User Interface for Video Observation over the Internet *

Bor Prihavec, Franc Solina
Computer Vision Laboratory
Faculty of Computer and Information Science,
University of Ljubljana

Abstract

We present the design and application of a system for live video transmission and remote camera control over the World Wide Web. Extensive testing of the Internet Video Server (IVS) prompted us to improve its user interface. We developed the GlobalView extension of IVS which enables the generation of panoramic images of the environment and a more intuitive control of the camera. The live video frame is superimposed on a 360^o panoramic picture. By interactively moving a rectangular frame in the panoramic picture the user locally selects the new direction of the camera. Once the view is selected the users prompts the selection and the command is issued over the Internet to the remotely controlled camera. Visual summaries of activities on an observed location can be generated and custom queries are possible with a similar intuitive user interface.

*This work was supported by the Ministry of Science and Technology of Republic of Slovenia (Project L2-8721) and by the European Cultural Month Ljubljana 1997 (Project Netropolis – Cyborg's Eye).

1 Introduction

Live video transmission over the Internet and interactivity are becoming more and more popular. This very moment we can find on the World Wide Web hunderts of cameras all across the world that we can use as our remote eyes [1, 2]. Video can give us information that static images can not (telepresence) and with further development of technology and Internet infrastructure the speed of transmission and the amount of video imagery will only increase. Therefore, intelligent control of video capture by means of changing the view direction of the camera, spatial structuring of visual information, as well as generation of visual summaries are essential for successful application of this technology. To study user-interface issues of remotely operable cameras and provide a testbed for the application of computer vision techniques (motion detection, tracking, security) we developed the Internet Video Server (IVS) system [3] which we recently expanded with the GlobalView interface.

The next chapter describes the IVS system in detail. The third chapter is on generating panoramic images which are used in the GlobalView extension of IVS. Chapter 4 describes the GlobalView interface which enables a much more intuitive remote control of the camera, especially if the connection is slow. The fifth chapter contains the discussion on IVS and GlobalView can be used to make visual summaries. Conclusions in chapter six give a short summary and some ideas for future research.

2 Internet Video Server

IVS enables live video transmission and remote camera control (pan & tilt) over the Word Wide Web.

In designing the system certain objectives were followed:

- client side platform independence,
- optimization for slow connections, and
- remote control of the camera.

Platform independence of clients was easily achieved by using the World Wide Web technology - HTTP (Hyper-Text Transfer Protocol) and HTML (Hyper-Text Markup Language): on almost any platform one can find a Web browser capable of displaying text and graphics.

For greater flexibility the camera can be placed at a location without or with a slow Internet connection. This is made possible by a two-level IVS system architecture. The first level is the distribution level and the second

is the camera level (Fig. 1). The camera sends processed images to the level 1 which distributes them to all clients on the Internet. Therefore the distribution level has to have a relatively fast Internet connection as it has to serve many clients simultaneously. Requests that come from the clients on Internet are filtered and processed by level 1 and only necessary data and control commands are sent to level 2. Therefore, the main task of level 2 is digitizing and compressing the picture. The channel between the two levels is optimized for data transfer.

IVS can also operate in a single-level mode. In this mode we loose the advantages of parallel processing in the two-level mode. The camera level would have to serve also the requests that come from the clients and this would cause a reduction of performance.

2.1 IVS Architecture

IVS consists of four specialized modules (Fig. 1) and a few general modules which are part of the system software.

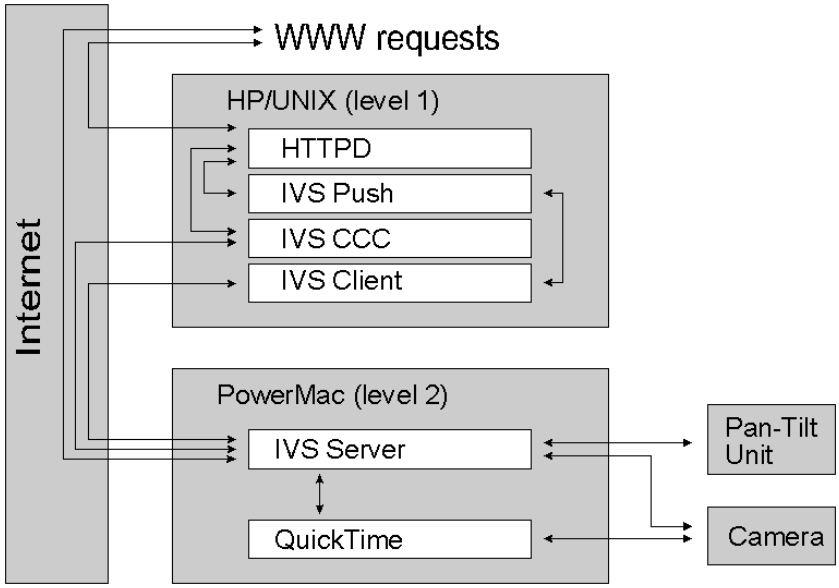

Figure 1: IVS architecture

2.1.1 IVS-Server

The heart of the IVS is the IVS-Server module which runs on the camera level. Its main tasks are serving the requests that come from clients (internal

or external), digitizing and processing the image and controlling the camera and the pan-tilt unit. Images are digitized in resolution 120×160 pixels and then transformed into the JPEG format.

IVS-Server can serve two sets of requests:

- The first set consists of requests sent by IVS-Client and IVS-CCC module which runs on the distribution level. They are referred to as IVS (internal) requests.
- The second set are HTTP (external) requests that come directly from clients on the Internet. HTTP requests are served when operating in the single-level mode.

In both sets are also requests for: single image, continuous image stream, control of the camera and pan-tilt unit, and status of the module.

2.1.2 IVS-Client

This module is located on the distribution level and transports images between the two levels. At first, the persistent connection is established with the IVS-Server module with request for continuous image stream. Every image that this module receives is accessible by multiple concurrently running IVS-Push modules. IVS-Client has to be harmonized with IVS-Push modules.

2.1.3 IVS-Push

This module sends the image stream to clients on the Internet. For each client a separate instance of this module is executed which enables independence between serving requests. The speed of image transmission can be selected which allows every client to customize the amount of data transmitted over the network.

2.1.4 IVS-CCC – Camera Control Center

The Camera Control Center is the front end of the system (Fig. 2). Through this module the user interaction is carried out. Using this module the camera can be moved in all directions (left, right, up and down) or turned to some predefined position depending on the actual location of the camera. Only one user can interact with the system at the time and therefore a queue has been implemented to allow a fair time sharing. Every user gets a time slot of 5 minutes to control the camera. Then the control is passed to the next request in the queue.

Figure 2: The plain IVS user interface showing buttons for controlling the camera direction

2.1.5 Interaction between modules

At system startup the IVS-Server and IVS-Client modules are started. When IVS-Server is ready for accepting the requests, the IVS-Client module connects to it with the request for a continuous image stream. Each subsequent image is digitized and compressed by the IVS-Server module and then transmitted over network to the IVS-Client module to store it and make it available for further distribution to clients all over Internet.

There are two kinds of clients on the Internet: passive and active. Passive clients are only observers while active clients are also interacting with the system by means of controlling the camera.

When a client (passive or active) starts a session a separate instance of the IVS-Push module is invoked. This instance is responsible for transmitting the video to this client only. In this way few things are gained. Every client can receive the image stream with different speed depending on connection throughput and its own settings. Image streams transmitted to clients are independent which improves the performance of the whole system. Images transmitted to clients are supplied by the IVS-Client module. Since there can be more than one IVS-Push module running the synchronization with the IVS-Client module is required.

An active client enables in parallel also the control of the camera. Re-

quests for controlling the camera are served by the IVS-CCC module. Since only one user can control the camera at the time, camera might be occupied by another user when a request arrives. In this case the user is added into the waiting queue and informed about the situation (number of users in the queue, position in the queue, estimated waiting time). When the user operating the camera (operator) stops controlling the camera (either its time run out or he pressed the quit button) the next user in the queue gets the control. When the IVS-CCC module receives the request for a camera movement from the current operator the appropriate command is issued to the IVS-Server module which then performs the operation requested.

2.2 IVS hardware

For image capture we used different types of CCD cameras with lenses of different focal length and with or without automatic aperture control. For turning the camera in the desired direction we used the pan-tilt unit PTU-46-17.5 by Directed Perception, Inc. The camera level (level 2) of the system was implemented on an Apple Macintosh computer with a Power PC processor which handles the control of the camera and the pan-tilt unit, image capture and image compression. The server (level 1) is a HP UNIX workstation.

2.3 Application of IVS

The Internet Video Server has been tested quite extensively several times covering different events and using almost all possible means of connecting the camera level and the distribution level [4, 5, 6]. So far, we used modem connections over analog and digital (ISDN) telephone lines and GSM mobile telephone as well as direct Internet connections. In the future we would like to experiment also with microwave connections.

We observed that the users of IVS had some problems operating the camera with the interface shown in Fig. 2 which became more severe when the response of the system was slow due to slow connections. A user would, for example, press the button for moving the camera one step to the left. If nothing happened in a few seconds, he would press again the same button or try a different operation. Due to buffering, he would suddenly get a new image which could be a result of just the first command, or a combination of any number of subsequent commands. Due to slow and uneven reaction of the IVS system to user's actions the operation does not seem to be very predictable from the users viewpoint.

Another problem with the user interface is more perceptual. If the focal length of the lens is large, one can easily loose the notion where the camera

is pointing at. Observing a distant location through IVS or a similar system gives a somewhat limited perception akin to looking at the surrounding through a long tube. Due to the precisely controlled position of the camera by means of the pan-tilt unit, individual images acquired by IVS can be assembled in a panoramic view of the entire surroundings which can be used as a backdrop for the current live image.

We expanded the IVS system with the GlobalView interface to give IVS users a better spatial perception of the observed location and a more intuitive control of the camera platform. We first describe how panoramic views are acquired and then how the panoramic view is used in the IVS interface.

3 Panoramic views

Panoramic views have been traditionally generated by special photographic cameras and photographic techniques by means of rotating the whole camera or just the aperture in the camera lens. To be usefull in a computer system the photographic film must be first scanned which, needless to say, prevents any real-time application.

To generate panoramic images using a video camera two general approaches are known:

1. using special omnidirectional sensors and
2. using conventional image-based systems.

3.1 Omnidirectional video

The first approach, which is becoming more and more popular, is using specialized cameras or camera systems that are able to acquire omnidirectional visual information [7]. Optics of such sensors use a fish-eye lens or combine standard lens on a video camera with a conic mirror [8], a spherical mirror [9], or a paraboloidal mirror [10].

These images, which cover a complete half sphere, must be mathematically processed to free them of severe distortions and get a proper perspective view. The advantage of this approach is that the whole panorama is imaged at once and that several users can each move their own "virtual camera" over this image to observe the part of the scene they are interested in. However, the benefit of such single step image capture is reduced by a very uneven resolution of these panoramic images. The majority of the image covers the sky or the ceiling of indoor spaces while the usually more interesting parts of the image are on the boundaries where the resolution is the poorest. To get

usefull information from both hemispheres, two such parabolic mirrors and cameras must be applied at once.

3.2 Camera rotation

The second approach involves camera rotation and/or integration of overlapping images taken with a regular camera. By panning the camera over a scene and composing the video frames, large panoramic images of arbitrary shape and detail can be created [11]. To automatically construct those panoramic images, however, we must derive the alignment (warping) transformations based directly on images or some other parameters that are gained automatically, rather than relying on manual intervention. If the camera direction information is automatically available it can be used as a warping parameter. This makes possible fast automatic panoramic image composition which can be applied on static scenes. This method is inappropriate for dynamic scenes since the panoramic image is generated gradually.

Even if the camera direction information is available, as in our case, we still need some additional parameters to perform fast panoramic image construction without the need to search for a translation vector between two consecutive images. We need to know the horizontal and vertical view angles of the camera lens. By knowing these two parameters we can perform image composition automatically without the need to calculate the relative position between two consecutive images from the images them selves [11]. Using the pan-tilt unit we know the precise position of the captured image within the whole panoramic image.

3.3 Generating panoramic views with a pan-tilt manipulator

We generate 360^o panoramic views by turning the camera (in horizontal and, if necessary, in vertical direction) and assembling the pictures into a single slit. To get a rectangular panoramic image the individual images must be transformed from sphere to cylinder coordinates. If we are scanning only the area in the level of the camera horizon with a small vertical angle this transformation is not necessary since the image distortion is small. In our case, the vertical angle of panoramic images is about 90^o and therefore the transformation is necessary to assure smooth panoramic images. The panoramic images obtained in this way have a uniform resolution.

If the camera is rotating around its optical center we can assume that the optical center is located in the center of a sphere and the camera can observe

the inner surface of the sphere. From the camera's point of view every scene can be represented as an image mapped onto the inner surface of the sphere - a spherical panoramic image (Fig 3). Next, spherical panoramic image must be transformed to fit onto the surface of a cylinder - cylindrical panoramic image. Fig. 4 shows how this transformation is performed.

A transformation consists of three steps:

1. mapping of an image I onto the projection plane (Cartesian coordinates) ($I \rightarrow I_p$)
2. transformation into spherical coordinates ($I_p \rightarrow I_s$)
3. transformation into cylindrical coordinates ($I_s \rightarrow I_c$)

Individual image I is defined as a matrix of pixel values $I(i,j)$ where $i \in [-\frac{W}{2}, \frac{W}{2}]$ and $j \in [-\frac{H}{2}, \frac{H}{2}]$. W and H are image width and image height.

In spherical coordinates every point on the surface of the sphere can be represented as $I_s(\varphi, \vartheta)$, where φ is angle in the horizontal direction ($\varphi \in [-\pi, \pi]$) and ϑ is angle in the vertical direction measured, from the sphere horizon ($\vartheta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$).

Every point on the cylinder surface can be represented as $I_c(\psi, v)$, where ψ is the angle in horizontal direction ($\varphi \in [-\pi, \pi]$) and v is the elevation.

3.3.1 Transformation to spherical coordinates

First, it is necessary to find out which part of the sphere is on the image I taken and for that, the horizontal and vertical view angles of the camera are needed. The method for obtaining these two camera parameters is described in the following section. Since we control the movement of the camera we know the exact orientation of the camera's optical axis (φ_0, ϑ_0). For the sake of simplicity we assume that the optical axis goes through the center of the image $I(0,0)$. The center point of the image matches the point $I_{s_0} = (\varphi_0, \vartheta_0)$ in sphere coordinates and $\vec{r_0}$ in Cartesian coordinates.

$$\begin{aligned} x_0 &= R \cdot \cos(\varphi_0) \cdot \cos(\vartheta_0) \\ y_0 &= R \cdot \sin(\varphi_0) \cdot \cos(\vartheta_0) \\ z_0 &= R \cdot \sin(\vartheta_0) \end{aligned}$$

$\vec{r_0} = (x_0, y_0, z_0)$ is the intersection of the camera optical line and the sphere's surface. The center of the projection plane is also moved into this point. The correct mapping for all other points in the image $I(i,j)$ to the spherical coordinates has to be calculated from the camera orientation and other camera parameters.

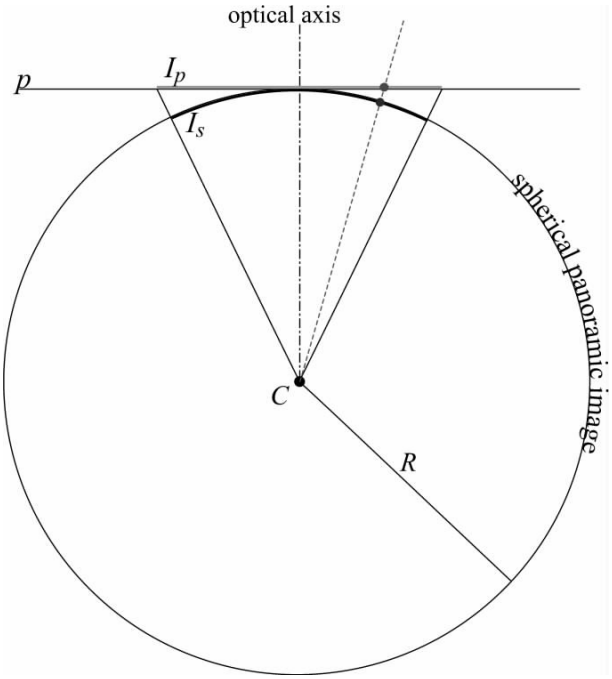

Figure 3: Image I_p on the projection plane p is mapped onto the sphere surface - I_s.

In the next step we find the point $\vec{r_1}$ on the projection plane which corresponds to the point $I(0, \frac{H}{2})$. From $\vec{r_0}$ and $\vec{r_1}$ the bases of the image in projection plane ($\vec{b_x}$ and $\vec{b_y}$) are obtained.

$$\vec{b_y} = (\vec{r_1} - \vec{r_0}) \cdot \frac{2}{H}$$

$$\vec{b_x} = (\vec{b_y} \times \vec{r_0}) \cdot \frac{2 \cdot \tan\alpha}{|\vec{b_y}| \cdot W}$$

Now every point $I(i,j)$ can be mapped into the point $I_p(i,j) = \vec{r_p}'(i,j)$ on projection plane p and then into the point $\vec{r_p}(i,j)$ on the surface of the sphere:

$$\vec{r_p}'(i,j) = \vec{r_0} + i \cdot \vec{b_x} + j \cdot \vec{b_y}$$

$$\vec{r_p}(i,j) = \frac{\vec{r_p}'(i,j)}{|\vec{r_p}'(i,j)|} \cdot R$$

From $\vec{r_p}(i,j) = (x_p, y_p, z_p)$ the spherical coordinates $I_s(\varphi, \vartheta)$ of every point $I(i,j)$ can be calculated:

$$\varphi = arctan2(y_p,\ x_p)$$

$$\vartheta = arcsin(\frac{z_p}{R})$$

where $arctan2(y,x)$ is defined as $\arctan(\frac{y}{x})$ and returns a value in the range -π to π radians.

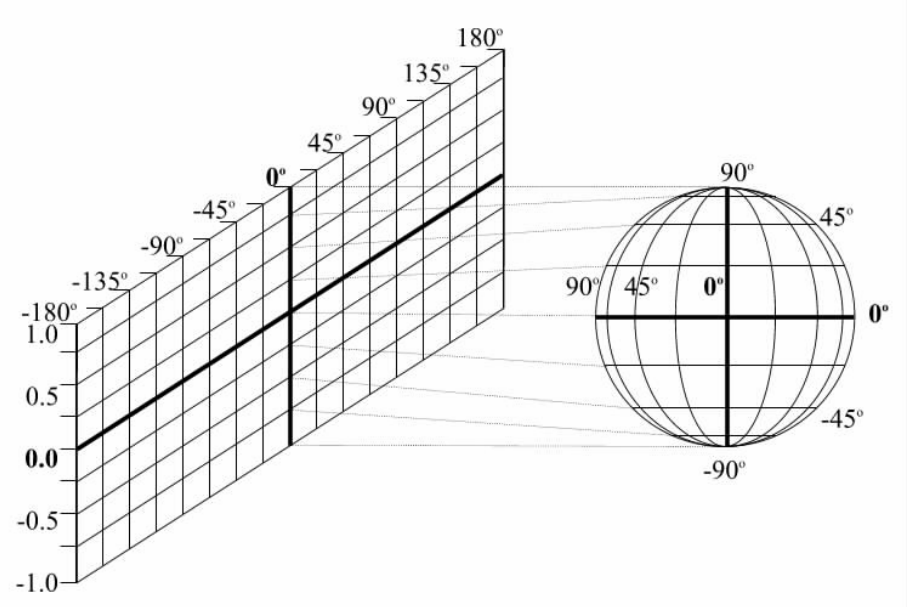

Figure 4: Projection of the spherical images onto the cylindrical surface.

3.3.2 Cylindrical panoramic image

To transform the image on the spherical surface I_s into the cylindrical panoramic image I_c, every point $I_s(\varphi, \vartheta)$ is transformed into the corresponding point $I_c(\psi, v)$ on the cylinder surface as follows:

$$\psi = \varphi$$

$$v = \frac{\vartheta}{\frac{\pi}{2}} \cdot R$$

R represents the radius of the sphere and can be set to 1.

In order not to loose resolution in the vertical direction the elevation v is mapped from ϑ as the vertical arc length between the corresponding

point on the sphere surface and the point on the horizon of the sphere with the same horizontal angle φ. Elevation v is now in range of $[-1, 1]$. The horizontal resolution is decreasing from maximum at the elevation 0 to the minimum at the elevations 1 and -1. This horizontal resolution decrement can be formulated as the function of vertical angle ϑ in this way:

$$Res_H = Res_{Hmax} \cdot \cos(\vartheta)$$

One can observe that the resolution at elevation -1 and 1 (the north and the south pole of the sphere) is 0. This means that the north and south poles of the sphere are transformed into the top-most and bottom-most line in the cylindrical panoramic image.

The number of images needed to generate a panoramic image is $N_h \times N_v$ where N_h is the number of images taken in each horizontal plane and the N_v is the number of horizontal scans. For example, our camera's horizontal and vertical view angles are 20^o and 15^o, respectively, and we want to produce the panoramic image which would cover 360^o in the horizontal direction and 90^o in the vertical direction. The minimum number of images taken for that would be $(\frac{360^o}{20^o}) \times (\frac{90^o}{15^o}) = 108$ images.

3.3.3 Brute-force scanning

To achieve smooth lines and at the same time avoid the need of any geometric transformation of images, a brute-force scanning approach can be used. In the brute-scanning approach only a few center columns are taken from each image since the distortion increases from the center vertical line outward and only the center column has no distortion at all. Therefore, the brute-force scanning approach increases the number of images significantly. The number of center columns taken from each image is a compromise between quality and time that we need for scanning the whole panoramic image. The properties of the scene that we are scanning should be considered also (scene dynamics, structure, light sources, etc.).

3.4 Camera calibration

To be able to automatically register images directly from knowing the camera viewing direction, the camera lens horizontal and vertical view angles are required. We have developed an algorithm that calculates these two camera parameters and is designed to work with cameras where zoom settings and other internal camera parameters are unknown. The algorithm is based on the accuracy of the pan-tilt unit on which the camera is mounted. The basic idea of the algorithm is to calculate the translation between two images while

the camera has been rotated in the horizontal or vertical direction. When the exact angle of rotation is known, the image angles can be calculated.

The complete calibration algorithm for one of the axes consists of the following steps:

1. Position the pan-tilt unit to the base position for the calibration and get a reference image.

2. Turn the pan-tilt unit for a very small angle in the direction which is being calibrated. Get an image from this position.

3. Calculate the translation between these two images and calculate the first raw approximation of the camera view angle.

4. Turn the pan-tilt unit to the calculated position that should correspond to some predefined offset (for example 1/4 of the image) and get the image from this position.

5. Calculate the translation between this and the reference image and calculate the corrected view angle.

6. If the view angle correction is small enough or some constant number of steps has been performed then finish, otherwise go to step 4.

The resolution of the pan-tilt unit used in our experiments is 185.1428 arc seconds per pan or tilt position, which means that one pan or tilt position corresponds to 0.0514 degrees.

For calculation of the translation between two images the combination of two algorithms is used [12]. Basic incremental-motion estimator can obtain estimates for motion, given that frame-to-frame displacements are small. The precision and range of the estimator are enhanced through coarse-fine motion tracking within a pyramid structure.

Using this camera calibration algorithm few things should be noted. First, the selection of the scene on which calibration is performed is very important. For example: estimation of the vertical camera view angle would fail when using a scene with parallel vertical lines. This can be observed in table 1 for the estimation estimation of β - vertical view angle for the target. The calibration was performed on a wood pattern with clean vertical lines. In horizontal direction the estimation was successful. Since this algorithm performs estimation of the horizontal and the vertical camera view angles separately it enables the selection of different area (viewing direction) for each direction.

	α	β	estimation of α		estimation of β	
Near target	33,73	24,06	Average:	33,730	Average:	22,666
	33,73	22,22	St.Dev:	0,000	St.Dev:	0,863
	33,73	21,80				
	33,73	22,42				
	33,73	22,83				
Far target	34,35	26,13	Average:	34,350	Average:	26,130
	34,35	26,13	St.Dev:	0,000	St.Dev:	0,000
	34,35	26,13				
	34,35	26,13				
	34,35	26,13				

Table 1: Algorithm results - α is the estimated horizontal angle of camera and β is the estimated vertical view angle of the camera. The near target was a wood pattern with clean vertical lines located approximately 40 cm from the camera. The far target was aproximately 4 meters away from the camera.

Second, the distance from camera to objects on the scene on which calibration is performed is important. Since normal cameras do not have telecentric lenses [10] the view angles change when objects with different distance from the camera are focused. Therefore, the best solution is to perform the camera view angle estimation directly on the scene which is to be scanned. In table 1 this change of view angle is quite obvious when comparing the estimated view angles between a far and a near target.

3.5 Results

On our equipment the scanning of panoramic pictures in the brute-force scanning manner (without geometric transformation) takes from a few seconds (building the panorama out of complete, non-overlapping images) to about 3 minutes (taking only the 5 vertical lines in the center of each image) resulting in coarser or finer vertical image joints.

Using the described geometrical transformation the panoramic image generation can be quite slow. Fortunately, it can be speeded up by using look-up tables, precalculated for every horizontal scan and then used instead of applying the actual transformation. For every pixel on the digitized image $I(x, y)$ its relative position on the cylindrical panoramic image is calculated.

Panoramic picture on Fig. 5 was generated without applying any geometrical tranform. Fig. 6 shows a panoramic picture taken in a single horizontal scan with a fish eye lens using the brute force scanning approach.

Figure 5: Raw panoramic image generated without any tranformation applied.

Figure 6: 360^o panoramic image generated with only one horizontal scan using a wide angle lens.

4 Integration of the panoramic view into the IVS

GlobalView interface (Fig. 7) is a combination of a static panoramic view and live images received from IVS. Live images arriving from IVS are transformed from spherical to cylinder coordinates and superimposed on the corresponding position in the panoramic view.

At the system startup, the panoramic image is generated first by scanning the complete surroundings of the camera. When a user starts interacting with the system he is shown the whole panoramic image. A rectangular frame in the panoramic image indicates the current direction of the camera. Inside this attention window the live video image is seamlessly superimposed onto the panoramic image. By means of a mouse the user can move the attention window and in this way control the direction of the camera. When the attention window is at the desired location the user prompts this to the system. From the position of the attention window within the panoramic image the physical coordinates of the next camera direction are computed and the appropriate command is issued to the pan-tilt unit.

The whole interaction with the IVS system is carried out through the

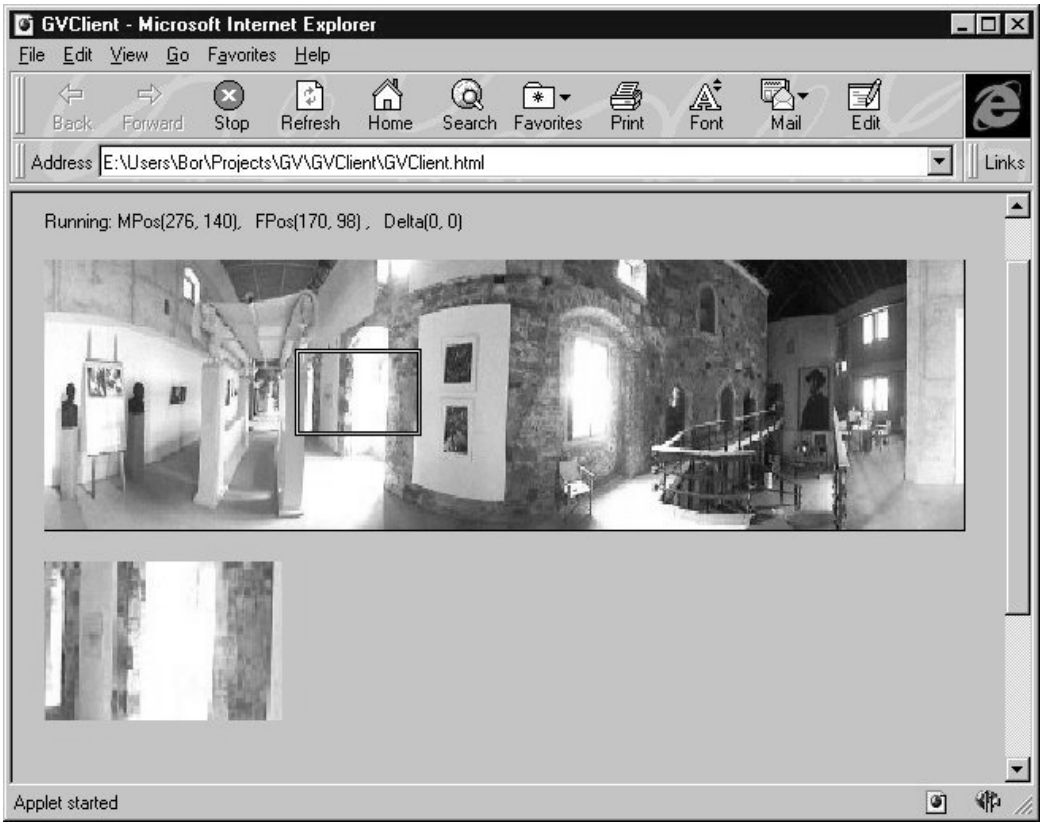

Figure 7: GlobalView interface of IVS. The rectangle in the panoramic image indicates the attention window which in turn indicates the current direction of the camera. By moving the rectangle the user controls the direction of the camera. On the bottom of the web page an enlarged live video image is shown.

attention window. Moving this attention window results in moving the camera. At any time, only one user can move the camera since allowing more than one user to interact with it could be confusing.

To allow a fair time sharing every user can control the camera for a few minutes and after that, the control is passed to the next user waiting in the queue. Other users can only observe the remote area. Their control of the attention window is disabled. Since only the first user in the queue – the active user – can move the attention window. In remote surveillance applications where only one operator is required this multi-operator feature can be disabled.

Since the panoramic image is rather large in comparison to the live video image and the attention window is small, a separate window for live video can be present to make a more detailed observation possible. In this way the attention window acts like a magnifier glass over the remote location.

The client-side user interface was written in Java (Java applet) which is run within any web browser that supports Java. This approach enables platform independence. With minor changes, it can be run on any Java virtual machine.

At present, only one focus rectangle is present since only one camera is available. In general, several cameras could be controlled by the same number of attention windows. A combination of cameras with different focal length would be possible. Zoom lenses could be controlled by resizing the attention window. An interesting combination would be a live panoramic image attainable by a system such as the Omnicam [10] and the IVS system offering a combination of peripheral, low resolution image, and a high resolution focal image. Such an interface would allow a better overview of remote areas without a complicated user interaction.

5 Discussion

When using the IVS we sometimes wish to see what was happening at a remote location in the past. For example, one would want to check who entered a building in some specified time interval, or just to keep track of the number of people on a specific location.

To enable this kind of functionality the live video frames from the remote location should be saved in some appropriate form together with a time stamp and information about the camera direction.

The system could operate in two modes: in active or passive mode. When operating in passive mode the control over the camera would be completely in the hands of an operator who would control the direction of the camera.

The system's only job would be the recording of the user actions and saving of the video images.

When operating in active mode, the system should autonomously perform continuous observation of the areas of interest. These areas of interest could be predefined (like doors, windows, paintings on the walls, etc.) or could be extracted automatically. Automatic extraction of the areas of interest could be done by finding the areas of high levels of change. In this way, a priority list of the areas of interest could be generated. The system would check the locations higher on the list more often. If new areas of large change would arise the priority list could be updated dynamically. Of course, the entire area could be defined as an area of top interest and the system would then continuously scan the entire area.

Different intelligent schemes for visual surveillance using IVS are still under consideration. We are integrating into the IVS system a simple motion detection method which would enable the camera to automatically track a moving object.

In addition, a tool for visual report generation which will allow custom queries is under construction. The basic feature of the system will be the playback facility which will paste images on the appropriate place in the panoramic image in chronological order. Different options for this playback will be available:

- selecting the speed of the playback,
- selecting the time frame,
- playing back only those images in which a change occurred,
- playing back only images which are in a specific subframe of the panoramic image,
- tracking of selected objects over time.

For the generation of visual summary reports we are using a SQL database which enables image base queries and reports. The Informix Universal Server seems to be a good solution since it is a powerful SQL database which can be fully customized with so called DataBlades. A DataBlade is a set of functions which can be applied on the data within a database. This enables the definition of user defined filters which can be used as image evaluation functions in queries and reports. It has a built-in web support so clients could request and see the results of different kinds of customized queries within their favorite WEB browser.

6 Conclusion

IVS is a system which enables live video transmission and remote camera control over the Internet. With the GlobalView extension, which generates a panoramic image of the whole environment, and its user interface interface the observer gains a better understanding of the observed location and a more intuitive control of the camera. Video-conferencing and remote surveillance are examples of applications that would certainly benefit from such an interface.

Our future work is directed to integration of visual information from several cameras (i.e. relay-tracking) and visually-based teleoperation of a mobile platform.

References

[1] Webcam32 Gallery
`http://www.kolban.com/webcam32/forms/gallery_read.hts`

[2] Tommy's List of Live Cam Worldwide,
`http://www.rt66.com/ ozone/cam.html`

[3] B. Prihavec, A. Lapajne, and F. Solina 1996. Active video observation over Internet. In B. Zajc and F. Solina, editors, *Proceedings Fifth Electrotechnical and Computer Science Conference ERK'96*, Vol. B, 117–120, Portorož, Slovenia, IEEE Slovenia Section.

[4] Jožef Plečnik exhibition at Prague Castle,
`http://razor.fri.uni-lj.si:8080/Plecnik/`

[5] ROTAS-TENET, `http://razor.fri.uni-lj.si:8080/Rotas/`

[6] Srečo Dragan 1997. Netropolis – Cyborg's eye, artinternet installation project, European Cultural Month Ljubljana 1997,
`(http://razor.fri.uni-lj.si:8080/Netropolis-ECML97)`

[7] F. Hamit 1997. New video and still cameras provide a global roaming viewpoint. *Advanced Imaging*, March, 50–52.

[8] Y. Yagi, S. Kawato and S. Tsuji 1994. Real-time omnidirectional image sensor (COPIS) for vision-guided navigation. *IEEE Trans. on Robotics and Automation*, 10(1), 11–22.

[9] J. Hong, X. Tan, B. Pinette, R. Weiss, E. M. Rieseman 1991. Image-based homing. *Proc. IEEE Int. Conf. on Robotics and Automation 1991*, 620–625.

[10] Shree K. Nayar 1997. Catadioptric Omnidirectional Camera. Proc. of 1997 Conference on Computer Vision and Pattern Recognition, 482–488.

[11] R. Szeliski 1996. Video Mosaics for Virtual Environments. *IEEE Computer Graphics and Applications*, 16(2), 22-30.

[12] J. R. Bergen, P. J. Burt, R. Hingorani and S. Peleg 1990. Computing Two Motions from Three Frames. Technical report, David Sarnoff Research Center, Subsidiary of SRI International, Princeton, NJ 08543-5300