

# Ein Fenster in die Wirklichkeit

## A Window into Reality

Hynek Bakstein<sup>1</sup>, Federico Diaz<sup>2</sup>, and Tomáš Pajdla<sup>1</sup>

<sup>1</sup>Center for Machine Perception, Dept. of Cybernetics,  
Faculty of Electrical Eng., Czech Technical University in Prague,  
Karlovo nám. 13, 121 35 Prague, Czech Republic  
tel.: +420 2 2435 7637, fax: +420 2 2435 7385

e-mail: {bakstein,pajdla}@cmp.felk.cvut.cz, web: <http://cmp.felk.cvut.cz>

<sup>2</sup>E-Area, Marie Cibulkové 30, 140 00 Prague 4, Czech Republic  
e-mail: federico@e-area.cz

### Zusammenfassung

Dieser Artikel beschreibt ein komplettes System des Aufnehmens, der Bearbeitung und der Visualisierung realer Rechnerumgebung. Der Benutzer kann diese Umgebung durch ein virtuelles Fenster in der realen Welt beobachten. Dieses Fenster wird durch den Rechnerbildschirm oder durch den Projektionsschirm präsentiert. Die Position des Benutzers muss beobachtet sein. Verschiedene Applikationen stellen das digitale Kulturerbe, die Unterhaltung, die interaktive Kunst und Spiele dar. Dieser Artikel beinhaltet vor allem die Beschreibung der Aufnahmeeinrichtung, das Prinzip der Darstellung und die nötige Vorbearbeitung des Bildes

### Abstract

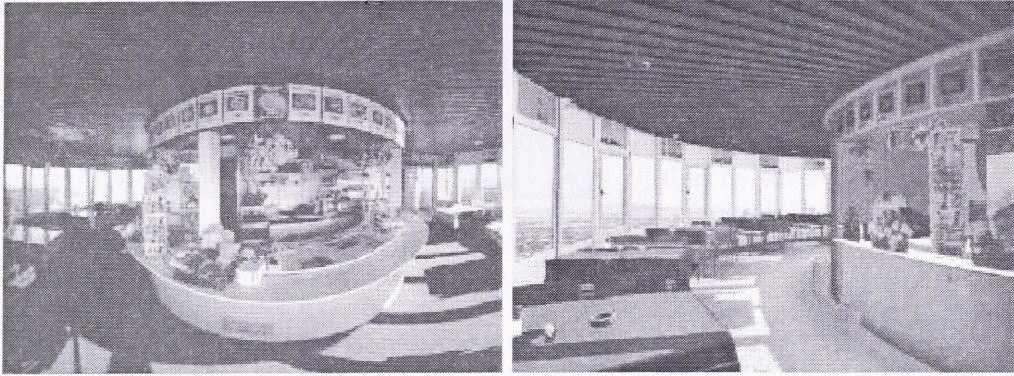
We present a complete acquisition-processing-visualization system to represent real environments in a computer. A user can observe the environment through a virtual window into a real world. This virtual window is generated by a computer on a projection screen. The user motion is tracked, and the views through the window are generated in real time to correspond to his actual position. Possible applications include interactive exhibits of cultural heritage, entertainment, interactive art, and games. The paper focuses on technical description of the acquisition system, the visualization principle and necessary preprocessing of the acquired images.

### 1. Introduction

A computer based representation of a real environment can be used to create illusion of a real world. This representation can be based on a 3D model, but this requires a powerful graphics hardware for visualization and extensive manual work to construct the 3D model. Although modern computer games, working as a showcase of real-time 3D rendering, take place in very convincing 3D environments, it is very easy for a human to distinguish it from reality. Especially organic structures like trees look still artificial when rendered by the computer, not mentioning complex lighting issues.

On the other hand, image based rendering (IBR) methods [7] allow the users to move in an environment represented by a collection of images, giving a photorealistic impression. The users can observe the scene also from positions different from the locations, where the camera acquired the images in the collection. Thus, a relatively small number of images can be used to represent a large number of viewpoints. The scene can be arbitrarily complex, the IBR methods do not require any knowledge of the scene geometry or a very rough approximation of the scene depth is sufficient. The user has a freedom of movement in some specified region of exploration (REX). Moreover, the IBR methods do not require any special rendering hardware. Their main

disadvantage is a huge memory requirement, but today's computers are already equipped with sufficient memory capacity. See Figure 1 for example outputs from IBR and 3D model methods.



**Fig. 1** Comparison between image based rendering (left) and a 3D model (right).

This paper focuses on a method of implementing a virtual window, where a computer or projection screen serves as the window and a user can move freely in front of the window. Provided that the user motion is tracked, the screen displays an image of a real world as if observed through a real window. This paper gives an introduction to the technology involved, the reader is referred to relevant literature for more technical details.

The rest of the paper is organized as follows. Section 2 describes the idea of a virtual window. Section 3 presents the IBR method employed for the realization of the virtual window. Practical realization of the virtual window is described in Section 4.

## 2. Virtual window

Consider the following scenario. A person walks in front of a computer screen, mounted on a wall. The picture on the screen changes as the person moves. When he goes closer, he sees more of the scene, sideways or vertical motion uncovers hidden parts of the scene and occlusions change realistically. The scene is rendered in a photorealistic quality, with correct lighting and shadows. Everything moves smoothly. Figure 2 is an illustration of the above principle and Figure 3 shows some example viewing direction from an implementation of our method presented at CeBit 2005.

It is clear that the photorealism and real-time rendering vote in favour of the IBR methods. Instead of manual creation of a 3D environment, IBR methods allow scene representation by a collection of photographs. The data acquisition is then only a question of taking a sufficient number of images on a specified trajectory. Because the camera moves during the image acquisition, scene objects are observed from different viewpoints and this enables the IBR method to model the occlusions. To give the viewer a complete freedom of movement, we propose to capture the environment using an omnidirectional camera, e. g. a fish-eye lens. Then, when the viewer moves close to the screen, we have information in all viewing directions. Fish-eye lenses can be easily calibrated [2].

As a drawback, the IBR methods require a lot of memory and the lighting conditions remain fixed, as opposed to the 3D models, where the lighting can be arbitrary. On the other hand, the lighting is realistic including shadows and reflections, effects which can be hardly modelled in real-time. The next section describes in detail the IBR method we have chosen for the virtual window realization.



Fig. 2 Illustration of the virtual window.

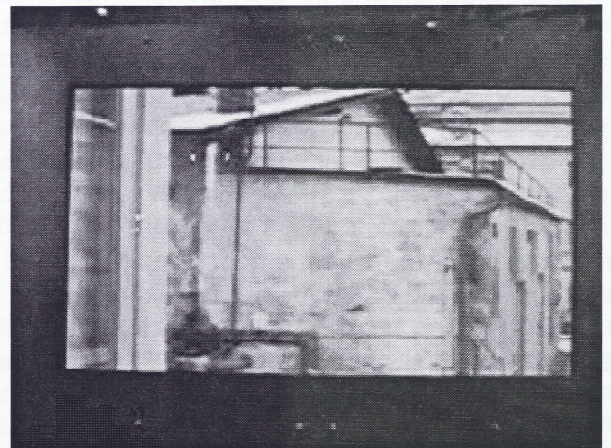
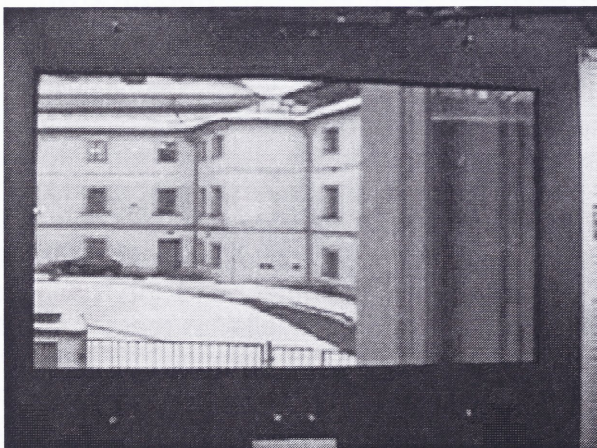
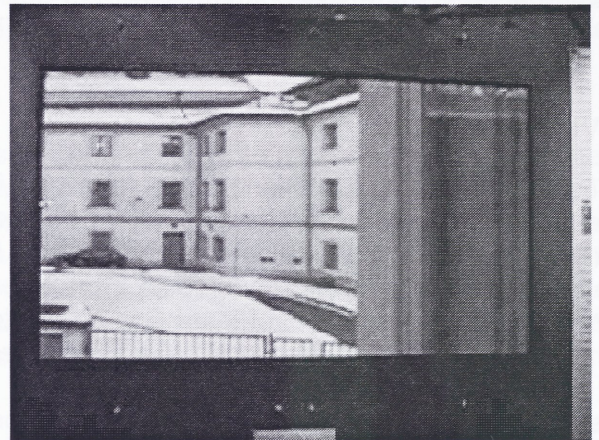
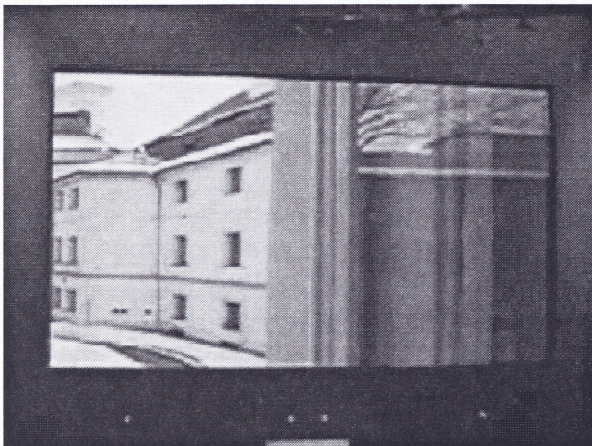


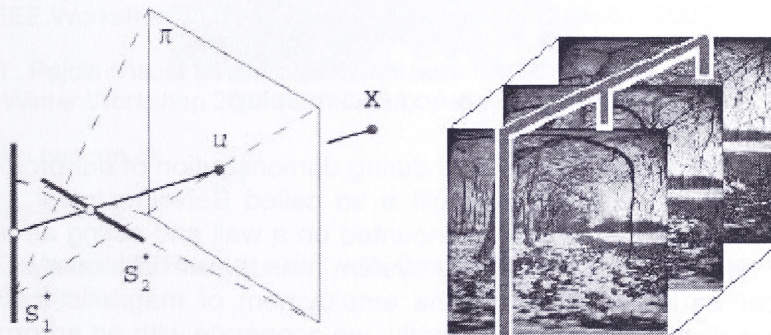
Fig. 3 Selected frames from a virtual window demonstration demonstrating the changes in the viewing direction and corresponding virtual images.

### 3. X-slits rendering

The method used for an implementation of the virtual window is called X-slits [9]. Let the camera move on a linear path, looking in perpendicular direction. The camera captures images in regular steps. It should be noted that motion and intervals can be irregular [1], but regular motion and intervals can be easily achieved and the practical implementation is much simpler. The input images are organized in a so called image volume, which is just a collection of images as they were acquire in time, see Figure 4 right. It was shown in [9] that by arranging the columns of the volume images into virtual images, so that the columns are selected by some linear function, a virtual image corresponding to a position separate from the input path is created. Every linear function defined on the columns corresponds to such an image.

More formally, let the input path be denoted by  $S_2$ . Then, linear selection of the image columns choses light rays intersecting the line  $S_2$  and also a line  $S_1$  perpendicular to the line  $S_2$ . These two lines are called slits, and the position of the vertical slit  $S_1$  corresponds to a position of a virtual camera. The slits are depicted in Figure 4 left. Each column in the volume images contains light rays with the same angle  $\beta$  between the light rays and the middle column, where  $\beta=0$ . Then, for a column in the virtual image  $V$ , we take the same column in an image  $A_i$  from the input volume. The position of  $A_i$  depends linearly on the angle  $\beta$ , see Figure 5. There is no extensive computation behind this principle and therefore it can be implemented in real-time on common PCs.

The rearrangement of the columns simulates the occlusion changes, but forward and backward motions also require the scene objects to become larger or smaller in the virtual images. For this, some scaling of the columns has to be defined [9]. It is out of scope of this paper to describe this scaling in details, but we should note that this scaling is correct only for objects in a certain, user defined, depth range. Objects closer or more distant that this depth range have different aspect ratios. Methods for approximating the scene depth are discussed in [6]. For most of the scene containing natural objects, this effect is hardly observed by people [3]. The same scene depth approximation is used to reduce the number of the images in the image volume [2].



**Fig. 4** Left - virtual views are composed from rays crossing two „slits“, the camera path and a vertical line corresponding to the position of the viewer. Right - each virtual image corresponds to a linear cut through a volume of input images.

An important issue is the number of images that has to be acquired for a given virtual window. The window is represented by some computer screen with a given horizontal resolution. Looking at the Figure 5, one can observe that we always take columns from images at positions  $A_i$ . These positions correspond to columns in the virtual image. Linear function between the columns and these positions give us a rule of thumb that the number of images in the volume should be the same as the number of columns in the virtual image. For most computer screens this means 1024 images.

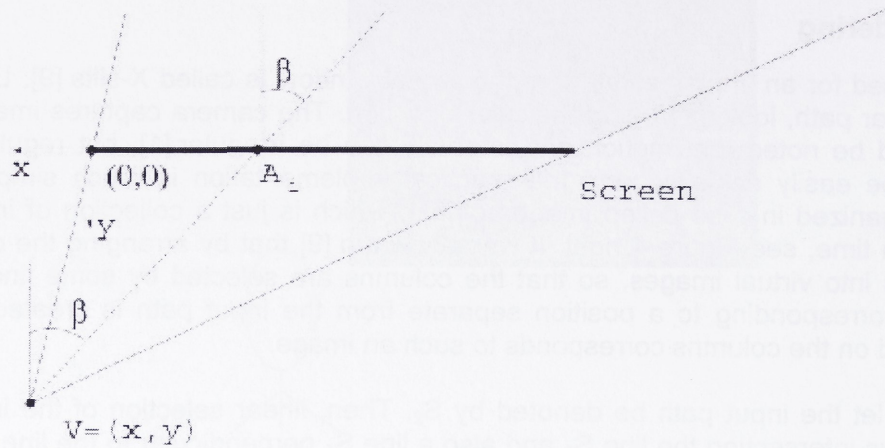


Fig. 5 X-slits principle.

#### 4 Practical realization

In the scope of the 5-th framework EU project BeNoGo [4], we have assembled an acquisition setup, which can capture high resolution omnidirectional images on a linear path up to 6 meters long. The setup is depicted in Figure 6. It employs a Canon 1Ds digital camera with 11Mpix resolution equipped with a Sigma 8mm fish-eye lens providing 180 field of view. The camera is moved using a motorized platform on a rail, assuring regular intervals between images and a linear motion.

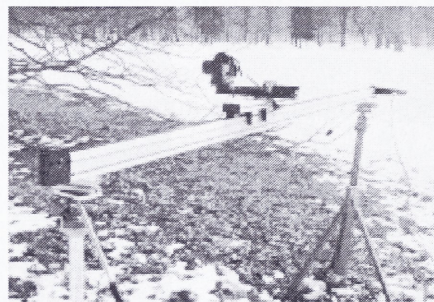
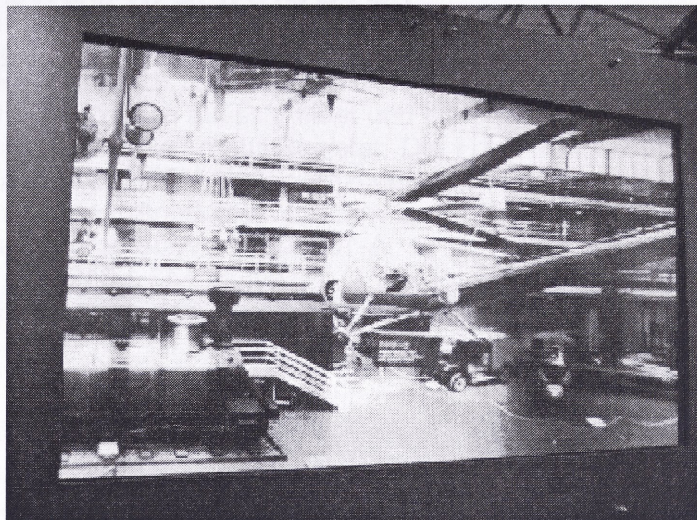


Fig. 6 Acquisition setup

Images captured using this device were used during demonstration of our project at the CeBit 2005 fair. Our partner, the Aalborg university, built a so called BeNoGo tower, which, among other demonstrators, contained a plasma screen mounted on a wall and acting as a window. The tower and the screen can be seen in Figure 7. The system runs in real time on a common PC with 2GB of memory. Practical issues arising from the employment of magnetic tracker and a study of perceptual aspect are discussed in [8]. Presently, we cooperate with an art group E-area [5] on an interactive installation employing the principle of a virtual window and computer generated contents.

#### 5 Conclusion

We have presented a concept of a virtual window, where a computer screen acts as a window into a real world. The images presented on the screen are created from a set of omnidirectional images acquired on a linear path using an IBR method called X-slits. This allows real-time photorealistic rendering of complex natural scenes. Practical realization of the proposed virtual window exists and was exhibited at the CeBit 2005 fair. Possible applications include interactive exhibits of cultural heritage, entertainment, interactive art, and games.



**Fig. 7** Left - the BeNoGo tower at CeBit 2005. Right - the plasma screen showing the technical museum in Prague.

## Acknowledgements

This work was supported by the following projects: BeNoGo IST-2001-39184, Kontakt ME 678, STINT Dur IG2003-2 062, project MSM 6840770013 and project MSM6840770012.

## References

- [1] A. Rav-Acha, Y. Pritch, and S. Peleg, Online Video Registration of Dynamic Scenes using Frame Prediction, Hebrew University Technical Report HUJI-CSE-LTR-2005-78, June 2005.
- [2] H. Bakstein and T. Pajdla Panoramic mosaicing with a 180 field of view lens. In A. D. Williams, editor, Proceedings of the IEEE Workshop on Omnidirectional Vision, pages 60-67, 2002.
- [3] H. Bakstein and T. Pajdla Visual fidelity of image based rendering. In D. Skočaj, editor, Proceedings of the Computer Vision Winter Workshop 2004 (CVWW04), pages 139-148, 2004.
- [4] BeNoGo. <http://www.benogo.dk>.
- [5] E-area. <http://www.e-area.cz>.
- [6] D. Feldman and D. Weinshall, Real-time IBR with Omnidirectional Crossed-Slits Projection, Accepted to ICCV'05, October 2005, Beijing.
- [7] H.-Y. Shum and S. B. Kang. A review of image-based rendering techniques. In IEEE/SPIE Visual Communications and Image Processing (VCIP) 2000, pages 2-13, June 2000.
- [8] J. Weikop, R. Stenholt, R. E. Laursen, and E. Granum, Creating a Virtual Window using Image Based Rendering, Proceedings: 8th Annual International Workshop on Presence (PRESENCE 2005), London, United Kingdom, September 2005.
- [9] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection. IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(6):741-754, June 2003.