

# Floor-Plan Reconstruction from Panoramic Images

Dirk Farin  
Univ. of Technol. Eindhoven  
Signal Processing Systems  
5600 MB Eindhoven  
Netherlands  
d.s.farin@tue.nl

Wolfgang Effelsberg  
Univ. Mannheim  
Dept. Computer Science IV  
68159 Mannheim  
Germany  
effelsberg@informatik.uni-  
mannheim.de

Peter H.N. de With  
LogicaCMG and  
Univ. of Technol. Eindhoven  
5600 MB Eindhoven  
Netherlands  
p.h.n.de.with@tue.nl

## ABSTRACT

The capturing of panoramic 360° images has become a popular photographic technique. While a panoramic image gives an impressive view of the environment, many people have difficulties to understand the spatial scene arrangement from this flat image. In this paper, we present a new visualization technique for panoramic images based on a coarse reconstruction of the indoor environment, in which the panorama was captured. Applications of this are, for example, real-estate or hotel advertising, featuring virtual tours through the apartment. We use a semi-automatic reconstruction process, in which the user marks the room corners in the panoramic images. These can be translated into viewing-angle measurements, from which our algorithms can compute the exact sizes of the walls, based on a pre-defined geometric model.

## Categories and Subject Descriptors

I.4.8 [Image Processing and Computer Vision]: Scene Analysis; I.2.10 [Artificial Intelligence]: Vision and Scene Understanding3D/stereo scene analysis

## General Terms

Algorithms

## Keywords

3D scene reconstruction, panoramic images.

## 1. INTRODUCTION

The capturing and display of panoramic 360° images has become a standard technique which is used in a variety of applications. The most frequently-used model for capturing panoramic images is to project the environment around a fixed camera location onto a virtual cylindrical surface. There are two commonly used ways to display these panoramic images. The first approach shows the unrolled surface of the cylinder in a very wide image. Since this image shows all directions around the camera at the same

time, the panoramic image can be very confusing to the viewer. Another popular form of presentation is through an interactive viewer. These *Panoramic Image Browsers* (PIB) show a rectified sub-view of the scene, where the user actively controls the viewing direction of a virtual camera. The disadvantage of this representation is that it is not possible to get a fast overview of the scene, and it is not possible to see the complete environment on a static medium like a paper copy.

In [5], different projections of the whole panoramic image are compared with respect to the visual quality for the viewer. These include projections known from geography, such as the Mercator projection. They further propose to use a perspective multi-plane projection, where walls are projected onto different planes. This eliminates geometric distortion and results in a more pleasing viewing experience than the traditional panoramic images with bent lines. However, for 360° panoramas, the user will still be confused by looking in all directions at the same time.

A comparable approach was chosen in [2], where a new visualization technique for panoramic images based on a coarse 3-D reconstruction was proposed. It is based on the observation that the spatial arrangement of the environment is an important feature that helps the observer to orientate himself within the scene. For this reason, it was assumed that the panoramic image is recorded within a rectangular room of unknown dimensions. By marking the four corners of the room in the panoramic image, the room dimensions and the camera position could be recovered and the panoramic image could be projected onto a virtual room of the same geometry as the original.

In this paper, we further extend this approach by concentrating on panoramic images that were recorded inside more complex indoor environments, not necessarily limited to rectangular room. Our reconstruction algorithm also allows the scene to span several rooms. This is an important special case that enables many new application areas like hotel room advertising, or presentation of real-estate.

Since it is not always possible to cover the whole scene within a single panoramic image, our algorithm allows to reconstruct the 3-D layout of the floor-plan from a set of panoramic images. Once the geometry of the floor-plan is known, a 3-D model of the room walls can be synthesized and the wall textures can be added using the image data from the panoramic image. The proposed presentation provides a flexible way to visualize the scene. On one hand, the virtual camera can be placed outside the room, such that the viewer obtains an overview of the whole scene appearance and room layout. On the other hand, the virtual camera can also be placed at the position of the original camera. Interactively rotating this virtual camera at this position provides views that equal the output of the PIB technique.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'07, September 23–28, 2007, Augsburg, Bavaria, Germany.  
Copyright 2007 ACM 978-1-59593-701-8/07/0009 ...\$5.00.

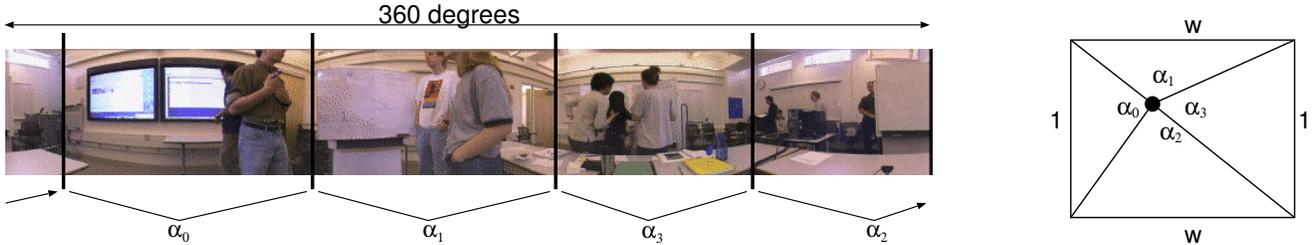


Figure 1: Panoramic input image of a rectangular room with marked room corners. The measured angles are also indicated in the simple floor-plan on the right side.

## 2. RECONSTRUCTION OF FLOOR-PLANS

Several algorithms for 3D reconstruction have been proposed. They can be divided into algorithms without pre-knowledge about the scene and algorithms making use of a scene model. Algorithms of the first class are usually very complex to implement [3] and they are probably not robust enough in cases of low-textured surfaces. Algorithms of the second class employ a complete geometric model of the object or scene, and they only adjust the sizes in the model based on the observed images. An algorithm of this second class is described in [1]. Another algorithm [4] considers specifically the reconstruction of room shapes from panoramic images. Compared to our algorithm, it supports more general geometries than a collection of walls, but compared to our proposal, it is more complex to implement and to use.

### 2.1 Reconstruction algorithm concept

Our algorithm falls into the second class mentioned above, but even though the reconstruction process is semi-automatic, it is easy and fast. It starts with the user drawing a coarse top-down floor-plan of the scene, where the sizes of the rooms need not be correct. Furthermore, the user marks the room corners seen in each panoramic image and indicates to which corner in the floor-plan this corner corresponds. From the positions of the corners marked in the panoramic image, the algorithm can deduce the angle between these corners, seen from the camera position. Using these angle measurements, our algorithm automatically adjusts the sizes of the walls and also finds the correct camera positions, from which the scene was captured. With the geometry known, a textured 3-D model of the scene (usually an apartment or office) is created and used for visualization.

The principal idea of the reconstruction algorithm is derived from the observation that the horizontal dimension of the panoramic image is equivalent to the rotation angle of the camera. Hence, if the distance between two room corners in the panoramic image is, e.g.,  $1/5$  of the image width, the angle between those two corners at the camera are  $72^\circ$ . For the simple case of a rectangular room, we obtain 4 angles, as shown in Figure 1. Since all angles sum to  $360^\circ$  degrees, this only gives three independent measurements. However, this is enough to reconstruct the three parameters of the setup (ratio of wall sizes, and camera position). This is the special case, for which an efficient reconstruction algorithm is proposed in [2]. The algorithm described in the following extends this to more general floor-plans.

Our algorithm starts with an initial floor-plan that defines the room layout (position of walls and constraints about perpendicular walls), but that does not yet include the correct wall sizes. For an example, see Fig. 2. This figure shows a user-supplied geometric room model, where the outline of the room is specified, in which the correct wall sizes are still unknown. The basic principle of

the algorithm is to compute the *corner-to-corner* angles from the current model and compare them with the angles measured in the panoramic image. Subsequently, an optimization process is carried out to adapt the wall sizes such that the differences between angles in the model and the measured angles are as small as possible.

### 2.2 Model Parameterization

The floor-plan reconstruction algorithm uses two types of information for the estimation:

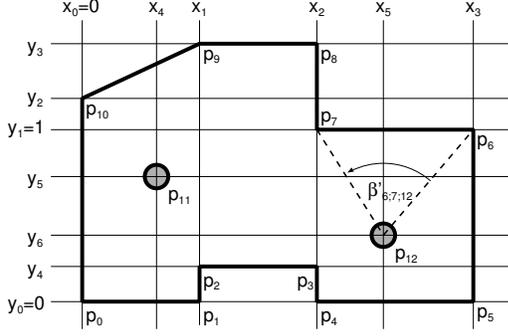
- the angles between room corners measured from their position in the panoramic images, and
- the predefined geometrical layout of the room. This geometrical model includes the relative position of the walls, but not their size. The model also implicitly contains pre-knowledge about right angles between walls.

A floor-plan is parameterized by the 2-D positions of the room corners and the camera positions. The camera positions are required to carry out the texture mapping.

Let us first consider a simple example of a rectangular room and one camera. This configuration gives  $4 \times 2$  parameters for the room corners plus two parameters for the camera position. However, the absolute placement of the room in our coordinate system is arbitrary, so that we can fix one corner to a predefined position, like  $(0, 0)$ . Moreover, we can fix the overall rotation angle of the floor-plan, and as absolute size cannot be determined, we can also fix the length of one wall to, e.g., unity. The easiest way to do this is to fix the position of a second corner to, e.g.,  $(0, 1)$ . In total, this reduces the number of degrees of freedoms by four.

The reduction from ten parameters to only six was obtained by eliminating superfluous degrees of freedom in the parameterization. On the other hand, we can add more pre-knowledge about the room geometry. For example, we can assume that the room shape is rectangular. This pre-knowledge can be expressed with three constraints, each forcing one wall to be perpendicular to another wall. These three constraints further reduce the number of free parameters from six to three. This makes a reconstruction of the geometry possible, since we have three independent measurements of corner-to-corner angles.

For the general floor-plan reconstruction, we enforce the constraints for perpendicular walls implicitly through the parameterization. We use the convention that the rotation of the complete floor-plan is such that (most) walls will be aligned along the horizontal and vertical coordinate axes. This is basically transparent to the user, since this is the way he would draw the floor-plan anyway. Each wall that is aligned to the coordinate axes can be parameterized with only three parameters. For example, a vertical wall is parameterized by the two corner positions, but both positions share the same  $x$  coordinate.



**Figure 2:** Room corners are specified by coordinates  $x_i, y_i$ . Horizontal and vertical walls will reuse the same  $x_i$  or  $y_i$  coordinate for both corners. This implicitly encodes the pre-knowledge that these walls have to be horizontally or vertically aligned. Camera positions are assigned their own pair of  $x_i, y_i$  coordinates.

Furthermore, we also add the normalization of the floor-plan position and size as hard constraints in the parameterization. For this, we select one vertical wall and define one corner position to  $(x_0, y_0) = (0, 0)$  and the other corner position to  $(x_0, y_1) = (0, 1)$ .

An example is depicted in Fig. 2. The room shape has eleven walls, but it is parameterized with only six free parameters  $x_1, \dots, x_3, y_2, \dots, y_4$ . Additionally, the two camera positions add four parameters  $x_4, y_5, x_5, y_6$ . From the image of the left camera, we can obtain nine independent angle measurements, since the corner  $p_4$  is occluded. The right camera can contribute seven angle measurements (note also the angle  $\beta_{7;9;12}$ ). In total, we have 16 measurements for 10 parameters and the reconstruction is possible. Note that a reconstruction would also be possible with only the left camera. In this case, we would only have nine measurements, but also only eight parameters, since the position of the right camera is not included. On the other hand, a reconstruction from only the right camera is impossible, since we would have eight parameters to estimate from only seven measurements.

### 2.3 Geometry-parameter estimation

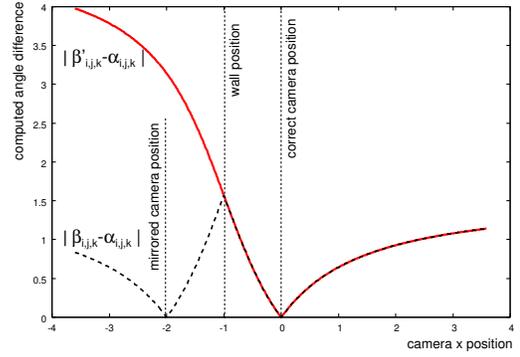
The central task in the floor-plan reconstruction is to estimate the model parameters based on the angle measurements that were measured in the panoramic images. The model parameters consist of the coordinates  $x_i, y_i$  of the wall corners and the camera positions. According to the geometric constraints, some of these coordinates can appear in the specification of several positions. All coordinate values that appear in the model are collected in a parameter vector

$$\mathbf{v} = (x_0 = 0, x_1, x_2, x_3, \dots, y_0 = 0, y_1 = 1, y_2, y_3, \dots), \quad (1)$$

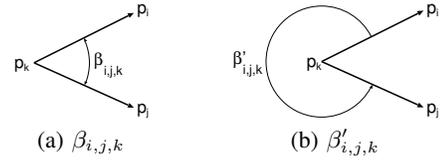
in which three entries are fixed (namely  $x_0 = y_0 = 0$ , and  $y_1 = 1$ ) to remove the superfluous degrees of freedom. To find the corresponding coordinates for a position  $\mathbf{p}_i$ , we use two index sets  $m_i$  and  $n_i$  into the parameter vector  $\mathbf{v}$  to define  $\mathbf{p}_i = (x_{m_i}, y_{n_i})^\top$ .

From the captured panoramic images, we obtain a set of angle measurements  $\mathcal{M} = \{(i, j, k)\}$ . Each measurement gives an angle  $\alpha_{i,j,k}$  between corners  $\mathbf{p}_i$  and  $\mathbf{p}_j$ , as seen from camera position  $\mathbf{p}_k$ . Furthermore, we can compute corresponding angles  $\beta_{i,j,k}$  from the geometric model as

$$\beta_{i,j,k} = \arccos \frac{\mathbf{d}_{ik}^\top \cdot \mathbf{d}_{jk}}{\|\mathbf{d}_{ik}\| \cdot \|\mathbf{d}_{jk}\|}, \quad (2)$$



**Figure 3:** With the non-oriented angle  $\beta_{i,j,k}$ , it cannot be distinguished on which side of a wall the camera is located. The oriented angle  $\beta'_{i,j,k}$  has a single minimum at the correct side.



**Figure 4:** Definition of angle differences. While  $\beta_{i,j,k}$  is the inner angle between the two vectors (a),  $\beta'_{i,j,k}$  is defined as the angle from  $\mathbf{p}_i$  to  $\mathbf{p}_j$  in counterclockwise direction (b).

where  $\mathbf{d}_{ik} = \mathbf{p}_i - \mathbf{p}_k$  and  $\mathbf{d}_{jk} = \mathbf{p}_j - \mathbf{p}_k$  are the vectors from the camera position  $k$  to the corners  $i$  and  $j$ . This equation defines  $\beta_{i,j,k}$  as the inner angle between these vectors. We can now define a cost function, computing the total error between the measured angles and the angles present in the current floor-plan model, as

$$E(\mathbf{v}) = \sum_{(i,j,k) \in \mathcal{M}} |\beta_{i,j,k} - \alpha_{i,j,k}|. \quad (3)$$

This error is minimized with the vector  $\mathbf{v}$  as variable using a Quasi-Newton optimization.

The convergence robustness of this optimization depends mainly on the convexity of the cost function  $E$ . When considering again the definition of  $\beta_{i,j,k}$  from Eq. (2), we notice that the definition gives the inner angle  $\beta_{i,j,k} \in [0; \pi]$  between two vectors. This means that the angle is the same for the camera being on either side of the wall. In the optimization process, this has the disadvantage that there is a local minimum on each side of the wall (Fig. 3).

To prevent this effect, we do not actually use the definition of the angle  $\beta_{i,j,k}$ , but we apply the *oriented angle*  $\beta'_{i,j,k}$ , which is defined as the angle from corner  $\mathbf{p}_i$  to corner  $\mathbf{p}_j$ , measured in counter-clockwise direction (see Fig. 4). We can compute the oriented angles as

$$\beta'_{i,j,k} = \begin{cases} \beta_{i,j,k} & \text{if } \det[\mathbf{d}_{ik} \mid \mathbf{d}_{jk}] \leq 0, \\ 2\pi - \beta_{i,j,k} & \text{if } \det[\mathbf{d}_{ik} \mid \mathbf{d}_{jk}] > 0. \end{cases} \quad (4)$$

Using this angle definition, this term introduces a single minimum at the correct side of the wall, and the error increases monotonically with increasing distance from the optimal position (Fig. 3).

When  $\mathbf{v}$  is computed, the exact sizes of the walls and the camera positions are known, and the panoramic images can be projected back to the wall planes. We obtain a set of textured quadrilaterals that can be rendered as a 3-D model.

### 3. EXPERIMENTAL RESULTS

An example result for the floor-plan reconstruction is shown in Figure 5. The input images for the floor-plan reconstruction were captured with a digital still camera and combined into a panoramic image later. The computation time for the reconstruction was clearly below one second in all of our experiments. We evaluated the accuracy of the reconstruction result by comparing the normalized size of the walls in the reconstruction with their real sizes. The average deviation was about 4%, which is probably mainly due to an inaccurate alignment of the input images. Moreover, for simplicity, we assumed that the walls itself have no thickness, which is obviously wrong in reality and which also leads to small deviations in room size.

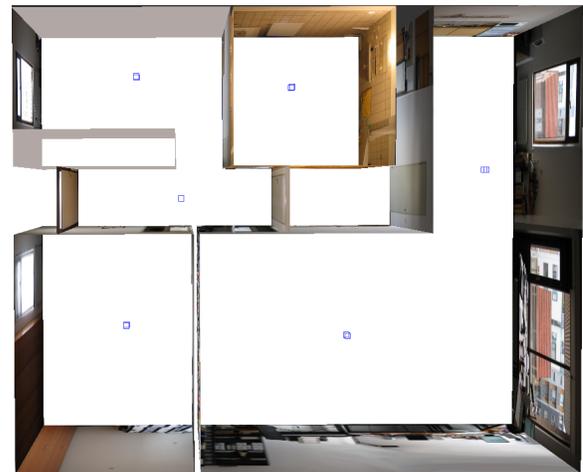
### 4. CONCLUSIONS

We have proposed a visualization technique for panoramic images recorded in an indoor environment. Our algorithm reconstructs the room geometry from the panoramic images and presents the panoramic image as the projection onto the room walls. The reconstruction algorithm requires only minor user assistance. Our conclusion is that the proposed visualization can provide a more comprehensive presentation of the scene to the user than a flattened panoramic image where the room geometry is not visualized. Applications of our proposal, especially for the floor-plan reconstruction, are the advertisement of apartments, or digital museums for which virtual tours could be made available online. Another application could be the reconstruction of scenes in surveillance systems, in which the objects are extracted from the video and inserted into the 3-D model at their corresponding real-world position.

It should be noted that both reconstruction algorithms can also be used directly with panoramic video instead of single images, providing video textures on the walls of the 3-D model. Even for the video application, the geometry model only has to be computed once, if the positions of the cameras are fixed.

### 5. REFERENCES

- [1] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *SIGGRAPH'96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20, New York, NY, USA, 1996. ACM Press.
- [2] Dirk Farin and Peter H. N. de With. Reconstructing virtual rooms from panoramic images. In *26<sup>th</sup> Symposium on Information Theory in the Benelux*, pages 301–308, May 2005.
- [3] M. Pollefeys, R. Koch, M. Vergauwen, B. Dknuydt, and L. Van Gool. Three-dimensional scene reconstruction from images. In *SPIE Electronic Imaging, Three-Dimensional Image Capture and Applications III*, volume 3958, pages 215–226, 2000.
- [4] H.-Y. Shum, M. Han, and R. Szeliski. Interactive construction of 3D models from panoramic mosaics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–433, June 1998.
- [5] L. Zelnik-Manor, G. Peters, and P. Perona. Squaring the circle in panoramas. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1292–1299, 2005.



**Figure 5:** Example reconstruction for a complete apartment from six panoramic images. The estimated camera positions are indicated with small cubes inside the rooms.