Focus of Attention in Video Conferencing

Jie Yang   Leejay Wu   Alex Waibel

June, 1996
CMU-CS-96-150

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This research was sponsored by the Advanced Research Projects Agency under the Department of the Navy, Naval Research Office under grant number N00014-93-1-0806. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of the Department of the Navy, or the U.S. government.

Keywords: Video tele-conferencing, Low bitrate, Focus of attention, Face tracking, Real-time tracking

Contents

1 Introduction 1

2 Focus of Attention for Teleconferencing 3

3 Tracking Human Faces in Real-Time 6

3.1 Skin-Color Model 6

3.2 Motion Estimation and Prediction 10

3.3 Model-Based Camera Control 11

4 A Prototype System 13

4.1 Vic and Its Modification 13

4.2 A Interface for Selecting a Face 15

5 Experimental Results 18

6 Conclusion 22

Acknowledgment 22

References 23

List of Figures

Figure 1 An example of a video transmission system 3
Figure 2 Focus attention for video transmission 4
Figure 3 An example of face and analyzed area 8
Figure 4 The color distribution of a human face in chromatic color space 8
Figure 5 Locating faces using a combination of color, geometry, and motion information 10
Figure 6 An example of motion estimation 11
Figure 7 Camera control scheme: model-based predictive control 12
Figure 8 An example of an image in different modes: (a) original; (b) pseudo-cropping; (c) slicing; (d) blurring 15
Figure 9 The control panel for face selection and manual camera control 16
Figure 1 Averages of Trials 1 and 2 (Equal Weights) 21

Abstract

In this report we present an approach to low bitrate video teleconferencing by focusing attention on important information. We show that by selectively degrading the quality of less important regions, more important regions can be sent without loss of quality but with greatly reduced bandwidth requirements. Low bitrate transmission for real-time video delivery over a dynamic network is achieved by region blurring and cropping. A prototype system has been developed to demonstrate the concept. We assume that a human facial area is the most interesting area in the system. The system can automatically focus its attention on a given face and its adjustable surrounding area. The selected area is then fed to the coding system and sent to the receiver. The selection function of the system is fulfilled by a real-time face tracker. The face being tracked can be selected by a mouse or finger pointing if a touch screen is used. It can track a person's face while the person moves freely (e.g., walking, jumping, sitting and rising) in a room. Based on the information provided by the face tracker and the network traffic, a window surrounding the face can be determined. The window size can reflect the network traffic. The image outside of the window will be either cropped or blurred. The preprocessed image is then fed to a tele-conferencing software package- *vic*, a real-time multimedia application for video conferencing over the Internet. The experimental results show significant savings of required bandwidth for video subjected to the changes.

1 Introduction

The demand for video communication between geographically distant sites has increased tremendously in the last decade. Applications depending on video delivery include video teleconferencing and telephony, interactive multimedia and multimedia e-mail. Although rapid progress has been made in technologies of digital communications, demand for data transmission bandwidth is still beyond the capabilities of available techniques. Very low bitrate tele-conferencing is another research field for addressing such a problem [1].

Video delivery depends not only on the bandwidth and data size, but also the network traffic. While sending the same amount of data via even the same network, different network traffic can result in different delivery times. For example, video transmission over a wireless network requires not only adaptation to changes in bandwidth and channel characteristics as the sender and receiver move around, but also to the amount of other traffic which can neither be controlled nor ignored. In a video conference, it might be more important to keep updating the most interesting information than sacrificing frame rate to send the entire frames. Partial or lower quality real-time images might be preferable to complete but delayed images. It is desirable for a video conference system to optimally keep sending the most important information at a reasonable speed based on network characteristics.

In this paper we present an approach to low bitrate video teleconferencing by focusing attention on important information. The key idea is to use an object-centered representation to extract the most important information in video image and then send only such information through the network. In this way a system can make better use of available network resources to achieve optimal performance without losing important information. The technique can be implemented with existing coding/network. This new approach differs from the model-based coding approach [2] in that no model is needed at the receiver.

In order to demonstrate the proposed approach, we have developed a face tracker-based tele-conferencing system. The system can automatically focus its attention on a given face and its adjustable surrounding area. The selected area is then fed to the coding system and sent to the receiver. The selection function of the system is fulfilled by a real-time face tracker [3] developed in Carnegie Mellon University. The face tracker has achieved a rate of

30+ frames/second using a workstation with a framegrabber and a video camera. It can track a person's face while the person moves freely (e.g., walking, jumping, sitting and rising) in a room. The face being tracked can be selected by a mouse or finger pointing if a touch screen is used. Based on the information provided by the face tracker and the network traffic, a window surrounding the face can be determined. The window size can reflect the network traffic. The image outside of the window will be either cropped or blurred. The preprocessed image is then fed to a tele-conferencing software package -- *vic*, a real-time multimedia application for video conferencing over the Internet developed by the Lawrence Berkeley National Laboratory in collaboration with the University of California at Berkeley [4]. Some modifications have been made to make it send the selected images.

The system has been successfully running in our lab across different Alpha and HP machines. We have performed several experiments to evaluate the system performance. To minimize other factors such as network traffic and machine loads, two separate 100-frame sequences were recorded through some functions customized to write YUV data and corresponding coordinates to files. Similar routines enabled binaries to substitute these files as input sources, ignoring the camera and face tracker, and thus run trials with comparable results. The experiments have shown some significant measurement results. The savings in frame sizes through slicing (see 4.1) in both of two trials was approximately a 50% reduction on total bits sent through the network when the selected area is about 25% of the entire image.

The remainder of this paper is organized as follows. Section 2 discusses the concept of focus attention and its application to video tele-conferencing. Section 3 addresses how to track human face in real-time. Section 4 and Section 5 introduces a prototype system, and shows experimental results. We close with a discussion of future work. Section 6 presents the conclusions.

2 Focus of Attention for Teleconferencing

Rapid progress in digital communications systems performance and mass-storage density has provided opportunities for new network-based multimedia applications. Current packet-switched networks such as the Internet, however, offer only a best-effort service, where the performance of each session can degrade significantly when the network is overloaded. Although ATM (Asynchronous Transfer Mode) networking has been chosen as the standard technology for future integrated services networks, it will co-exist with other networking technologies for a long time to come.

Coding is another way to minimize the communication capacity required for transmission of high quality/rate video signals, and to minimize the storage capacity required for saving such video data in fast storage media and in archival databases. Video coding (or compression) is the process of reducing the number of bits required to represent video data subject to a fidelity criterion on the final representation. Video coding can be based on either statistical redundancy in the video data, or subjective irrelevance of certain features in the video data.

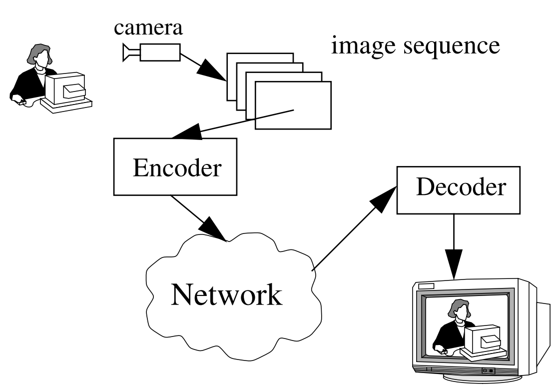

Figure 1 **An example of a video transmission system**

Figure 1 shows an example of a video transmission system. Video delivery depends on the codec and network bandwidth as well as network traffic. Both increasing bandwidth and data compression have certain physical limitations. They may exceed the capabilities of available techniques or be unreasonably costly. For example, it requires a compression ratio of ~240:1 to transmit or receive video data with 15 frames/second 144 x 176 pixels/frame

(8+2+2)=12 bits/pixel over telephone lines with modems at a rate of 19200 bits/second. Compression of raw color video data with 30 frames/second 288 x 352 pixels/frame (8+2+2)=12 bits/pixel to a rate of 1Mbits/second, requires a compression ratio of ~35:1. The problem becomes more complex when the network cannot provide network services with performance guarantees. Therefore, one of the most challenging problems in multimedia communications is how to keep video streams at a reasonable level of quality while the network cannot provide performance guarantee service. We present a solution to such a problem by adding a selection module on the top of a codec as shown in Figure 2. The technique is the use of feature-indicating interest images to focus attention on specific areas of the video imagery. The motivation comes from scientific studies on selective visual attention [5].

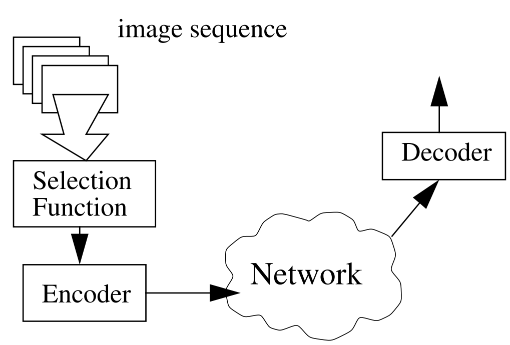

Figure 2 **Focus attention for video transmission**

Humans outperform computers in many pattern recognition and navigation tasks. One explanation is that computers have to process all available information, whereas a human quickly focuses his/her attention to the most important information without paying too much attention to less relevant information. The limited processing capacity of humans has imposed the need for selectivity in processing [6] [7]. The same strategy can be used by a machine when it is subject to the same limitations as a human. Focus of attention has been investigated in a variety of contexts. One of the largest branches of study has studied attention in static images [8] [9]. The concept of “spotlight” was used to explain how information is excluded from further analysis. Computational models of the spotlight mechanism have been implemented using neural networks [10] [11]. For a sequence of images, the process of focusing attention becomes more challenging because objects can move and change.

Many studies on focus of attention have directed to selecting relevant inputs to improve performance and robustness of a system. In this study, the concept of focus of attention is used to select important information for achieving optimal performance under the constraints of limited bandwidth and varying network traffic. In a video tele-conference, not all information is equally important at any particular time. Using an object-centered representation, it is possible for the system to focus its attention to the most important information. Then the system can send only such information through the network. People are typically important objects in a tele-conference.; therefore, we can partition the entire scene into several sub-scenes based on each individual's position and then select these sub-scenes to send to the receiver based on their priorities and network traffic. For example, if we are interested in a speaker, the speaker's face and surrounding area can be contains the most important information. The speaker's face and surrounding area have the highest priority to be sent to the receiver.

Two modes have been proposed to implement the concept: cropping and blurring. In the cropping mode, the system will crop the image outside of the selected window while the system will blurring the image outside of the selected window in the blurring mode. The mechanism of the blurring mode is similar to that of the human eye, which is not equipped with a uniform resolution over the whole visual field. Near the optical axis it has the fovea where the resolution (over a one degree range) is higher by an order of magnitude than that in the periphery. A human can view a large visual field by moving the fovea rapidly.

Since the system has to focus and maintain attention on moving and changing objects, a real-time face tracker is essential for the proposed approach.

3 Tracking Human Faces in Real-Time

A human face provides a variety of different communicative functions such as identification, perception of emotional expressions, and lip-reading. In order to track a human face, the system needs to be able to not only locate a face, but also find the same face throughout a sequence of images. This requires the system to have the ability to estimate the motion while locating the face. Furthermore, to track faces outside a certain range the system needs to control the camera, e.g., panning, tilting, and zooming. We have developed a real-time face tracker [4]. The camera's panning, tilting, and zooming are controlled by the computer via a serial port. Images are obtained by a framegrabber which digitizes the analog video signal into RGB values. The system can provide the following functions in real-time:

- Locating arbitrary human faces in various environments in real-time;
- Tracking the face in real-time by controlling camera position and zoom after selecting a face;
- Adapting model parameters based on individual appearance and lighting conditions in real-time;
- Providing face locations for user modeling applications in real-time.

Three types of models have been employed in developing the system. First, a stochastic model is used to characterize skin-colors of human faces. The information provided by the model is sufficient for tracking a human face in various poses and views. This model can adapt in real-time to different people and different lighting conditions. Second, a motion model is used to estimate image motion and therefore determine a search window. Third, a camera model is used to predict and to compensate for camera motion.

3.1 Skin-Color Model

To locate human faces, facial features, such as eyes, nose and mouth, are natural candidates. But these features may change from time to time. Occlusion and non-rigidity are basic problems with these features; note that many motion estimation algorithms work only for a rigid object., but a face cannot be regarded as a rigid object because the eyes and mouth are deformable. Color is another feature on human faces. Using skin color as a feature for tracking a face has several advantages. Processing color is much faster than processing other facial features. Under certain lighting conditions, color is orientation invariant. However, tracking human faces using color as a feature has several problems because the color representation of a face obtained by a camera is influenced by many factors such as ambient light, object movement, etc. Different cameras produce significantly different color values even for the same person under the same lighting condition. Human skin colors differ from person to person.

We have had two important observations about skin-colors which make it possible to develop a parametric skin-color model for characterizing human faces. First, we have found that distributions of skin-colors of different people are clustered in chromatic color space. Although skin colors of different people appear to vary over a wide range, they differ much less in color than in brightness, contradictory to popular belief. We have further discovered that skin-color distributions of different people under different lighting conditions can be approximated by Gaussian distributions in the chromatic color space. Based on these observations, we have developed an adaptive model for characterizing skin-colors.

Most video cameras use a RGB representation; other color representations can be easily converted into a RGB representation. In order to remove brightness from the skin-color representation while preserving an accurate but low dimensional color information, we can represent skin-color in the chromatic color space. Chromatic colors (r, g) [12], known as "pure" colors in the absence of brightness, are defined by a normalization process:

$$r = R / (R + G + B), \tag{EQ 1}$$

$$g = G / (R + G + B). \tag{EQ 2}$$

In fact, (EQ 1) and (EQ 2) define a $\mathbf{R}^3 \rightarrow \mathbf{R}^2$ mapping. Color blue is redundant after the normalization because r+g+b =1.

Figure 3 shows a face image and corresponding area for histogram analysis. The histogram of the skin-color is illustrated in Figure 4. The distribution of the skin-colors is clustered in a small area of the chromatic color space, i.e., only a few of all possible colors actually occur in a human face. The color distribution can be represented by a Gaussian distribution in the chromatic color space, i.e., $N(m,\Sigma^2)$, where $m = (\bar{r}, \bar{g})$ with

$$\bar{r} = \frac{1}{N}\sum_{i=1}^{N} r_i , \tag{EQ 3}$$

$$\bar{g} = \frac{1}{N}\sum_{i=1}^{N} g_i \ , \qquad \text{(EQ 4)}$$

and

$$\Sigma = \begin{bmatrix} \sigma_{rr} & \sigma_{rg} \\ \sigma_{gr} & \sigma_{gg} \end{bmatrix} \ . \qquad \text{(EQ 5)}$$

The procedure for creating the skin-color model is as follows:

1. Take a face image, or a set of face images if a general model is needed
2. Select the skin-colored region, e.g. Figure 3 (b), interactively
3. Estimate the mean and the covariance of the color distribution in chromatic color space based on (EQ 3) - (EQ 5)
4. Substitute the estimated parameters into the Gaussian distribution model
5. Since the model only has six parameters, it is easy to estimate and adapt them to different people and lighting conditions.

(a) (b)

Figure 3 **An example of face and analyzed area**

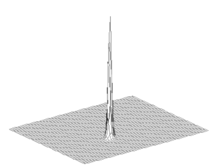

Figure 4 **The color distribution of a human face in chromatic color space**

Most color-based systems are sensitive to changes in viewing environment. Although human skin colors fall into a cluster in the chromatic color space, skin-color models of different persons differ from each other in mean and/or variance. Even under the same lighting conditions, background colors such as colored cloths may influence skin-color appearance.

Furthermore, if a person is moving, the apparent skin colors change as the person's position relative to camera or light changes. Therefore, the ability of handling lighting changes is the key to success for a color model. The adaptive approach provides a way to make a color model useful in a large range. Instead of emphasizing the recovery of the spectral properties of light sources and surfaces that combine to produce the reflected lights, the goal of adaptation is to transform the previously developed color model to the new environment. We have developed a method to adapt the skin-color model. Based on the identification of the skin-color histogram, the modified parameters of the model can be computed as follows:

$$\hat{\bar{r}}_k = \sum_{i=0}^{N-1} \alpha_{k-i} \bar{r}_{k-i}, \qquad \text{(EQ 6)}$$

where $\hat{\bar{r}}_k$ is the adapted mean value of r at sampling time k; $\alpha_i \le 1$, $i = k, k-1, ..., k-N-1$, are weighting factors; $\bar{r}_k$ is the estimated mean value of r at sampling time k; N is a computational window.

$$\hat{\bar{g}}_k = \sum_{i=0}^{N-1} \beta_{k-i} \bar{g}_{k-i}, \qquad \text{(EQ 7)}$$

where $\hat{\bar{g}}_k$ is the adapted mean value of g at sampling time k; $\beta_i \le 1$, $i = k, k-1, ..., k-N-1$; are weighting factors; $\bar{g}_k$ is the estimated mean value of g at sampling time k; N is a computational window.

$$S_k = \sum_{i=0}^{N} \gamma_{k-i} \Sigma_{k-i}, \qquad \text{(EQ 8)}$$

where S_k is the adapted covariance matrix of color distribution at sampling time k; $\gamma_i \le 1$, $i = k, k-1, ..., k-N-1$, are weighting factors; Σ_i, $i = k, k-1, ..., k-N-1$, are the estimated covariance matrix of color distribution at sampling time k; N is a computational window. The weighting factors α, β, γ in (EQ 6) - (EQ 8) determine how much the past parameters will influence current parameters.

A straightforward way to locate a face is to match the model with the input image to find the face color clusters. Each pixel of the original image is converted into the chromatic color space and then compared with the distribution of the skin color model. Since the skin colors occur in a small area of the chromatic color space, the matching process is very fast. However, the background may contain skin colors, too. A variety of distributions of energy quanta of photons can be perceived as the same color. This means that many points in the color space representing the different physical distributions of photon energy quanta can be mapped onto a single point in the color space. In other words, the mapping between the physical spectrum and the color space can be many-to-one. It is impossible to locate faces simply from the result of color matching.

When faced with a many-to-one mapping problem, it is natural to use other mappings to get rid of uncertainties. This requires additional information. Three types of information are available from a sequence of images: color distribution, geometric, and motion information. An example of locating faces using color, size, and motion is shown in Figure 5. The sequence of images was taken from a laboratory with a complicated background. By combining color, geometry, and motion information, three faces are accurately located.

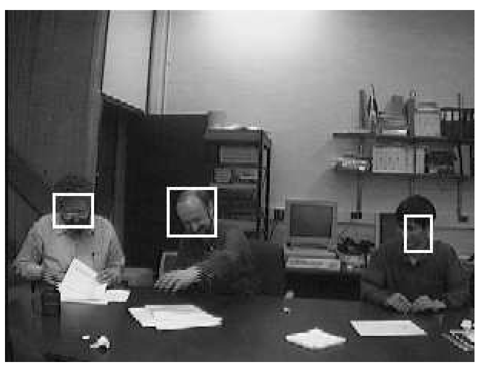

Figure 5 **Locating faces using a combination of color, geometry, and motion information**

3.2 Motion Estimation and Prediction

Under the assumption that the image intensity doesn't change between adjacent frames, color is an orientation invariant feature. By using the skin color as a feature, a translation model is needed to characterize image motion. In this case, only one corresponding point, in theory, is needed to determine the model parameters. In practice, two or more points can be used for robust estimation. We can obtain these corresponding points by the face correspondence between adjacent image frames as shown Figure 6.

Since tracking can be formulated as a local search problem, the system can search for the feature locally within a search window instead of the entire image. The window size and position are two important factors in real-time tracking. An unduly large search window results in unnecessary searching while a too small search window may easily lose the face. Several factors may influence search window size. For example, the search window size grows with the square of the maximum velocity of the face. An effective way to increase tracking speed is to use an adaptive search window. With any given zoom, the face size can be a criterion to determine search window size. If a person is close to the camera, a small motion may result in a large change in the image, whereas if the person is far away from the camera, the same motion will have less influence on the image.

Figure 6 **An example of motion estimation**

Motion prediction is effective in increasing tracking speed. The tracker only has to search small regions to find the features as long as the predictions are reliable. Some motion modeling techniques such as Kalman filters can help predict future position. These methods, however, are computationally expensive. A simple way of predicting the motion is based on the current position and velocity. If the sampling rate is high enough, the location of a point in the current image and the displacement prediction based on the current image speed produce a very good approximation for the location in the next image.

3.3 Model-Based Camera Control

In order to achieve high quality tracking performance, the face tracker uses a Canon VC-C1 camera with pan, tilt, and zoom control. There are two major problems with this camera: (1) the camera cannot pan and tilt simultaneously; (2) response of the camera is much slower compared to the real-time sampling rate. We have developed several methods to solve these problems. Instead of directly controlling the camera, the camera is controlled through a socket-based server. With the server, client code does not have to deal with the complex RS-232 port and client code can ignore the fact that the VC-C1 does not have simultaneous pan, tilt or zoom. If we use a conventional feedback control scheme, we can hardly achieve good performance because of the time-delay. To overcome time-delay, we have developed a model-based predictive feedback scheme as shown in Figure 7.

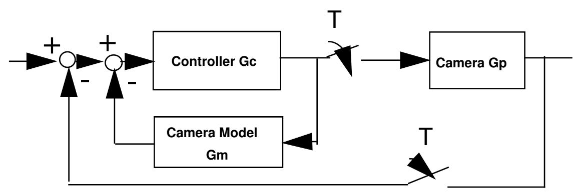

Figure 7 **Camera control scheme: model-based predictive control**

A camera model is used to predict the camera motion and compensate for egomotion. With the pinhole camera model, the face tracker can effectively control the camera and compute the egomotion. The errors caused by the model can be reduced by feedback control.

4 A Prototype System

4.1 Vic and Its Modification

In order to demonstrate the proposed approach, a prototype system has been developed in our lab. The chosen video-conferencing application was Vic-2.6, written and maintained by Steve McCanne (mccanne@ee.lbl.gov) and Van Jacobson (van@ee.lbl.gov) as one of a suite of multimedia conferencing utilities. Vic provides the video interface to communications by accepting video data from a number of possible frame grabbers supported on various platforms, and transmitting this data to recipients who in turn may be sending back video. In this particular setup, a DEC Alpha equipped with a J300 card and a VC-C1 Canon camera was directly supported with several session encoding methods -- nv, CellB, scr, JPEG, and H261, each of which could be used to correspond with another program that supported the appropriate format. From a programming standpoint, a benefit of using vic is that the video data for the nv, CellB, and scr sessions all store the video data in YUV format in the same type of buffer; hence, modifications to the data at that stage affect these three types.

The source code for the jv300 grabber routines was modified to use sockets-based communication with the server to check for new coordinates of a region to target. The rectangle thus selected could be subjected to a number of modifications, depending on the options selected by the user via a Tcl/Tk script. It was determined that the YUV image data consisted of sets of four bytes, each set corresponding to two horizontally adjacent pixels whose values were mathematically interwined. Hence, the rectangle selected was often adjusted so that its borders did not split paired columns; in addition, all modifications were fairly gross and simple, for the sake of speed.

The target is chosen by the selection function based on the face tracker. Each modification takes place right before the target frame is encoded. A function is used to obtain coordinate information, read the configuration file, and call the appropriate image editing routines before passing the data to the encoder. In addition to those major modifications, a fourth option could be combined with any or none of those to add a green box around the interesting region, mostly as a debugging tool.

Three modes have been implemented in the prototype system as shown in Figure 8 and they can be switched at any time

Pseudo-cropping: This option causes the entire area outside the selected rectangle to be turned black, saving a significant amount of transfer time, as fewer pixels change between frames and thus the delta-based encoding of each session type can send fewer pixels. At the cost of aesthetics and a large portion of the image, this provides fairly significant savings with little cost in processing.

Slicing: This option causes the 25% of the image including and around the facial region to be sent as a smaller frame image, with the rest being discarded. Here, the YUV data is extracted as a replacement for the data buffer, and several global variables that depend on the image size are temporarily updated before encoding, and then returned back to normal, so the encoder is essentially fooled into accepting a smaller size image. Unlike the previous option, this choice always discards the same amount of data, and there is minimal loss in appearance, as no editing is evident on the receiving end. However, significant motion of the head will cause it to appear as if the camera were panning or tilting, and this saves slightly less bandwidth at a slightly greater cost in processing time.

Blurring: This option causes the area outside the facial region to undergo an averaging process, with coarseness set by the user. The background is essentially divided into a number of rectangles, each of which is filled with the average values of the component pixels. Even at the very low levels of coarseness, where the loss of detail is noticeable but not particularly bad, the savings in bandwidth tend to be considerable. At the highest levels, the background becomes almost completely uninformative and the savings approximate those of simply discarding the background as in the pseudo-cropping method.

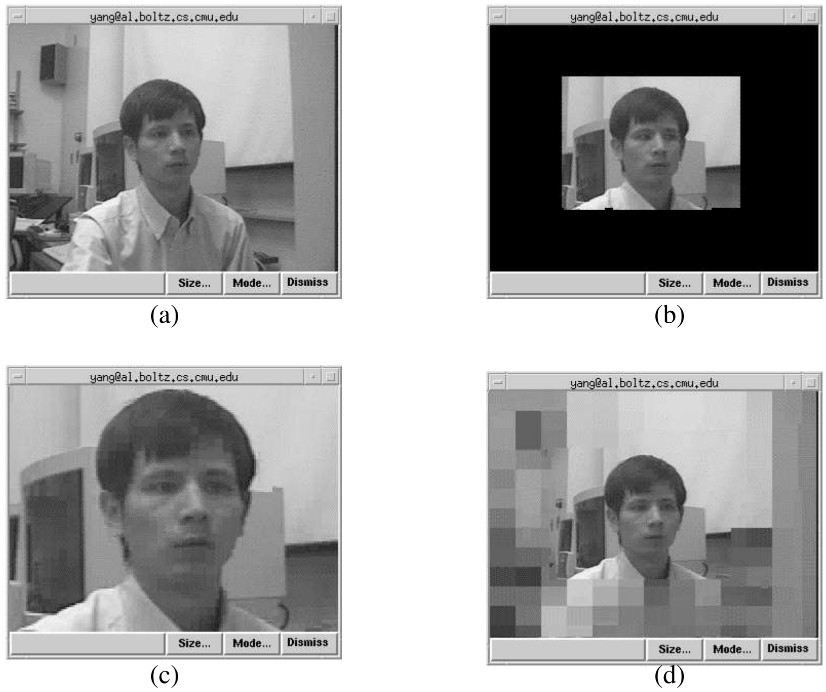

Figure 8 **An example of an image in different modes: (a) original; (b) pseudo-cropping; (c) slicing; (d) blurring**

4.2 An Interface for Selecting a Face

In a video-conferencing session, there might be multiple people on each end, and one party might wish to track the current speaker in the other, regardless of that speaker's relative position to the camera. By default, the face-tracking algorithm selects the largest face to start tracking. If two faces are present, and one is significantly closer to the camera than the other, the closer face will be chosen (assuming that the faces are of equal physical size). The tracker itself does not directly provide the controls for specifying preferences, but does allow any point to be used as a focal point for the search. We have developed a user interface to allow the user to select a face to start tracking.

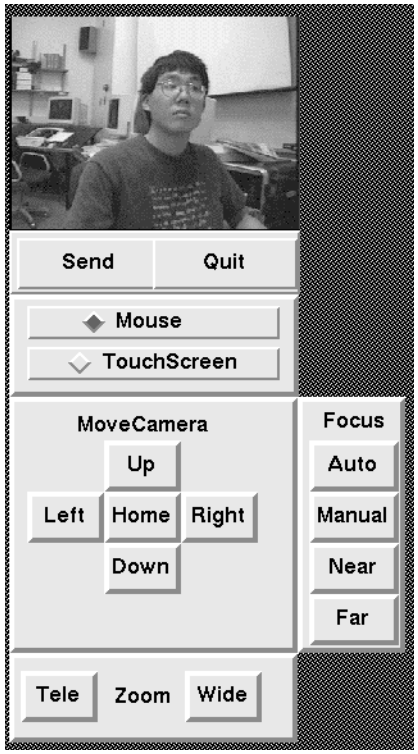

Figure 9 **The control panel for face selection and manual camera control**

The interface is a Tcl/Tk-based program used for specifying focal points for the face-tracker's search algorithm. It includes a window for the camera's video output, to allow the user to select a point; a pair of buttons that controls whether the input device is the mouse or a touch-screen; and buttons for panning, tilting, focusing, and zooming the camera. The implementation is twofold; the GUI routines are written in Tcl/Tk, but the Tcl/Tk script is controlled by C code, which uses the Tcl/Tk code to draw everything and receive the mouse clicks, but then controls the actual effects after translating the mouse clicks and screen touches to coordinates.

The interface is designed to work with several other programs, of which the most important is the face-tracker. Given that the windows are immoveable, the coordinates yielded from using either input device allow selection to determine which function is desired. If the selected point is within the video window, for instance, that point will be highlighted with a white circle around it. The user may then use the Send button to transmit the point to the tracker, via a socket server as intermediary. As visual feedback, the white circle turns red briefly.

The next portion of the interface provides the switching between the mouse (default) and the touch screen. The mouse support is built in with the Tcl/Tk script, and the touch-screen support is provided through a program run in the background, which simply waits for the user to touch the screen and passes on the coordinates to point. The user may switch between the devices at will with this section, although confirmation is required via a preemptive dialog box. The last part deals with the camera controls. These controls provide the ability to pan and tilt the camera, as well as to adjust the focus or zoom; note, however, that the tracker also communicates with the camera and tends to adjust it to facilitate tracking. The user selects the desired button, the point program queries the server as to the camera's current status, and the appropriate instruction is sent in return.

5 Experimental Results

To test the effectiveness of the proposed method, we have performed various experiments. The system has been running on different platforms such as Alpha and HP workstations in our lab. In order to minimize the effects of load and network traffic, we used as metric vic's count of bytes transmitted for each image, instead of frames per second. Two sets of data were collected for such comparison. Additional code was written to create two more versions, one in which it behaved normally but also recorded the unmodified YUV image data to a file, along with the corresponding facial coordinates, and a second version which replaced the camera and socket input with that read from the YUV and coordinate file. These were used to insure that in two trial sets, every modification used the exact same frames and face coordinates.

Trial 1: First set of 100 frames

Table 1 summarizes the results of the data set under different modifications. The higher levels of blurring approach the same results as that of discarding the entire background. This is because essentially the background becomes one largish rectangle in a highly blurring case. Significant savings in excess of 30-50% over normal images using the same compression technique even without much loss of subjective detail.

Trial 2: Second set of 100 frames

The second trial collected a different set of 100 frames taken at a different time, with a somewhat different distance from the camera. In addition, as movements were not scheduled or restricted in either sequence, a different set of tracking coordinates was generated. Table 2 summarizes these results.

Figure 10 shows the normalized averages of Trials 1 and 2, with the modifications denoted by the 3-character labels -- Nor meaning normal and unchanged, Cro meaning pseudo-cropping, Sli being slicing and Bx meaning blurring to intensity x; B** is blurring at the maximum intensity of 100. Several interesting observations have been discovered in experiments. There are fairly significant differences in the results of the two trials. This can generally be explained in terms of the differing face-to-frame-size ratios; the closer the face to the camera and thus the higher ratio, the less background there is to reduce, and therefore less reduction of frame size from background-oriented methods.

The savings in frame sizes through slicing was essentially identical, at approximately 50% when sending 25% of the image; here, distance to the camera has no effect except on the efficacy of the face tracker and the possible need to focus further away; as the program sent the same-sized section, the size reductions were essentially equivalent. The degree of movement of the subject also matters significantly, since the efficacy of the face tracker may be adversely affected by sharp, abrupt movements, and changes in the background (such as no longer being blocked by the subject) affect the different modifications to different degree.

In both trials, blurring and cropping provided marked decreases in the frame size, with the higher levels of blurring essentially sacrificing a recognizable background to quickly send whatever region is designated interesting, such as an area denoted by the coordinates from the face tracker. A progressive blurring technique, fine near the face and coarser further away, might instead be feasible, to keep some of the image quality in the vicinity but retain much of the savings in bytes transferred of the higher blurring intensities.

Table 1: Summary of Trial 1

	Total Bytes	**Relative Average**
Normal	1203148	100%
Cropped	321365	27%
Sliced	597433	50%
Blurred, 2%	695972	58%
Blurred, 5%	487568	41%
Blurred, 10%	419898	35%
Blurred, 20%	411514	34%
Blurred, 30%	395542	33%
Blurred, 40%	385405	32%
Blurred, 50%	331779	28%

Table 1: Summary of Trial 1

	Total Bytes	Relative Average
Blurred, 60%	347700	29%
Blurred, 70%	330210	27%
Blurred, 80%	317393	26%
Blurred, 90%	294347	24%
Blurred, 100%	279574	23%

Table 2: Summary of Trial 2

	Total Bytes	Relative Average
Normal	1557339	100%
Cropped	913048	59%
Sliced	684929	44%
Blurred, 2%	1200918	77%
Blurred, 5%	1031649	66%
Blurred, 10%	969410	62%
Blurred, 20%	958626	62%
Blurred, 30%	956547	61%
Blurred, 40%	949820	61%
Blurred, 50%	915667	59%
Blurred, 60%	930376	59%
Blurred, 70%	934616	60%
Blurred, 80%	938228	60%
Blurred, 90%	959600	62%
Blurred, 100%	942630	61%

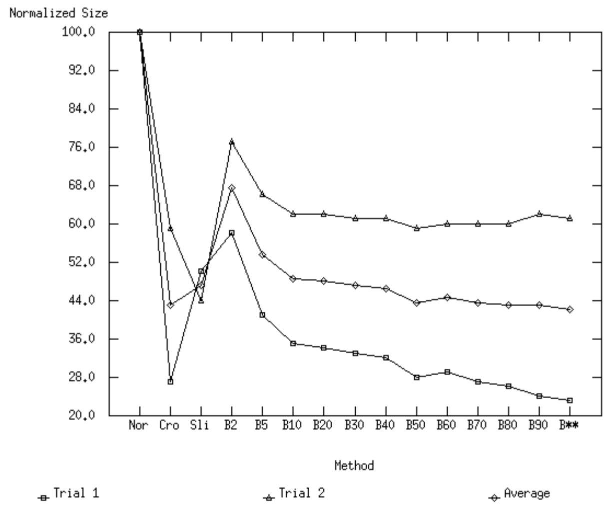

Figure 10 **Averages of Trials 1 and 2 (Equal Weights)**

6 Conclusion

We presented an approach to low bitrate video tele-conferencing by focusing attention on the most important information. We have demonstrated that a selection mechanism on the top of a normal codec to help choose particular regions can yield significant savings in the required bandwidth. We have identified three ways to cut off the data rate, i.e., blurring, slicing, and pseudo-cropping. More specifically, the blurring modification at the lower intensities results in a very acceptable trade-off between only a minor degradation of image quality, and none in the selected area, versus a marked improvement in bandwidth requirements. Slicing, which yields a constant-size extract around the selected area as a new, smaller image, also provides significant savings at low quality cost, with the minor drawback of an apparent camera movement whenever the selected region is moved within the frame. Pseudo-cropping, the most severe of the methods used, saves the greatest amount of bandwidth and is the simplest to implement, but sacrifices everything outside the selected area. Thus, in a video-conferencing situation where low bandwidth is a more restrictive constraint than computing power, the use of such measures as blurring or even outright discarding the data outside an important region can preserve the most significant information to minimize transfer requirements, all without requiring a special client on the receiving side.

Acknowledgments

We would like to Dr. Hui Zhang for his valuable comments. This research was sponsored by the Advanced Research Projects Agency under the Department of the Navy, Naval Research Office under grant number N00014-93-1-0806.

References

[1] H. Li, A. Lundmark, and R. Forchheimer, “Image sequence coding at very low bit rates: a review,” IEEE Transactions on Image Processing, Vol.3, No.5, pp. 589-609, 1994.

[2] K. Aizawa and T.S. Huang, “Model-based image coding advanced video coding techniques for very low bit-rate applications,” Proceedings of the IEEE, Vol. 83, No.2, pp. 259-71, 1995.

[3] J. Yang and A. Waibel, “Tracking human faces in real-time,” CMU CS Technical Report, CMU-CS-95-210, November, 1995.

[4] S. McCanne and V. Jacobson, “vic: a flexible framework for packet video,” Proceedings of ACM Multimedia ‘95.

[5] C. Bundesen, “A theory of visual attention,” Psychological Review, Vol. 97, No. 4, pp. 523-547, 1990.

[6] D.E. Broadbent, “Task combination and selective intake of information,” Acta Psychologica, Vol. 50, pp. 253-290, 1982.

[7] A. Allport, “Visual attention,” in: Posner, M., ed., Foundations of Cognitive Science, MIT Press, Cambridge, Ma., pp. 631-683, 1989.

[8] A. Triesman, & G. Gelade, “A feature integration theory of attention,” Cognitive Psychology: Vol. 12, pp. 97-136, 1980.

[9] A. Hulbert and T. Poggio, “Spotlight on attention,” MIT AI Laboratory Memo AI-817. Massachusetts Institute of Technology, Cambridge, MA, 1985.

[10] C. Koch, and S. Ullman, “Shifts in selective visual attention: towards the underlying neural circuitry,” Human Neurobiology, Vol. 4, pp. 219-227, 1985.

[11] M.C. Mozer, “A connectionist model of selective attention in visual perception,” Technical Report CRG-TR-99-4, University of Toronto, Toronto, Canada, 1988.

[12] G. Wyszecki and W.S. Styles. Color Science: Concepts and Methods, Quantitative Data and Formulae, Second Edition, John Wiley & Sons, New York, 1982.