Egocentric Navigation for Video Surveillance in 3D Virtual Environments

Gerwin de Haan*
TU Delft

Josef Scheuer
TU Delft
TNO Science And Industry

Raymond de Vries
TNO Science And Industry

Frits H. Post
TU Delft

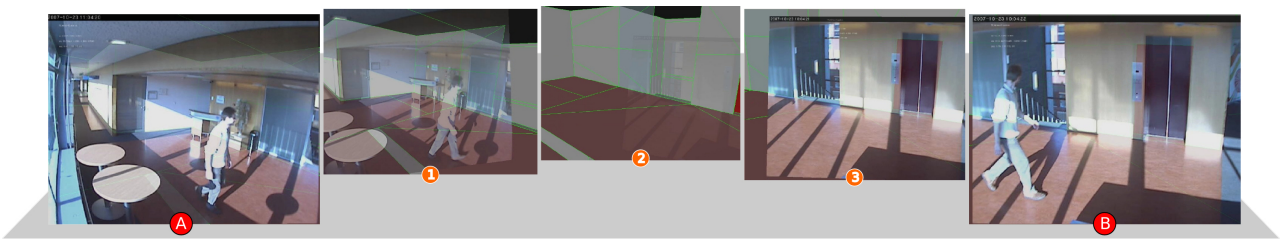

Figure 1: View transition sequence of tracking a person between camera A and adjacent camera B in an office hallway surveillance scenario (see video). The guided 3D navigation enables simple first-person video observation while ensuring a good visual flow for spatial context. Our method dynamically embeds and blends video canvases in a 3D VE which consists of characteristic model landmarks and perspective lines.

Abstract

Current surveillance systems can display many individual video streams within spatial context in a 2D map or 3D Virtual Environment (VE). The aim of this is to overcome some problems in traditional systems, e.g. to avoid intensive mental effort to maintain orientation and to ease tracking of motions between different screens. However, such integrated environments introduce new challenges in navigation and comprehensive viewing, caused by imperfect video alignment and complex 3D interaction. In this paper, we propose a novel, first-person viewing and navigation interface for integrated surveillance monitoring in a VE. It is currently designed for egocentric tasks, such a tracking persons or vehicles along several cameras. For these tasks, it aims to minimize the operator's 3D navigation effort while maximizing coherence between video streams and spatial context. The user can easily navigate between adjacent camera views and is guided along 3D guidance paths. To achieve visual coherence, we use *dynamic video embedding*: according to the viewer's position, translucent 3D video canvases are smoothly transformed and blended in the simplified 3D environment. The animated first-person view provides fluent visual flow which facilitates easier maintenance of orientation and can aid in spatial awareness. We discuss design considerations, the implementation of our proposed interface in our prototype surveillance system and demonstrate its use and limitations in various surveillance environments.

Keywords: video surveillance, virtual environments, navigation

Index Terms: H.5.1 [Information Systems]: Information Interfaces And Presentation—Multimedia Information Systems;

1 Introduction

Video surveillance control rooms have to deal with an ever-growing amount and diversity of data from complex environments. Currently, operators observe the video streams directly on large matrix display arrangements, combined with an interactive camera layout plan, e.g. see Figure 2. With densely placed cameras in a complex environment, it is hard to maintain orientation and coherence between individual streams. Our main application is a surveillance system in such a complex 3D environment, a newly-built soccer stadium with over 200 cameras. Cognitive overload can easily occur and it can become hard to perform tasks such as selecting regions of interest in a camera image or tracking motions between different cameras. Our current research concerns the design of user interfaces to enhance the support of surveillance tasks.

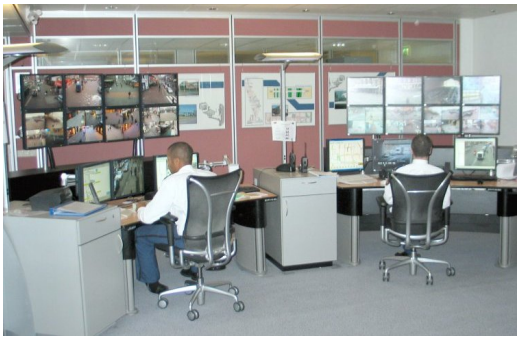

Figure 2: Surveillance control room with a traditional video matrix display arrangement. Operators select sets of individual videos from 2D maps.

We observed that a major bottleneck in many surveillance tasks lies in the difference between *view reference frames* [15] in different camera views and the map view. Where operators can naturally reason and act upon the *first-person* or *egocentric* views from a single camera, they have to mentally "translate" their reasoning back to the other camera views or the map, a *third-person* or *exocentric* representation. This dual representation can hinder direct integration of video information with map data and requires two modes of interaction . An example is to track a person which walks outside of the current camera range to the left. The operator needs to select a camera to the left-side of the current camera. The location of the current camera has to be found on the exocentric map view and its left neighbour selected. In an egocentric view, the operator would naturally indicate the system to move the viewpoint to the left.

In this paper, we propose an egocentric or first-person 3D interface for video surveillance monitoring in VEs. The goal of this interface is to provide improved *situational awareness* and to serve

*Corresponding author, g.dehaan@tudelft.nl

IEEE Symposium on 3D User Interfaces 2009
14-15 March, Lafayette, Louisiana, USA
978-1-4244-3965-2/09/$25.00 ©2009 IEEE

as a basis for new egocentric interaction techniques. It provides easy navigation along the available video streams, as if the operator "flies" from camera to camera in the 3D environment. Our main contribution is the enhanced *visual flow* during navigation along integrated images in the VE. This is obtained by combining our method of *dynamic embedding* of video streams and guided navigation between optimal views. The dynamic, distortion-reducing blending transitions between camera views provide spatial coherence when image distortions are too large. Finally, we propose a *context graph* to represent and manage spatial relationships between cameras, trajectories and the 3D model.

The remainder of this paper is organised as follows: In the next section we discuss related work. In section 3, we describe the embedding of video streams and our dynamic video embedding technique. We then discuss the guided navigation for exploring the 3D environment in section 4. The context graph concept is described in section 5. We describe the implementation details of our prototype and results of our interface in section 6. Finally, in section 8, we draw conclusions and describe future work.

2 Related Work

New types of interfaces have been proposed for spatial coherence in multi-camera video surveillance systems [5, 7]. The geographical context of camera streams is generally conveyed through placement of video thumbnails, camera icons and coverage indicators on a 2D map of the environment. In the Spatial Multi-Video player [6], the spatial coherence between video streams is provided by carefully arranging spatially related video thumbnails around the current, selected camera stream. In a sense, these interfaces convey spatial context through embedding multiple egocentric representations (video streams) into the exocentric 2D map representation. Their user study revealed that spatial context, provided by either a map or spatially related videos, improved user performance and acceptance for tracking persons walking on an office floor across multiple cameras. We embed complete video streams directly in the spatial context in which they have been captured by incorporating them directly into a 3D environment model, with our motivation being to alleviate duality in the navigation interface. In a similar fashion, the recent DOTS system [4] displays segmented images of tracked persons directly in 3D.

For embedding video surveillance images in a 3D model, regular texture mapping often produces poor results as it requires a-priori knowledge or live segmentation for mapping the image to the 3D model. Many techniques from image based modelling and rendering are used to integrate information from acquired images and 3D models [12]. In the *Augmented Virtual Environment* (AVE) system [8, 10] and the *Video Flashlight* technique [9], live video streams are integrated with a 3D environment model using *Projective Texture Mapping* [11]. This technique, originally devised for lighting effects and shadow generation, projects an acquired image from the camera viewpoint as a texture on the 3D model. Although objects that are not part of the model are projected back onto a different part of the model, it is reported that people have little difficulty in dealing with those effects. The use of multiple video streams are also demonstrated, although effects of dense camera placement and overlapping images are not discussed. [17] explore the design space of combining videos with environment models. Their testbed combines existing, static rendering techniques and provides basic navigation and report on possible usage patterns. In their recent user study, Wang et al. [16] show that 3D integration can indeed be a useful addition in surveillance tasks.

To avoid complete reconstruction in *model space* we aim at guiding the user into correct *view space* reconstructions. To achieve this, we use an specific form of dynamically transforming texture billboards, inspired by the work in Photo Tourism [14]. The use of many video billboards can lead to confusing, cluttered scenes, as the images can be strangely overlaid on the 3D model. Also in Photo Tourism, dynamic view-dependent filtering is used to show only those images that possibly contribute to the current view.

The user should be able to smoothly navigate to this position instead of *teleporting*, in order to maintain spatial context and orientation [2]. Issues on supporting the reconstruction of a mental map by navigating spatial related information are also addressed by [1]. The experiment performed in this study reveals that for inspecting information arranged in space, animations and smooth transitions of the viewpoint can help the user to build up the mental map of spatial object relations. The use of a 3D environment does introduce the problem of fast and easy 3D navigation, e.g. losing orientation and awkward controls. The important role of automated and guided navigation for a complex surveillance context was already emphasized in [17]. The 3D video player in [4] provides arbitrary 3D viewing or camera transitions to follow automatically tracked persons. Our method focuses on simplifying interactive navigation, such that the user is guided or constrained along the views in a comprehensible manner, see also [3]. In a recent extension of Photo Tourism, Snavely et al. [13] also combine path-based, egocentric navigation and blended photographs.

3 Dynamic Video Embedding

For our approach to visualise multiple video streams, this section describes the principles of representing the images. We currently make three main assumptions on the availability of camera and scene information. First, we assume images have been taken by stationary cameras, conforming to the *pinhole camera*-model, without optical distortion. Second, we assume there is available information on the cameras, such as the position, pose and perspective parameters. Finally, we assume an abstract, virtual 3D model is available that provides the landmarks of the environment.

To render video images in a 3D environment, the visualisation has to generate geometric objects that can be added to the specification of a 3D scene. The embedding of these objects in $3D$ space relates to the context from which the images have been generated. By additionally rendering a model of the environment in the background, the generated visualisation objects augment the virtual scene of the model with information from the real site.

Furthermore, this section covers the issues on viewing such arranged images. Reconstructing the position and settings of the camera for the user's viewpoint achieves an optimal representation of an image. On the one hand, viewpoints other than the camera's position give overview by providing the depth cues of the 3D model. On the other hand, such viewpoint also introduce a list of misalignment effects that make the image arrangement appear wrong. Therefore, our visualisation applies view dependent rendering methods in combination with guided user navigation to counteract these effects.

3.1 Static Video Embedding

We use the term *spatial embedding* for arranging a set of generated $2D$ images or projections in a $3D$ space or model. For each of these images, the parameters of the generating camera are used to determine the $3D$ position, orientation and scaling of a canvas object. On this canvas, the created image is applied as a texture. Figure 3 illustrates this arrangement concept within a $3D$ stadium model. First, three images are generated by rendering from three virtual cameras. Then, the generated images are placed on $3D$ canvases, whose position and pose are abstracted from the parameters of the corresponding three cameras. In a new view, the resulting textured $3D$ canvases are rendered together with a wire frame representation of the model. For clarity, a glyph illustrates the camera viewpoint and its view frustum for each camera.

With spatial embedding, regular perspective projection transformations of the $3D$ canvases display an image in a non-cluttered way.

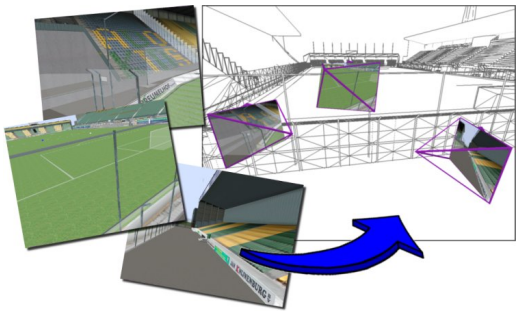

Figure 3: Camera information is used to statically embed three images as textured canvases in a $3D$ context. Glyphs emphasise the original camera placement.

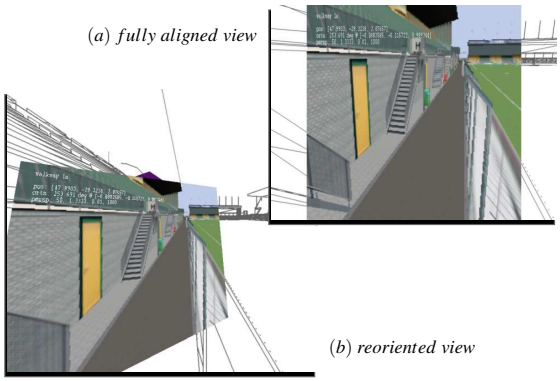

Figure 4: Viewing an image from the exact camera position. In (a) the user's view corresponds to the camera settings. Although the orientation and perspective differs in (b), the projection is still aligned with the background.

This is the main difference with billboards, where the textured canvas is explicitly re-oriented to face the viewer. This is achieved by orienting the plane perpendicular to the viewing direction. In spatial embedding, the lack of this viewer-dependent orientation infers that large differences between viewing direction and camera optical axis can result in large visible distortions. In the following sections we will address the placement of the canvases and these distortion effects.

3.2 Canvas Object Creation

A canvas, a geometrical surface object, is used to display the $2D$ images. One can freely define the resulting image size and its arrangement by modifying the canvas object. As the $3D$ space offers a lot of freedom to choose the parameters for the canvas' placement, the camera parameters are used to constrain this process. In our approach, the parameters for the cameras' ***position***, ***orientation***, ***viewing angle*** α, and ***aspect ratio*** are assumed to be available. From these camera parameters, the missing parameters ***height*** h and ***width*** w can be determined, which are used to setup the canvas object geometry. Depending on the choice for the distance δ from the eye point, this results in a projection surface, which is parallel or even identical to the camera's *near*-plane. Finally, the image data is used as a warped texture to fit the canvas surface.

3.3 Image Canvas Viewing

When a viewpoint corresponds to a camera position, the projection exactly matches the perspective view of this camera while taking the picture. A consequence of this is that the optimal presentation of the projected picture in the $3D$ scene can be achieved by restoring the initial camera properties as the user's view. In essence, the viewpoint in the visualisation system resembles the original camera in the real world. The obvious limitation of this method for optimal image display is this strict dependence on the user's view. For viewpoints other than the exact camera position and orientation, the image representation suffers mainly from two effects.

First, the $3D$ perspective projection causes image distortions that can drastically change the appearance of objects visible in the image. Estimations of sizes, angles and distances of these objects to each other could become more difficult.

Second, for these viewpoints a parallax effect occurs. This apparent shift of the canvas against its background could lead to a perception of seemingly misplaced images. Under such conditions, the visible background, which is either the neighbouring images or the scenery of the environment model, does not exactly match the information shown in the picture. This shift may result in image projections that overlap in the user's view. If the canvases occlude each other, valuable information is hidden. In other cases, images that depict the same object from different points of view could be seen placed next to each other. For the user, the object then appears twice, which might lead to misinterpretation.

To analyse the problem of misalignment let us first assume that the user has an optimal view to an image projection. (see Figure 4a). This means that the location of the virtual view conforms to the camera position.

Here, it is important to see that the picture represents the perspective projection of the real scenery with exactly this point as the centre of projection. The picture is rendered by a canvas, arranged according to the principles of section 3.2. Like all other objects in the scene, this canvas is naturally affected by the perspective distortion in this point. As the projection distorts with the canvas, the image will be displayed perfectly matching the background. Due to the properties of the central projection, this holds even if the user's viewing direction or perspective does not correspond to the camera's original properties any more (see Figure 4b).

For user viewpoints other than the camera position, the following approach categorises the cause of misalignment effects by two main cases: a view displacement along the optical axis of the camera and a displacement away from this line. The reason for dividing these two cases is the fact that they contribute differently to the total misalignment.

The first type of view displacement along the optical axis results in a perspective scaling of the canvas. In case of moving backwards, as depicted in Figure 5a, the projection becomes smaller. At the same time, additional details emerge from the background environment. As the viewpoint's distance to surrounding geometry differs with its distance to the canvas, the appearing background does not scale at the same rate as the canvas. This means, the projection appears too small compared to the background scenery, and does not align any more. A similar effect occurs likewise for the case of moving the viewpoint forward.

The situation changes for the second displacement type away from the camera's optical axis, see Figure 5b. As illustrated, viewpoints apart from this line result in a motion parallax effect. Due to the increasing angle γ of the view to the original line of sight, the canvas gets a perspective shift against the background. For such a point of view, the background scenery appears to shift in the opposite direction, what makes it of course impossible for the canvas' projection to keep alignment.

3.4 View Dependent Embedding

To counteract the effects illustrated above, the projection surfaces should be arranged more carefully. The goal is to improve the projection's match to its context by adjusting the size and placement of

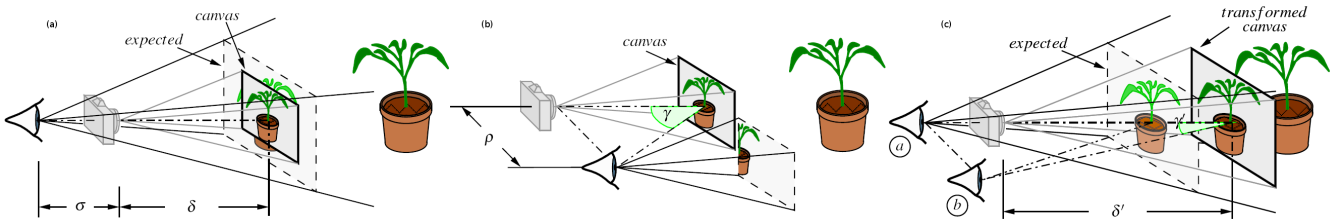

Figure 5: Distortion effects for embedding . If the view position moves along the viewing axis, the embedded video appears to have incorrect scale (a). If the view position moves away from the viewing axis, the embedded video appears warped (b). View dependent embedding (c) transforms the canvas to reduce both these distortion effects.

the canvas. These parameters are modified according to the current view of the user and thus, the following explanations also refers to these canvas corrections by "*view dependent rendering*".

The underlying idea for compensating the misalignment makes use of one central principle for perspective projections: the canvas could be located arbitrarily along the optical axis of the camera as long as it scales according to the properties of the viewing frustum. Deriving the quantities for locating and scaling the canvas from the camera's frustum and the displacement σ along its optical axis (compare Figure 5a), evaluating such transformation comes down to:

$$\delta' = \delta + \|\sigma\| \quad (1)$$

$$scale_{height} = 2\delta' \cdot \tan\left(\frac{\alpha}{2}\right) \quad (2)$$

$$scale_{width} = scale_{height} \cdot aspect \quad (3)$$

The result of this transformation is illustrated in Figure 5c. For increasing the canvas' distance to the camera and performing an appropriate scale, the projection appears unchanged in the eye-point of the camera. However, these transformations do have effect on the misalignment seen above:

- For the first type of view displacement (a), at a certain depth the applied scaling compensates the scale caused by the perspective. Hence, for background geometry at this depth the projection regains alignment.
- In case of an additional displacement corresponding the second case (b), increasing the distance to the canvas reduces the resulting angle γ' and thus, reduces the perspective shift against the background.

Applying these canvas transformations can not fully compensate the resulting misalignment for every viewpoint other than the camera's eye. Even worse, this simple approach also introduces new and undesirable effects. These view dependent transformations try to achieve a fit even for viewpoints that differ just too much from the original camera location. Such canvas placement could be counterproductive and hamper the overview by cluttering the scene or make the projections intersecting each other. The following sections describe our approach to overcome these problems.

3.5 Viewing Multiple Canvases

Figure 6a outlines a more complex situation in which the user's view is located somewhere in-between two neighbouring camera views. In this setting, the cameras are pointing to the same object which makes their optical axes adjusted inwards. Unlike for parallel optical axes, during a straight-line view transition between the camera locations, a point projection of the intermediate view onto each of these lines results in a displacement according to type a (see Figure 5c). Because of equation 1 applying the norm of the displacements σ_a and σ_b this yields the same canvas transformation as discussed above. Such transformed canvas objects are likely to intersect with others or with background geometry. The latter case of collision can be corrected easily by changing the rendering order. Overlapping projections however occlude valuable information and might cause confusing views. Hence, it appears sensible to perform an alpha-blend for these projections and to fuse all canvas information that is available for an intermediate view.

3.6 Setting Canvas Focus

In Figure 6a, the main object is shown both as a visible 3D model and on both camera images. However, a 3D model of this object is usually not available in the 3D environment so only two camera images are shown, see Figure 6b. As illustrated, the projection canvases intersect and, for the given *intermediate view*, both canvases are visible at the same time. Both cameras are oriented inwards and take pictures from different sides of the object. Especially when combined with projection blending, the object appears twice, projected for each of the two camera viewpoints. Therefore, watching the pictures from such intermediate point of view produces a virtual cross-eye effect that "*doubles*" information and thus, might lead to confusion.

Allowing such intermediate views aims at giving the user a chance to perceive the spatial relation of the camera locations even by means of the smooth transition. Nevertheless, with the described methods for correcting misalignment and blending intersections such intermediate viewpoints still result in strong image distortions and interfering projections. One of the main reasons for this is the visualisation's unawareness of the environment. The canvas transformation does place the projections in a view-dependent way, but at a default distance from each camera. In the example of Figure 6b the red lines indicate the axes where the projections converge if the canvases were calibrated. Hence, increasing the distance to the camera further for each of the projections would yield convergence for these axes, see Figure 6c. This would eventually arrange the projections at the virtual location that relate with the real world position of the object.

Such calibration has at least two direct consequences for the sketched example. Firstly, even for intermediate views the cross-eye effect shows both object projections aligned at these axes, which results in blending them to a single object instead. Secondly, as the axes converge to the same location in the virtual space, the corresponding parts of the projections get invariant to motion parallax during the whole view transition. In case this invariant axis complies with some characteristic edge in the projected picture, as in this case, the stem of the plant, this gives the user additional indication on how the projection changes along the path. It is important to notice that the convergence of image projections is an effect which is caused by giving an additional hint about the geometry of the surrounding environment. Such knowledge can not be assumed for the general case. However, if there is more information avail-

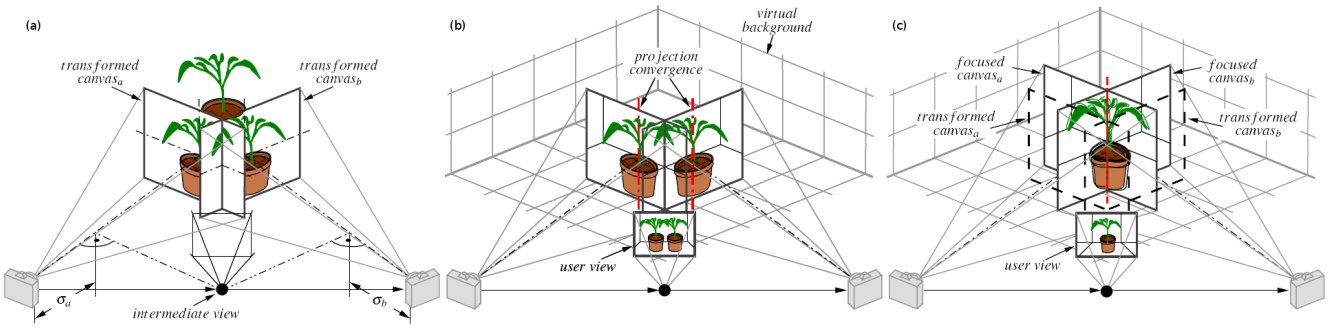

Figure 6: Transition view between camera A and camera B looking at a single object. With view dependent embedding, canvases can intersect and overlap (a). Dynamic blending of translucent canvases can compensate for overlap, although cross-eye effects may occur (b). If the view dependent embedding is supplied with a focus point, the canvases can be aligned as good as possible (c).

able about the scene and the distances at which the cameras focus at, it is possible to give such hints and obtain correct transitions for objects at this depth. For instance, in situations of neighbouring cameras, the spots often share the view to at least some distinctive points or edges. By using these common points for calibrating the canvases *focus depths*, it is possible to reduce misalignment effects and to get intermediate projections much more consistent.

4 User Navigation

The visualisation technique described in the previous sections binds the rendered information firmly to certain locations within a three-dimensional representation space. Thereby, this method aims to present the captured pictures related to the spatial context of their real site origin. To preserve this context, the visualisation system has to avoid an arbitrary rearrangement of these images and to prevent the user from inspecting the information out of this logical order. In combination with the egocentric view onto the representation space, selecting a particular piece of information comes down view navigation in 3*D* space.

Navigating freely in a 3D virtual environment might lead to awkward views and disorientation of the user. Various studies[3] have examined this problem and presented several variations of *guided navigation* to avoid such situations. One approach is to restrict the degrees of freedom over which the user has direct control. For providing the needed mobility instead, these methods offer a predefined set of automated manoeuvres for controlling the view.

The proposed visualisation system also approaches the strategy of *guided navigation* by gradually granting and refusing *permissions* for view navigation. The main navigation style is a guided movement over a trajectory along a set of cameras, controlled through simple mouse movements along the x-axis.

These permissions are used to map configurable input axes to certain view transformations. Common examples for input axes might be dragging the mouse or a progressing time-value (e.g. for automated transitions). Furthermore, we can apply additional smoothing, for example by *slow-in/slow-out*, before generating an appropriate response transformation.

Specifying user navigation via permissions also allows us to merge *constraint trajectories* and *free navigation* by just considering them as two different permission sets. An example of this is to allow for free view rotations during guided translations.

Such permission sets are composed to conform the user's needs in different situations. This permission type considers the differences between two camera views and creates a specific manoeuvre of DOF modifications by which to transform the viewpoint. By determining a specific view which to approach and using the current position as the start, the input for navigating among these locations reduces to *forward* and *backward* along one input axis.

5 The Context Graph

The meta information of the camera images is the most important input for the visualisation and guided navigation algorithms. We propose a *context graph* that provides this knowledge during runtime and thus, helps to abstract from the complexity of the surrounding geometry. The following list gives an overview of the data that the context graph nodes keep available during runtime.

- camera properties (position, attitude, perspective)
- captured visual data (e.g. photo, video stream)
- spot meta data (e.g. time stamp, description, notes)
- list of all contexts this spot is part of

In addition, this data structure links subsets of cameras within task specific *contexts*. Such contexts allows one to separately define the type of visualisation and navigation for a specific group of cameras. The context graph is therefore defined as a *directed graph* that comprises the set of cameras as nodes and possibly multiple layers of edges that represent the camera relationships. These layers, which correspond with the contexts named above, are kept apart to allow the iteration over a specific group of camera locations. For the example of navigation, this allows to define camera trajectories along these camera links and to provide the algorithms with optimised parameters for guiding the user among the cameras of such context.

Analysing camera configurations, often found for crowd observation reveals relations in the arrangement between the individual spots. For instance, the cameras might observe the same object, seen from different directions or form a sequence that follows the course of a corridor. Hence, the cameras can be grouped by patterns that reflect these common alignment properties (see Figure 7). The patterns define an ordered subset of camera spots, whereby each camera gets a neighbourhood of a predecessor and a successor that share a certain alignment. In the context graph, these patterns appear as camera contexts for which we can define specific visualisation and navigation. An example of this is providing a smooth navigation along stitched camera images of a panoramic view.

6 Prototype Description

We developed a prototype 3D visualisation application for video surveillance, in which we implemented the proposed techniques. This 3D visualisation application and the interface to configure the provided functionality are described first. We then describe the surveillance scenarios which provide insight in the results of our techniques.

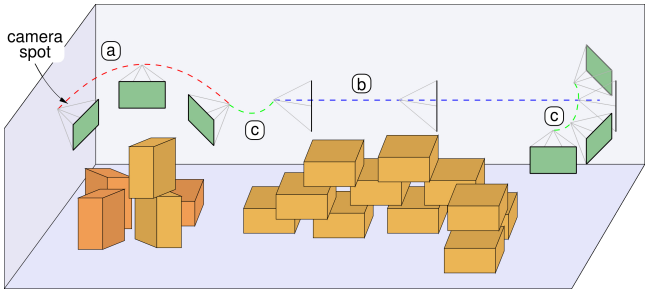

Figure 7: Camera spots are grouped into one or more patterns. Pattern (a) denotes a subset of cameras, observing the same scene from different angles, while (b) describes a camera sequence along a wall and (c) refers to a camera pattern with a panoramic view.

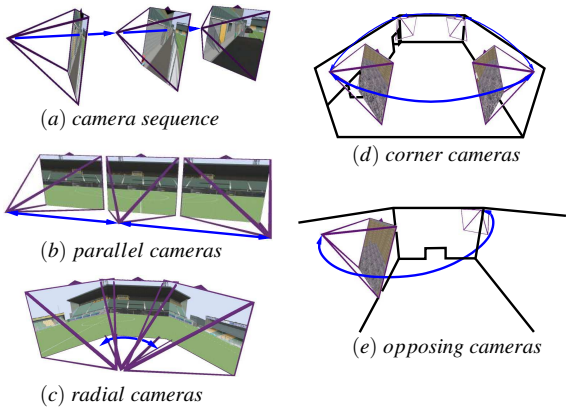

Figure 8: Common camera arrangements for monitoring different areas.

6.1 Implementation

The implementation of our visualisation application is based on the OpenSceneGraph library, which provides comprehensive support for low level rendering and prepares ground for further development. For a specific environment, the visualisation functionality is configured via an XML description file. Within this file, an environment designer specifies the visualisation behaviour and the specific navigation interface. The specification of available camera spots includes their perspective settings, references to associated video sources, and further meta information. Declarations of different contexts form the layers of the context graph that link the provided camera nodes to groups. For each of these groups, the designer specifies a set of algorithms and settings. According to these properties, the application decides on which rendering technique to apply for representing the image or meta data kept in the nodes. Furthermore, a configuration of such context for applying a navigation algorithm enables the user later to "visit" the corresponding camera locations and to navigate along the trajectories specified for these views.

The data associated with each of the configured camera spots can be visualised in various ways. For rendering image data, the application currently supports still images as well as video sequences, which are kept either on disk or streamed over the network. Furthermore, functionality is provided for rendering views of only the 3D model to texture, which allows the designer to define purely virtual camera spots. Such renderings of the virtual environment make it possible to test camera placements without having real footage available. Displaying descriptions or other extra information can be integrated by overlay renderings, such as text or glyph objects. This method renders the provided information in separate layers which appear blended in the user's view.

The navigation algorithms configured for a certain camera context use the spot information to define the trajectories along which to transform the user's view. These trajectories interpolate the camera settings by blending them either linearly or following the characteristics of given functions to transform the camera attributes individually. Smoothing functions such as *slow-in/slow-out* are used to control the acceleration of the view transition. As the trajectories always consider the current information that is provided for a camera, the application can also handle the navigation among cameras that are adjustable in their settings, such as pan-tilt-zoom cameras.

For testing the concept on the basis of virtual camera views, we defined animated objects which are only visible in the "*recorded videos*". This can be used to simulate population in the area and to reproduce surveillance scenarios, such as following persons and default round tours throughout the environment.

6.2 Surveillance Scenarios

We applied our proposed techniques on different surveillance scenarios to get insight in their properties. The three real scenarios used are a soccer stadium, an airport departure hall and an office hallway. Aside from legal and security issues, we experience that in practice it can be difficult to obtain and use both real video footage and 3D models of actual surveillance sites. Therefore, most prototyping is done on pure virtual environment models, augmented with synthetic video created with the pre-rendering functionality.

Original camera planning documents of several of the surveillance sites provided insight into the general strategies for camera arrangements. Different arrangements are used to monitor a greater variety of environment layouts, such as long narrow corridors, locations with highly repetitive geometry or those with missing landmarks. Also, alternate camera types such as wide angle cameras and pan-tilt-zoom cameras were taken into account. Figure 8 illustrates some of the camera arrangements we use in our environment models. These general arrangements were first modelled in a 3D mock-up virtual environment as virtual cameras and accompanying guided trajectories. This allowed us to recreate common surveillance scenarios and to experiment with various parameters of navigation and rendering.

The concept for rendering camera spots aims to allow configurations for any given environment. Hence, the framework provides various options for adjusting the layout of rendered video images. Due to the view-dependent rendering, the appearance of the projected images is strongly related to the methods for guided navigation. Controlling the fading of images and setting the projections' focus are only two of the options for tuning the results for intermediate views in transfers. For a configuration to visualise and to navigate through a given environment we found different measures for evaluating a setup. Aside from assessing the support for user's orientation and guided mobility, we emphasise assumed incident scenarios and following objects throughout the site. The results gained by such setup evaluation could help for planning operator rounds within the virtual monitoring system and configuring the location sequences for the guided navigation.

In the ***stadium scenario***, four video cameras standing several meters apart, monitor crowds at the opposite side of the stadium, at a distance of approximately 50 meters. Figure 9 gives an overview of the video streams, the 3D model and one example of a final embedding. The 3D visualisation application streams the video from a network server, set up to mimic realistic networked camera streams. As accurate camera calibration information was absent, the cameras were manually positioned in the 3D model to provide necessary spatial context. This led to the discovery that some of the dimen-

Figure 9: In the stadium scenario, four video streams (left) are spatially arranged within a 3D model (middle). The video streams are integrated such to provide a stitched view (right). The first-person view allows for integrated information in the 3D model, a heads-up display and the use of various known egocentric interaction metaphors.

sions of the 3D model did not correspond to the real-world environment. Nevertheless, a regular setup of the guided trajectories leads to a comfortable fly-through along the camera views. As the distance to the crowds was relatively large compared to the distance of the cameras, we also configured a panoramic fly-through trajectory, see Figure 9, right image. By showing all four videos canvases simultaneously at the correct distance, this configuration effectively stitches streams together to form a single panoramic overview. Visible alignment artifacts are a result of known errors in the dimensions of the modelled 3D environment and inaccurately positioned cameras.

The second scenario deals with observing incidents in ***airport departure hall***. Four cameras are used to observe a target location from varying angles and distances where incidents were enacted. More specifically, this set of recorded videos is used for performance evaluation of automated detection algorithms. From the available camera calibration data and marker positions, camera positions were determined in 3D world coordinates. It proved difficult to grasp the spatial relationships when viewing only camera images spatially arranged in an empty 3D environment. Therefore, a simple and roughly aligned abstract 3D model of the environment was reconstructed in a 3D modelling application. Characteristic landmarks, such as a column or the staircase, and vertical and horizontal lines provide strong hints on perspective and create strong visual flow and maintain spatial context during transitions, see Figure 10. In long or slow transitions, for example from cameras standing far apart while facing in opposite directions, it can still be hard to maintain spatial context. In this setup, it would be beneficial for the spatial context to use more intermediate cameras to avoid long trajectories and large angles. To counteract some of the effects, we use curved view trajectories and zooming out during transitions to keep sufficient 3D model information on screen.

The third scenario deals with the observation of an ***office hallway*** in which we use five camera streams, either live or recorded. A simple 3D model was reconstructed from old CAD drawings and camera parameters were estimated and roughly aligned to fit the model. In Figure 1 and the second accompanying video, an interactive first-person session of the final result is shown. Here, the operator interactively follows two of our colleagues as they walk along the hallway. The context graph is defined to link the camera views from left to right, the operator uses simple mouse dragging to indicate travel direction. In this scenario, it becomes clear that the small translations and angle changes between cameras are beneficial to fast and comprehensive transitions. Part of current work is to support more complex navigation trajectories through better path selection, e.g. to follow someone down the stairs.

7 Evaluation

We used formative evaluation throughout the development process of our prototype system and the construction of the diverse scenarios, including hypothetical, virtual scenarios. The prototype application was demonstrated and used by approximately 10 people with computer graphics background. Screencasts of the interface showing the visual transitions were shown to approximately 50 people with backgrounds in video surveillance. In general, observers have the impression that guided viewing along different camera streams integrated in the 3D model does indeed provide a surprisingly compelling sense of spatial context and coherence between videos and the 3D environment.

The effectiveness of individual transitions is dependent on the trajectory speed and the pattern, focus, placement, and orientation of the cameras. Observers noted that some transitions were hard to grasp because they were too fast, had too large displacement or angle change, or lacked in reference landmarks in either videos or 3D model. Early evaluation led to the wide array of settings for fine-tuning transition parameters for dealing with specific camera configurations. These parameters include navigation and transition speed, blending factors and focus placement. We noted that after transitions were repeated only a few times, observers could cope with more difficult situations at higher navigation speeds. Surprisingly, the quality of the 3D model and the camera alignment to the 3D model does not seem to have a significant influence on spatial cues.

Finally, we observed that, in contrast to matrix displays, in our egocentric interface the use of more and densely placed cameras is beneficial for the spatial context. This allows us to decrease the time every video is shown and to increase the speed of comfortable navigation. A more formal evaluation of these findings requires carefully designed user experiments and is part of future work.

8 Conclusions and Future Work

We presented an egocentric 3D interface for video surveillance monitoring. The dynamic embedding method places the video streams on canvases in the abstract 3D model. The operator can navigate through the 3D model, guided via a trajectory along camera views. The use of the context graph allows for rapid definition and querying of camera information, patterns, transitions and trajectories. During guided navigation, multiple video streams are selected, transformed and blended to provide a smooth visual transition between views. When distortions are too large during transitions, the camera views are fully transparent while the motion perspective cues of the 3D model provide the necessary spatial context. The implemented prototype demonstrates the results of the interface on various scenarios with both synthetic and real cameras, arranged in different camera patterns and environments. Given a simple, abstract 3D environment model and a context graph, the prototype can be used directly when connected to already available video streams.

The original goal of our work was to provide an interface for improved situational awareness and to provide a basis for egocentric interaction techniques. The proposed egocentric 3D interface should not be seen as a replacement but as a valuable addition to

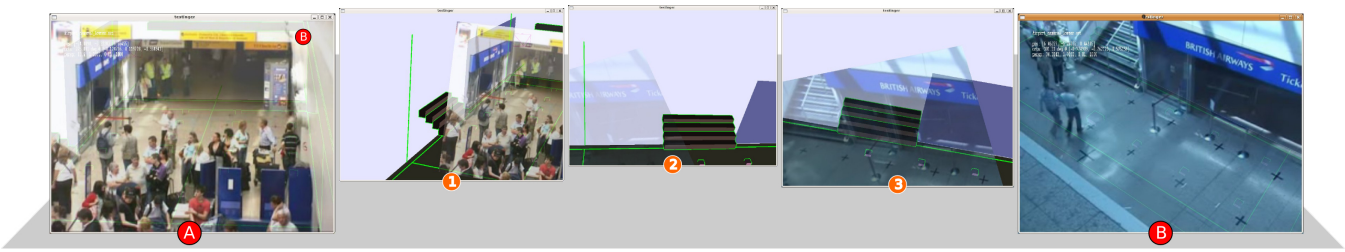

Figure 10: View transition sequence between camera A and opposing camera B in an airport departure hall scenario. Even though most transitions involve a large change in camera position and viewing angle, the use of dynamic video embedding and characteristic landmarks keeps the visual experience fluent (see video).

current practice in the surveillance of complex, crowded environments. As exocentric displays such as the video matrix display remain indispensable to keep global environment overview, our interface focuses on interactive observation and coherence on a more local level. Our results indicate that guided egocentric viewing along dynamically arranged canvases does indeed provide a compelling sense of coherence between videos and the 3D environment. In contrast to projective texturing techniques, the quality of the 3D model and integrated camera alignment does not seem to have a significant influence on spatial cues. We note that the quality of individual transitions depends on the trajectory and the pattern, focus, placement, and orientation of the cameras. Sufficient visual flow and spatial context can only be reached with enough density and overlap of camera views. Given the trend in surveillance of an increasing number of densely placed cameras, it is important to observe that in a first-person metaphor this appears to be beneficial for the overall spatial context, which is in contrast with regular matrix displays.

The main egocentric interaction technique that we provide is guided navigation, which provides smooth transitions along sets of camera views with little or no effort. Our prototype implementation also facilitates the integration of other new egocentric 3D interaction techniques. Our current work therefore focuses on live annotation, interactive camera selection and indicating regions of interest, all directly within video streams. A hybrid interface allowing smooth transitions between pure egocentric and exocentric representations is work in progress. Also, the direct integration of the control of other sensors such as pan-tilt-zoom cameras, microphone arrays and high-resolution cameras is part of future work. Finally, we see great potential in augmenting video streams, 3D models and additional 3D information overlays, e.g. from automated tracking systems. When combined, 3D interfaces in these *mixed-reality* environments may help restore the productivity of surveillance operators under the increasing load of video data.

Acknowledgements

Part of this research has been funded by the Dutch BSIK/BRICKS project. The airport departure hall video sequences are part of the PETS 2007 benchmark dataset.

References

[1] B. B. Bederson and A. Boltman. Does animation help users build mental maps of spatial information? In *Proc. IEEE INFOVIS '99*, page 28, 1999.

[2] D. A. Bowman, D. Koller, and L. F. Hodges. Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. *Proc. VRAIS '97*, 00:45, 1997.

[3] T. A. Galyean. Guided navigation of virtual environments. In *Proc. SI3D '95*, pages 103–104, 1995.

[4] A. Girgensohn, D. Kimber, J. Vaughan, T. Yang, F. Shipman, T. Turner, E. Rieffel, L. Wilcox, F. Chen, and T. Dunnigan. DOTS: support for effective video surveillance. In *Proc. MULTIMEDIA '07*, pages 423–432, 2007.

[5] A. Girgensohn, F. Shipman, A. Dunnigan, T. Turner, and L. Wilcox. Support for effective use of multiple video streams in security. In *Proc. VSSN '06*, pages 19–26, 2006.

[6] A. Girgensohn, F. Shipman, T. Turner, and L. Wilcox. Effects of presenting geographic context on tracking activity between cameras. In *Proc. CHI '07*, pages 1167–1176, 2007.

[7] Y. Ivanov and C. Wren. Toward spatial queries for spatial surveillance tasks. In *Pervasive: Workshop on Pervasive Technology Applied to Real-World Experiences with RFID and Sensor Networks*, 2006.

[8] U. Neumann, S. You, J. Hu, B. Jiang, and J. Lee. Augmented virtual environments (AVE): dynamic fusion of imagery and 3d models. *Proc. IEEE Virtual Reality*, pages 61–67, Mar. 2003.

[9] H. S. Sawhney, A. Arpa, R. Kumar, S. Samarasekera, M. Aggarwal, S. Hsu, D. Nister, and K. Hanna. Video flashlights: real time rendering of multiple videos for immersive model visualization. In *Proc. EGRW '02*, pages 157–168, 2002.

[10] I. O. Sebe, J. Hu, S. You, and U. Neumann. 3d video surveillance with augmented virtual environments. In *Proc. IWVS '03*, pages 107–112, 2003.

[11] M. Segal, C. Korobkin, R. van Widenfelt, J. Foran, and P. Haeberli. Fast shadows and lighting effects using texture mapping. In *Proc. SIGGRAPH '92*, pages 249–252, 1992.

[12] H. Shum and S. Kang. A review of image-based rendering techniques. *IEEE/SPIE Visual Communications and Image Processing (VCIP)*, 213, 2000.

[13] N. Snavely, R. Garg, S. M. Seitz, and R. Szeliski. Finding paths through the world's photos. *ACM Transactions on Graphics*, 27(3):11–21, 2008.

[14] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *Proc. SIGGRAPH '06*, page 42, 2006.

[15] P. Thorndyke and B. Hayes-Roth. Differences in spatial knowledge acquired from maps and navigation. *Cognitive Psychology*, 14(4):560–589, 1982.

[16] Y. Wang, D. Bowman, D. Krum, E. Coelho, T. Smith-Jackson, D. Bailey, S. Peck, S. Anand, T. Kennedy, and Y. Abdrazakov. Effects of video placement and spatial context presentation on path reconstruction tasks with contextualized videos. *IEEE TVCG*, 14(6):1755–1762, 2008.

[17] Y. Wang, D. M. Krum, E. M. Coelho, and D. A. Bowman. Contextualized videos: Combining videos with environment models to support situational understanding. *IEEE TVCG*, 13(6):1568–1575, 2007.