Journal of Network and Computer Applications (1998) **21**, 219–237
Article No. ma990074

User interface for video observation over the internet

Bor Prihavec and Franc Solina

Computer Vision Laboratory, Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia

This paper presents 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. The GlobalView extension of IVS was developed 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° static 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. The static panoramic image is constantly updated in areas where new live video information gets available. Two methods are described for static panoramic image generation: one uses geometric transformation and the other is the brute-force scanning approach. We discuss how visual summaries of activities on an observed location can be generated and custom queries made with a similar intuitive user interface.

1. Introduction

Live video transmission over the Internet and interactivity are becoming more and more popular. At this moment hundreds of cameras all across the world can be found on the World Wide Web that can be used as remote eyes [1, 2]. Video can provide 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) the Internet Video Server (IVS) system [3] was developed which was recently expanded with the GlobalView interface.

The next section describes the IVS system in detail. The third section concerns generation of panoramic images which are used in the GlobalView extension of

Tel: +386 61 1768 381; Fax: +386 61 1264 647; E-mail: {Bor.Prihavec/Franc.Solina}@fri.uni-lj.si

1084–8045/98/040219 + 19 $30.00/0

IVS. Here two methods are described for static panoramic image generation: one uses geometric transformation and the other is a brute-force scanning approach. Section four describes the GlobalView interface which enables a much more intuitive remote control of the camera, especially if the connection is slow. The fifth section contains the discussion on how IVS and GlobalView can be used to make visual summaries. Conclusions in section 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 and tilt) over the World Wide Web.

In designing the system, certain objectives were followed:

- client side platform independence;
- optimization for slow connections;
- 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.

To achieve greater flexibility of operation, the camera can be placed at a location with or without 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 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 the 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 the advantages of parallel processing in the two-level mode are lost. The camera level would also have to serve 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.

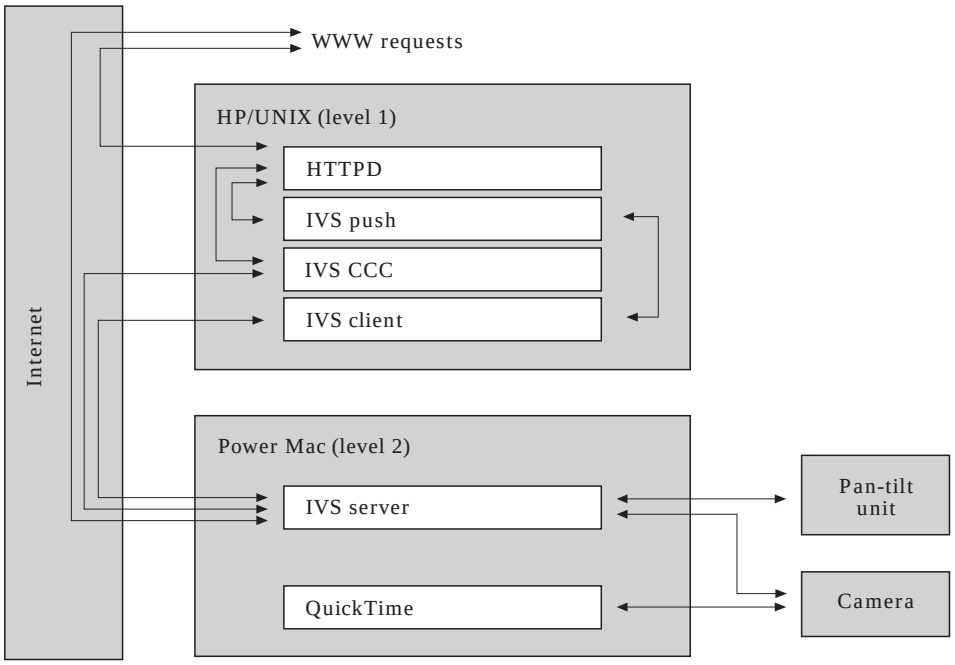

Figure 1. IVS architecture consists of two levels: distribution (1) and camera level (2).

2.1.1 *IVS-server*. The heart of 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 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 there 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. For image stream transmission the server-push technology is used. Image stream is similar to motion JPEG stream where each image (frame of video sequence) is treated as a separate image and a completely new and separate JPEG compression on each image is performed. The size of each compressed image is around 5 KB and therefore the network requirements are very straightforward to calculate: for 10 frames per second approximately 50 KB per second is required.

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 a time and therefore a queue has been implemented to allow a fair time-sharing. When a user posts a request for camera control, the IVS-CCC module is invoked. At first

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

it verifies if the user is already in the queue and if not the user is added to the queue. If the user is first in the queue then the request is submitted to the camera level. If the user is not the first in the queue then the request is discarded and the user is informed about its position in the queue and the estimated waiting time. Every user gets a time slot of 5 minutes to control the camera and then the control is passed to the next user in the queue.

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. Video frames are digitized and compressed by the IVS-server module and then transmitted over network to the IVS-client module to store them and make them 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 speeds, 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 control of the camera in parallel. Requests for controlling the camera are served by the IVS-CCC module. Since only one user can control the camera at the time, the 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 it pressed the quit button) the next user in the queue gains 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, different types of CCD cameras were used with lenses of different focal length and with or without automatic aperture control. To turn the camera in the desired direction, the pan-tilt unit PTU-46-17.5 (Directed Perception, Inc.) was used. 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) was 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]. So far, modem connections have been used over analogue and digital (ISDN) telephone lines and GSM mobile telephone as well as direct Internet connections. Further experiments may be carried out with microwave connections.

Between August 1996 and May 1998, the IVS system was accessed and used more than 40 000 times from more than 1000 different computers all over the world.

It was 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 lose the notion to which part of the scene 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.

The commands for 'left' and 'right' move the camera for 15° in horizontal direction and the commands for 'up' and 'down' move the camera for 5° in vertical direction. To move the camera for 180° in horizontal direction the command for 'left' or 'right' has to be applied 12 times and to move the camera for 45° in vertical direction the command for 'up' or 'down' has to be applied nine times. These figures tell us that if we want to find some object on the scene and for that we need to to preview the whole scene at least 100 camera movements are required. Using an updatable panoramic view as a part of the user interface the number of camera movements required to find some target can be decreased significantly: to one move only.

The IVS system was expanded 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. How panoramic views are acquired will be described 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 useful 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;
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 [6]. Optics of such sensors use a fish-eye lens or combine standard lens on a video camera with a conical mirror [7], a spherical mirror [8], or a paraboloidal mirror [9].

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 useful 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 [10]. To automatically construct those panoramic images, however, the alignment (warping) transformations must be derived 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 somewhat inappropriate for very dynamic scenes since the panoramic image is generated gradually.

Even if the camera direction information is available, as in our case, some additional parameters are still needed to perform fast panoramic image construction without the need to search for a translation vector between two consecutive images. The horizontal and vertical view angles of the camera lens need to be known. By knowing these two parameters, image composition can automatically be performed without the need to calculate the relative position between two consecutive images from the images themselves [10]. Using the pan-tilt unit, the precise position of the captured image is known within the whole panoramic image.

3.3 *Generating panoramic views with a pan-tilt, manipulator*

Static 360° panoramic views are generated by turning the camera (in horizontal and, if necessary, also 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 scanning is done of 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° 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 centre, it can be assumed that the optical centre is located in the centre 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). After this, a spherical panoramic image must be transformed to fit onto the surface of a cylinder which is a cylindrical panoramic image. Fig. 4 shows how this transformation is performed.

The transformation consists of two steps:

1. transformation into spherical coordinates ($I \rightarrow I_s$);
2. transformation into cylindrical coordinates ($I_s \rightarrow I_c$).

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

In spherical coordinates every point on the surface of the sphere can be represented as $I_s(\phi, \vartheta)$, where ϕ is angle in the horizontal direction ($\phi \in [-\pi, \pi]$)

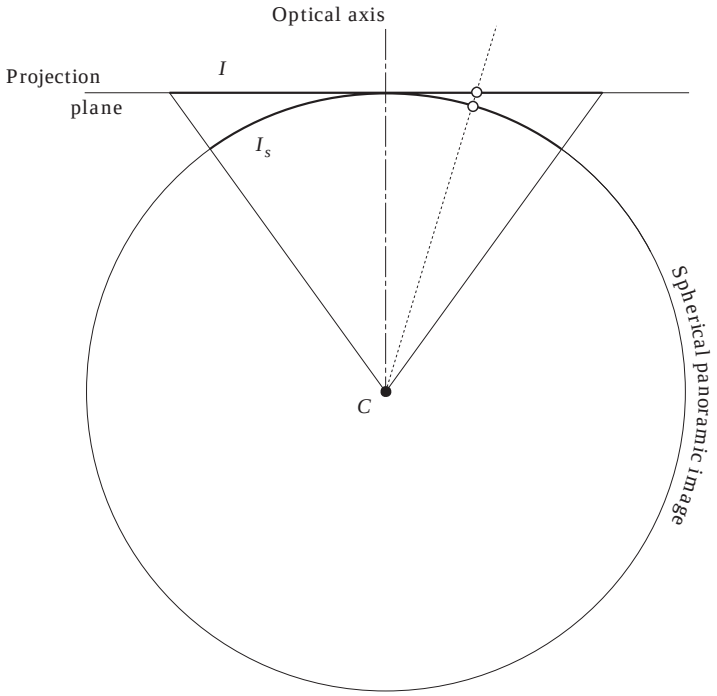

Figure 3. Image I on the projection plane is mapped onto the sphere surface—I_s.

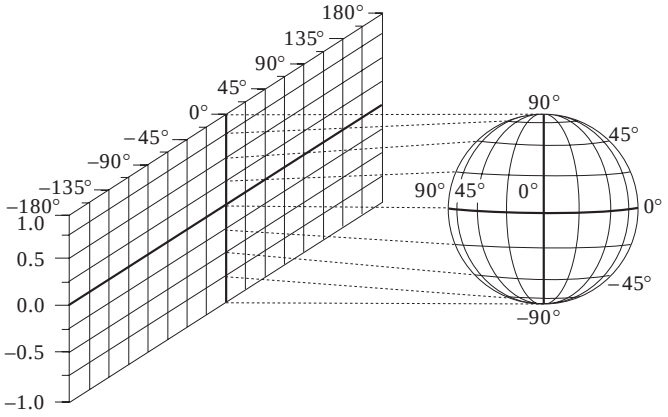

Figure 4. Projection of spherical images onto cylindrical surface.

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 $I_c(\psi, \upsilon)$, where ψ is the angle in horizontal direction ($\phi \in [-\pi, \pi]$) and υ is the elevation.

3.3.1 *Transformation to spherical coordinates*. Since we control the movement of the camera we know the exact orientation of the camera's optical axis (ϕ_0, ϑ_0) and we also know the exact position of the camera's optical centre (C). From ϕ_0 and ϑ_0 a 3×3 rotation matrix $\mathbf{R}$ can be obtained where the column vectors r_i of the rotation matrix $\mathbf{R}$ represent the unit direction vectors of the coordinate axis of the camera-centered reference frame [11]. The origin of the camera-centered reference frame corresponds to the camera optical centre and the z-axis corresponds to the camera optical axis. The projection plane is the plane with equation $z=1$ and the coordinate axes in the image are parallel to the x-and y-axis.

If a scene point P has coordinates (X, Y, Z) and the position vector of camera's optical centre C has coordinates (X_C, Y_C, Z_C) then the projection of the scene point P to the point r with coordinates (u, v) on projection plane is:

$$\varrho \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{R}^T (P - C)$$

where $\varrho = r_{13}(X - X_C) + r_{23}(Y - Y_C) + r_{33}(Z - Z_C)$ is a non-zero real number and r_{ij} is the (i, j)th entry of the rotation matrix $\mathbf{R}$.

The transition from the geometrical coordinates (u, v) to the pixel coordinates (x, y) represented by the column vector $p = (x, y, 1)^T$ is possible if the intrinsic camera parameters (represented by the camera calibration matrix $\mathbf{K}$) are known:

$$\varrho p = \mathbf{K}\mathbf{R}^T (P - C)$$

where

$$\mathbf{K} = \begin{pmatrix} k_x & s & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Here, (x_0, y_0) are the pixel co-ordinates of the optical centre of the image, the parameter s is usually referred to as skewness of the pixel ($s=0$ corresponds to rectangular pixels) and k_x and k_y indicate the number of pixels per unit length in the horizontal and vertical directions respectively. The values of k_x and k_y are calculated from horizontal and vertical camera view angles. The method for obtaining these two parameters is described in the following section.

For every image point p in I, the direction in which all possible scene points P lay needs to be known, that is, which scene points P are projected into the image point p. The direction of the ray of sight creating the image point p is given by a 3-vector $\mathbf{K}^{-1}P$ in the camera-centered reference frame. In world frame this direction is $\mathbf{K}^{-1}P$ and the parameter equation of the ray of sight in the world frame is:

$$P = C + \varrho \mathbf{K}^{-1} P \qquad \text{for some } \varrho \in \mathbb{R}.$$

From direction of the ray of sight $\tilde{P}$ in world frame:

$$\tilde{P}=\begin{pmatrix}\tilde{X}\\ \tilde{Y}\\ \tilde{Z}\end{pmatrix}=\mathbf{K}^{-1}P$$

the spherical coordinates (ϕ, ϑ) of every image point p can be calculated:

$$\phi=\arctan 2(\tilde{X}, \tilde{Z})$$
$$\upsilon=\arcsin\left(\frac{\tilde{Y}}{\sqrt{\tilde{X}^2+\tilde{Y}^2+\tilde{Z}^2}}\right)$$

where $\arctan 2(y, x)$ is defined as $\arctan(\frac{y}{x})$ and returns a value in the range $-\pi$ to π radians. In this way the image I is transformed into spherical coordinates $-I_s(\phi, \vartheta)$.

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 $I_s(\phi, \vartheta)$ is transformed into the corresponding point $I_c(\psi, \upsilon)$ on the cylinder surface as follows:

$$\psi=\phi$$
$$\upsilon=\frac{\vartheta}{\frac{\pi}{2}}\cdot R$$

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

To prevent any lose of resolution in the vertical direction, the elevation υ is mapped from ϑ as the vertical are 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 υ 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:

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

It can be observed 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 lines 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 scan and the N_v is the number of horizontal planes. For example, our camera's horizontal and

vertical view angles are 20° and 15°, respectively, and we want to produce the panoramic image which would cover 360° in the horizontal direction and 90° in the vertical direction. The minimum number of images taken for that would be $(\frac{360^\circ}{20^\circ}) \times (\frac{90^\circ}{15^\circ}) = 108$ 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 centre columns are taken from each image since the distortion increases from the centre vertical line outwards and only the centre column has no distortion at all. Therefore, the brute-force scanning approach increases the number of images significantly. The number of centre columns taken from each image is a compromise between quality and time that is needed for scanning the whole panoramic image. The properties of the scene that are being scanned should also be considered (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. An algorithm has been developed 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 obtain 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 one quarter 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.

Table 1. *Results of the calibration algorithm: 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 approximately 4 metres away from the camera*

	α	β	Estimation of α		Estimation of β	
Near target *c*. 40 cm	33.73	24.06	Average:	33.730	Average:	22.666
	33.73	22.22	SD:	0.000	SD:	0.863
	33.73	21.80				
	33.73	22.42				
	33.73	22.83				
Far target *c*. 4 m	34.35	26.13	Average:	34.350	Average:	26.130
	34.35	26.13	SD:	0.000	SD:	0.000
	34.35	26.13				
	34.35	26.13				
	34.35	26.13				

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

For calculation of the translation between two images the combination of two algorithms is used [12]. Basic incremental-motion estimators 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, a few points 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 of β which is the vertical view angle for the target. The calibration was performed on a wood pattern with clean vertical lines. In the 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. Second, the distance from camera to objects for the scene on which calibration is performed is important. Since normal cameras do not have telecentric lenses [9], 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 the equipment used, the scanning of panoramic pictures in the brute-force scanning manner (without geometric transformation) takes from a few seconds

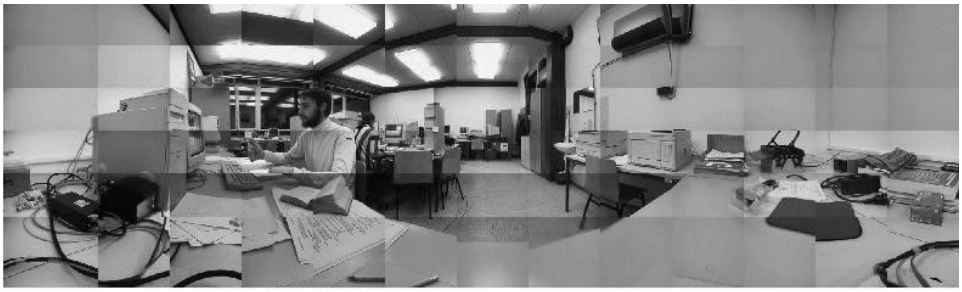

Figure 5. Raw panoramic image assembled out of 78 images without any transformation applied.

Figure 6. 360° panoramic image generated with only one horizontal scan out of 200 images using a wide angle lens.

(building the panorama out of complete, non-overlapping images) to about 3 minutes (taking only the five vertical lines in the centre 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 I_c is calculated.

Panoramic picture on Fig. 5 was generated without applying any geometrical transform and therefore the edges where individual images join are broken. It is composed out of 78 images and the process took around 20 seconds. Figure 6 shows a panoramic picture taken in a single horizontal scan with a fish-eye lens using the brute force scanning approach. Approximately 2 minutes and 200 images were required for this process.

4. Integration of the panoramic view into the IVS

Global View 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.

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 the live video image in actual resolution (left) and a zoomed panoramic view (right) are shown.

At the system startup, the panoramic image is first generated 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 select 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. When the camera is moved to a new direction, the last image from the old position is pasted to the static panoramic image. In this way the panoramic image is constantly updated in the areas of most interest to observers.

The whole interaction with the IVS system is carried out through the 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 in actual resolution can be present to make a more detailed observation possible (see Fig. 7). Since the panoramic image is too large to fit onto the screen in its original size, a separate window is present which acts like a magnifier glass over the remote location. In this way the user has a complete overview over the remote location without losing any details due to panoramic image resolution reduction.

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 superimposed on the same panoramic image. A combination of cameras with different focal length would be possible resulting in attention windows of different sizes. Zoom lenses could be controlled by interactively re-sizing the attention window. An interesting combination would be a live panoramic image attainable by a system such as the Omnicam [9] 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 it sometimes needs to be seen what was happening at a remote location in the past. For example, a user might 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 (e.g. doors, windows, paintings on the walls, etc.) or could be extracted automatically. Automatic extraction of the areas of interest could be carried out 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 arose, the priority list would 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. A simple motion detection method which would enable the camera to automatically track a moving object is being integrated into the IVS system.

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 report a SQL database is being used 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 built-in web support so clients can request and see the results of different kinds of customized queries within their favourite WEB browser.

6. Conclusion

The Internet Video Server (IVS) and its Global View Extension have been built. IVS is a system which enables live video transmission and remote camera control over the Internet. With the Global View extension, which generates a panoramic image of the whole environment, and its user-friendly interface, the observer gains a better understanding of the observed location and a more intuitive control of the camera. To preview the whole scene using only IVS at least 100 camera movements are required. By means of the Global View extension the number of camera movements required to locate and direct the camera to some target can be decreased to one. Video-conferencing and remote surveillance are examples of applications that would certainly benefit from such an interface.

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

Acknowledgements

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).

References

1. Yahoo: Web Cams, Found at URL: http://dir.yahoo.com/Computers_and_Internet/Internet/Interesting_Devices_Connected_to_the_Net/Web_Cams/
2. Live streaming camera views and live video webcams around the world (livecams), Found at URL: http://arie.net/cam/livecams.htm
3. B. Prihavec, A. Lapajne and F. Solina 1996. Active video observation over Internet. In *Proceedings Fifth Electrotechnical and Computer Science Conference ERK'96*, Vol. B, B. Zajc and F. Solina, (eds), Portorož, Slovenia, IEEE Slovenia Section, 117–120.
4. Srečo Dragan 1998. Exhibitions 1993/98. Ljubljana, ZDSLU.
5. Srečo Dragan 1997. Netropolis—Cyborg's eye, artinternet installation project, European Cultural Month Ljubljana 1997, which can be found at URL: http://razor.fri.uni-lj.si:8080/Netropolis-ECML97.
6. F. Hamit 1997. New video and still cameras provide a global roaming viewpoint. *Advanced Imaging*, March, 50–52.
7. Y. Yagi, S. Kawato and S. Tsuji 1994. Real-time omnidirectional image sensor (COPIS) for vision-guided navigation. *IEEE Transactions on Robotics and Automation*, **10**, (1) 11–22.
8. J. Hong, X. Tan, B. Pinette, R. Weiss and E. M. Rieseman 1991. Image-based homing. *Proceedings of the IEEE International Conference on Robotics and Automation 1991*, 620–625.
9. S. K. Nayar 1997. Catadioptric Omnidirectional Camera. *Proceedings of 1997 Conference on Computer Vision and Pattern Recognition*, 482–488.
10. R. Szeliski 1996. Video Mosaics for Virtual Environments. *IEEE Computer Graphics and Applications*, **16**, (2) 22–30.
11. T. Moons 1998. A guided tour through multiview relations. *Proceedings of SMILE '98 Workshop, LNCS* **1506**, 304–346.
12. J. R. Bergen, P. J. Burt, R. Hingorani and S. Peleg 1990. Computing Two Motions from Three Frames. *Technical report, David Sarnoff Research Centre*, Subsidiary of SRI International, Princeton, NJ 08543–5300.

Franc Solina is professor of computer science at the University of Ljubljana and head of Computer Vision Laboratory at the Faculty of Computer and Information Science. He received a B.Sc. and a M.Sc. in electrical engineering from the University of Ljubljana, Slovenia in 1979 and 1982, respectively, and a Ph.D. in computer science from the University of Pennsylvania in 1987. His research interests include range image interpretation, 3D shape reconstruction and computer vision applications on the Internet.

Bor Prihavec is a graduate student at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. He received a B.Sc. in computer science from the University of Ljubljana, Slovenia in 1996. His research interests include image mosaicing, user interfaces, networking and real time applications.