.

CONCLUSIONS

In this paper a fundamental dual tradeoff between awareness and privacy, and between awareness and disruption or resource utilization has been introduced and discussed. Simply stated, this tradeoff involves the information sent or received. The more information sent by a person the more their co-workers can be aware of them. However, the more information one sends, the greater effect this can have on one's privacy. Similarly, the more information one receives about others, the more aware one can be of them. However, this information then also has greater potential for disruption of "real work", either by direct interruption, or by consuming resources needed elsewhere.

We believe this tradeoff is fundamental and in some sense unavoidable. However, by viewing awareness problems through the lens of this tradeoff space — in particular, by carefully examining the nature of the information being transmitted and received with respect to this tradeoff — it is possible find good tradeoff points which provide awareness, while still providing good privacy or disruption properties. This paper has illustrated this with four specific techniques. Each technique was designed to address one or both of the dual tradeoffs in some way, each was designed using this notion of "information analysis", and each illustrates that more fidelity and more bandwidth does not necessarily produce a better result.

FUTURE WORK

One important remaining challenge for the work presented here is to integrate the techniques described into a unified system. We are currently working to do this in a web-based framework using Java applets. This framework will allow components such as the ones described here to be quickly "plugged together" and presented within a set of web pages.

REFERENCES

[1] Adler A., Henderson A., A Room of Our Own: Experiences From a Direct Office Share. In Proceedings Of ACM Conference On Computer Human Interaction, 1994, pp. 138-144.

[2] Bly S., Harrison S., Irwin S. Media Space. In Communications Of The ACM, 36(1), January 1993, pp. 28-47.

[3] Borning A., Travers M., Two Approaches To Casual Interaction Over Computer And Video Networks. In Proceedings Of ACM Conference On Computer Human Interaction, 1991, pp. 13-19.

[4] Dourish P, Bly S. Portholes: Supporting Awareness in a Distributed Work Group. In Proceedings Of ACM Conference On Computer Human Interaction, 1992, pp. 541-547.

[5] Fish R., Kraut R., Root R. Evaluating Video As A Technology For Informal Communication. In Proceedings Of ACM Conference On Computer Human Interaction, 1992, pp. 37-48.

[6] Fish R., Kraut R., Root R, Rice R., Video as an Architecture for Informal Communications, In Communications Of The ACM, 36(1), January 1993. pp. 48-61.

[7] Fredrick, R., Experiences with Real-Time Software Video Compression. In Proceedings of the Sixth International Workshop on Packet Video, Portland, OR, Sept. 26-27, 1994.

[8] Gaver W., Smith R., O'Shea T. Effective Sounds in Complex Systems: The ARKola Simulation. In Proceedings Of ACM Conference On Computer Human Interaction, 1991, pp. 85-90.

[9] Gaver W., Moran T., MacLean A., Lovstrand L., Dourish P., Carter K., Buxton W. Realizing A Video Environment: EuroPARC's RAVE System. In Proceedings Of ACM Conference On Computer Human Interaction, 1992, pp. 27-35.

[10] Gaver W. The Affordances of Media Spaces for Collaboration. In Proceedings of the ACM CSCW '92 Conference on Computer Supported Cooperative Work, 1992, pp. 17-24.

[11] Heath C., Luff P., Disembodied Conduct: Communication Through Video in a Multi-Media Office Environment. In Proceedings Of ACM Conference On Computer Human Interaction, 1991, pp. 99-103.

[12] Issacs E., Tang C. What Video Can And Can't Do For Collaboration: A Case Study. In Proceedings Of ACM Conference On Multimedia, 1993, pp. 199-206.

[13] Mantei M., Baecker R., Sellen A., Buxton S., Milligan T., Wellman B. Experiences In The Use Of A Media Space. In Proceedings Of ACM Conference On Computer Human Interaction, 1991, pp. 203-208.

[14] Pier K., Newman W., Redell D., Schmandt C., Theimer M., Want R. Locator Technology in Distributed Systems: The Active Badge. In Proceedings Of Conference on Organizational Systems, 1991, pp. 285-287.

[15] Root R. Design Of A Multi-media Vehicle For Social Browsing. In Proceedings Of ACM Conference On Computer Supported Cooperative Work, 1988, pp. 25-38.

[16] Smith I., Hudson S., Mynatt E., Selbie J. Applying Cryptographic Techniques To Problems In Media Space Security. In Proceedings Of ACM Conference On Organizational Computing Systems, August 1995.

[17] Smith, I., Hudson, S., Low Disturbance Audio for Awareness and Privacy in Media Space Applications. in Proceedings of ACM Multimedia '95, Nov. 1995.

[18] Tang, J., Rua M., Montage: Providing Teleproximity for Distributed Groups. In Proceedings Of Computer Human Interaction, 1994, pp. 37-43.

[19] Whittaker S., Frohlich D., Daly-Jones O. Informal Workspace Communication: What Is it Like And How Might We Support It. In Proceedings Of ACM Conference On Computer Human Interaction, 1994, pp. 131-137.