VERSITA

OPTO-ELECTRONICS REVIEW **21**(1), 01–22

DOI: 10.2478/s11772-013-0079-5

Wide-angle vision for road views

F. HUANG*[1], K.-K. FEHRS[2], G. HARTMANN[3], and R. KLETTE[3]

[1]Computer Science and Information Engineering, National Ilan University, 1, Sec. 1, Shen-lung Road, Yi-Lan, 260, Taiwan, R.O.C.
[2]Vision Systems E-2, Technical University Hamburg-Harburg, 95, Schwarzenbergstraße, D-21073 Hamburg, Germany
[3]Tamaki Innovation Campus, The University of Auckland, 261, Morrin Rd., St Johns, Auckland 1072, New Zealand

The field-of-view of a wide-angle image is greater than (say) 90 degrees, and so contains more information than available in a standard image. A wide field-of-view is more advantageous than standard input for understanding the geometry of 3D scenes, and for estimating the poses of panoramic sensors within such scenes. Thus, wide-angle imaging sensors and methodologies are commonly used in various road-safety, street surveillance, street virtual touring, or street 3D modelling applications. The paper reviews related wide-angle vision technologies by focusing on mathematical issues rather than on hardware.

Keywords: panoramic views, fisheye lenses, wide-angle vision, stereo vision, road views, driver-assistance systems.

1. Introduction

A single wide-angle image (also referred to as a panoramic image) contains more information or features than a "normal" image. This is advantageous for understanding the geometry of three-dimensional (3D) scenes, and for estimating the locations of panoramic sensors within a 3D scene. Thus, wide-angle (or panoramic) imaging sensors and methodologies are commonly used in various static and dynamic street applications, including road surveillance, driver assistance, virtual touring, and 3D city model reconstruction.

Wide-angle (or panoramic) cameras have been built since the second half of the 19th century. In England in 1860 Thomas Sutton designed a wide-angle (120°) camera whose "lens" was a hollow glass sphere filled with water; that camera was then built by Paul Eduard Liesegang in Germany. Albrecht Meydenbauer (1834–1921), a German architect, designed in 1867 a camera which used the first wide-angle (105°) optical lens. As an alternative to a single wide-angle shot, cameras could also take a continuous shot during a full 360° rotation. For example, the sophisticated "Cyclographe" panoramic cameras of Jules Damoizeau, built between 1890 and 1894, rotated by means of a spring mechanism as the film was fed past the shutter at the same speed, but in the opposite direction. A camera with a pivoting lens, called a "périphote", was built in 1901 by Lumiére in Lyon. During exposure, the lens rotated 360°. In Ref. 1, besides providing such historic reminiscences, different panoramic cameras and their properties are discussed as used today in science and technology.

Due to the rapid development of camera technology, panoramic images are already part of our daily lives. They are generated with relatively inexpensive tools, and basically by anyone with a digital camera after spending a few minutes reading a manual. In this article, we categorize today's wide-angle imaging devices by the number of cameras involved while capturing a single panoramic image.

1.1. Single camera approaches

Conventional cameras using standard lenses have a focal--length in the range of 30 to 40 mm and a sensor with a similar side length. Consider a camera with the focal length f and the sensor side length s, both of 40 mm, depicted in Fig. 1. The field-of-view angle α for this camera can be calculated by the following equation

$$\alpha = 2\tan^{-1}\left(\frac{s}{2f}\right), \tag{1}$$

and equals about 53°.

The field-of-view can be increased by decreasing the focal length f. Therefore, specific wide-angle lenses are mounted to the camera. These lenses enable the camera to have a focal-length of less than 20 mm. Consider a camera with a focal length f of 20 mm and a sensor side length of 40 mm. In this setup the vertical and horizontal field-of--view angle α of 90° is achieved.

*e-mail: fay@niu.edu.tw

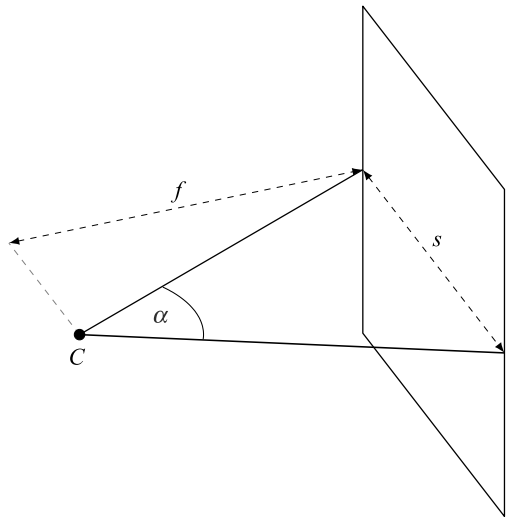

Fig. 1. Field-of-view of conventional cameras.

A further reduction of the focal length is technically hard to achieve. To obtain a wider field-of-view, the side length of the sensor plate can be increased. However, increasing the side lengths leads to issues of lens distortions of the optical system (and would also increase the size of the camera). Therefore, increasing the size of the sensor plate is a challenging technological option on its own. In summary, a field-of-view of about 180° is not achievable with a common sensor size and the usual lower bound for the focal length. Therefore, other camera or lens designs are needed to produce a wide-angle vision.

Catadioptric cameras are cameras using an image sensor which incorporates a curved mirror [2,3]. The mirror is mounted above the lens; the camera is "looking into this mirror". The mirror enables the camera to capture an omnidirectional view. Catadioptric image sensors map an omnidirectional field-of-view into a circular shaped image. However, the mounting of the mirror usually prevents the camera from obtaining scene information in the direction of the optical axis. The recorded images can be projected onto a cylindrical surface to obtain panoramic images with a 360° field-of-view. The major drawbacks of the catadioptric approach include low resolution near the centre of an image, non-uniform spatial sampling, inefficient usage of images (e.g., there is a self-occluded or mirror-occluded area in each captured image), severe distortions and image blurring due to aberrations caused by coma, astigmatism, field curvature, and chromatic aberration. These drawbacks suggest that catadioptric panoramas are not suitable for those recognition or inspection types of applications where high accuracy or high image resolution is required (as in close-range photogrammetry).

In contrast to catadioptric cameras, a rotating sensor-line panoramic camera is capable of obtaining a high resolution omnidirectional field-of-view image. This type of visual sensor consists of a vertical sensor-line ("the camera") that is rotated around a fixed axis. By merging the images of the sensor-line recorded during a full rotation, the panoramic image is obtained. Due to the time dependent image acquisition, sensor-line cameras can be used for capturing static scenes, but create motion artefacts for dynamic scenes.

Fisheye lenses enable cameras to capture scene points located at angles greater than $\alpha/2$, or even more than 90° in relation to the optical axis. These lenses ensure projections which map a semi-sphere onto a plane. In Fig. 2, a fisheye camera and an image acquired by a fisheye lens are shown. The use of fisheye lenses has advantages compared to the other image sensors mentioned before (e.g., more uniform resolution, and dynamic scene capability).

A panoramic image can also be composed from multiple images captured by a single camera (normally assuming constant intrinsic camera parameters) at different times [4–7]. Composition is known as *image stitching* or *image mosaicing*. A basic requirement for merging two images is that they have an overlapping field-of-view. Image mosaicing usually refers to methods of merging motion - uncontrolled image sequences (e.g., aerial imaging), while image stitching is usually applied to images captured by known camera motion (e.g., during rotation on a tripod).

Fig. 2. Fisheye camera and fisheye image. Left: Basler camera with Fujinon fisheye lens. Right: Fisheye image capturing a field-of-view of 185°.

1.2. Multiple camera approaches

Today, due to the availability of inexpensive digital video cameras, various multiple camera systems have been created. Usually, more than three cameras are bundled to provide a wide field-of-view image. Common panoramic images, produced by multiple-camera systems, are cylindrical, spherical, or bird's-eye view images.

There are various proposals for constructing bird's-eye views. Ref. 8 produced nearly seamless bird's-eye view images that were limited to displaying objects on, or very near the ground plane, due to a reliance on homography with the ground plane. While Ref. 8 uses an arrangement of fisheye cameras around the roof of a car, Ref. 9 used catadioptric cameras mounted on a tractor trailer vehicle. The approach of Ref. 2 also relies on an assumption that observed objects are "in the ground plane". Thus, the projection of recorded images into the bird's-eye view causes significant object distortions if objects do not adhere to this limitation. Ref. 10 produces a 3D perception aiming at a driver-assistance system (DAS) using fisheye lenses, where the poses of a camera pair are following the traditional, roughly "parallel configuration" of stereo camera systems. Ref. 11 bundled eight video recorders, a quadruple of cameras looking to each side, and mounted them on the top of a car to capture image sequences for 3D reconstruction of the street.

Ref. 12 describes a novel technique for stitching wide field-of-view images, thus, capable of producing 360° panoramic cylindrical images (see Fig. 3). This method extends the plane-sweep technique employed by Ref. 13 to produce seamlessly stitched images, by sweeping right circular cylinders instead of planes, and working on video sequences instead of static images.

The Ladybug2 camera developed by PointGrey consists of six cameras with five of the cameras looking horizontally outwards and one looking upward. The resolution of each image is 1024×768. Each adjacent camera pair has been designed to visually overlap by about 10%. Six images are then stitched together using the multi-perspective plane sweep (MPPS) approach of Ref. 13. This allows for the production of a spherical panoramic image, and it is of the resolution of 2048×1024 if transformed into a rectangular presentation. The system is capable of capturing 30 panoramic images per second for creating panoramic video sequences.

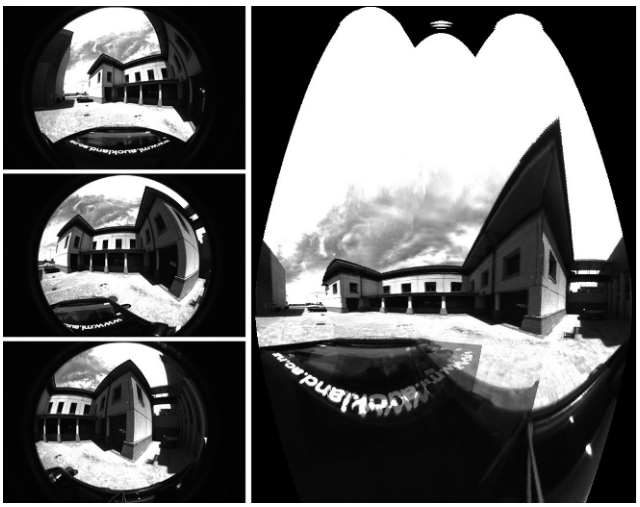

Fig. 3. Left: Three fisheye input images, recorded (*top to bottom*) either in direction of vehicle motion, or tilted 45° clockwise or 90° clockwise relative to the direction of vehicle motion. Right: Stitched cylindrical image.

1.3. Applications

Driver assistance systems (DAS) are additional electronic devices in vehicles, which support the driver through acoustic, visual or tactile information and aim to warn of potential dangers and to handle certain driving situations more easily. Systems integrated in modern vehicles such as lane departure warning and lane keeping assistance need to be aware of the environment surrounding the vehicle. A vision-based driver assistance approach, especially production of wide-angle or bird's-eye view images, is the most natural approach for these kinds of tasks. Other potential applications for vision-based driver assistance are obstacle warning, collision avoidance, and road sign detection. Some of these tasks require distance or depth information. To obtain distance information, multiple cameras are installed in the system. By capturing the same scene from different viewpoints, distance information can be extracted from the recorded images. The extraction of distance information from digital images is referred to as stereo vision or structure from motion.

Stereo vision overcomes the limitations of distance measurements by radar and LiDAR sensors. The discrimination between objects and the ground can be achieved by image analysis. Furthermore, the sensitivity of distance information is in general not dependent on the object's material (ignoring strongly reflective or transparent surfaces). Distance information for the complete field-of-view of the camera can be obtained. A vision based approach is also appropriate as the traffic infrastructure is based on visual perception. Lane markings and road signs can only be detected visually. Radar and LiDAR sensors are blind to this information. Using stereo vision to obtain distance information has an additional benefit, lanes and road signs can be detected as well. Furthermore, cameras can be produced in a cost efficient manner.

To obtain stereo vision, cameras can be mounted horizontally or vertically in or on the vehicle. In particular, if stereo catadioptric cameras are considered, they should be arranged vertically in order to optimize the useful image portions due to the given image geometry constraint (see further below). Another possible approach is to use multiple fisheye cameras [8].

Virtual touring applications have become very popular these days. For static indoor and outdoor scenarios, a set of sparse high-resolution panoramic images accompanied with location information would be sufficient and useful. Such

high-resolution panoramic images can be captured by a rotating line-camera. Moreover, stereo panoramic images for stereo visualization of the scenes can be achieved if the camera consists of two sensor lines and is placed at some constant distance from the rotation axis [14]. For scenes with dynamic motions, or for moving imaging platform, panoramic cameras must be able to acquire a wide field-of--view image within a very short period of time. Increasing image capturing speed usually forces a trade-off in reduced resolution. The Ladybug camera is such a camera which has a relatively quick image acquisition rate of 30 Hz coupled with a somewhat low panoramic resolution of 2048×1024. Panoramic images recorded with Ladybug cameras have recently been used in the Google Streetview application. In this application a large set of street view panoramic images are obtained while the car is moving, and depth information can also be obtained by the equipped laser range finders (see Fig. 4). This system is also capable of generating a panoramic video of the environment if the camera is kept static.

3D models of large urban environments are needed in many applications such as virtual fly/drive-throughs, augmented reality, urban planning, and for documentation purpose. Inventing a fully or semi-automatic method for fast building model reconstruction has lately become a vivid research topic in many fields such as computer graphics and vision. Due to the recent explosion of digital photography, various image-based modelling approaches have drawn a great deal of attention from many researches [11,15]. The use of panoramic images instead of multiple planar images for modelling applications has the following advantages. First, the pose recovery result is more accurate and stable due to the wide field-of-view nature of the panoramic images. Second, since each face texture of any building can be extracted from a single panoramic image, there is no need to deal with the colour blending problem while it is unavoidable if textures are extracted from different planar images captured from different perspectives.

Fig. 4. A Point Grey Ladybug3 camera was mounted on the top of a car for panoramic image acquisition. Three laser range finders were arranged below the camera to obtain depth information for detailed refinement of the buildings surface models, but which is not in the scope of this paper.

In some applications, detailed building reconstruction is not necessary. One specific example is GPS-based car navigation system. Such systems mainly use simplified aerial street maps incorporated with speech to guide drivers to their destinations. In many practical situations drivers might feel that it is difficult to link the aerial 2D map with the visual impression of the environment. Thus, supplying realistic street views of the route can be very useful, and this could be achieved by a set of simple texture-mapped 3D building models.

2. Wide-angle imaging

To understand the process of image acquisition, a camera model is needed. In general, a camera model describes the mathematical relationship between a scene point and its projection on the image plane. In this section, camera models, imaging geometry, and image formation of different panoramic cameras are presented.

2.1. Fisheye camera

For conventional cameras the pinhole camera model is used to describe the projection. In this camera model the aperture is assumed to be a single point. Every light ray emanating from a scene point through the aperture point is captured. The projection of a scene point can be described mathematically by a projection function $r(\theta)$, where θ is the angle between the incoming light ray and the optical axis. The projection function $r(\theta)$ calculates the distance to the optical centre in the image plane. The pinhole camera model is defined by the projection function given in the following equation, where f denotes the focal length of the camera

$$r(\theta) = f \tan(\theta). \tag{2}$$

Therefore, the perspective projection is limited to a field-of-view of less than 180° and, thus, unable to describe the image acquisition model for fisheye cameras [16].

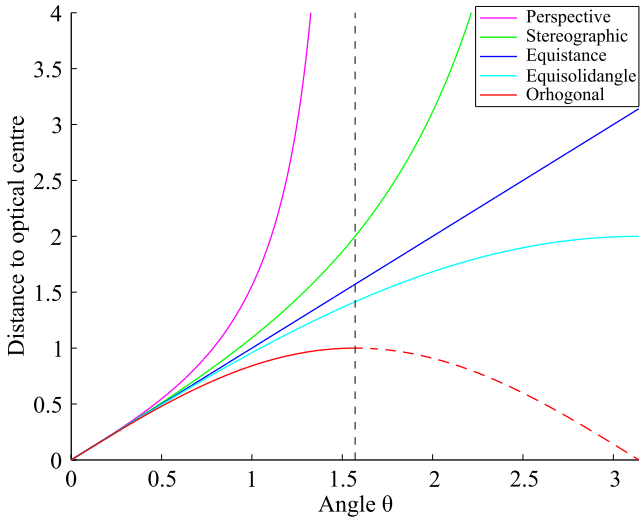

Fig. 5. Projection functions.

Fisheye cameras are cameras using wide-angle lenses, which enable the camera to capture a semi-spherical field--of-view. Fisheye lenses are constructed to follow specific projection functions that map a sphere or a hemisphere onto a plane [17,18]. The different spherical mappings are defined by the following equations

$$r(\theta) = 2f \tan(\theta/2), \tag{3}$$

$$r(\theta) = f\theta, \tag{4}$$

$$r(\theta) = 2f \sin(\theta/2), \tag{5}$$

$$r(\theta) = f \sin(\theta). \tag{6}$$

The projection functions are referred to as stereographic Eq. (3), equidistant Eq. (4), equisolid angle Eq. (5) and orthogonal projections Eq. (6). Due to the spherical projection functions, the recorded images are hugely distorted. Straight lines occurring in a scene are curved in the image plane. Depending on the distance of the line to the optical centre, the curvature differs. Lines in the vicinity of the optical centre are less curved than lines further away. Additionally, the projection function influences the curvatures.

To describe the process of image acquisition for fisheye cameras a specific camera model is needed. The camera model needs to be able to handle a hemispherical or larger field-of-view. Based on the fact that fisheye lenses are built to follow a spherical mapping, it is straight forward to use a camera model that describes the fisheye lens as a hemisphere. Several calibration methods [19–21] are based on such a camera model. In the fisheye camera model a hemisphere is located at the distance of the focal length f from the image plane with the optical axis running through its centre. Without loss of generality, a unit-hemisphere can be assumed, because the size of the sphere has no influence on the projection. Every scene point **P** that can be projected by a central projection onto a point $\mathbf{p}_l$ on the surface of the sphere is captured by the camera. A scene point **P** is projected onto the image point $\mathbf{p}_i$ dependent on the angle θ to the optical axis. In Fig. 6, a camera model for fisheye cameras is shown, where the optical axis is in the z-direction.

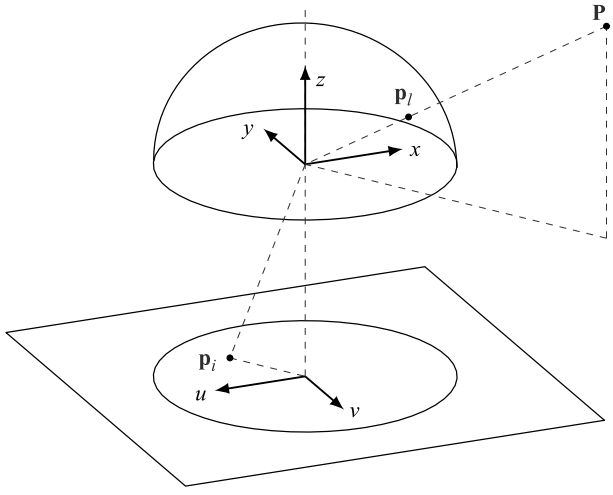

Fig. 6. Fisheye camera model.

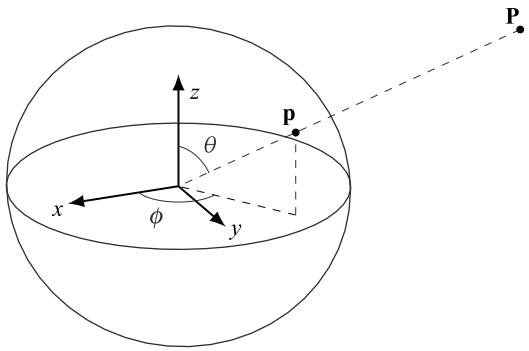

Fig. 7. Spherical camera model.

Another camera model to describe wide-angle cameras is the spherical camera model. A spherical camera is a camera in which the image points are located on the surface of a sphere. Figure 7 depicts a spherical camera capturing a scene point **P**. An image point **p** of a spherical camera can be expressed by the longitude angle ϕ and the latitude angle θ using spherical coordinates.

The images of spherical cameras are referred to as spherical images. Spherical images are intermediate images that can be obtained from every *central camera* (see definition in Sect. 3.1). To transform a captured image into a spherical image, a mapping function is needed that relates the image points of the captured image to points on the surface of a sphere. In Sect. 3.1, a mapping from fisheye images to a spherical surface is presented.

2.2. Rotating sensor-line camera

A "normal digital camera" combines a (CCD or CMOS) sensor-matrix with some optics, all packed nicely into a box containing various electronic components. Now imagine that the sensor matrix, consisting of $M \times N$ sensor elements (each recording a single pixel), degenerates in a way that there is only one column of sensor elements (say, $N = 1$; for example, similar to those used in a flatbed scanner). The benefit of such a configuration is that sensor technology enables us to produce such a *sensor-line* for very large values of M, say M greater than 10,000, but producing sensor-matrices of 10,000×10,000 elements is still a challenge today.

A digital camera, with the sensor-matrix "shrunk" into a single sensor-line, may now be placed on a tripod and rotated, taking many images, "column by column" during such a rotation. This defines a *rotating sensor-line camera*, a panoramic camera which may record 360° panoramic images within the time frame needed for taking many shots during one full rotation. Such a sensor is not only more economical (compared to the use of a, say, 10,000×10,000 sensor-matrix camera), it also comes with several performance benefits (e.g., the option of having stereo pairs) for recording panoramic images [1].

The camera model is depicted in Fig. 8. The projection centre of the sensor-line is denoted as $\mathbf{C}_i$ (for $i \in N$), which

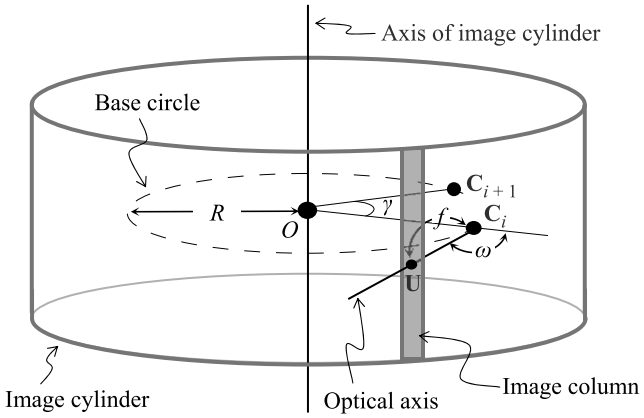

Fig. 8. Imaging geometry of a rotating sensor-line camera.

describes the position of the sensor-line camera. As the camera is rotated 360 degrees along a pre-specified axis, the trajectory of the camera projection centre defines a circle called the *base circle*. Ideally, we assume that the plane of the base circle is perpendicular to the rotation axis, the camera's optical axis remains coplanar to the base circle at all of its positions during the rotation, and the sensor-cell array is configured parallel to the rotation axis.

Through such a 360° rotation, the sensor-cell array of the camera describes (in some abstract sense) a cylindrical surface. The *image cylinder* describes the mathematical abstract location of rotating tri-linear sensor-lines (*tri-linear* because of one line for each of the three RGB colour channels). Parameters M and L are used to describe the *size of a panoramic image*, captured by a rotating sensor-line camera, where M denotes the number of pixel sensors in the line, and L denotes the total number of lines captured for generating this panoramic image.

The rotation axis is the axis of the image cylinder, and the point O on the axis denotes the centre of the base circle. The base circle has the radius R, which is called the *off-axis distance*. The optical axis of a camera at position the $\mathbf{C}_i$ forms a *principle angle* ω with the ray emitting from O and passing through $\mathbf{C}_i$. The angle defined by two adjacent camera positions, e.g., $\angle \mathbf{C}_i O \mathbf{C}_{i+1}$, is called the *angular increment* and is denoted by γ. Moreover, $\mathbf{U}$ defines the point where the optical axis intersects with the image cylinder. The Euclidean distance between $\mathbf{C}_i$ and $\mathbf{U}$ identifies the *focal length* f of the camera at the position $\mathbf{C}_i$. In the ideal case, the focal length f, the principle angle ω, and the angular increment γ are assumed to remain constant during a rotation of a sensor-line camera (e.g., during the recording of one panoramic image).

Fig. 9. Top-left: A commercial catadioptric camera system. Shape of the mirror has been emphasized by an added black curve for clarity. Top-right: An image taken by this panoramic sensor. Bottom: A panorama produced by rectifying the above image (courtesy by N. Ohnishi and A. Torii).

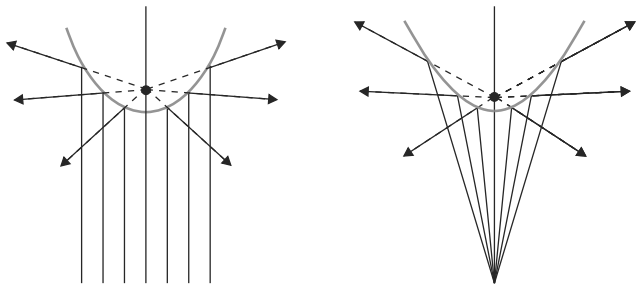

Fig. 10. Catadioptric panoramas: Parabolic mirror with orthographic projection on the left, and hyperboloidal mirror with perspective projection on the right.

This model generalizes various panoramic imaging models [22–25]. The four intrinsic sensor parameters, R, f, ω, and L characterize how a panoramic image is acquired. These notations will be used in later sections when referring to the panoramic images captured by a sensor-line camera.

2.3. Catadioptric camera

A catadioptric camera system enables us to record a full half sphere image in one shot. The word catadioptric means pertaining to or involving both the reflection and the refraction of light. A catadioptric camera system is engineered as a combination of a quadric mirror and a conventional sensor-matrix camera; see Fig. 9 top-left. Catadioptric camera systems provide real-time and highly portable imaging capabilities at an affordable cost. There are only two possible combinations which satisfy a single projection centre constraint: one is a hyperboloidal mirror used in conjunction with a sensor-matrix camera, and the other is the (more theoretical) configuration of a paraboloidal mirror with an (assumed) orthographic projection camera. Both catadioptric sensors allow that all the reflected projection rays intersect at a single point, and, hence possess a simple computational model which supports various applications. Both sensor models are illustrated in Fig. 10. The projection formulas and the mapping of the circular image onto a cylindrical image are reported in Ref. 26. The major shortcoming of such panoramic cameras is the non-uniform and the low image resolution after mapping into a cylindrical form. In the remaining sections, catadioptric camera systems will not be discussed any further due to their limited applications for road views.

3. Camera calibrations

Camera calibration is an essential preliminary step towards many vision-related applications. For driver assistance system applications, fisheye cameras are the most common imaging sensor to be installed in the vehicles. For virtual touring applications, (stereo) high-resolution panoramic images,captured by a rotating line-camera, are able to offer the best viewing impression of a scenario. For large area 3D reconstruction purposes, a panoramic camera device must be able to be mounted on a moving platform, and to capture images during motion. The Ladybug camera system has been designed for this purpose. The following camera calibration, stereo analysis, and camera pose estimation sections will mainly focus on these types of panoramic cameras.

3.1. Fisheye camera

The calibration of fisheye cameras has been researched in several studies and a variety of different calibration methods exist. Some calibration methods consider fisheye lenses as hugely radially distorted lenses and aim to undistort the image to a perspective view [27,28]. Other calibration methods are based on the fisheye camera model described in the previous section [19–21].

The fisheye camera calibration used in Ref. 27 is an adaptation of the pinhole camera calibration method proposed in Ref. 29, which is integrated into the OpenCV[1] library. In the adapted calibration method for fisheye cameras the distortion model is changed. The polynomial degree of the radial distortion model is increased to handle the huge distortion and the tangential distortion is omitted completely. In Fig. 11, a fisheye image is projected on planes at different distances to the image plane. Through the projection a perspective view is obtained. For image areas at wide angles, the image information is hugely stretched and thereby these areas are very error-prone for stereo vision. Furthermore, the image information for narrow angles is compressed if a reasonable image size is used. Due to these properties, the pinhole camera model is not sufficient for the process of image rectification for fisheye cameras if a wide field-of-view is desired.

See Ref. 21 for a calibration method for omnidirectional cameras and an introduction of a MATLAB toolbox. Omnidirectional cameras are defined as “a vision system providing a 360° panoramic view of the scene. Such an enhanced field of view can be achieved by either using catadioptric systems, or employing purely dioptric fisheye lenses” [21]. The applied camera model, referred to as the general *central camera* model, is an extension of the camera model described in Sect. 2.1.

The calibration method distinguishes between two different projection planes, a hypothetical sensor plane and the image plane. The sensor plane is orthogonal to the image sensor[2] with the sensor axis intersecting the sensor plane in its centre. The sensor plane and image plane are related by an affine transformation. The distinction between these two planes is motivated by a possible misalignment of the image centre and the optical axis. Due to the affine transformation, the calibration method is also capable of handling cameras with non-square pixel sensors which leads to a small deformation.

[1] Open Computer Vision Library, http://www.sourceforge.net/projects/opencvlibrary.

[2] In our case, the sensor refers to the fisheye lens and the sensor axis coincides with the optical axis.

Fig. 11. Fisheye images projected onto a plane at different distances to the image plane.

In Fig. 12, the general central camera model is shown for fisheye lenses. Let $\mathbf{P} = (X, Y, Z)^T$ be the scene point and let $\mathbf{p}_s = (u_s, v_s)^T$ be its projection on the sensor plane. Assume that $\mathbf{p}_i = (u_i, v_i)^T$ is the corresponding point on the image plane. The scene point **P** is projected through the sensor onto the point $\mathbf{p}_s$ on the hypothetical sensor plane. The projected point $\mathbf{p}_s$ is then mapped onto the point $\mathbf{p}_i$ on the image plane. The relation between the sensor plane and the image plane is defined by the affine transformation $\mathbf{p}_i = \mathbf{A}\mathbf{p}_s + \mathbf{o}$, where **A** is a 2×2 rotation matrix and **o** is a 2×1 translation vector.

We consider **q** being the vector pointing from the camera centre *O* in the direction of the scene point **P**. By introducing the imaging function $g: \mathbb{R}^2 \mapsto \mathbb{R}^3$ the relation between the vector **q** and the corresponding point $\mathbf{p}_s$ on the sensor plane is obtained. The imaging function *g* is described by the following equation

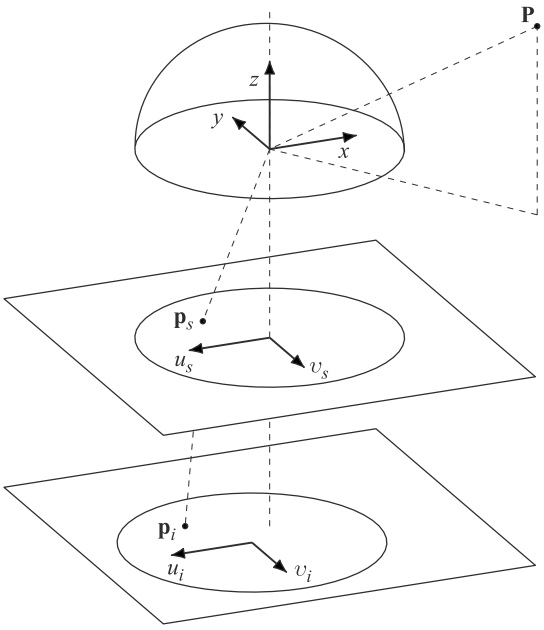

Fig. 12. Camera model used in the calibration method proposed in Ref. 21.

$$\mathbf{q} = g(u_s, v_s) = [u_s, v_s, h(u_s, v_s)]^T . \tag{7}$$

Assuming that the sensors are rotationally symmetrical to the sensor axis, the function *h* depends on u_s and v_s only through $\psi = \sqrt{u_s^2 + v_s^2}$. The function *h* is proposed to have the following polynomial form.

$$h(u_s, v_s) = a_0 + a_1\psi + a_2\psi^2 + \ldots + a_N\psi^N . \tag{8}$$

Consider the scene point **P** being the point on a planar calibration object, such as a checkerboard. Assume that $\mathbf{M} \in \mathbb{R}^4$ describes the same point in the homogeneous world coordinates and that the origin of the world coordinate system is located on the calibration object. Let $\mathbf{Q} \in \mathbb{R}^{3,4}$ be the perspective projection matrix in homogeneous coordinates. The relation between the image point $\mathbf{p}_i$ and the scene point **P** is then defined by the following equation, where $\lambda > 0$ is a scale factor for the direction vector **q**

$$\mathbf{P} = \lambda \cdot \mathbf{q} = \lambda \cdot g(\mathbf{u}_s) = \lambda \cdot g(\mathbf{A}\mathbf{u}_i + \mathbf{o}) = \mathbf{Q}\mathbf{M}. \tag{9}$$

Camera calibration is achieved by estimating the parameters $[\mathbf{A}, \mathbf{o}, a_0, a_1, a_2, \ldots, a_N]$. To calibrate the camera, a planar calibration object with known geometry is captured in different positions or from different viewpoints. The relative orientation of the calibration object to the camera is described by a rotation matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ and a translation vector **t**. The extrinsic parameters, defining the matrix **R** and the vector **t**, can be combined to the matrix **Q** using homogeneous coordinates.

Let I^i be an observation image of the calibration object in the position *i* and let $\mathbf{m}_{ij} = [u_{ij}, v_{ij}]^T$ be the image coordinates of points on the calibration object. The corresponding scene points are given by $\mathbf{M}_{ij} = [X_{ij}, Y_{ij}, Z_{ij}, 1]^T$ expressed in homogeneous world coordinates. From the assumption that the calibration object is planar it follows that $Z_{ij} = 0$. For simplicity[3], we assume that the sensor plane and the image plane coincide. This implies that the affine transformation is described by $\mathbf{A} = \mathbf{I}$ and $\mathbf{o} = \mathbf{0}$. By Eq. (9), the rela-

[3] For a more detailed description see Ref. 21.

tion between the image points $\mathbf{m}_{ij}$ and the scene points $\mathbf{M}_{ij}$ is described by the following equation

$$\lambda_{ij}\mathbf{m}_{ij} = \begin{bmatrix} u_{ij} \\ v_{ij} \\ a_0 + \ldots + a_N \psi_{ij}^N \end{bmatrix} = [\mathbf{r}_1^i \mathbf{r}_2^i \mathbf{r}_3^i \mathbf{t}^i] \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \\ 1 \end{bmatrix} = \tag{10}$$

$$= [\mathbf{r}_1^i \mathbf{r}_2^i \mathbf{t}^i] \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 1 \end{bmatrix}.$$

Multiplying both sides of Eq. (10) vectorial with m_{ij}, the scale factor λ_{ij} can be eliminated

$$\begin{bmatrix} u_{ij} \\ v_{ij} \\ a_0 + \ldots + a_N \psi_{ij}^N \end{bmatrix} \times [\mathbf{r}_1^i \mathbf{r}_2^i \mathbf{t}^i] \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 1 \end{bmatrix} = 0. \tag{11}$$

This relation is then used to create systems of linear equations for determining the calibration parameters. The calibration method is divided into several stages. First, the extrinsic parameters of the calibration object for each observation image I^i are determined. This is achieved by solving a system of linear equations, which is constructed by one of the three resulting equations of Eq. (11). Using the singular value decomposition (SVD), the constructed equation system can be solved by minimizing the least squares criterion.

Next, the intrinsic parameters, the coefficients of the imaging function *g*, are estimated. Another linear equation system is created from Eq. (11). The remaining two equations are rewritten and all unknown variables are stacked into one vector. The least-squares solution of the equation system can be obtained by using the pseudo-inverse. At this stage, the intrinsic and extrinsic parameters are estimated. To achieve a more accurate calibration, the parameters are refined by a linear minimization.

The pixel position of the optical centre is next calculated. The image point of the optical centre is determined in an iterative search. Therefore, the sum of squared reprojection errors (SSRE) is calculated for each point *j* in each observation image *i*. Assuming that the minimum SSRE is obtained at the potential centre, a particular image region is uniformly sampled. For each sampling point, calibration is performed and the SSRE is computed. The potential image region is decreased around the sampling point with the minimal SSRE at the centre. The process is repeated until the difference between two potential optical centres is less than a certain threshold.

The last stage of the calibration method is a non-linear refinement. The optimization process minimizes the error function *E*. The error function *E* sums up the reprojection error of each calibration image. To minimize this error function, the Levenberg-Marquardt algorithm is applied.

3.2. Rotating sensor-line camera

This type of panoramic sensor consists of a line camera and a rotatable platform. The line camera used for panoramic image acquisition can be accurately calibrated in advance by using some commonly available toolbox or during the camera production. Thus, the focal length *f* (in pixel) and central row j_c are assumed to be known. The task here is to calibrate the off-axis distance *R* and the principle angle ω of our sensor setup. Due to the fact that only one single image column is acquired, it is impossible to recover *R* and ω by using the functions provided in standard camera calibration toolboxes.

A *parallel line approach* that uses geometric properties of parallel line segments (calibration lines) available in the scene is presented in this section. A panoramic sensor is well posed if the axis of the image cylinder is parallel to those straight line segments. It is a standard procedure to ensure that both the camera and the rotating rig (and, thus, the panoramic sensor) are both levelled during image acquisition. This requirement is normally achievable by using a "bulls' eye" or a more advanced levelling device. Therefore, it is possible to use any vertical edges available in the scene to recover the sensor parameters. The advantage of this approach is that no extra calibration object is needed. Already available objects may be used instead, and it is also common to attach (circular) calibration labels.

We assume there are at least three pairs of parallel straight line segments in the scene (e.g., straight edges of doors or windows) which are parallel to the axis of the image cylinder. For the purpose of the calibration, it is reasonable to perform the calibration process in an environment which contains a sufficient number of straight lines. For each straight line segment we further assume that both end points are visible (from the camera) and identifiable in the panoramic image, and that we have an accurate measurement of the distance between these two end points. Attaching small circular labels at the end points of line segments in the scene would ease the identification process. Note that the projected line segment (in the panoramic image) should ideally be in a single image column. Finally, we assume that the distance between each selected pair of parallel lines are measurable and known.

The general intention is to find a single linear equation that links 3D geometric scene features to the image cylinder such that (by providing sufficient scene measurements) we are able to calibrate *R* and ω with acceptable accuracy.

1. *Distance constraint*: Any usable straight line segment in the 3D scene is denoted as $\mathcal{L}$ and indexed where needed for the distinction of multiple lines. The (Euclidean) distance of two visible points on the line $\mathcal{L}$ is denoted as *H* (like "height"). The length of a projection of a line segment on the image column *k* is denoted as *h* and measured in pixels. Examples of H_k and corresponding h_k values are illustrated in Fig. 13, where $k \in [1,\ldots,5]$.

The distance D_{kl} between two parallel lines $\mathcal{L}_k$ and $\mathcal{L}_l$ is the length of a line segment that connects both and is per-

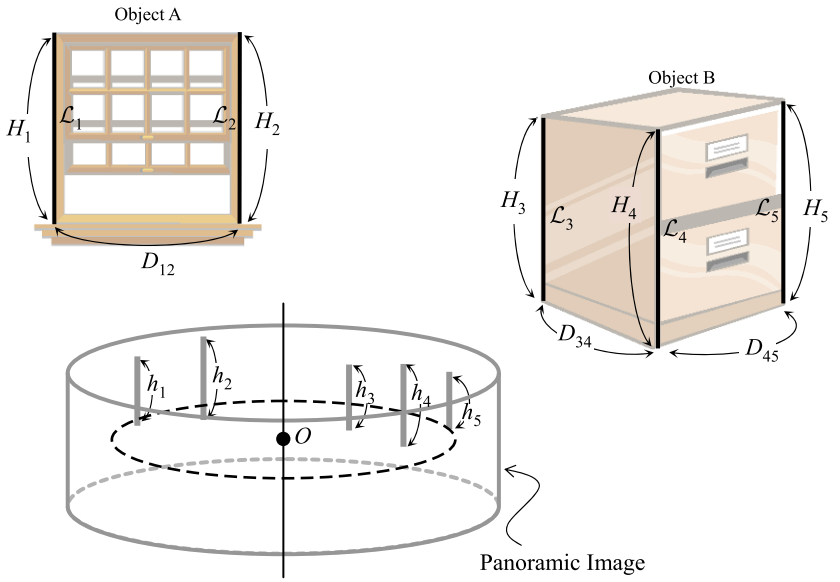

Fig. 13. Configurations of parallel straight lines in 3D scene and on image cylinder.

pendicular to them. If the distance between two straight line segments is available, then we say that both lines form a *pair of lines*. A line segment may be paired up with more than just one other line segment. Figure 13 shows three pairs of lines, namely $(\mathcal{L}_1, \mathcal{L}_2)$, $(\mathcal{L}_3, \mathcal{L}_4)$, and $(\mathcal{L}_4, \mathcal{L}_5)$.

Consider two straight segments $\mathcal{L}_k$ and $\mathcal{L}_l$ in 3D space and the image columns of their projections, denoted as i_k and i_l, respectively. The optical centres associated with image columns i_k and i_l are denoted as $\mathbf{C}_k$ and $\mathbf{C}_l$, respectively. The distance of the two associated image columns is $d_{kl} = |u_k - u_l|$ (in pixels). The angular distance of two image columns, associated with line segments $\mathcal{L}_k$ and $\mathcal{L}_l$, is the angle between line segments $\overline{\mathbf{C}_k O}$ and $\overline{\mathbf{C}_l O}$, where O is the centre of the base circle. We denote the angular distance of a pair $(\mathcal{L}_k, \mathcal{L}_l)$ of lines as θ_{kl}. Examples of angular distances for two pairs of lines are given in Fig. 14. The angular distance θ_{kl} and d_{kl} are related by

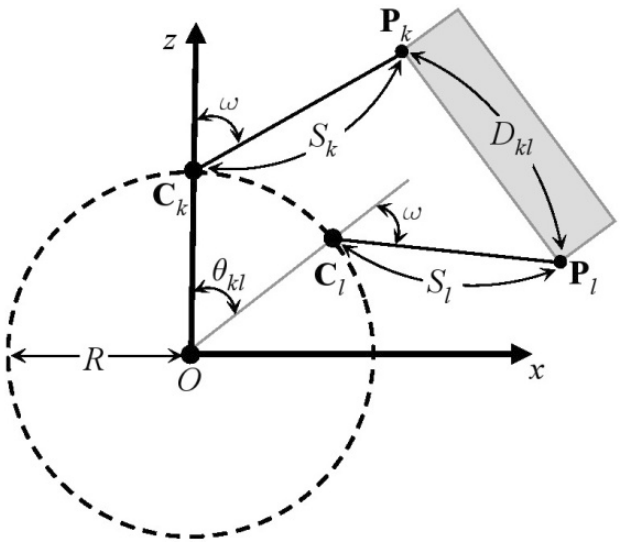

Fig. 14. Coordinate system of a pair of lines.

$$\theta_{kl} = \frac{2\pi d_{kl}}{L},$$

where L is the width of the panorama in pixels.

The distance S between the line segment $\mathcal{L}$ and the associated optical centre (which "sees" this line segment) is defined by the length of the line segment starting at the optical centre, ending on $\mathcal{L}$ and being perpendicular to $\mathcal{L}$. We have that

$$S = \frac{f_\tau H}{h},$$

where f_τ is the pre-calibrated effective focal length of the camera.

2. *Geometric relation*: Now we are ready to formulate a distance constraint by combining all the previously described geometric information. A 2D coordinate system is defined on the xz-plane for every pair of lines $(\mathcal{L}_k, \mathcal{L}_l)$; see Fig. 14. Note that even though all the measurements are defined in a 3D space, the geometric relation of interest can be described in a 2D space since all the straight segments are assumed to be parallel to the axis of the image cylinder. The origin of the coordinate system is O, and the z-axis is incident with the camera focal point $\mathbf{C}_k$. The x-axis is orthogonal to the z-axis and the axis of image cylinder. (This coordinate system coincides with the camera coordinate system previously defined but without the y-axis.) Such a coordinate system is defined for each pair of lines.

The position of $\mathbf{C}_k$ can now be described by coordinates $(0, R)$, and the position $\mathbf{C}_l$ can be described by coordinates $(R\sin\theta_{kl}, R\cos\theta_{kl})$. The intersection point of line $\mathcal{L}_k$ with the xz-plane, denoted as $\mathbf{P}_k$, can be expressed by a sum of vector $\overline{O\mathbf{C}_k}$ and vector $\overline{\mathbf{C}_k\mathbf{P}_k}$. Thus, we have the following

$$\mathbf{P}_k = \begin{bmatrix} S_k \sin\omega \\ R + S_k \cos\omega \end{bmatrix}.$$

Analogously, the intersection point of line $\mathcal{L}_l$ with the xz-plane, denoted as $\mathbf{P}_l$, can be described by a sum of vectors $\overline{O\mathbf{C}_l}$ and $\overline{\mathbf{C}_l\mathbf{P}_l}$. We have the following

$$\mathbf{P}_l = \begin{bmatrix} R\sin\theta_{kl} + S_l \sin(\theta_{kl}+\omega) \\ R\cos\theta_{kl} + S_l \cos(\theta_{kl}+\omega) \end{bmatrix}.$$

The distance D_{kl} between points $\mathbf{P}_k$ and $\mathbf{P}_l$ has been measured. We have the following equation

$$D_{kl}^2 = [S_k \sin\omega - R\sin\theta_{kl} - S_l \sin(\omega+\theta_{kl})]^2 + [R + S_k \cos\omega - R\cos\theta_{kl} - S_l \cos(\omega+\theta_{kl})]^2.$$

This equation can be expanded and rearranged. Finally, we obtain

$$\begin{aligned} 0 = {} & (1-\cos\theta_{kl})R^2 + (S_k+S_l)(1-\cos\theta_{kl})R\cos\omega \\ & -(S_k+S_l)\sin\theta_{kl}R\sin\omega \\ & + \frac{S_k^2 + S_l^2 + D_{kl}^2}{2} - S_k S_l \cos\theta_{kl}. \end{aligned} \tag{12}$$

3. *Error function*: Basically we use Eq. (12) as an error function. The values of S_k, S_l, D_{kl}, and θ_{kl} are known. Thus, Eq. (12) can be arranged into the following linear form

$$A_1X_1 + A_2X_2 + A_3X_3 + A_4 = 0,$$

with coefficients A_n, $n = 1,2,3,4$, defined as follows

$$\begin{aligned} A_1 &= 1-\cos\theta_{kl}, \\ A_2 &= (S_k+S_l)(1-\cos\theta_{kl}), \\ A_3 &= -(S_k-S_l)\sin\theta_{kl}, \\ A_4 &= \frac{S_k^2 + S_l^2 - D_{kl}^2}{2} - S_k S_l \cos\theta_{kl}. \end{aligned}$$

For the three linearly independent variables X_n, $n = 1,2,3$, we have

$$\begin{aligned} X_1 &= R^2, \\ X_2 &= R\cos\omega, \\ X_3 &= R\sin\omega. \end{aligned}$$

In this case we can solve for absolute (not just relative) values R and ω by using all three equations. (If more than three equations are provided, then it is possible to apply a linear least-square technique.) The values of R and ω may be calculated by

$$R = \sqrt{X_1} = \sqrt{X_2^2 + X_3^2}$$

and

$$\omega = \arccos\left(\frac{X_2}{\sqrt{X_1}}\right) = \arcsin\left(\frac{X_3}{\sqrt{X_1}}\right) = \arccos\left(\frac{X_2}{\sqrt{X_2^2+X_3^2}}\right).$$

The given dependencies among variables X_1, X_2, and X_3 define multiple solutions of R and ω. To tackle this multiple-solution problem, we constrain the parameter estimation process further by

$$X_1^2 = X_2^2 + X_3^2,$$

which is valid because

$$R^2 = (R\cos\omega)^2 + (R\sin\omega)^2.$$

Assume that N copies of Eq. (12) are given. We want to minimize the following

$$\sum_{n=1}^{N} (A_{1n}X_1 + A_{2n}X_2 + A_{3n}X_3 + A_{4n})^2 \tag{13}$$

subject to the equality constraint $X_1 = X_2^2 + X_3^2$, where the values of A_{1n}, A_{2n}, A_{3n}, and A_{4n} are calculated based on measurements in the real scene and in the image. We also know that $X_1 = R^2$, $X_2 = R\cos\omega$, and $X_3 = R\sin\omega$. Now, the values of R and ω can be uniquely (!) calculated as

$$R = \sqrt{X_1}$$

and

$$\omega = \arccos\left(\frac{X_2}{\sqrt{X_1}}\right).$$

Note that even though the additional constraint forces a use of a non-linear optimization method, the accuracy of the method remains at the quality level of a linear parameter estimation procedure.

4. Stereo analysis

Multiple images of different views are required in order to obtain depth information from vision-based approaches. This section presents the geometrical relationship between two panoramic imaging cameras, expressed by epipolar lines or curves. Stereo matching algorithms are applied along derived epipolar lines or curves to identify pairs of corresponding image points in two images. Depth information of scene points is then calculated by a triangulation defined by projection centres and corresponding points.

4.1. Fisheye camera

We consider a stereo vision setup with two fisheye cameras which capture spherical images. This supports wide-angle stereo vision. The basic idea of this type of *spherical stereo vision* is "to straighten" the epipolar curves such that state-of-the-art stereo matching algorithms can be applied. In the following, a spherical stereo vision is presented as proposed in Ref. 30. Straight epipolar lines are obtained by sampling along the epipolar curves in the spherical image, and by mapping sampling points onto straight lines. This projection ensures straight epipolar lines in the resulting image.

The captured fisheye images are projected onto the surface of a unit sphere assumed to be around the camera's projection centre. The geometric relation between fisheye images and spherical images is identified by calibration. Consider two spherical cameras and the scene point **P**. Let the projections of the scene point **P** onto the spherical

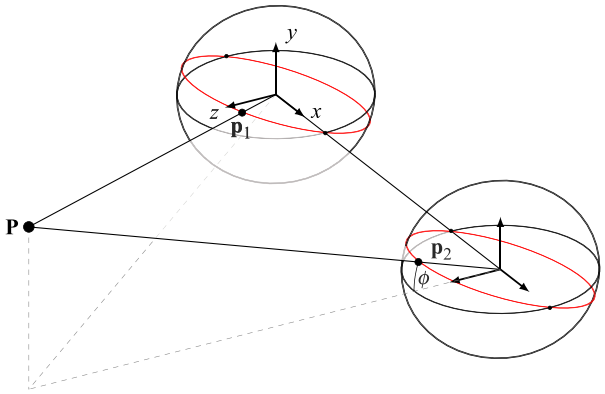

Fig. 15. Spherical stereo.

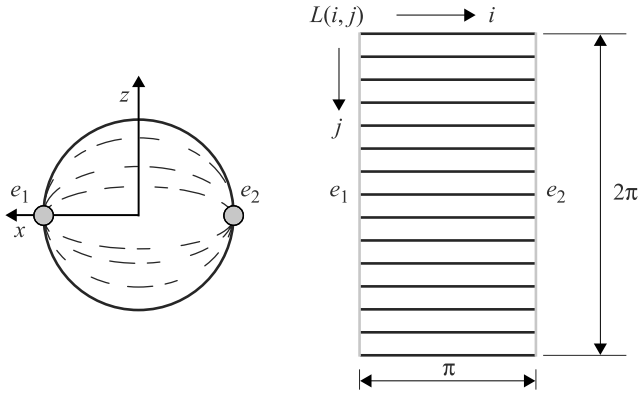

Fig. 16. Latitude-longitude sampling [30].

images be denoted by $\mathbf{p}_1$ and $\mathbf{p}_2$. Assume aligned camera coordinate systems. Recall that an epipolar plane is defined by the scene point **P** and the two camera projection centres. Epipolar lines are determined by intersections of the spherical images and the epipolar plane. Due to the fact that the camera centres lie in the epipolar plane, and that the spherical image points lie on the sphere around the camera centre, the epipolar curves are great circles around the camera centres. In Fig. 15, two spherical cameras capturing the scene point **P** are shown. The epipolar curves in the spherical images are highlighted in red.

Knowing the shape of the epipolar curves in the spherical images, a projection function that maps the epipolar curves onto straight lines can be derived. Assume that the baseline is collinear with the x-axis of the camera coordinate systems. Thus, the epipoles are located at the intersections of the spherical images with the x-axis. Let the spherical image points $\mathbf{p}_1$ and $\mathbf{p}_2$ be expressed in spherical coordinates with the zenith in the positive x-direction. Note that the longitude angle ϕ lies in the yz-plane and that the longitude angle is equal for every pair of corresponding image points. Furthermore, the longitude angle is fixed to the epipolar plane between the epipoles. In Fig. 15, the longitude angle ϕ is shown for the spherical image point $\mathbf{p}_2$.

The epipolar curves can be sampled between the epipoles by changing the latitude angle while the longitude angle stays fixed. By mapping the sampling points onto straight lines, an image is obtained with straight horizontal epipolar lines. For a horizontal camera setup, the resulting image shows a horizontal field-of-view of 180°. The projection is referred to as latitude-longitude sampling. Figure 16 illustrates the latitude-longitude sampling.

Next, a mathematical description of the latitude-longitude sampling is presented. Consider $\mathbf{L}(i, j)$ as the latitude-longitude sampled image, where i and j denote the pixel position in the sampled image. Assume that the resulting image has a resolution $m \times n$ of pixels and that $\mathbf{p} = (x, y, z)$ denotes the spherical image point. The relation between the spherical image point **p** and a point in the latitude-longitude sampled image is described by the following equations

$$x = \cos\left(\frac{i}{m}\pi\right), \tag{14}$$

$$y = \sin\left(\frac{i}{m}\pi\right)\cos\left(\frac{j}{n}2\pi\right), \tag{15}$$

$$z = \sin\left(\frac{i}{m}\pi\right)\sin\left(\frac{j}{n}2\pi\right). \tag{16}$$

By using this relation, the intensity value for each pixel position (i, j) in the resulting image can be obtained at the spherical image point $\mathbf{p} = (x, y, z)$. The latitude-longitude sampled image is derived by sampling the intensity value for each image point. As aforementioned, the mapping from spherical images to fisheye images is determined during the calibration process. Thus, the captured fisheye images can be transformed into latitude-longitude sampled images. Figure 17 shows a pair of stereo images recorded with a horizontal camera setup.The images capture the field-of-view of 180° both in horizontal and vertical direction.

Due to the straight horizontal epipolar lines in the resulting images, state-of-the-art stereo matching algorithms can be applied without a need for alterations. By comparing two latitude-longitude sampled images, a disparity map is obtained that differs from conventional disparity maps. We refer to the obtained disparity maps as spherical disparity maps. The spherical disparity maps denote the horizontal displacement of the image points in the longitude-latitude sampled image. This horizontal displacement is linearly related to the latitude difference of the spherical image points in spherical coordinates.

For wide-angle stereo vision, distance approximation as known from conventional stereo vision is insufficient. We quote in this paragraph from Ref. 30: "Two problems occur when this approach is applied to fisheye cameras with a wider-than-hemispherical FOV. Firstly, computational errors in the disparity may be caused if "a very small disparity is computed from two very large horizontal coordinates. The aforementioned problem will occur if the conventional pinhole camera model is applied to semi-spherical-FOV fisheye stereo images". Secondly, the depth "is usually used to describe how far an environment point is located from the camera... This is because the depth of an environment point imaged by a narrow-FOV camera is similar to the distance of the point from the camera. However, the depth may be significantly different from the distance of an environment

Fig. 17. Latitude-longitude sampled stereo images.

point to the camera for spherical stereo, because an environment point may be in any direction relative to a spherical camera and, thus, may have a great distance but a small depth value." Therefore, the spherical disparity must be defined accordingly.

Consider an epipolar plane in a stereo vision setup with two spherical cameras. Let **P** denote a scene point in the epipolar plane. Assume that the projections of **P** are given by $\mathbf{p}_1$ and $\mathbf{p}_2$. Let the latitude angle of $\mathbf{p}_1$ and $\mathbf{p}_2$ be denoted by θ_1 and θ_2, where the zenith is in positive x-direction. Figure 18 shows the epipolar plane of a scene point **P** in a spherical stereo setup.

A spherical camera was introduced by using a unit sphere around the camera centre (e.g., the spherical image lies on the surface of this unit sphere). We normalize the disparity map so that the disparities describe the displacement on the surface of the unit sphere. The displacement on the unit sphere coincides with the latitude difference of the spherical image points. Consider that the latitude-longitude sampled images capture a field-of-view of π in the horizontal direction and are w pixels wide. The normalization from pixel differences to angular differences is described by the following equation, where d_n denotes the normalized disparity and d the disparity expressed in pixels

$$d_n = \frac{d}{w}\pi. \tag{17}$$

With this normalization, a relation between the latitude angles of the spherical image points is given. The normalized displacement d_n coincides with the difference of the latitude angles θ_1 and θ_2.

$$d_n = \theta_1 - \theta_2. \tag{18}$$

By applying this relation, we calculate the distance of a scene point to a reference camera. Assume that ρ_1 and ρ_2 denote the distance of the scene point **P** to the camera centres O_1 and O_2, respectively. Distances ρ_1 and ρ_2 are defined by the following equations

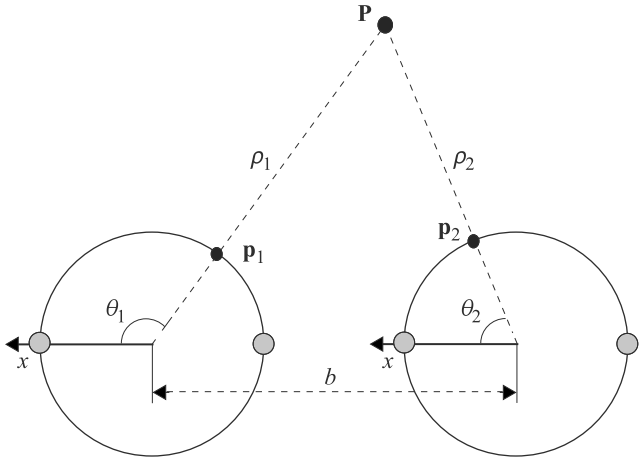

Fig. 18. Disparity definition [30].

$$\rho_1 = b\frac{\sin(\theta_2)}{\sin(d_n)} = b\frac{\sin(\theta_1 - d_n)}{\sin(d_n)}, \tag{19}$$

$$\rho_2 = b\frac{\sin(\pi - \theta_1)}{\sin(d_n)} = b\frac{\sin(\theta_2 + d_n)}{\sin(d_n)}. \tag{20}$$

By changing orientations of the camera coordinate systems, the spherical stereo vision approach can also be applied to a vertical camera setup. Due to the epipoles, spherical stereo vision in a horizontal camera setup is limited in its field-of-view to in horizontal direction. Whereas in a vertical camera setup, stereo vision for the complete horizontal view can be obtained. Limitations are "swapped" for the field-of-view in the vertical direction.

4.2. Rotating sensor-line camera

Consider an arbitrary pair of multi-view panoramas $E_{\mathcal{P}1}$ (R_1, f_1, ω_1, L_1) and $E_{\mathcal{P}2}$ (R_2, f_2, ω_2, L_2). Subscripts $_1$ and $_2$ are used to indicate that these values of camera parameters, as-

sociated with these two panoramas, may be different. A multi-view panoramic pair defines a *general epipolar curve equation* because the epipolar geometry of other types of cylindrical panoramic pairs can be derived from this equation. Given is the image point (x_1, y_1) on $E_{\mathcal{P}1}$. This image point is a projection of an (unknown) point in a 3D space, which (assuming it is visible) will project into the corresponding image point (x_2, y_2) in $E_{\mathcal{P}2}$. Without knowing the projected 3D point, the knowledge about sensor parameters and (x_1, y_1) allows us to specify the possible locations of (x_2, y_2).

The origin of the sensor coordinate system, defined for image $E_{\mathcal{P}1}$, is denoted by O_1, and for the image $E_{\mathcal{P}2}$ it is denoted by O_2. Let a 3×3 rotation matrix $\mathbf{R}$ and a 3×1 translation vector $\mathbf{t}$ specify the orientation and the position of the O_2 coordinate system with respect to the O_1 coordinate system. The rotation matrix $\mathbf{R}$ is decomposed into its three row vectors as $[\mathbf{r}_1^T\ \mathbf{r}_2^T\ \mathbf{r}_3^T]^T$, and the translation vector is represented by its three elements as $(t_x, t_y, t_z)^T$. The general epipolar curve equation is then stated as in the following paragraph.

Let (x_1, y_1) and (x_2, y_2) denote the image coordinates of the projection of a 3D point in the source image $E_{\mathcal{P}1}$ and the destination image $E_{\mathcal{P}2}$, respectively. Consider x_1 and y_1 as being given. Let $\alpha_1 = (2\pi x_1)/L_1$, $\alpha_2 = (2\pi x_2)/L_2$, $\delta_1 = (\alpha_1 + \omega_1)$, $\delta_2 = (\alpha_2 + \omega_2)$, and $\beta_1 = \arctan(y_1/f_1)$. The corresponding epipolar curve on the destination image $E_{\mathcal{P}2}$ can be represented by the equation

$$y_2 = \frac{f_2\,\mathbf{r}_2^T \cdot \mathbf{V}}{\sin\delta_2\,\mathbf{r}_1^T \cdot \mathbf{V} + \cos\delta_2\,\mathbf{r}_3^T \cdot \mathbf{V} - R_2\,\cos\omega_2},$$

which is only valid if the value of the denominator is greater than zero. The vector $\mathbf{V}$ is defined as follows

$$\mathbf{V} = \mathbf{A} + \frac{R_2\,\sin\omega_2 + \cos\delta_2\,\mathbf{r}_1^T \cdot \mathbf{A} - \sin\delta_2\,\mathbf{r}_3^T \cdot \mathbf{A}}{\sin\delta_2\,\mathbf{r}_3^T \cdot \mathbf{B} - \cos\delta_2\,\mathbf{r}_1^T \cdot \mathbf{B}}\,\mathbf{B},$$

where

$$\mathbf{A} = \begin{pmatrix} R_1\sin\alpha_1 - t_x \\ -t_y \\ R_1\cos\alpha_1 - t_z \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} \sin\delta_1\cos\beta_1 \\ \sin\beta_1 \\ \cos\delta_1\cos\beta_1 \end{pmatrix}.$$

Figure 19 illustrates an example of epipolar curves for a general case of a pair of multi-view panoramic images. The shown curves demonstrate the geometric complexity of those objects.

The internal sensor parameters of $E_{\mathcal{P}1}$ (R_1, f, ω_1, L) and $E_{\mathcal{P}2}$ (R_2, f, ω_2, L) are as follows: R_1 = 500 mm, ω_1 = 45°, R_2 = 250 mm, ω_2 = 65°, f = 35 mm, and L = 1,000 pixels. The affine transform between both sensor coordinate systems (associated with these two panoramas) can be described by the translation vector $\mathbf{t}$ and the rotation matrix $\mathbf{R}$. Let $\mathbf{t} = (2000, 300, 1500)^T$ in mm and

$$\mathbf{R} = \mathbf{R}_x(-1°)\mathbf{R}_y(-1°)\mathbf{R}_z(2°),$$

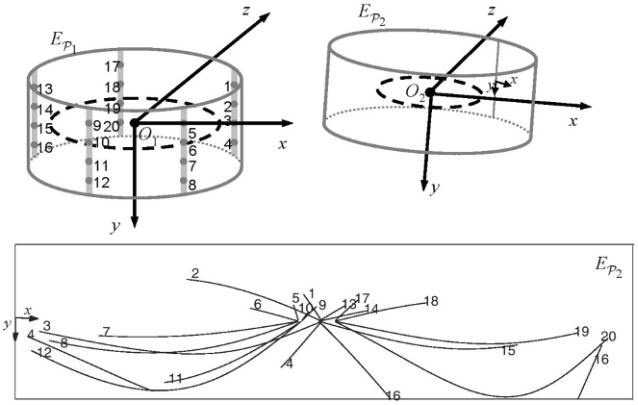

Fig. 19. A pair of multi-view panoramas and epipolar curves in the destination image originating from 20 selected points in the source image.

where $\mathbf{R}_x$, $\mathbf{R}_y$, and $\mathbf{R}_z$ are the rotation matrixes with respect to each of the three axes. The top of Fig. 19 shows the specified geometric relation between these two panoramas in a 3D space.

20 points have been selected in the image $E_{\mathcal{P}1}$, labelled by numbers. The corresponding epipolar curves in the image $E_{\mathcal{P}2}$, also labelled by numbers, are shown at the bottom of Fig. 19. An interesting observation is that epipolar curves do cross each other (in this case), and these intersections are not necessarily epipoles (for example, curves 3 and 8). This is a new situation compared to epipolar geometry of a pinhole-model camera where epipolar lines only intersect at epipoles. However, if we only consider epipolar curves associated with image points on the same column of the source image $E_{\mathcal{P}1}$, for example curves labelled 17, 18, 19 and 20, then they only intersect (if at all) at epipoles.

5. Camera pose recovery

In order to understand scene geometry from multiple panoramic images, it is essential to know the relative camera orientations. Having calibrated each camera accurately, the spatial relation between two camera locations can be calculated.

5.1. Fisheye camera

In this section, a method is proposed that precisely calculates the relative camera orientation of a multicamera system. The method is divided into two steps. First, an estimation of the spatial relation between the two cameras is calculated. The estimation is obtained from the extrinsic parameters of the calibration object. In the second step the estimation is optimized. Using a non-linear refinement, the optimal camera orientation is calculated with respect to a modelling function.

Consider a stereo vision system with two cameras capturing a calibration object. Assume that the cameras are located at O_1 and O_2 and that the cameras are displaced by base distance b. Let $\mathbf{P} \in \mathbb{R}^3$ be the scene point located on the

calibration object. Note that $\mathbf{P}$ can be expressed with respect to three different coordinate systems, two camera coordinate systems and a world coordinate system located at the calibration object. In the following a scene point is annotated with a superscript indicating the respective coordinate system in which the point is expressed. Let the world coordinate system be located at O_w on the calibration object. Assume that the spatial orientation of the two cameras is described by the translation vector $\mathbf{t}_{12} \in \mathbb{R}^3$ and the rotation matrix $\mathbf{R}_{12} \in \mathbb{R}^{3\times3}$. In Fig. 20, the relation between two camera coordinate systems and the reference coordinate system attached to a calibration object is shown.

The transformation between the two camera coordinate systems is defined by the relative camera orientation. Thus, the scene point $\mathbf{P}^{(1)}$ can be transformed into $\mathbf{P}^{(2)}$ and vice versa. The following equations describe the coordinate system change between the camera coordinate systems

$$\mathbf{P}^{(2)} = \mathbf{R}_{12}\mathbf{P}^{(1)} + \mathbf{t}_{12}, \tag{21}$$

$$\mathbf{P}^{(1)} = \mathbf{R}_{12}^{-1}(\mathbf{P}^{(2)} - \mathbf{t}_{12}). \tag{22}$$

Assume that the extrinsic parameters of a calibration object are defined by the translation vectors $\mathbf{t}_k \in \mathbb{R}^3$ and the rotation matrices $\mathbf{R}_k \in \mathbb{R}^{3\times3}$, where $k \in \{1,2\}$ refers to the respective cameras. Knowing the extrinsic parameters, the scene point $\mathbf{P}^{(w)}$ on the calibration object can be expressed in camera coordinates. The transformation of the scene point $\mathbf{P}$ from world coordinates to camera coordinates is described by the following equation

$$\mathbf{P}^{(k)} = \mathbf{R}_k\mathbf{P}^{(w)} + \mathbf{t}_k, \quad k \in \{1,2\}. \tag{23}$$

The solutions of Eq. (23) for $\mathbf{P}^{(w)}$ for both camera coordinate systems contain the relative camera orientation implicitly. The relative camera orientation can be obtained by setting the transformations from camera to world coordinates equal to each other

$$\mathbf{P}^{(w)} = \mathbf{R}_1^{-1}(\mathbf{P}^{(1)} - \mathbf{t}_1) = \mathbf{R}_2^{-1}(\mathbf{P}^{(2)} - \mathbf{t}_2). \tag{24}$$

By solving Eq. (24) for $\mathbf{P}^{(2)}$, the coordinate transformation from Camera 1 to Camera 2 can be expressed in terms of the extrinsic parameters.

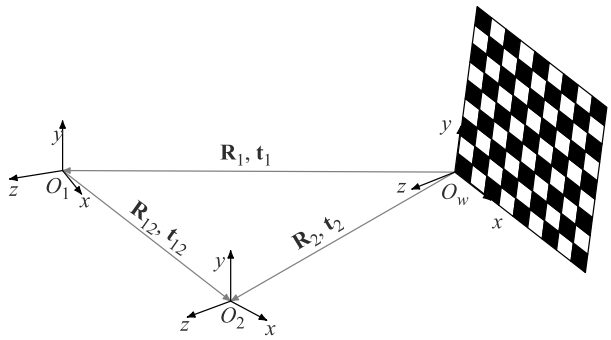

Fig. 20. Relation between two camera coordinate systems and a world coordinate system with its origin on a planar calibration object.

$$\mathbf{P}^{(2)} = \mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{P}^{(1)} - \mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{t}_1 + \mathbf{t}_2. \tag{25}$$

Based on Eqs. (21) and (25), the relative camera orientation can be determined by the following

$$\mathbf{R}_{12} = \mathbf{R}_2\mathbf{R}_1^{-1}, \tag{26}$$

$$\mathbf{t}_{12} = \mathbf{t}_2 - \mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{t}_1. \tag{27}$$

From the camera calibration process, the extrinsic parameters of both cameras have been recovered, and, thus, the relative camera orientation can be estimated from Eqs. (26) and (27).

In the next step, a non-linear optimization is derived to refine the estimated camera orientation. Given an estimation of the relative camera orientation, the scene point $\mathbf{P}^{(1)}$ can be transformed into $\mathbf{P}^{(2)}$ and vice versa. Let $\tilde{\mathbf{R}}_{12}$ and $\tilde{\mathbf{t}}_{12}$ denote the estimation of the relative camera orientation. Assume that $\mathbf{R}_k$ and $\mathbf{t}_k$ are the extrinsic parameters of camera $k \in \{1,2\}$. The transformation between the camera coordinate systems can be described with the camera orientation, estimated by $\tilde{\mathbf{R}}_{12}$ and $\tilde{\mathbf{t}}_{12}$, as follows

$$\tilde{\mathbf{P}}^{(2)} = \tilde{\mathbf{R}}_{12}\mathbf{P}^{(1)} + \tilde{\mathbf{t}}_{12} = \tilde{\mathbf{R}}_{12}(\mathbf{R}_1\mathbf{P}^{(w)} + \mathbf{t}_1) + \tilde{\mathbf{t}}_{12}, \tag{28}$$

$$\tilde{\mathbf{P}}^{(1)} = \tilde{\mathbf{R}}_{12}^{-1}(\mathbf{P}^{(2)} - \tilde{\mathbf{t}}_{12}) = \tilde{\mathbf{R}}_{12}^{-1}(\mathbf{R}_2\mathbf{P}^{(w)} + \mathbf{t}_1 - \tilde{\mathbf{t}}_{12}). \tag{29}$$

Due to inaccuracies of the recovered camera extrinsic parameters and the estimated camera orientation, the spatial positions of $\mathbf{P}^{(k)}$ and $\tilde{\mathbf{P}}^{(k)}$ differ slightly. The spatial differences for all the calibration points are calculated. By minimizing the spatial displacement with the least squares criterion, the optimal camera orientation is obtained. In Fig. 21 the spatial displacement of a calibration object is shown. The positions of the black and the grey checkerboard should be the same. The spatial orientation of both checkerboards is calculated with respect to the camera located at O_1.

The relative camera orientation can be expressed by six parameters: three parameters for the translation vector $\mathbf{t}_{12}$ and three parameters for the rotation vector $\mathbf{r}_{12}$. We combine the parameters of the camera orientation to the six-dimensional parameter vector $\mathbf{x}$ by stacking the vectors $\mathbf{r}_{12}$ and $\mathbf{t}_{12}$. This defines

$$\mathbf{x} = \begin{pmatrix} \mathbf{r}_{12} \\ \mathbf{t}_{12} \end{pmatrix}. \tag{30}$$

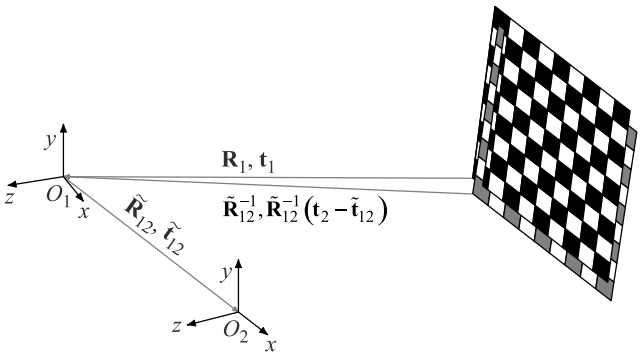

Fig. 21. Spatial displacement of planar calibration object.

The optimal camera orientation is obtained by minimizing an error function depending on the parameter vector $\mathbf{x}$. We establish the error function $e(\mathbf{x})$ as the function that calculates the spatial differences of all calibration points estimated from both cameras. The spatial differences are obtained from Eqs. (28) and (29).

Assume that the scene point $\mathbf{P}_{ij}^{(k)}$ describes the j-th scene point of the calibration object in the position i with respect to the camera coordinate system k. Let $\tilde{\mathbf{P}}_{ij}^{(k)}$ be the transformation of the scene point $\mathbf{P}_{ij}^{(k)}$ obtained from Eqs. (28) and (29). Let us consider the calibration object in all positions i. Therefore, we annotate the extrinsic parameters of the calibration object with a superscript that indicates the position i. The extrinsic parameters of the calibration object in position i are denoted by the rotation matrix $\mathbf{R}_k^{(i)}$ and the translation vector $\mathbf{t}_k^{(i)}$, where k relates to the respective camera.

The transformation $\tilde{\mathbf{P}}_{ij}^{(k)}$ of the scene points are dependent on the relative camera orientation. During the optimization process the relative camera orientation is adapted. Therefore, the transformed scene points $\tilde{\mathbf{P}}_{ij}^{(k)}$ need to be written as a function of the parameter vector $\mathbf{x}$. Let us introduce an auxiliary function $\psi(\mathbf{x}, i, j, k)$ that determines the scene points $\tilde{\mathbf{P}}_{ij}^{(k)}$ from Eqs. (28) and (29) depending on the vector $\mathbf{x}$. The following equation defines the function $\psi(\mathbf{x}, i, j, k)$, where $\mathbf{R}(\mathbf{x})$ calculates the rotation matrix $\mathbf{R}_{12}$ and $\mathbf{t}(\mathbf{x})$ calculates the translation vector $\mathbf{t}_{12}$ depending on the parameter vector $\mathbf{x}$. We have that $\psi(\mathbf{x}, i, j, k)$ equals

$$\mathbf{R}(\mathbf{x})^{-1}(\mathbf{R}_k^{(i)}\mathbf{P}_{ij}^{(w)} + \mathbf{t}_k^{(i)}) - \mathbf{R}(\mathbf{x})^{-1}\mathbf{t}(\mathbf{x}), \tag{31}$$

for $k = 1$, or equals

$$\mathbf{R}(\mathbf{x})(\mathbf{R}_k^{(i)}\mathbf{P}_{ij}^{(w)} + \mathbf{t}_k^{(i)}) + \mathbf{t}(\mathbf{x}), \tag{32}$$

for $k = 2$.

Note that we have N observation image pairs with M calibration points in each image. For each corresponding point pair two error equations can be obtained by calculating the differences between $\mathbf{P}_{ij}^{(k)}$ and $\tilde{\mathbf{P}}_{ij}^{(k)}$ for both cameras. The points $\mathbf{P}_{ij}^{(k)}$ are determined from the extrinsic parameters of the calibration object with respect to the camera k. The points $\tilde{\mathbf{P}}_{ij}^{(k)}$ are estimated from the extrinsic parameters relating to the other camera and the relative camera orientation. Thus, we have $2MN$ equations to optimize the relative camera orientation which is described by the parameter vector $\mathbf{x}$.

We define the error function $e(\mathbf{x})$ as the function that calculates the spatial differences of the scene points $\mathbf{P}_{ij}^{(k)}$ and $\tilde{\mathbf{P}}_{ij}^{(k)}$ for each calibration point pair. The error function $e(\mathbf{x})$: $\mathbb{R}^6 \mapsto \mathbb{R}^{2MN}$ that is to be minimized is described by the following equation where $k \in \{1,2\}$ denotes the camera, $i \in \{1, \ldots, N\}$ denotes the observation image and $j \in \{1, \ldots, M\}$ refers to the point on the calibration object.

$$e(\mathbf{x}) = \begin{pmatrix} \mathbf{P}_{11}^{(k)} - \psi(\mathbf{x},1,1,k) \\ \vdots \\ \mathbf{P}_{ij}^{(k)} - \psi(\mathbf{x}, i, j, k) \\ \vdots \\ \mathbf{P}_{NM}^{(k)} - \psi(\mathbf{x}, N, M, k) \end{pmatrix}. \tag{33}$$

At the global minimum of $e(\mathbf{x})$ the optimal camera orientation is obtained. For the parameter vector $\mathbf{x}_{\min}$ the spatial differences between the calibration object positions are minimal. A scene point expressed in one camera coordinate system can be transformed into the other camera coordinate system with a minimal error. The global minimum $\mathbf{x}_{\min}$ can be obtained by using a non-linear optimization technique, such as the Levenberg-Marquardt algorithm.

$$\min_{\mathbf{x} \in \mathbb{R}^6} \|e(\mathbf{x})\|_2^2 . \tag{34}$$

Note that the intrinsic parameters of both cameras and the extrinsic parameters of the calibration object for each position are constant during the optimization process. Only the relative camera orientation is adapted during the minimization process.

5.2. Rotating sensor-line camera

While capturing cylindrical panoramic images using a rotating sensor-line camera system, it is a standard practice to aim for a set of levelled panoramas. The only constraint is that all associated axes of image cylinders have to be parallel, to be guaranteed by a leveller. Figure 22 sketches a levelled pair of panoramas.

Levelled panoramas are common for virtual navigation [31,32] or reconstruction [22,33] of large scale environments. Levelled panoramas support large "overlapping" fields of view. The larger the common field of view, the higher the probability that object surfaces are visible in more than one panorama. This supports more reliable stereo reconstruction and smooth view-transitions between multiple panoramas in, for example, a walk-through simulation.

The sensor pose estimation criteria of a levelled pair is specified in the following. Both levelled panoramas are acquired by two sensors with the same intrinsic parameters, and the sensor poses are related by the single rotation angle ϕ with respect to the rotation axis and the translation vector $(t_x, t_y, t_x)^T$. The five variables to be recovered, are $X_1 = \cos\phi$, $X_2 = \sin\phi$, $X_3 = t_x$, $X_4 = t_z$, and $X_5 = t_y$. In the equational system, the following nine coefficients are also used

$$\begin{aligned}
c_{1i} &= y_{2i} R \sin(\delta_{1i} - \alpha_{2i}) + y_{1i} R \sin(\delta_{2i} - \alpha_{1i}) \\
c_{2i} &= y_{1i} R \cos(\delta_{2i} - \alpha_{1i}) - y_{2i} R \cos(\delta_{1i} - \alpha_{2i}) \\
c_{3i} &= -y_{2i} \cos\delta_{1i} \\
c_{4i} &= y_{2i} \sin\delta_{1i} \\
c_{5i} &= y_{1i} \cos\delta_{2i} \\
c_{6i} &= -y_{1i} \sin\delta_{2i} \\
c_{7i} &= f \sin(\alpha_{2i} - \alpha_{1i}) \\
c_{8i} &= f \cos(\alpha_{2i} - \alpha_{1i}) \\
c_{9i} &= -(y_{1i} + y_{2i}) R \sin\omega,
\end{aligned}$$

where $\alpha_{ki} = \dfrac{2\pi x_{ki}}{L}$, $\delta_{ki} = (\alpha_{ki} + \omega)$, and $k = 1$ or 2.

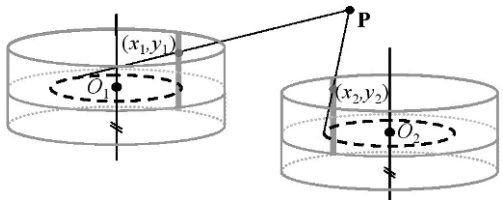

Fig. 22. A pair of levelled panoramas and a pair of corresponding image points.

Given a set of the corresponding pairs of points (x_{1i}, y_{1i}) and (x_{2i}, y_{2i}), where $i = 1,2,\ldots,n$, the values of ϕ, t_x, t_y, and t_z can be estimated by minimizing the following sum

$$\sum_{i=1}^{n}(c_{1i}X_1 + c_{2i}X_2 + c_{3i}X_3 + c_{4i}X_4 + c_{5i}X_1X_3 + c_{6i}X_1X_4 + c_{7i}X_1X_5 + c_{6i}X_2X_3 - c_{5i}X_2X_4 + c_{8i}X_2X_5 + c_{9i})^2$$

subject to the constraints $X_1^2 + X_2^2 = 1$, $X_1^2 \le 1$, and $X_2^2 \le 1$.

The above equation can be derived from a pair of the corresponding image points (x_1, y_1) and (x_2, y_2) of levelled cylindrical panoramas $E_{\mathcal{P}1}$ and $E_{\mathcal{P}2}$, respectively. Given x_1 and y_1, the corresponding epipolar curve in $E_{\mathcal{P}2}$ can be expressed as follows

$$\begin{aligned} &y_2 R\sin(\alpha_1 + \omega - \alpha_2 - \phi) - y_2 R\sin\omega \\ &-y_2\cos(\alpha_1 + \omega)t_x + y_2\sin(\alpha_1 + \omega)t_z \\ &+f\sin(\alpha_2 - \alpha_1 + \phi)t_y - y_1 R\sin\omega \\ &+y_1 R\sin(\alpha_2 - \alpha_1 + \omega + \phi) \\ &+y_1\cos(\alpha_2 - \omega + \phi)t_x \\ &-y_1\sin(\alpha_2 + \omega + \phi)t_z = 0. \end{aligned} \tag{35}$$

This result can be derived directly from the general epipolar equation. The cost function is defined by the row difference between an actual corresponding image point and the point on the same column incident with the epipolar curve. In short, by algebraic rearrangements of Eq. (35), the second-order algebraic representation for the minimization can be obtained.

6. Camera trajectory estimation

In the applications of street visualization or a 3D city street reconstruction, a dense set of panoramic images is required. The common approach to achieve such requirement is by mounting a panoramic camera on the top of a vehicle and capture spherical images during motion. Several specially designed multi-camera systems are commercially available for this purpose. The major focus while considering a large set of panoramic images is to estimate the camera's moving trajectory based on the pairwise camera relative pose estimation results.

The spherical representation of the resulting panoramic images is preferred in this case, rather than a cylindrical image. It is because the pairwise relative camera pose can be recovered by directly adopting the well-known camera self--calibration methods (such as the 8-point algorithm) originally developed for the "normal" planar images. The reason for that is because a single projection centre has been assumed in the spherical image representation. Those algorithms have shown to be robust to error, and it is possible to achieve real-time performance speed. Note, in order to adopt those camera self-calibration algorithms to the panoramic image situation, a single-centre of projection constraint must be ensured. This means, that the accuracy of the mapping, from a set of planar images captured by the multi--camera system, to the resulting spherical panoramic image is critical. Usually this mapping function is provided along with the purchase of the panoramic camera.

The framework of camera trajectory estimation approach is as follows: First, a feature detection algorithm is applied to each of the source panoramic images, and, then, a feature point matching search is performed between each pair of successive images. The matching results enable us to recover the essential matrix describing the spatial relationship between two imaging coordinate systems. The relative orientation, represented by a rotation matrix, and position, represented by a unit vector, of two successive panoramic images can be derived from the essential matrix. Camera trajectory can be recovered through point cloud reconstruction of the scene and bundle adjustment based on the available global positioning system (GPS) information.

The image point correspondences can be established by scale-invariant feature transform (SIFT) feature detection plus SIFT-based matching. Although speeded up robust features (SURF) was claimed to have similar performance to SIFT, while at the same time being faster, however, in the case of complex environmental objects such as trees, our experiential results showed that SURF-based matching led to more false matches than the SIFT approach within the image regions of plants. When street trees are often unavoidable and this task is usually performed off-line (in such case, computation time is not a critical issue), it is a better choice to rely on SIFT-based matching result. A single threshold D_{SIFT} is used to determine if a match is acceptable in the SIFT-based matching algorithm. The smaller the value, the more image correspondences are identified, and the higher possibility that the result would include false matches.

The 8-point algorithm is employed to estimate the essential matrix. The algorithm utilizes the epipolar constraint, more specifically the coplanarity constraint, to solve for the essential matrix. The coplanarity constraint can be assured by vector arithmetic; thus, the implementation of the 8-point algorithm is independent from the geometrical form of the image surface. A 2-pass approach is proposed to obtain the final essential matrix. First, an initial essential matrix is derived according to a smaller set of corresponding image points, which is the matching result associated to the relatively large threshold value D_{SIFT}. Those sparse corresponding points are believed to be more accurate but less

descriptive. Next, a smaller threshold value is assigned to obtain a larger set of point matches. The initial essential matrix is then used to serve as a constraint to filter out the incorrect matches. In other words, the matching outliers are filtered by the epipolar constraint. Remaining point matches are then used to compute the final essential matrix.

The derived essential matrix is used to solve for the external camera parameters **R** and **t**, which stand for the rotation matrix and the translation vector, respectively. The algorithm leads to two valid solutions of **R** and two solutions of **t** pointing in opposite directions. The desired solution of the translation **t** can be obtained by assuming forward motion of the camera. To identify the correct solution of **R**, the scene points based on the already processed panoramic image were reconstructed with respect to the 3D world coordinate system. Each valid solution of **R** is used to calculate the new scene points. Ideally, majority of those new scene points should coincide with the previously reconstructed scene points. The correct solution of **R** can be determined by evaluating the 3D reconstruction results.

Pairwise external camera parameters are integrated one by one to obtain the global camera motion and thus the camera's moving trajectory. The major drawback of such a method is that the camera parameter estimation error would propagate through the integration process. One way to correct such drift is by a path-closing strategy. The idea is to evenly distribute the offset at the closing position to all the intermediate camera location. However, this method does not always work well. In order to deliver an efficient and relatively more accurate solution to this problem, we have chosen to incorporate GPS information. Since the accuracy of GPS varies from 1 to 5 meters, it is sensible to correct the trajectory drift every 50 meters based on the GPS reading. The concept of this approach is to apply a "loop-closing method" to every selected camera location. Instead of actually closing the camera path, the estimated camera's position at each selected location is shifted to coincide with the GPS reading. The amount of displacement is evenly distributed along the piecewise path.

7. Performance evaluations

This section presents performance evaluation of camera pose recovering methods, introduced in Sect. 5, and of the camera trajectory estimation approach, introduced in Sect. 6. All the experiments were performed on Windows XP operating system running on a Intel(R) Core(TM) i7 CPU 920 2.67 GHz with 3G of RAM.

7.1. Fisheye camera pose recovery

To evaluate the performance of the proposed methods, a vertical camera setup was considered. Two vertically displaced fisheye cameras were mounted on the roof of the university's test vehicle. The test vehicle is referred to as HAKA1 (acronym for high awareness kinematic vehicle number 1) and was facilitated by a partnership with Daimler AG. HAKA1 provides a fully road worthy car as a mobile platform for simulating a passenger car in all driving situations, where video analysis systems and algorithms can be tested and research conducted. The test vehicle can be fitted with an onboard computer and several cameras. In Fig. 24, the test vehicle HAKA1 and our vertical camera setup are shown.

For the experiments, the two cameras need to be calibrated precisely. For camera calibration, the method proposed in Ref. 21 was applied by using the provided MATLAB toolbox. A checkerboard printed on a planar surface was used to determine the intrinsic parameters of the cameras. To obtain corresponding calibration images, the checkerboard was placed in front of the camera setup so that it was captured by both cameras. In Fig. 23, a pair of corresponding calibration images is shown.

To evaluate the accuracy of the camera calibrations, the reprojection error of all calibration points was considered. The reprojection error of a calibration point is the Euclidean pixel distance of the detected pixel position and the reprojection onto the image. The pixel position of the reprojection is obtained from the extrinsic parameters of the checkerboard and the calculated intrinsic parameters of the camera.

In total, 24 pairs of corresponding calibration images were considered with calibration points in each image. Thus, 2,400 calibration points were evaluated to measure the accuracy of each camera calibration. In Table 1, the reprojection errors of all calibration points for both cameras

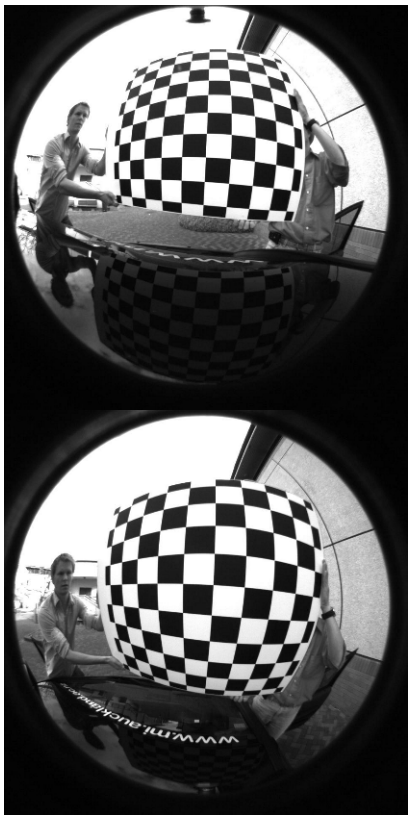

Fig. 23. Pair of corresponding calibration images for a vertical camera setup. Up: Top camera. Down: Bottom camera.

Fig. 24. Stereo vision setup. Left: Test vehicle HAKA1. Right: Vertical camera setup mounted on HAKA1.

were statistically analysed. It can be seen that the top camera was calibrated slightly more accurately than the bottom camera. The top camera has an average reprojection error of 0.207 ± 0.168, whereas the bottom cameras has an average reprojection error of 0.266 ± 0.320. Note that the median value for both cameras is less than 0.2 pixels, which means that half of all calibration points are reprojected with a minimal error. Thus, it can be concluded that the cameras are calibrated very precisely. Outliers can be related to inaccuracies in the corner detection of the camera toolbox. Those outliers especially occur in overexposed image areas in the calibration images. In these image areas the corners between checkerboard boxes cannot be determined distinctly.

In Sect. 5.1, an estimation and a non-linear refinement of the relative camera orientation were proposed. Both estimation and optimization were considered in our evaluation. To evaluate the performance, the spatial displacement of all calibration points was calculated. The spatial displacement of a calibration point is the Euclidean distance between the scene points, which was estimated from both cameras, and describe the calibration points in space. A mathematical description of the spatial displacement was given in Sect. 5.1.

Table 1. Reprojection error of all calibration points for both cameras.

	mean e	std e	median e	min e	max e
Top camera	0.207	0.168	0.183	0.004	3.430
Bottom camera	0.266	0.320	0.193	0.003	3.051

In Table 2, the spatial displacement of all calibration points is statistically analysed. Three estimations of random checkerboard positions, the average estimation of all checkerboard positions, and the non-linear optimization were considered. The results of the estimations vary significantly. In comparison to the optimal result, estimation 1 shows good results. An average spatial displacement of 0.824 ± 0.650 mm was achieved. Estimation 2 and estimation 3 show poor results, the calibration points in space are in average displaced by 1.128 ± 0.852 mm and 1.721 ± 1.089 mm. The variation of the checkerboard position in space is either related to the inaccuracy during image acquisition or to inaccuracies of the extrinsic parameters obtained from the camera calibration process.

A more stable estimation was obtained by averaging over all estimations for the camera orientation. The results of the averaged estimation is close to the optimal result. On average, the calibration points were displaced by 0.821 ± 0.479 mm for the averaged estimation. Through the non-linear optimization proposed in Sect. 5.1, the optimal camera orientation was obtained. The resulting camera orientation has an average spatial displacement of 0.779 ± 0.492 mm. Our tests have shown that, using the Levenberg-Marquardt algorithm, the non-linear optimization reaches the optimal solution regardless of the initial estimation.

In Fig. 25, a histogram of the spatial displacement error is shown. The average value is depicted by a dashed line and the median value is depicted by a dotted line. Although the sum of squared displacement errors is minimal, the average

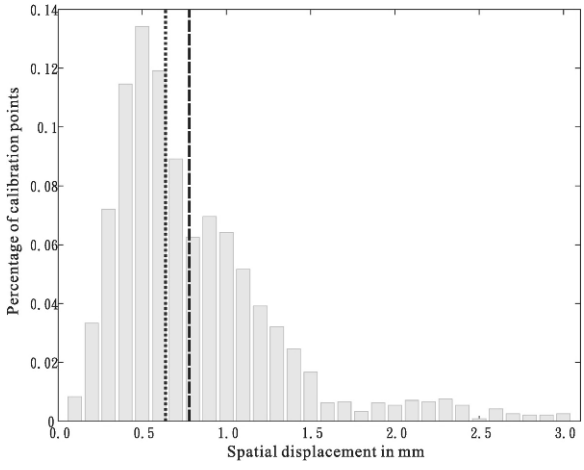

Fig. 25. Histogram of spatial displacement error for optimized camera orientation. Dotted line indicates median value and dashed line indicates average value.

displacement is still high. The majority of calibration points were displaced by less than 1.5 mm, but the percentage of point pairs near zero displacement is small.

Table 2. Spatial displacement error in millimetres of all calibration points.

	mean e	std e	median e	min e	max e
Estimation 1	0.824	0.650	0.587	0.000	3.453
Estimation 2	1.128	0.852	0.869	0.000	4.220
Estimation 3	1.721	1.089	1.634	0.000	5.274
Averaged estimation	0.821	0.479	0.695	0.045	2.987
Non-linear optimization	0.779	0.492	0.639	0.030	3.056

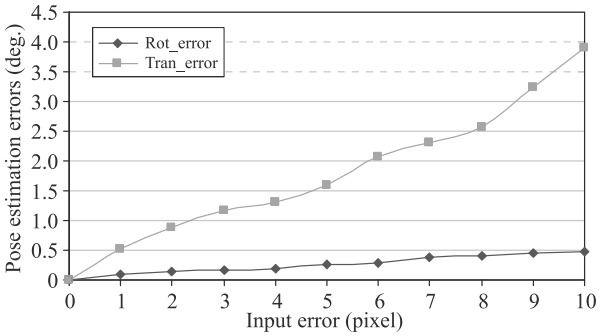

Fig. 26. Plot shows how errors of the given corresponding image points affect the estimated sensor poses in levelled panoramas.

7.2. Sensor-line camera pose recovery

Error sensitivity analysis was carried out to evaluate the performance of the relative camera pose recovering method proposed in Sect. 5.2 while a pair of levelled sensor-line cameras were used for capturing panoramic images. It is almost certain that there will be minor deviations from our ideal model when dealing with rotating sensor-line cameras in real-world applications. For instance, sensors may not be perfectly levelled or the identified corresponding points are erroneous.

Two synthetic experiments were conducted by using MatLab to answer the question of how each of the following affects the pose estimation results:

1) error in coordinates of the given corresponding image points;
2) some non-parallelity of rotation axes.

In particular, the first stated error is mostly likely to occur, which may be caused by low image resolution, incorrect feature allocation or even a general result of erroneous camera calibration.

The setup of the synthetic experiments for a pair of levelled panoramas is as follows: R = 320 mm, ω = 90°, f = 8.75 mm, and L = 1,800 pixels. Each of these was chosen based on the considerations provided in Ref. 14 and the commonly available equipment. The relative poses between these two panoramas are described by the following: $\mathbf{t} = (1000, 0, 500)^T$ in mm and $\mathbf{R} = \mathbf{R}_y(30°)$. This setup reflects common outdoor situations.

The estimated sensor pose is characterized by $\hat{\mathbf{R}}$ and $\hat{\mathbf{T}}$. The error measurement for rotation is defined b

$$\text{Rot\_error} = \arccos\left(\frac{\text{tr}(\mathbf{R}\hat{\mathbf{R}}^{\text{T}}) - 1}{2}\right),$$

and the error measurement for translation is defined by

$$\text{Tran\_error} = \arccos\left(\frac{\mathbf{t}\cdot\hat{\mathbf{t}}}{\|\mathbf{t}\|\cdot\|\hat{\mathbf{t}}\|}\right)$$

in degrees in both cases. These two quantities represent the angle difference and the direction difference, respectively, which were used in Ref. 34. Due to the nonlinear constraints, the quadratic programming optimization approach is not directly applicable. Thus, the sequential quadratic programming method is used instead for optimization (e.g., function *fmincon* in MatLab).

The pose estimation result of the first experiment (e.g., about erroneous corresponding points) is illustrated by the plots in Fig. 26. Up to ten-pixel errors were introduced (the errors are modelled by additive Gaussian-distributed random numbers). Under those circumstances, the proposed approach is still able to achieve less than five-degree error in the estimation of **t**; see the Tran_error as plotted by means of a grey line. In a real situation, it is very unlikely that such poor correspondence results would be obtained. For a two-pixel error in the given corresponding pairs, the algorithms are able to guarantee an error of less than one degree for the estimated values of **R** and **t**.

The second synthetic experiment is to test how robust the method is in incorrectly levelled panorama situations. In the experiments, a total 4-degree levelling error was assumed, defining different orientations represented, for example, by $\mathbf{R}_x(\pm 4°)$, $\mathbf{R}_x(\pm 1°)\mathbf{R}_z(\pm 3°)$, or $\mathbf{R}_x(\pm 2°)\mathbf{R}_z(\pm 2°)$. For each of those n-degree levelling errors, the average errors introduced by all different combinations of possible rotation matrices were calculated. The results are shown in Fig. 27 which suggest that an "unleveledness" in the image acquisition greatly influences the estimation of the translation vector **t**. The levelling process and its accuracy are critical for the precision of the pose estimation results.

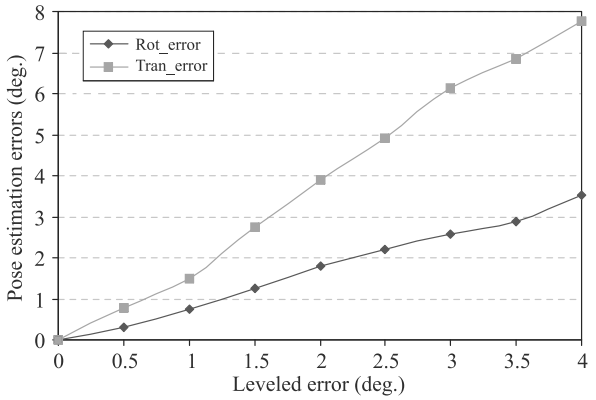

Fig. 27. Plots show how "unleveledness" results into errors of estimated sensor poses.

7.3. Panoramic camera trajectory estimation

A Point Grey Ladybug3 camera was used to capture dense spherical panoramic images. The camera was mounted on top of a car as shown in Fig. 4. The car was moving at an average speed of 35 kilometres per hour on the street. This way, adjacent panoramic images are captured at locations approximately one meter apart. This car was also equipped with a GPS system.

We recorded thousands of panoramic images this way on different streets, however, for image experiments there is no sufficiently accurate ground truth data available for evaluation. As described in the previous sections, we aim to reconstruct a rough 3D street model for which accuracy was not our major concern. An example of a reconstruction is shown in Fig. 28.

To demonstrate the accuracy of the camera trajectory recovery result, one small experiment was performed on the secure area of a sidewalk instead of on a (busy) road such that the path can be planned ahead and carefully measured. Figure 29 illustrates the camera trajectory recovery result of a 200 meter path. Here, an ideal ground-truth image acquisition path is illustrated by the black curve, and the estimated camera path is plotted in a white dotted curve.

Fig. 28. A 3D street reconstruction example.

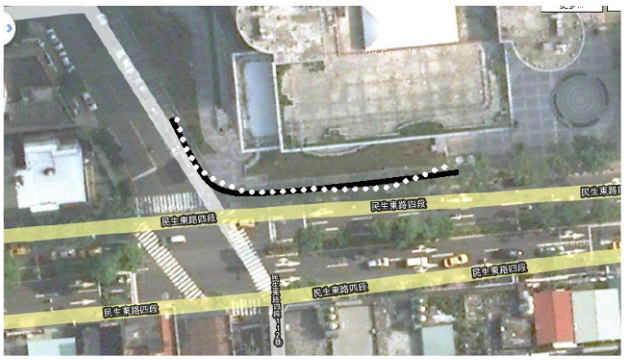

Fig. 29. Camera trajectory recovery result of real image experiment. Black curve indicates ideal path of the camera, and white dotted curve illustrates the recovered trajectory.

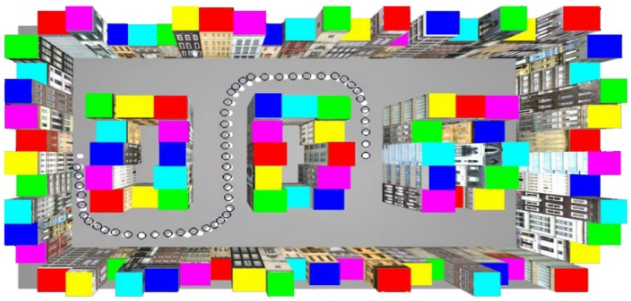

Fig. 30. Camera trajectory recovery result of synthetic experiment.

To evaluate the performance of the proposed camera trajectory recovery approach, we have also conducted some synthetic experiments. The 12×20 units (note: the unit is as used in the software) virtual street model was built by Maya and all the buildings were texture mapped with real images. A virtual camera was implemented to capture the panoramic images in the virtual world. For the experiment illustrated in Fig. 30, 50 panoramic images were generated at the locations indicated by white dots. The estimated camera path is represented by a set of black circles. The average drifts of the resulting camera path to the actual path is equal to 0.324 units. Since this is an example of a short open path, no bundle adjustment nor loop closing was performed. In the other experiment, a closed path was used for evaluation, and bundle adjustment was used to refine the resulting estimated path. The average error is reduced to 0.18 units.

8. Conclusions

The paper reviews camera projection geometry and calibration methods of various wide-angle vision systems which are particularly suitable for road-related applications. These vision systems are able to capture images with a wide field--of-view, often referred to as panoramic images. Stereo analysis of a pair of captured panoramic images and the relative camera pose recovery methods is also presented for selected imaging sensors.

Single or multiple fisheye cameras are starting to be used for driver assistance applications due to their relatively low cost and high flexibility. High-resolution panoramic images, captured by a rotating line-camera, offer the best viewing impression of a static scenario and are often used for virtual touring applications. By following these approaches, it is possible to generate stereo panoramic images viewable by any type of stereoscopic visualization method. Various multi-camera systems have been developed to be mounted on a moving car, for example the Point Grey Ladybug camera, to capture panoramic images during motion, which is especially suitable for large-area 3D reconstruction purposes.

Performance evaluations were carried out to test the robustness of the presented camera pose recovery methods for fisheye and rotating line-cameras. The results show that these methods are robust for practical situations. In order to evaluate the accuracy of the presented camera trajectory

estimation approach for spherical panoramic images, a synthetic 3D street model was built. The experiments show that, in such an ideal environment, "good" trajectory estimation results are obtained. However, for real experimental image sequences, captured with a Ladybug camera, GPS data should be incorporated into the location estimation process to correct for drift produced by error propagation.

Acknowledgements

Part of the street 3D reconstruction project was financially supported by the National Science Council, Taiwan (Grand no. NSC 100-2221-E-197 – 028). Special thanks to A. Tsai for the support on the Ladybug camera experiments.

References

1. F. Huang, R. Klette, and K. Scheibe, *Panoramic Imaging: Sensor-Line Cameras and Laser Range-Finders*, Wiley, Chichester, 2008.
2. K. Daniilidis and R. Klette, *Imaging Beyond the Pinhole Camera*, Springer, New York, 2007.
3. S. Nayar, "Catadioptric omnidirectional camera", *Proc. Conf. Comput. Vision Pattern Recogn.*, pp. 482–488, San Juan, Puerto Rico, 1997.
4. D.G. Lowe, "Automatic panoramic image stitching using invariant features", *Int. J. Comput. Vision*, **74**, 59–73 (2006).
5. S. Peleg, "Panoramic mosaics by manifold projection", *Proc. Conf. Comput. Vision Pattern Recogn.,* pp. 338–343, San Juan, Puerto Rico, 1997.
6. R. Szeliski, "Image alignment and stitching: A tutorial", Technical Report MSR-TR-2004-92, Microsoft Research, 2004.
7. R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and texture-mapped models", *Proc. SIGGRAPH*, ACM Press, pp. 251–258, Los Angeles, 1997.
8. Y.-C. Liu, K.-Y. Lin, and Y.-S. Chen, "Bird's-eye view vision system for vehicle surrounding monitoring", *Proc. Robot Vision* **4931**, pp. 207–218, Springer-Verlag Heidelberg, 2008.
9. T. Ehlgen, T. Pajdla, and D. Ammon, "Eliminating blind spots for assisted driving", *IEEE T. Intell. Transp.* **9**(4), 657–665 (2008).
10. S. Gehrig, C. Rabe, and L. Krueger, "6D vision goes fisheye for intersection assistance", *Proc. Canadian Conf. Comput. Robot Vision*, pp. 34–41, Windsor, 2008.
11. M. Pollefeys, D. Nister, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, and H. Towles, "Detailed real-time urban 3D reconstruction from video", *Int. J. Comput. Vision* **78**, 143–167 (2008).
12. G. Hartmann and R. Klette, "Cylinder sweep: Fisheye images into a bird's-eye view", Technical Report MI-tech-TR 69, The University of Auckland, New Zealand, 2011.
13. S.-B. Kang, R. Szeliski, and M. Uyttendaele, "Seamless stitching using multi-perspective plane sweep", Technical Report MSR-TR-2001-48, Microsoft Research 2001.
14. F. Huang and R. Klette, "Stereo panorama acquisition and automatic image disparity adjustment for stereoscopic visualization", *Multimed. Tools Appl.* **47**, 353–377 (2010).
15. C. Frueh, S. Jain, and A. Zakhor, "Data processing algorithms for generating textured 3D building facade meshes from laser scans and camera images", *Int. J. Comput. Vision* **61**(2), 159–184 (2005).
16. M. Fleck, "Perspective projection: The wrong imaging model", Technical Report, Dep. Computer Science, University of Iowa, 1995.
17. J. Kumler and M. Bauer, "Fisheye lens designs and their relative performance", *Proc. SPIE* **4093**, pp. 360–369 (2000).
18. K. Miyamoto, "Fish-eye lens", *J. Opt. Soc. Amer.* **54**, 1060–1061 (1964).
19. H. Bakstein and T. Pajdla, "Panoramic mosaicing with 180° field of view lens", *Proc. IEEE Workshop Omnidirectional Vision*, pp. 60–67, Copenhagen, 2002.
20. J. Kannala and S.S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish--eye lenses", *IEEE Trans. Pattern Anal. Machine Intell.* **28**, 1335–1340 (2006).
21. D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras", *Proc. IEEE/ RSJ Int. Conf. Intell. Robots Systems*, pp. 5695–5701, Beijing, 2006.
22. H. Ishiguro, M. Yamamoto, and S. Tsuji, "Omni-directional stereo", *IEEE T. Pattern Anal. Machine Intell.* **14**, 257–262 (1992).
23. Y. Li, H.Y. Shum, C.K. Tang, and R. Szeliski, "Stereo reconstruction from multiperspective panoramas", *IEEE T. Pattern Anal. Machine Intell.* **26**, 45–62 (2004).
24. D. Murray, "Recovering range using virtual multicamera stereo", *Computer Vision Image Understanding* **61**, 285–291 (1995).
25. S. Peleg and M. Ben-Ezra, "Stereo panorama with a single camera", *Proc. Conf. Comput. Vision Pattern Recogn.*, pp. 395–401, Fort Collins, 1999.
26. F. Huang, A. Torii, and R. Klette, "Geometries of panoramic images and 3D vision", *Machine Graphics & Vision* **9**, 463–477 (2010).
27. J.-Y. Bouguet, "Camera calibration toolbox for MATLAB". http://www.vision.caltech.edu/bouguetj/calib_doc/, 2010.
28. T.-H. Ho, C.C. Davis, and S.D. Milner, "Using geometric constraints for fisheye camera calibration", Proc. *IEEE Workshop Omnidirectional Vision*, pp. 17–21, Beijing, 2005.
29. J.-Y. Bouguet, "Visual methods for three-dimensional modelling", PhD thesis, California Institute of Technology, 1999.
30. S. Li, "Binocular spherical stereo", *IEEE T. Intell. Transp.* **9**, 589–600 (2008).
31. S.E. Chen, "QuickTime - An image-based approach to virtual environment navigation", *Proc. SIGGRAPH*, pp. 29–37, Los Angeles, 1995.
32. S.B. Kang and P. Desikan, "Virtual navigation of complex scenes using clusters of cylindrical panoramic images", *Proc. Graphics Interface*, pp. 223–232, Vancouver, 1998.
33. S.B. Kang and R. Szeliski, "3-d scene data recovery using omnidirectional multibaseline stereo", *Int. J. Comput. Vision* **25**, 167–183 (1997).
34. H. Li, R.I. Hartley, and J.H. Kim, "A linear approach to motion estimation using generalized camera models", *Proc. IEEE Comput. Society Conf.* on *Comput. Vision Pattern Recogn.*, pp. 1–8, Anchorage, 2008.

