# DISTANCE MEASUREMENT IN PANORAMA

*Zhongding Jiang, Nan Jiang, Yijie Wang, Binyu Zang*

Computer Graphics Laboratory, Software School, Fudan University
No.825 Zhangheng Rd., Shanghai 201203, China
{zdjiang,byzang}@fudan.edu.cn

## ABSTRACT

Computing world distances of scene features from the captured images is a common task in image analysis and scene understanding. Previous projective geometry based methods focus on measuring distance from one single image. Hence, the scope of measurable scene is limited by the field-of-view (FOV) of one single camera. In this paper, we propose one method of measuring distances of line segments in real world scene using panorama representation. With a full view panorama, the scope of measurable scene is increased and can fully cover the sphere of $360 \times 360$ FOV. With panorama representation, distance of long-range features, which can not be fully captured by a single image, can be measured from the panoramic image. A prototype system called *PanoMeasure* is developed for enabling user to interactively measure the distances of line segments. Experiments with simulated data and real measurement results verify that the method offers high accuracy.

***Index Terms—*** distance measurement, panorama

## 1. INTRODUCTION

Measuring distance in indoor/outdoor scene from the captured image has recently attracted much attention [1, 2, 3]. Those methods are based on projective geometry [4] and easier to use than on-the-spot measuring methods that use active devices (e.g., ultrasonic device, laser range finder and structured light device).

Existing projective geometry based methods focus on measuring distance from a single image [1, 2, 3]. The scope of measurable scene is limited by the field-of-view (FOV) of the camera. Since one perspective image of common camera provides narrow FOV, long-range features that are partly captured by the camera can not be measured. For improving FOV, a fish eye camera can be used for distance measurement [5]. Since the FOV of fish eye camera can not fully cover the visual sphere at the captured location, long-range features may suffer from being partly captured. Using image mosaics techniques, full view panorama (360×360 mosaics) that completely covers the visual sphere can be constructed [6, 7]. Since long-range features can be completely captured in the full view panorama, measuring their distances becomes possible.

In this paper, we propose one method for interactively measuring distances of line segments in real world scene using panorama representation. A prototype system, *PanoMeasure*, is developed for enabling user to interactively measure the distances of line segments. Our method is based on the concept of back-projected ray of scene point [4], and combines the strengths of automatic image mosaics [7] and plane recovery using geometry constraints [8, 9, 2]. In our system, user can interactively select the scene points for measurement with simple mouse operations. After that, the distance is automatically calculated using the recovered camera projection matrix and reference planes. As to our best knowledge, it is the first distance measure method that employs panorama representation for measuring distances of long-range features.

The paper is organized as follows. Section 2 describes the details of our distance measurement method with panorama representation. The accuracy of proposed method is validated with simulated experiments in Section 3 and real world data in Section 4, respectively. Conclusions and future work are pointed out in Section 5.

## 2. DISTANCE MEASUREMENT IN PANORAMA

We construct the panorama of real world scene from $n$ input images captured by one rotating camera[1] using automatical image mosaic technique [7]. For rendering in PanoMeasure, the panoramic image is converted into one cubic environment map and textured onto the faces of one cube.

When conducting distance measurement, user selects two image points of the rendered result of PanoMeasure with mouse click and drag. The corresponding real world scene points of the selected points can be on the same reference plane or on two different reference planes, respectively. Once the world coordinates corresponding to each selected point are determined, distance between scene points can be calculated from them using $L_2$ norm.

---

This work is partly supported by the 863 Program of China (Grant No. 2006AA01Z325) and gift from Microsoft Research Asia.

[1] The $i$th image captured by rotating camera is denoted as camera $i$ for brief description.

The back-projected ray of scene point [4] is invariant when user rotates, zooms in and zooms out for selecting image points using PanoMeasure. Hence, we exploit this property to determine the world coordinates of the selected point. The basic idea is that the world coordinates of the scene point can be determined by intersecting its back-projected ray and the plane containing the scene point (reference plane).

For measuring a line segment that can not be captured by one single image, user can zoom out the panorama to obtain a large FOV when specifying the two end points. Another approach for selecting the end points of long-range line segment is that user first selects one point, then rotates the panorama to select the other point.

The whole measurement process is divided into offline and online phases. The offline phase includes panorama generation and projection matrix recovery. In the online phase, world distance is measured interactively using PanoMeasure with simple mouse operations.

### 2.1. Panorama Generation and Camera Projection Matrix Recovery

For generating panoramic image, we first choose a registration coordinate frame $F$ with all cameras centered at its origin, then use automatical image mosaic technique [7] to estimate intrinsic matrix $\mathbf{K}_i$ and rotation matrix $\mathbf{R}_i^f$. $\mathbf{R}_i^f$ represents the orientation of each camera $i$ in $F$. The cubic environment map converted from the panoramic image is rendered in PanoMeasure.

To obtain the back-projected ray of scene point [4] for measurement, we recover all cameras' projection matrices. Since intrinsic matrix $\mathbf{K}_i$ has been determined in panorama generation process, we only need to estimate rotation matrix $\mathbf{R}_i^w$ and translation $\mathbf{T}_i$ for each camera $i$ in the world coordinate frame $W$. Instead of calculating extrinsic parameters for all cameras separately, we obtain $\mathbf{R}_i^w$ and $\mathbf{T}_i$ according to $\mathbf{R}_i^f$ indirectly since $\mathbf{R}_i^f$ represents relative orientation of all cameras. An appropriate camera (at least four world-to-image correspondences can be easily identified in its captured image) is chosen as the reference camera, then Matlab Camera Calibration Toolbox [10] is used to calculate its $\mathbf{R}_r^w$ and $\mathbf{T}_r$ in $W$. All other cameras' $\mathbf{R}_i^w$ and $\mathbf{T}_i$ in $W$ are determined from $\mathbf{R}_r^w$ and $\mathbf{T}_r$ with following equations:

$$\mathbf{R}_i^w = \mathbf{R}_i^f(\mathbf{R}_r^f)^{-1}\mathbf{R}_r^w \tag{1}$$

$$\mathbf{T}_i = \mathbf{R}_i^f(\mathbf{R}_r^f)^{-1}\mathbf{T}_r \tag{2}$$

The projection matrix of each camera $i$ is obtained by $\mathbf{P}_i = \mathbf{K}_i[\mathbf{R}_i^w|\mathbf{T}_i]$ .

### 2.2. Interactive Distance Measurement

Let $\mathbf{x}$ denote one point of the rendered image selected by user, and $\mathbf{X}$ be its corresponding scene point in real world. For two selected points $\mathbf{x_1}$ and $\mathbf{x_2}$, once we obtain the world coordinates of scene points $\mathbf{X}_1$ and $\mathbf{X}_2$, the distance between them can be calculated using $L_2$ norm.

When calculating the world coordinates of $\mathbf{X}$, we intersect the back-projected ray $\mathbf{L}$ [4] of the point $\mathbf{x}$ with the reference plane containing point $\mathbf{X}$. The equation of reference plane can be computed using plane recovery methods [8, 9, 2]. Since perpendicularity exists widely between planes, we implement simple methods for user interactively deducing the plane perpendicular to the reference plane.

The back-projected ray $\mathbf{L}$ of scene point $\mathbf{X}$ is calculated from camera center $\mathbf{C}$ and corresponding render image point $\mathbf{x}$ (Fig. 1). When $\mathbf{x}$ is selected, its 2D window coordinates is obtained. Since we set 3D render coordinates frame $G$ coincident with registration coordinates frame $F$ when rendering panorama, $\mathbf{L}$ also passes through render image point $\mathbf{x}$ and corresponding input image point $\mathbf{u}$ of camera $i$ (Fig. 1). To obtain direction of $\mathbf{L}$ in world coordinate frame $W$, two world points lying on it need to be determined. One is camera center $\mathbf{C}$, obtained by $\mathbf{C} = -(\mathbf{R}_i^w)^{-1}\mathbf{T}_i^w$. The other point $\mathbf{U}$ is calculated from $\mathbf{x}$ using coordinates transform. First, a point $(X, Y, Z)^\intercal$ on panoramic cube which is mapped to $\mathbf{x}$ in PanoMeasure is obtained using 3D graphics API (e.g., *gluUnproject* when using OpenGL for rendering) (Fig. 1). Since camera center $\mathbf{C}$ is $(0, 0, 0)^\intercal$ in $G$, the direction of back-projected ray $\mathbf{L}$ in $G$ can be denoted as $\mathbf{V} = (X, Y, Z)^\intercal$. Then its corresponding input image is determined by finding the corresponding camera $i$ that contains the ray $\mathbf{V}$. Let $\mathbf{u}$ denote the intersection point of input image captured by camera $i$ and $\mathbf{V}$. Point $\mathbf{U}$ is obtained using $\mathbf{P}_i^+\mathbf{u}$ [4], where $\mathbf{P}_i^+$ is the pseudo-inverse of camera projection matrix $\mathbf{P}_i$. Once $\mathbf{C}$ and $\mathbf{U}$ are obtained, the Plücker representation [4] of $\mathbf{L}$ is determined by joining $\mathbf{C}$ and $\mathbf{U}$ using following equation:

$$\mathbf{L} = \mathbf{U}\mathbf{C}_h^\intercal - \mathbf{C}_h\mathbf{U}^\intercal \tag{3}$$

where $\mathbf{C}_h$ is homogeneous representation of $\mathbf{C}$. Finally, the world coordinates of $\mathbf{X}$ is calculated by intersecting ray $\mathbf{L}$ and a reference plane $\boldsymbol{\pi_r}$ passing through $\mathbf{X}$ with following equation:

$$\mathbf{X} = \mathbf{L}\boldsymbol{\pi}_r \tag{4}$$

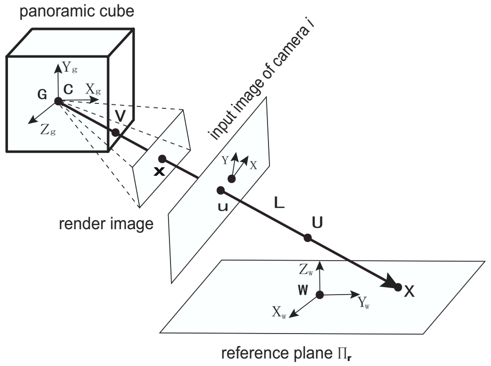


**Fig. 1**. The back-projected ray $\mathbf{L}$ of scene point $\mathbf{X}$.

To obtain reference planes for measurement, user can deduce world plane $\boldsymbol{\pi}_w$ interactively using PanoMeasure when

$\boldsymbol{\pi}_w$ is perpendicular to a reference plane $\boldsymbol{\pi}_r$. Suppose scene points $\mathbf{X}_1$, $\mathbf{X}_2$ are the two intersection points of $\boldsymbol{\pi}_r$ and $\boldsymbol{\pi}_w$. Since $\mathbf{X}_1$, $\mathbf{X}_2$ are on the reference plane $\boldsymbol{\pi}_r$, their world coordinates are determined using PanoMeasure. $\boldsymbol{\pi}_w$ is perpendicular to $\boldsymbol{\pi}_r$ and $\mathbf{X}_1$, $\mathbf{X}_2$ are also on $\boldsymbol{\pi}_w$, then $\boldsymbol{\pi}_w$ is calculated with following equations:

$$\begin{cases} \boldsymbol{\pi}_r^\intercal \boldsymbol{\pi}_w = 0 \\ \mathbf{X}_1^\intercal \boldsymbol{\pi}_w = 0 \\ \mathbf{X}_2^\intercal \boldsymbol{\pi}_w = 0 \end{cases} \tag{5}$$

If more than two intersection points can be identified in PanoMeasure (e.g., the intersection line of $\boldsymbol{\pi}_r$ and $\boldsymbol{\pi}_w$ is visible), Eq. (5) is over-determined and solved by least squares.

## 3. RESULTS WITH SIMULATED DATA

In the simulated experiment, we first construct a panoramic image, then test how the image noise and measure scope affect the accuracy of measurement results. To construct panorama, 7 cameras are located at $\mathbf{C} = (30, 30, 20)^\intercal$. They have the same intrinsic parameters with $f_u = 1600$, $f_v = 1600$, $s = 0$, $u_0 = 512$ and $v_0 = 384$. The image size is 1024×768 pixels. The horizontal and vertical FOV of each camera are about 35° and 27°, respectively. Each pair of adjacent cameras has a 30° angle between their optical axis. Thus they provide total 215° horizontal FOV under this arrangement. For image registration, the coordinate frame of camera 1 is chosen as the registration coordinate frame $F$ and all other cameras' $\mathbf{R}_i^f$ are estimated according to matching points of two adjacent cameras. Camera 1 is also selected as reference camera to retrieve all other cameras' extrinsic parameters. World plane $X = 0$, $Y = 0$ and $Y = 60$ are selected as reference planes and enough scene points are generated on them for image registration and projection matrix recovery (Fig. 2).

Without loss of generality, suppose there are two scene points $\mathbf{X}_1$ and $\mathbf{X}_2$, which are projected into image $i$ and $j$ $(1 \le i \le j \le 7)$, respectively. To denote the measure scope, we set a variable $k = j - i + 1$ to represent the number of sequential images between $i$ and $j$ (e.g., $k = 1$ means $\mathbf{X}_1$ and $\mathbf{X}_2$ are projected into the same image while $k = 7$ means they are projected into image 1 and image 7 respectively). Gaussian noise with mean 0 and standard deviation ranging from 0 to 4 pixels is added to the image points in image registration and projection matrix recovery processes. With a specified $k$ $(1 \le k \le 7)$ , we first perform image registration and projection matrix recovery under current noise level, then generate 1000 space lines to measure. The above process is repeated 50 times independently. The average relative error and standard deviation of results are shown in Fig. 3. They increase slightly under the same noise level when $k$ increases from 1 to 7 due to accumulated error in image registration and projection matrix recovery. From statistic results, our method is accurate enough even with high degree of noise.

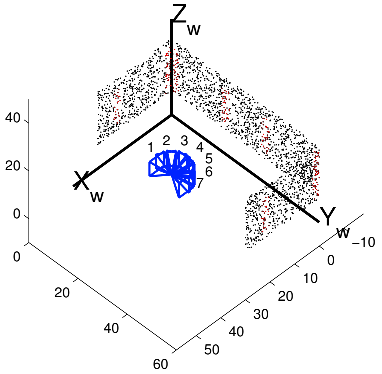


**Fig. 2**. Arrangement of 7 cameras in the simulated experiment: randomly generated scene points are marked by black color, and the detected matching points of two adjacent cameras are marked by red color.

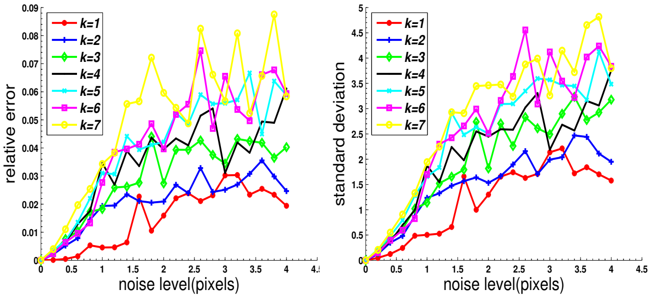


**Fig. 3**. Distance measurement in panoramic image generated by 7 cameras: relative error and standard deviation versus pixel noise level under various $k$.

## 4. REAL MEASUREMENT RESULTS

Measuring distance between two points is a common task in metrology applications, including city planning, 3D GIS, and culture heritage. Hence, our system is one useful tool for these applications. We construct a panorama of our office, then measure several line segments. The measured results are compared with their real lengths. We took 36 images of the office with an off-the-shelf webcam. The resolution of all input images is 640×480 pixels. The generated cubic map textures are shown in Fig.4. The world coordinate frame $W$ was chosen on the ground and an appropriate input image was selected as reference image for recovering its extrinsic parameters (Fig. 5). Then the extrinsic parameters of each camera pose when capturing the other input images were calculated from this reference image. At first, we could only measure distances on the reference plane $Z = 0$ (floorboard). We deduced other planes from $Z = 0$ interactively in PanoMeasure for measuring line segments that do not lie on $Z = 0$. In Fig. 6, world planes $\boldsymbol{\pi}_1$ (cabinet), $\boldsymbol{\pi}_2$ (front door), $\boldsymbol{\pi}_3$ (drinking trough), $\boldsymbol{\pi}_4$ (desk clapboard) and $\boldsymbol{\pi}_5$ (chalkboard) were determined according to their two intersection points and perpendicularity with $Z = 0$. Then $\pi_6$ (left wall), $\pi_7$ (right wall) and $\pi_8$ (ceiling) were determined from $\pi_2$. In Fig. 7, eight line segments were measured. The results are listed in Table 1. Note that the relative error of $S_7$ and $S_8$ are slightly high compared with other line segments. Since $S_7$ is far from the camera, it is difficult to accurately locate its end points in the rendered result using low resolution input images. The measurement result of the scene far from camera can be improved using input images with higher resolution. $S_8$ spans 7 of 12 images that fully cover the horizontal FOV. The accuracy of estimated length of $S_8$ was affected by accumulated error in image registration and projection matrix recovery.

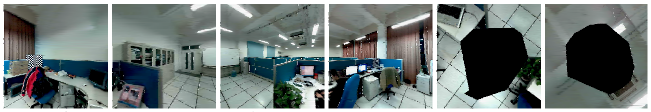

Left Front Right Back Bottom Top

**Fig. 4**. Cubic map textures of our office from 36 images.

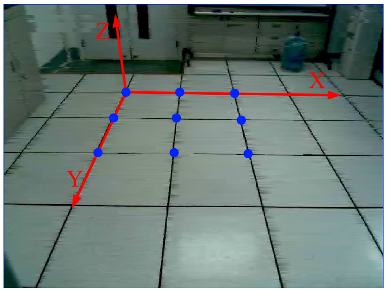


**Fig. 5**. The side length of each tile on the floorboard is 60cm and 9 image-to-world correspondences (marked by blue color) were selected to calculate $\mathbf{R}_r^w$ and $\mathbf{T}_r$ of camera when capturing this image.

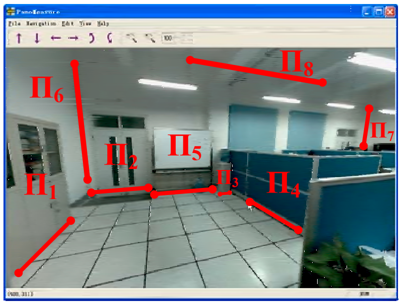


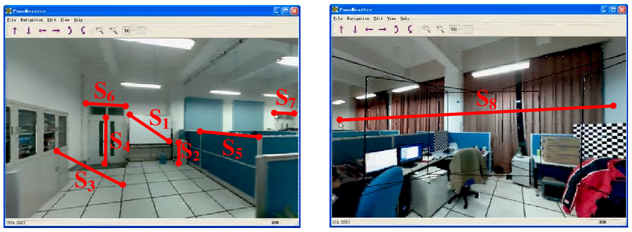
**Fig. 6**. World plane deduction from the reference planes.

**Table 1**. Measurement results of real world data.

| Line segment in image | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ | $S_8$ |
|---|---|---|---|---|---|---|---|---|
| Real distance(cm) | 216 | 90 | 165 | 203 | 162 | 160 | 160 | 1,200 |
| Measured distance(cm) | 217.5 | 90.2 | 165.4 | 202.4 | 163.2 | 158.8 | 163.3 | 1,218.2 |
| Relative error(%) | 0.69 | 0.22 | 0.24 | 0.30 | 0.74 | 0.75 | 2.06 | 1.52 |

## 5. CONCLUSIONS AND FUTURE WORK

This paper propose one method of measuring distances of scene features using panorama for scene representation. By constructing a full view panorama, the visual information of the scene at the captured location is completely recorded, then the distances of scene features can be measured without the limitation of FOV. Using PanoMeasure, user can interactively measure long-range features of real world scene with simple mouse operations. Simulated and real data experiments verify that our method offers high accuracy. In the future work, we will optimize the mosaics process for reducing the accumulation errors. Our method handles panoramic image, we will generalize it to measure distances of scene features in panoramic video.



(a) (b)

**Fig. 7**. Interactive distance measurement in PanoMeasure: (a) 7 line segments were measured by mouse click and drag. (b) A long line segment $S_8$ was measured which spans across 7 of 12 input images fully covering horizontal FOV.

## 6. REFERENCES

[1] A. Criminisi, I. Reid, and A. Zisserman, “Single view metrology,” *IJCV*, vol. 40, no. 2, pp. 123–148, 2000.

[2] G.H. Wang, Z.Y. Hu, F.C. Wu, and H.-T. Tsui, “Single view metrology from scene constraints,” *Image and Vision Computing*, vol. 23, no. 9, pp. 831–840, 2005.

[3] Nan Jiang and Zhongding Jiang, “Distance measurement from single image based on circles,” In *Proc. IEEE ICASSP2007*.

[4] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge Univ., 2nd edition, 2000.

[5] T. Morita, Y. Yasukaw, Y. Inamoto, T. Uchiyama, and S. Kawakami, “Measurement in three dimensions by motion stereo and spherical mapping,” in *Proc. CVPR*, 1989, pp. 422–428.

[6] Richard Szeliski and Heung-Yeung Shum, “Creating full view panoramic image mosaics and environment maps,” in *Proc. SIGGRAPH 97*, 1997, pp. 251–258.

[7] M. Brown and D. Lowe, “Recognizing panoramas,” in *Proc. ICCV*, 2003, pp. 1218–1225.

[8] Gang Xu, Jun ichi Terai, and Heung-Yeung Shum, “A linear algorithm for camera self-calibration, motion and structure recovery for multi-planar scenes from two perspective images,” in *Proc. CVPR*, 2000, pp. 474–479.

[9] Adrien Bartoli and Peter Sturm, “Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene,” *IJCV*, vol. 1, no. 52, pp. 45–64, 2003.

[10] *Camera calibration toolbox for Matlab*, http://www.vision.caltech.edu/bouguetj/calib_doc/.