Monocular 3D Reconstruction of Objects Based on Cylindrical Panoramas

Ralf Haeusler[1], Reinhard Klette[1], and Fay Huang[2]

[1] The University of Auckland, Computer Science Department, New Zealand
r.haeusler@cs.auckland.ac.nz
[2] CSIE, National Ilan University, Yi-Lan, Taiwan

Abstract. This paper discusses ways of using a single panoramic image (captured by a rotating sensor-line camera having very-high spatial resolution) for the geometric shape recovery of a shown object. The objective is to create a sparse polyhedral model, only allowing a few interactive user inputs for a given single panoramic image. The study was motivated by the general question whether a single panoramic image projection allows some kind of 3D shape recovery, possibly benefitting from available monocular approaches for standard (say, pinhole-type) camera models.

Keywords: Monocular 3D reconstruction, cylindrical projection, panorama, rotating sensor-line camera.

1 Introduction

The computation of 3D structure from stereo images receives increasingly attention due to the enormous progress recently in this area. However, the task of retrieving 3D information from a single image seems to be a rather ill-posed problem, yet scientific interest herein dates back many centuries [2]. In fact, so-called monocular reconstruction cannot work without some kind of a-priori knowledge (i.e., some assumptions about geometric properties or shapes of the shown objects, or about surface reflectance).

Apart from utilizing geometric constraints for specified classes of objects (see, for example, [7,8]), a popular approach to monocular 3D understanding applies the concept of vanishing points (see, for example, [4,5]), as introduced by painters in the renaissance.

Of course, talented artists may often be successful in modelling manually a scene from a single photograph, by using common 3D clues for the human visual system [10].

This paper deals with monocular reconstruction based on images of very high resolution and with a wide field of view. Such images may be recorded with so-called rotating sensor-line cameras [6], and the resulting images are also called *cylindrical panoramas*. The question arises whether such images, projected onto a straight cylinder, provide better opportunities for understanding the 3D structure from only a single image compared to images recorded with a 'normal' (say, *pinhole-type*) camera. [6] uses cylindrical panoramas for 3D modelling of (large)

T. Wada, F. Huang, and S. Lin (Eds.): PSIVT 2009, LNCS 5414, pp. 60–70, 2009.

Fig. 1. A 3D model of the throne room in castle Neuschwanstein [6]. Here, multiple laser range-finder scans and multiple cylindrical panoramas have been used. Of course, a single-view panoramic scan cannot provide this complexity of 3D information (not even close to this).

objects such as a castle, by fusion of data of a laser range-finder, yielding visually impressive results, see Figure 1.

However, the 3D information is in this case derived via purpose-designed measuring equipment (laser range-finder) whose application is characterized by difficult and labor-intensive manual handling of the involved equipment.

[3] reports about pioneering work on modeling a 3D scene directly from a panoramic image. However, the presented approach does not yet allow to reconstruct a broad range of objects, and did also not yet cover the recovery of aspect ratios.[1] Aspect ratios of recorded rectangles may be recovered from a single (pinhole-type) image; see [9].

The outline of this paper is as follows: Section 2 provides technical prerequisites related to panoramic imaging when projecting onto a straight cylinder. Section 3 presents a monocular reconstruction method and an example (image with resulting object model). Section 4 is pointing to particularities of cylindrical panoramas concerning monocular reconstructions. Section 5 contains conclusions.

2 Cylindrical Panoramas

A common cylindrical panorama results from some kind of image stitching, but to allow for very high-resolution cylindrical panoramas, a rotating sensor-line camera is an appropriate choice. A (typically, CCD) sensor-line and its projection center rotate about a defined axis, describing this way a cylindrical surface

[1] When mapping a rectangle into a trapezoid by perspective projection, the ratio of side lengths of the rectangle defines the unknown aspect ratio.

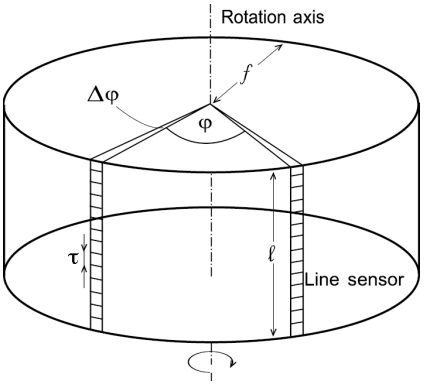

Fig. 2. Camera model of a cylindrical projection with a single projection center: $\Delta\varphi$ denotes the angular increment, f the effective focal length of the lens, τ is the physical size of a pixel on the sensor line (assumed to be constant), and l the total physical length of the sensor line

(with the recorded panorama) and a circular path, respectively. The recorded panorama is composed line by line, after (or during) such a rotation (typically of full 360°).

If the projection centers of all the recorded lines are at the rotation axis, then they all coincide, and the circular path degenerates into a single point. Such a case of a single projection center is illustrated in Figure 2.

The main advantage of such a camera system is its very large spatial resolution. By specifying the number of recorded lines (columns), the wide field of view of the recorded panorama may even extend beyond 360°, by recording into some directions more than once. The data volume of a single 360° panorama is in the range of several gigabytes for contemporary sensor lines of about 10k color pixels. The main disadvantage is the long exposure time, limiting its use for dynamic scenes (but also allowing interesting effects such as having a person repeatedly in a recorded panorama). Some of the intrinsic parameters (such as focal length, angular increment, size of a pixel) are also illustrated in Figure 2.

As it is most appropriate to record images with square pixels,[2] a common target is to specify the number of columns using an angular increase of

$$\Delta\varphi = 2 \cdot \arctan\left(\frac{\frac{1}{2}\tau}{f}\right)$$

for image recording. We assume (and used) a 360° image with pixels known to have square shape, and this specifies the used intrinsic parameters for monocular reconstruction (up to a scaling factor). Of course, this ignores some possible (minor) errors, such as having the projection center always exactly at the rotation axis. We assume a *camera center* **O** which identifies the unique origin of all projection rays.

[2] To be precise, these are actually 'cylindrical squares' on a cylindrical surface.

The principal point is defined by the intersection of the optical axis with the sensor line, and the actual position of this point will have no impact on the following discussion. Thus we simply assume that image coordinate $j = 0$ identifies the principal point (i.e., somewhere within this square pixel).

Projection rays, necessary for monocular reconstruction, can be calculated from pixel coordinates i and j in the recorded cylindrical image as follows:

$$\begin{aligned} t_\varphi &= \Delta\varphi * i \\ t_\Theta &= \arctan(\frac{j \cdot \tau}{f}) \\ t_\kappa &= \cos(t_\Theta) \end{aligned}$$

This defines a ray direction $\mathbf{t}$ in spherical coordinates, which is converted into Cartesian coordinates as follows:

$$\begin{aligned} t_x &= t_\kappa \sin t_\Theta \cos t_\varphi \\ t_y &= t_\kappa \sin t_\Theta \sin t_\varphi \\ t_z &= t_\kappa \cos t_\Theta \end{aligned}$$

A projection ray $\mathbf{r}$ is thus described by $\mathbf{r} = \mathbf{O} + \lambda \cdot \mathbf{t}$, for a real λ.

3 Monocular Reconstruction

Reconstruction is the process of determining an approximate geometric surface model of an object and its *pose* or *attitude* (i.e., position and direction) in 3D space.

3.1 Proposed Approach

The reconstruction approach based on projection rays, and using only a single image, is as follows: First, some prior knowledge about geometric properties is necessary, usually related to the shape of the shown objects. Then, a selected 3D shape prior has to fit the corresponding family of projection rays such that the image of the object's shape prior matches to the result of the given projection. In the 2D case (pinhole-type images), this was reported in [9] for rectangular objects by calculating a homography such that a given trapezoidal image of a rectangle was actually mapped into a rectangular shape.

We also discuss rectangular geometric primitives here, but apply it to the described cylindrical projection. The diagonals of a rectangle are bisecting each other, say in a 3D point r_d. Then we have that

$$r_d = \frac{r_1 + r_3}{2} = \frac{r_2 + r_4}{2}$$

for the four cyclically ordered vertices r_h of the rectangle, with $h \in \{1, 2, 3, 4\}$. As the corresponding projection rays of the image of r_h should be incident with r_h, it follows that

$$\lambda_1 \cdot t_1 - \lambda_2 \cdot t_2 + \lambda_3 \cdot t_3 = \lambda_4 \cdot t_4$$

Obviously, from a single image, a reconstruction is only possible up to a scaling factor. Thus, without restriction of the generality, it can be assumed that $\lambda_4 = 1$. This defines a linear equational system

$$\begin{bmatrix} t_{1x} & -t_{2x} & t_{3x} \\ t_{1y} & -t_{2y} & t_{3y} \\ t_{1z} & -t_{2z} & t_{3z} \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} t_{4x} \\ t_{4y} \\ t_{4z} \end{bmatrix} \tag{1}$$

The unique solution $\lambda_1, \ldots, \lambda_4$ describes the position of those 3D rectangular vertices up to a scale factor μ as follows: $r_h = \mathbf{O} + \mu \cdot \lambda_h \cdot \mathbf{t_h}$. Scale factor μ can be determined only if object dimensions are known for real world scenes (e.g., height or width of objects in the real world).

However, applied to an object that is composed of several 'connected' rectangles, a reconstruction result is not satisfactory if every single rectangle is reconstructed separately. The first reason is that every single rectangle would have a different scaling factor μ as one of the λ_h values was set to be equal to one. Adjusting the scale factors μ over all rectangles based on 'connectedness' (i.e., sharing of edges) properties of faces of the object still does not allow for a closed reconstructed object surface due to unavoidable reconstruction inaccuracies.

The following is now our proposition for solving this problem. From an object consisting of q rectangles with n vertices, a single linear equational system $\mathbf{T} \cdot \lambda = \mathbf{t}$ is derived as follows: An instance of vector $\mathbf{t}$ contains data from the 'first' projection ray to a vertex which may be incident with up to q rectangles. (The component λ_1 of vector λ is set to be equal to one due to scale ambiguity.) Assuming that q is the maximum for all considered rays, we have a matrix $\mathbf{T}$ composed of $n - 1$ columns and $3 \cdot q$ rows. These contain information about all the n projection rays, with up to q rectangles in each case.

All the equations of the derived system are as follows:

$$\begin{bmatrix} t^1_{2x} & -t^1_{3x} & t^1_{4x} & \cdots & t^1_{nx} \\ t^1_{2y} & -t^1_{3y} & t^1_{4y} & \cdots & t^1_{ny} \\ t^1_{2z} & -t^1_{3z} & t^1_{4z} & \cdots & t^1_{nz} \\ t^2_{2x} & -t^2_{3x} & t^2_{4x} & \cdots & t^2_{nx} \\ t^2_{2y} & -t^2_{3y} & t^2_{4y} & \cdots & t^2_{ny} \\ t^2_{2z} & -t^2_{3z} & t^2_{4z} & \cdots & t^2_{nz} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ t^q_{2x} & -t^q_{3x} & t^q_{4x} & \cdots & t^q_{nx} \\ t^q_{2y} & -t^q_{3y} & t^q_{4y} & \cdots & t^q_{ny} \\ t^q_{2z} & -t^q_{3z} & t^q_{4z} & \cdots & t^q_{nz} \end{bmatrix} \cdot \begin{bmatrix} \lambda_2 \\ \lambda_3 \\ \\ \vdots \\ \\ \lambda_n \end{bmatrix} = \begin{bmatrix} t^1_x \\ t^1_y \\ t^1_z \\ t^2_x \\ t^2_y \\ t^2_z \\ \vdots \\ \vdots \\ \vdots \\ t^q_x \\ t^q_y \\ t^q_z \end{bmatrix}$$

$\mathbf{T}$ is in general a sparse matrix, as one projection ray is often connected to not more than four rectangles. (For implementation, the `multimap`-datastructure from the Standard Template Library [1] may be recommended.)

In general we have that $3 \cdot q \geq n$, and an overdetermined system needs to be solved, by minimizing the Euclidean norm $\|\mathbf{T} \cdot \lambda - \mathbf{t}\|$. Due to a high sensitivity to outliers, this norm might be unsuitable for some objects, as it may violate our initial assumption of bisecting diagonals of the involved rectangles. Thus, we only use this for an initial solution for a subsequent nonlinear minimization. For this we apply as error metric a function Δ of the form $\Delta(e) = \log(1 + e^2/c)$ (for some constant c) which assigns smaller penalties to larger discrepancies e between vertices of rectangles.

Finally, after having computed a solution vector λ, the derivation of a list of reconstructed rectangles (from $\mathbf{T}$ and $\mathbf{t}$, using, for example, the `multimap`-datastructure) is kind of straightforward.

3.2 An Example

The proposed method can be used for (approximate) reconstructions of various objects defined by multiple rectangles. In the example shown below, a room of an indoor scene is approximated by a cuboid. Corresponding interactive user inputs (for identifying vertices of rectangles) are illustrated in Figure 3. In this case, a user selected eight corners of the room.

The shown arcs demonstrate the complexity of projected edges into such a panorama, basically illustrating that an automated extraction of vertices defines a challenging problem. Note that further rectangles such as windows or doors may be selected as well, leading in general to more robust 3D reconstructions.

Table 1 lists pixel coordinates (i, j) of the illustrated interactive user input and the corresponding coordinates (x, y, z) of reconstructed 3D points.

The maximum angular discrepancy in this example of a reconstructed cuboidal object is 1.4% (assuming right angles as the golden standard). This may be due to reconstruction inaccuracies in our optimization process, errors in the actual imaging process, or even deviations from an ideally cuboidal room in the shown historic architecture itself.

Figure 4 shows the reconstructed cuboidal room together with mapped textures using a projection of the image data available in the original (single) panorama.

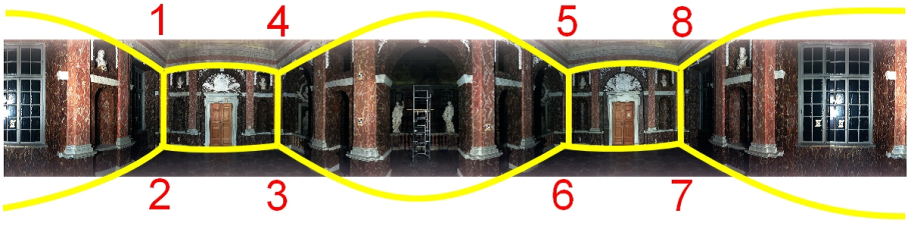

Fig. 3. Interactive user input (selection of eight points, or six geometric primitives). The shown arcs only illustrate how straight segments are curved in a cylindrical projection; they are not required for interactive input.

Table 1. Image coordinates of pixels selected in Figure 3 together with results of the 3D reconstruction process

Point	2D		3D		
	i	j	x	y	z
1	11189	2029	40.47	-50.11	16.84
2	11191	7368	40.49	-50.13	-21.66
3	18703	7414	-12.09	-61.76	-21.83
4	18711	2117	-12.04	-61.67	15.75
5	38702	2098	-36.27	51.42	15.64
6	38698	7409	-36.17	51.39	-21.83
7	46171	7314	16.94	62.70	-22.23
8	46171	2002	16.97	62.75	16.80

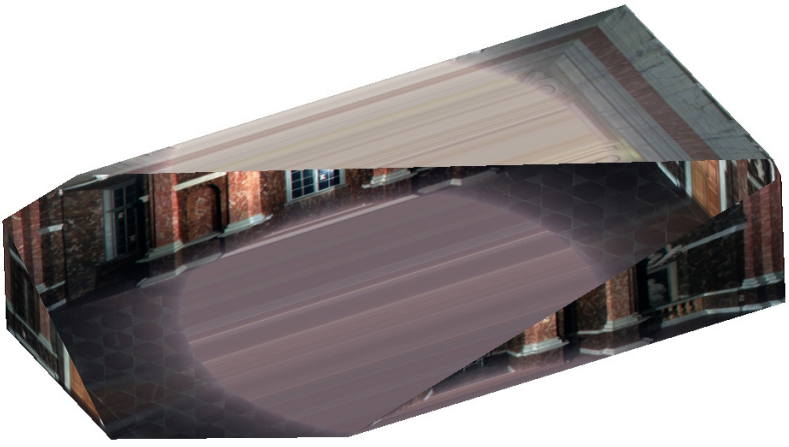

Fig. 4. Reconstructed cuboidal room with mapped textures. Circular regions on the floor and the ceiling were not recorded by the rotating sensor-line camera, and texture information is thus not available in these areas. (The ceiling is shown to indicate the reconstructed 3D volume.).

4 Pinhole-Type versus Cylindrical Camera

The example illustrated that it is possible to generate a full 3D volume model from a single 360° panoramic image, what is, of course, not possible with a single image of a pinhole-type camera. For pointing out whether the cylindrical projection itself is already advantageous compared to the standard pinhole model, we look at panoramic images with a viewing angle less than 360°.

For 360° cylindric images with square pixels, relevant intrinsic camera parameters were assumed to be given in Section 2. However, angular increment and focal length of a rotating line camera may also be estimated based on given (recorded) images.

4.1 Estimation of Angular Increment

The most obvious observation (that should be exploited) is that straight lines in the real-world are generally bent under cylindrical projection, in difference to pinhole-type cameras. In the description below we omit lens distortion effects and assume mathematical cylindrical projection.

Keeping in mind that the straightness of line segments is invariant under homographies, it is sufficient to ensure that line segments curved due to cylindrical projection become straight when projected into any plane (e.g., the one shown in Figure 5). A cylinder-to-plane projection involves the sought-after parameter $\Delta\varphi$, and this can be estimated iteratively.

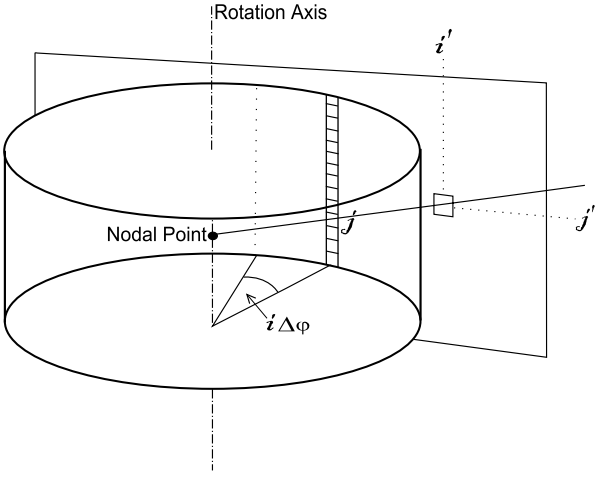

Fig. 5. Projection of an image cylinder into a tangential plane

Image coordinates (i, j) of the image cylinder are projected into planar image coordinates (i', j') (on a tangential plane) according to the following equations:

$$\begin{aligned} i' &= f \cdot \tan(i \cdot \Delta\varphi) \cdot \frac{1}{\tau} \\ j' &= \frac{j}{\cos(i \cdot \Delta\varphi)} \end{aligned} \tag{2}$$

The tangential plane coincides with the cylinder surface at $\varphi = 0$.

We refer to this as projection Π. It is obvious that only image data within a viewing angle of 180° can be projected onto a tangential plane.

Now, for any three points $(i_1, j_1), (i_2, j_2)$ and (i_3, j_3) on a 'curved line' in the cylindrical image, assumed to be a projection of a straight segment, the points $(i'_1, j'_1), (i'_2, j'_2)$ and (i'_3, j'_3), with

$$\begin{aligned} (i'_1, j'_1) &= \Pi((i_1, j_1)) \\ (i'_2, j'_2) &= \Pi((i_2, j_2)) \\ (i'_3, j'_3) &= \Pi((i_3, j_3)) \end{aligned}$$

have to be collinear. This infers that

$$\frac{j'_3 - j'_1}{i'_3 - i'_1} = \frac{j'_2 - j'_1}{i'_2 - i'_1}$$

Note that for $i'_3 = i'_1$ or $i'_2 = i'_1$, no information about $\Delta\varphi$ can be derived as vertical lines in the world remain straight on the image cylinder provided that the rotation axis is perfectly upright.

We are able to estimate $\Delta\varphi$ numerically by applying interval bisection, with

$$\frac{j'_3 - j'_1}{i'_3 - i'_1} - \frac{j'_2 - j'_1}{i'_2 - i'_1} \leq \epsilon \leq 10^{-5}$$

being the stop criterion.

Note that, although the method is usable for all 'bent straight segments' in the cylindrical image, it yields most accurate results for strongly bended 'horizontal' segments. In this case, precisions of up to 99.8 % were achieved in our experiments.

This is only the most simple method for estimating $\Delta\varphi$. Significant improvements concerning the precision can be made by taking more pixels into account (potentially all available pixels along a bended line segment), and also using more advanced approximation techniques.

4.2 Estimation of Focal Length

Concerning the focal length, from Equations (2) we see that parameter f is only a linear coefficient in the projection Π, and therefore cannot be estimated from curved lines. Normally we also do not know the length l of the sensor line. However, there is anisotropic scaling depending on the focal length, and this allows to estimate the (dimensionless) ratio l/f also using a-priori knowledge about aspect ratios of shown real-world objects (absolute length cannot be estimated in general due to scale ambiguity of the recorded 3D scene).

Given four vertices $r_1, ..., r_4$ of a rectangle and a-priori knowledge about the ratio

$$\Xi = \frac{|k_1|}{|k_2|} = \frac{|r_4 - r_1|}{|r_2 - r_1|}$$

of two of its edges, the ratio l/f can be estimated such that edge ratio Ξ', resulting from the reconstruction of image points of $r_1, ..., r_4$, is equal to Ξ. In the reconstruction process of image points of $r_1, ..., r_4$, value l/f is the only unknown as $\Delta\varphi$ was already estimated, independently from f, in the previous step. A square-pixel assumption (for the panoramic image) also supports an initialization for a computationally inexpensive iterative search procedure (e. g., interval bisection).

4.3 Use of Vanishing Points

Monocular reconstruction for pinhole-type cameras often utilizes vanishing points. Those are also of benefit for cylindrical images. As for pinhole-type camera images,

vanishing points allow to estimate object attitudes or the positioning of the camera coordinate system with respect to the scene.

A vanishing point is a point where two lines virtually intersect in an image, for two lines which are actually parallel in the 3D world. These lines (in general) do not project into straight lines in cylindrical panoramas. As a result, one pair of two parallel lines can actually have two vanishing points in the panoramic image.

If line segments are only considered in parts of a cylindrical panorama with a viewing angle less than 180°, then their vanishing points can be calculated conveniently using the projection Π as defined above, as well as its inverse projection Π^{-1}. Attention must be paid for choosing points in the cylindrical image with i-coordinates suitable for Π, as it is of little use when the calculation of the intersection of two lines (projected into the plane) is numerically unstable (e. g., when they are nearly parallel).

Now assume one line, containing points p_1 and p_2, and a second parallel line, containing p_3 and p_4; both vanishing points v_1 and v_2 are as follows:

$$v_1 = \Pi_{i_1}^{-1}\Psi^{-1}((\Psi\Pi_{i_1}(p_1) \times \Psi\Pi_{i_1}(p_4)) \times (\Psi\Pi_{i_1}(p_2) \times \Psi\Pi_{i_1}(p_3)))$$

$$v_2 = \Pi_{i_2}^{-1}\Psi^{-1}((\Psi\Pi_{i_2}(p_1) \times \Psi\Pi_{i_2}(p_4)) \times (\Psi\Pi_{i_2}(p_2) \times \Psi\Pi_{i_2}(p_3)))$$

where Ψ and Ψ^{-1} denote the transformation from Cartesian to homogeneous coordinates and vice versa, whereas the indices i_1 and i_2 of Π indicate that different cylinder coordinates i have to be used for obtaining both vanishing points.

Points p_1 and p_2 are unsuitable if the third component of the vanishing point in homogeneous coordinates is close to zero (i.e., parallel lines), and it is also critical if the Euclidean distance between v_1 and v_2 is very small (i.e., only 'one point'). In any of these cases, some permutation of assigned i-values may define a solution.

In case that a pair of bended line segments covers more than 180° in the given cylindrical panorama (what occurs, for example, on the ceiling or on the floor of a room), a plane being tangential to the cylinder surface is unsuitable for the considered projection Π; in this case we would prefer a plane with a normal vector almost parallel to the rotation axis. Apart from projection Π, the calculation of vanishing points remains the same.

An advantage of panoramic images in comparison to 'normal' images is that panoramas have a wider field of view, such also showing more projected lines, and thus, potentially, more vanishing points.

5 Conclusions

In [6] it is discussed how stereo pairs of cylindrical panoramas may be used for 3D reconstruction. In this paper we have specified a way how to use segmentations of 3D shapes into rectangles to ensure approximate 3D reconstruction just based on

a single cylindrical panorama. The use of the intersection point of both diagonals of a rectangle proved to be useful for this approach.

The 'bending' of straight lines, as occurring in panoramic images due to cylindrical projection, may be entirely characterized by two pixels on such an arc, the focal length, and the angular increment $\Delta\varphi$. Therefore, it is also possible to apply the concept of vanishing points for 3D reconstruction; see [4,5] for 'normal' images.

Object surfaces different from multiple rectangular faces are also possible for approximate monocular reconstruction; see [8]. These are, for example, spheres, circular discs, cylinders, or some specially shaped room corners (with a-priori knowledge about their geometry). The (manual) reconstruction of freeform shapes, which widely expands the functionality of a system for monocular reconstruction, is demonstrated in [10] and its incorporation for panoramic images was already proposed there.

Acknowledgments. The authors thank Karsten Scheibe from DLR (German Aerospace Center) for providing image data for experiments, and source code for efficient I/O operations for panoramic images of very-high spatial resolution.

References

1. Becker, T.: STL & generic programming: STL containers. C/C++ Users Journal 19 (February 2001)
2. Berkeley, G.: An essay towards a new theory of vision (1709), `http://www.gutenberg.org/etext/4722`
3. Chu, N.S.-H., Tai, C.-L.: Animating Chinese landscape paintings and panorama using multi-perspective modeling. In: Proc. Computer Graphics International, pp. 107–112 (2001)
4. Criminisi, A.: Single-view metrology: Algorithms and applications. In: Van Gool, L. (ed.) DAGM 2002. LNCS, vol. 2449, pp. 224–239. Springer, Heidelberg (2002)
5. Guillou, E., Meneveaux, D., Maisel, E., Bouatouch, K.: Using vanishing points for camera calibration and coarse 3D reconstruction from a single image. The Visual Computer 16, 396–410 (2000)
6. Huang, F., Klette, R., Scheibe, K.: Panoramic Imaging: Laser-Range Finders and Sensor-Line Cameras. Wiley, Chichester (2008)
7. Kanatani, K.: Group Theoretic Methods in Image Understanding. Springer, Berlin (1990)
8. Voss, K., Neubauer, R., Süße, H.: Monokulare Rekonstruktion für Robotvision. Shaker, Aachen (1994)
9. Wang, X., Klette, R., Rosenhahn, B.: Geometric and photometric correction of projected rectangular pictures. In: Proc. Image and Vision Computing, New Zealand, pp. 223–228 (2005)
10. Zhang, L., Dugas-Phocion, G., Samson, J.S., Seitz, S.M.: Single-view modelling of free-form scenes. J. Visualization Computer Animation 13, 225–235 (2002)