 doi:10.20944/preprints201612.0016.v1

Article

Line-Based Registration of Panoramic Images and LiDAR Point Cloud for Mobile Mapping

Tingting Cui [1,2], Shunping Ji [3,4]*, Jie Shan [5,6]*, Jianya Gong [1,3,5] and Kejian Liu [2]

[1] State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; cuitingting@ppsuc.edu.cn (T.C.); gongjy@whu.edu.cn (J.G.)
[2] Research Center of Remote Sensing in Public Security, People's Public Security University of China, Beijing 100038, China; liukejian@ppsuc.edu.cn
[3] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[4] RSISE, Australian National University (ANU), Canberra ACT 2600, Australia
[5] Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430079, China
[6] Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA
* Correspondence: jishunping@whu.edu.cn (S.J.); jshan@purdue.edu (J.S.)

Abstract: For multi-sensor integrated systems, such as a mobile mapping system (MMS), data fusion at sensor-level, i.e., the 2D-3D registration between optical camera and LiDAR, is a prerequisite for higher level fusion and further applications. This paper proposes a line-based registration method for panoramic images and LiDAR point cloud collected by a MMS. We first introduce the system configuration and specification, including the coordinate systems of the MMS, the 3D LiDAR scanners, and the two panoramic camera models. We then establish the line-based transformation model for panoramic camera. Finally, the proposed registration method is evaluated for two types of camera models by visual inspection and quantitative comparison. The results demonstrate that the line-based registration method can significantly improve the alignment of the panoramic image and LiDAR datasets under either the ideal spherical or the rigorous panoramic camera model, though the latter is more reliable.

Keywords: mobile mapping system; LiDAR point cloud; 2D-3D registration; panoramic sensor model

1. Introduction

Motived by applications such as vehicle navigation [1], urban planning [2] and autonomous driving [3], the need for 3D detailed photorealistic models in urban area has dramatically increased in recent years. Mobile mapping systems (MMS's), which collect 3D and/or 2D photographic data while vehicles move in a regular speed, have been widely used as the efficient data acquisition technology on street-level[4]. Short range laser scanners (e.g., 100 - 200 m) and electro-optical cameras are two primary sensors on a MMS's, each of which has its own characteristics. Laser scanner can acquire 3D information directly, but suffers from relatively low resolution and close range. While, a camera captures 2D images with high-resolution textures but the depth information is missing. Thus, such complementary characteristic between ranging and imaging sensors has gained broad study through the so called camera/LiDAR fusion [5-8].

A prerequisite of data fusion is to transform different datasets to a common coordinate system, which is often called a 2D-to-3D registration. Although the camera and LiDAR sensors are usually calibrated [9-12] in advance, there may still exist considerable misalignment between the two datasets. Possible reasons of such misalignment include: (1) System calibration errors. The relative orientation and position offsets between all these sensors, GPS, IMU (Inertial Measurement Unit), LiDAR and camera on a MMS, may lack precise measurements at manufactory. Further, the offsets and relative orientations determined prior to data collection may change due to mechanical vibrations [13]. (2) Different acquisition time. The image is collected at a certain exposure time while the corresponding point cloud data are actually collected via continuous scanning during a period of

 doi:10.20944/preprints201612.0016.v1

time span, indicating that the two sensors are not strictly synchronized. (3) Unreliable GPS signals make this even worse. At street level, the GPS signals are often suffered from ambiguity and loss due to multipath and canyon effects. Even IMU, DMI (Distance Measuring Indicator) or DGPS (Different GPS) are onboard, noticeable co-registration errors may still exist. All of these above unavoidable factors call for data-driven registration methods in the subsequent data processing procedure.

There have been a number of related works about 2D-3D registration. The recent review[14] is referred for a comprehensive coverage. The registration framework for LiDAR and optical images[15], as outlined in the concept of image registration[16], includes four components: registration primitives, similarity measure, registration transformation, and matching strategy.

The choice of primitives influences subsequent registration steps and accuracy. In order to evaluate current registration methods and their performance, the European Spatial Data Research (EuroSDR) in 2008 launched the project "Registration Quality – Towards Integration of Laser Scanning and their Performance". This led to a final technical report [17] in 2011. Through over 20 experiments, [17] found that the selection of the type of tie features (another way of saying registration primitive) is one of the main factors affecting registration accuracy: the point feature is not as good as other ones such as linear features or surfaces in general. Other studies also show linear features are potential to be reliably, accurately and automatically extracted from both 2D images and 3D point cloud [18,19]. In this work, we use straight-line features as registration primitives.

The principles and concepts of line-based registration come from the subject line photogrammetry, which was an active research topic in late 1980s and early 1990s. In this subject, linear features, especially straight lines are regarded as basic entity as traditional point features. [20] described the concepts, mathematical formulations and algorithms related to line photogrammetry, which is based on the straight lines extracted from digital images. [21] proposed a collinearity model to establish the relationship between lower level features such as edge pixels instead of fitting lines in image space and control lines in object space. [22] proposed a concept of "generalized points", a series of connected points representing a linear feature. Under this concept, the traditional collinearity model can accommodate more complicated features such as circles, rectangles, rather than only straight lines. [18,19] utilized linear features to determine camera orientation and position relative to 3D model. Vanishing points extracted from 2D images and 3D principle directions derived from 3D model were also used to estimate the camera orientation. In addition, there were also many studies using statistical similarity measures for the registration of LiDAR and images. [23] developed a registration strategy based on global mutual information and exploited the statistical dependency between the intensity and measured LiDAR elevation. While [24] investigated the effectiveness of both local and global mutual information.

All of the above studies were based on the popular frame cameras. However, a panoramic camera other than a frame camera is to be used in our mobile mapping system, which means a new panoramic sensor model should be taken into account. The biggest advantage of a panoramic camera is its 360° view angle, making it practically a standard component in recent MMS. Online street-view maps, provided by Google, Microsoft, Baidu and Tencent, were mostly generated from geo-registered panoramic imagery. Unfortunately, among limited works on LiDAR and image registration involving panoramic cameras, none of them considered rigorous panoramic camera model. [8,25] proposed an automatic registration of mobile LiDAR and spherical panoramas. However, only part of the panoramic image in limited viewport was used. Moreover, the registration was based on conventional frame camera model.

Besides registration, many applications based on panoramic images have been reported in recent years. [26] presented a structure-from-motion (SfM) pipeline for 3D modeling using Google Street View imagery with an ideal spherical camera model. [27] presented a piecewise planar model to reconstruct 3D scenes from panoramic image sequences. A quadrangular prism panoramic camera model was used for improved image matching. [28] explained both the ideal spherical cameral model and the rigorous panoramic sensor model for multi-camera rig and their corresponding epipolar geometry. [29] further compared the two models and their effects on the localization quality in object space and the quality of space resection. In this paper, we are to utilize the rigorous panoramic sensor

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

model for image and LiDAR registration. Since there were plenty of studies based on the ideal spherical camera model, we also tempt to demonstrate its limitations by comparing it with the rigorous model.

This paper proposes a rigorous line-based registration method for precise alignment of LiDAR point cloud and panoramic images. The rest of the content is structured as follows. Section 2 introduces a mobile mapping system and its sensor frames for the LiDAR and the panoramic camera. Section 3 presents registration models based on straight-line features for panoramic cameras. Section 4 addresses 3D line extraction from LiDAR point cloud, and Section 5 introduces the datasets and analyzes the registration results based on our model. Finally, we present the conclusions and future work in Section 6.

2. The Mobile Mapping System and Sensor Configuration

The MMS used in this work was jointly developed by Wuhan University and Leador Spatial Information Technology Corporation, Ltd., configured with a Ladybug 3 camera[30], three low-cost SICK laser scanners[31] (one for ground and two for facades) and GPS/IMU. The system and its sensor configuration are shown in Figure 1.

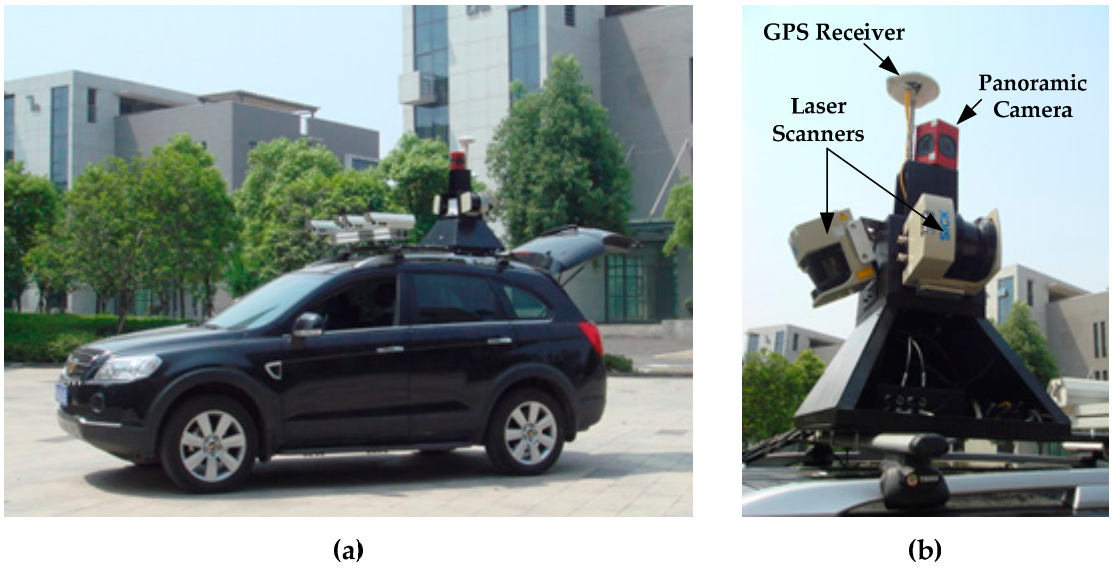


Figure 1. The Mobile Mapping System used in this study: (a) the vehicle; (b) panoramic camera, laser scanners and GPS receiver.

This section first introduces several key coordinate systems of the mobile mapping system and their geometric relationships, followed by the description of the geo-referenced LiDAR and the rigorous camera model for a multi-camera rig.

2.1. Coordinate Systems

As shown in **Figure 2**, there are three coordinate systems in our mobile mapping system: (1) world coordinate system, (2) vehicle platform coordinate system, and (3) camera-centered coordinate system. The world coordinate system is the reference for data management and organization. In our work, GPS/IMU records the translation and orientation from the world coordinate system to vehicle platform coordinate system denoted as $\boldsymbol{M}_1(\boldsymbol{R}_1, \boldsymbol{T}_1)$. LiDAR points have been geo-referenced to the world coordinate system (the left-bottom dotted line) according to $\boldsymbol{M}_1$ and the calibration parameters $\boldsymbol{M}_3$ between LiDAR sensor and the platform. $\boldsymbol{M}_2(\boldsymbol{R}_2, \boldsymbol{T}_2)$ is the transformation from camera to vehicle platform, which is also achieved through prior calibration.

The goal of registration is to determine the transformation $\boldsymbol{M}$ from LiDAR points to the panoramic image. Other than a static calibration which concerns only $\boldsymbol{M}_{23}$, the time series of localization information $\boldsymbol{M}_1$ should be considered. For simplification, we ignore the possible errors of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

M_3 and construct the transformation directly between the georeferenced LiDAR and camera (the bottom solid line). We assume that there exists $\Delta M(\Delta R, \Delta T)$ to meet:

$$M = \begin{bmatrix} \Delta R & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_2 & T_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix}. \quad (1)$$

ΔM compensates several aspects including M_3 (as is discussed in Section 1). Specifically, a line-based registration method is proposed in this paper. Line features are extracted in both images and LiDAR and then utilized to determine the optimal ΔM for an accurate registration. The solution procedure is based on the standard least squares technique imbedded with RANSAC for removal of possible gross errors in M_1.

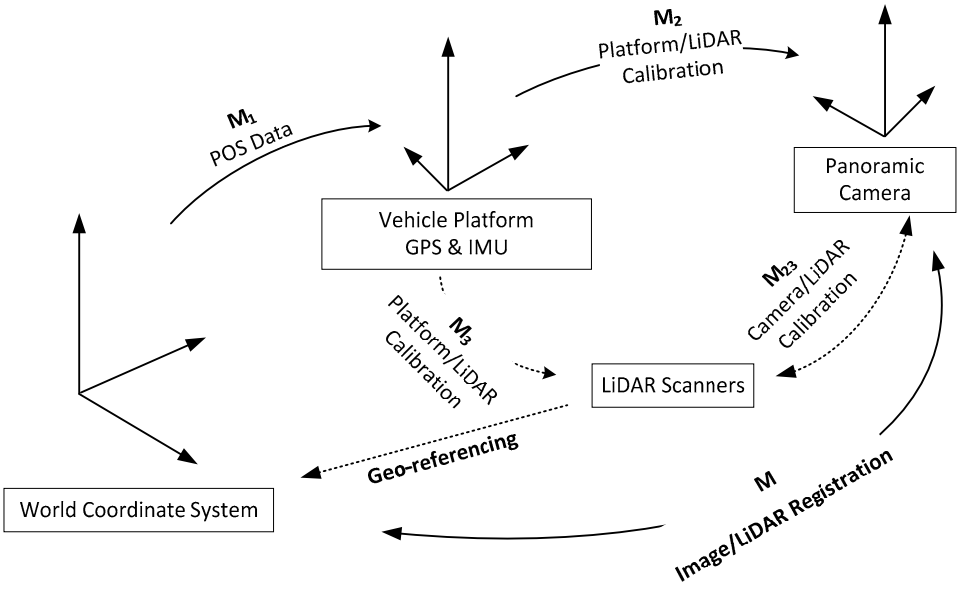


Figure 2. Coordinate systems of the Mobile Mapping System

2.2. Geo-refereneced LiDAR

The LiDAR points are geo-referenced to the world coordinate system by the interpolated rotation values $(\varphi_1, \omega_1, \kappa_1)$ recorded by INS at the corresponding position (X_1, Y_1, Z_1) from the GPS/IMU integrated navigation data and the calibration parameters M_3 between LiDAR and IMU [32]. In our system, three low-cost SICK laser scanners, all linear-array lasers, are equipped to acquire 3D point cloud of the object façade. The angular resolution (0.25 ~ 1.0°) and scan frequency (25 ~ 75 Hz) are fixed during data acquisition. The density of LiDAR points is uneven; the closer to the measured surface, the higher density of the points. For instance, points on the ground are much denser than those on the top façade. In addition, the point density in horizontal direction is dependent on the velocity of the mobile mapping vehicle.

2.3 Multi-Camera rig Models

The panoramic image in **Figure 3** covers a 360° view of surrounding scene, captured by the Ladybug 3 system composed of six fisheye cameras. Straight lines in reality are no longer straight on panoramic images, as compared to a common frame Figure 3(b).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

(a) **(b)**

Figure 3. Comparison between (a) a panoramic image and (b) a frame image.

Generally, the panoramic imaging process can be approximated by an ideal spherical camera model. However, since the entire image is technically stitched by 6 individual images through blending, stitching errors cannot be avoided. This section will first introduce the traditional ideal spherical camera model, and then extend it to the rigorous multi-camera rig panoramic model. We refer the **spherical** camera model as the ideal one and the **panoramic one** as the rigorous one.

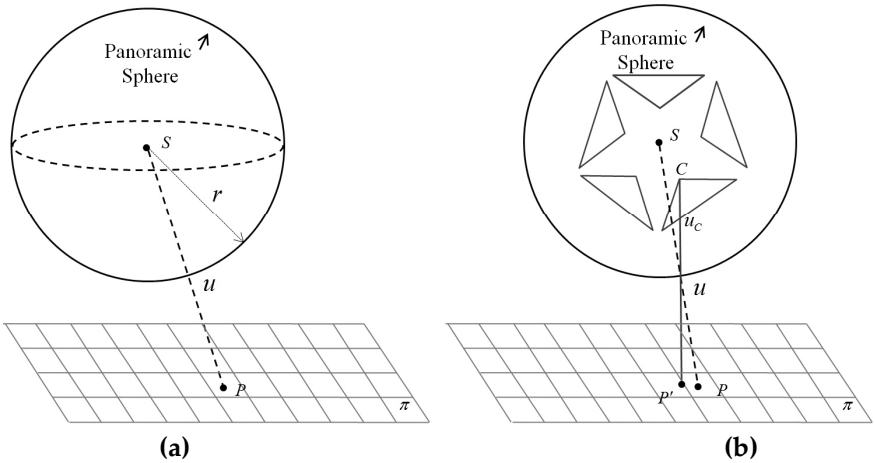
2.3.1. Spherical Camera Model

Under this model, the imaging surface is regarded as a sphere, whose center is the projection center. **Figure 4**(a) presents a schematic diagram of the spherical projection, where the sphere center S, the 3D points P in plane π, and the panoramic image point u are collinear[29]. Pixels on the panoramic image are typically expressed in polar coordinates. Assuming the width and height of the panoramic image is W and H respectively, the horizontal 360° view is mapped to $[0, W]$ and the vertical 180° view mapped to $[0, H]$. Thus each pixel (u, v) can be transformed to polar coordinate (θ, φ) by Equation (2):

$$\begin{cases} \theta = (2u - W) \cdot \dfrac{\pi}{W} \\ \varphi = (1 - \dfrac{2v}{H}) \cdot \dfrac{\pi}{2} \end{cases} \tag{2}$$

θ is the horizontal angle between $-\pi$ and π, and φ is the vertical angle between $-\pi/2$ and $\pi/2$. Let r be the radius of projection sphere, Equation (3) is used to calculate a set of Cartesian coordinates. In most cases, we define r = 20.0 meters for the best stitching accuracy[33].

$$\begin{cases} x = r \cdot cos\varphi \cdot sin\theta \\ y = r \cdot cos\varphi \cdot cos\theta \\ z = r \cdot sin\varphi \end{cases} \tag{3}$$



(a) **(b)**

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

Figure 4. Differences between the spherical and panoramic camera models. (a) The dashed line shows the ray through 3D point P, panoramic image point u, and the sphere center S; (b)the solid line shows the ray through 3D point P′, mono-camera image point $\boldsymbol{u}_c$, panoramic image point u, the mono-camera projection center C.

The sphere center S, 3D point $\boldsymbol{P}$, and edge pixel $\boldsymbol{u}$ are collinear. The relationship between $\boldsymbol{X}$ and $\boldsymbol{P}$ could be established by a perspective transformation in Equation (4).

$$\boldsymbol{X} = \lambda^{-1}\boldsymbol{R}^T(\boldsymbol{P} - \boldsymbol{T}) \tag{4}$$

where $\boldsymbol{P}$ is the coordinate of object point, and $\boldsymbol{X}(x, y, z)$ is the Cartesian coordinate of image point $\underline{u}$; $\boldsymbol{R}$ and $\boldsymbol{T}$ are respectively the rotation matrix and translation vector between object space and the panoramic camera space; and $\boldsymbol{\lambda}$ is the scale factor. Unlike the traditional camera model, the z coordinate of the image point is not equal to $-f$, where f is the focal length in traditional camera model. In the widely used spherical camera model, the image point u is under the spherical geometry constraint as Equation (5).

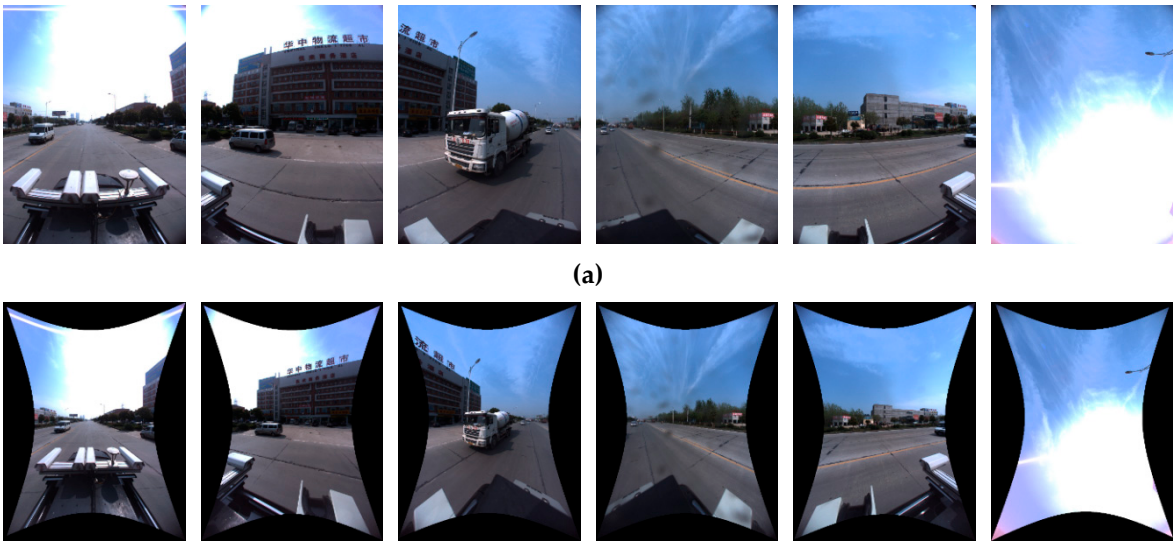
$$x^2 + y^2 + z^2 = r^2 \tag{5}$$

As a result, Equation (4) actually has two degrees of freedom, i.e., two independent equations.

2.3.2. Panoramic Camera Model

A multi-camera rig is consisted of several separate and fixed fish-eye lenses. Independent images are captured by each lens, and then stitched to form the entire panoramic image. As is shown in **Figure 4**(b), each lens has its own projection center $\boldsymbol{C}$, but it cannot be precisely located on the sphere center $\boldsymbol{S}$ due to the manufacturing constraints. The mono-lens center $\boldsymbol{C}$, instead of sphere center $\boldsymbol{S}$, panoramic image point $\boldsymbol{u}$ and 3D point $\boldsymbol{P}'$ in object space are collinear.

Specially, the panoramic camera used here is composed of 6 separate fish-eye lenses. **Figure 5**(a) shows an example of the 6 raw fisheye images, and **Figure 5** (b) shows the corresponding undistorted images rectified from the raw ones. Rectification from fisheye to ideal plane image only depends on the known camera calibration parameters $\boldsymbol{Kr}$, including projection model, radial and tangential distortion. The index r means that every fisheye camera has its own calibration parameters. Since the straight lines such as boundary of buildings are distorted in the raw fisheye image, we use rectified images for line extraction.



(a)

(b)

Figure 5. Images of Camera 0-5: (a) 6 fish-eye images; (b) 6 rectified images.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

As is shown in **Figure 6**, the global camera coordinate system is defined for the whole multi-camera rig, and six local coordinate systems are defined for each lens separately. The global coordinate system (see **Figure 6** (a)) of the panoramic camera is defined by three main directions. The X-axis typically is along the driving direction; the Z-axis is zenith direction; the Y-axis is orthogonal both to Y-axis and Z-axis. Each of the six lenses has its own local coordinate system (see **Figure 6** (b)): (1) The origin is the optical centre of the lens; (2) The Z-axis is the optical axis and points towards the scene; (3) The X- and Y-axes are parallel to the corresponding image coordinate. For each lens, there are 3 interior orientation elements, focal length f and image centre (x_0, y_0), and 6 exterior orientation parameters $(T_x, T_y, T_z, R_x, R_y, R_z)$ relative to the global coordinate system (the offsets between C and S in **Figure 4**(b) under spherical view). Both of them have been acquired in advance by careful manufactory calibration.

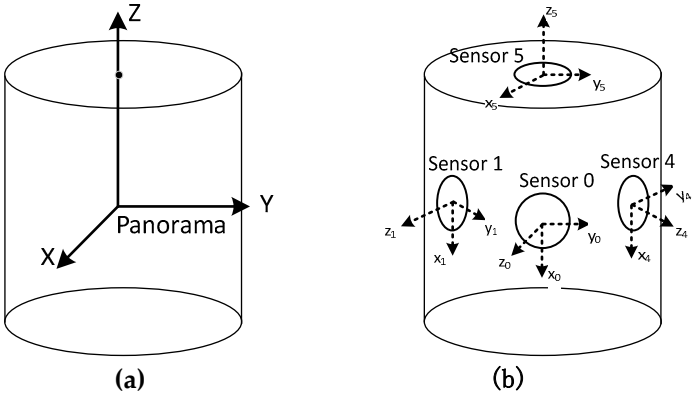


Figure 6. Global and local coordinate systems of multi-camera rig under a cylindrical projection. (a) The global panoramic camera coordinate system; (b) Six local coordinate systems of the rectified cameras.

Although each lens may have its own camera model, the advantages of panoramic imaging will be overlooked. To address this issue, we project all 6 images into the global spherical imaging surface to get uniform coordinates. Firstly, the coordinates of the rectified image of each separate lens are transformed to the global coordinate system. Each pixel $\boldsymbol{p}\,(x, y)$ in the rectified images forms a 3D ray in global coordinate system by Equation (6)

$$\mathbf{X}' = m\boldsymbol{R_r}\boldsymbol{X_r} + \boldsymbol{T_r} \tag{6}$$

where $\boldsymbol{X_r}(x - x_0, y - y_0, f)$ is the mono-camera coordinates of pixel $\boldsymbol{p}$; the translation vector $\boldsymbol{T_r}\,(T_x, T_y, T_z)$ and the local rotation matrix $\boldsymbol{R_r}$ are known, and the latter can be calculated by the following equation

$$\boldsymbol{R_r} = \begin{bmatrix} \cos R_z \cos R_y & \cos R_z \sin R_y \sin R_x - \sin R_z \cos R_x & \cos R_z \sin R_y \cos R_x + \sin R_z \sin R_x \\ \sin R_z \cos R_y & \sin R_z \sin R_y \sin R_x + \cos R_z \cos R_x & \sin R_z \sin R_y \cos R_x - \cos R_z \sin R_x \\ -sinR_y & \cos R_y \sin R_x & \cos R_y \cos R_x \end{bmatrix} \tag{7}$$

$\boldsymbol{X'}(x', y', z')$ is the coordinates transformed into the global panoramic coordinate system, and the scale factor m defines the distance from the rectified image plane to the projection surface (typically a sphere or cylinder). Combing Equation (5) and (6) we can resolve m and $\boldsymbol{X'}$ for a sphere projection.

In the next step, the collinearity equation based on the multi-camera rig can be established. As is shown in **Figure 4**(b), the real 3D ray is through CuP' instead of SuP, which can be vectorized as $(\mathbf{X}' - \boldsymbol{T_r})$. Translating the vector to the global camera coordinate system, we get:

$$\boldsymbol{T_r} + \boldsymbol{\lambda}(\mathbf{X}' - \boldsymbol{T_r}) = \boldsymbol{R}^T(\boldsymbol{P} - \boldsymbol{T}) \tag{8}$$

Equation (8) would be the same as the sphere projection (4) when $\boldsymbol{T_r}$ is small enough and vanishing. However, for the self-assembly panoramic cameras whose $\boldsymbol{T_r}$ are too large to ignore, the panoramic camera model should be a better choice.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

3. Line-based Registration Method

To simply the transformation in Equation (1), we introduce an auxiliary coordinate system, which is close to the camera-centered coordinate system but still has $\Delta \boldsymbol{M}$ bias. Using $\boldsymbol{M_1}$ and $\boldsymbol{M_2}$ in **Figure 2**, we can first transform the LiDAR point $\boldsymbol{P}_w$ in the world coordinate into the auxiliary coordinate $\boldsymbol{P}$, as is defined in Equation (9), which can be further discussed as below.

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{R_2} & \boldsymbol{T_2} \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{R_1} & \boldsymbol{T_1} \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \boldsymbol{P_w} \tag{9}$$

3.1. Transformation Model

Suppose a known line segment $\boldsymbol{AB}$ is given in the world coordinate (here actually the auxiliary coordinate). Its corresponding line on the panoramic image is detected as edge pixels. The projection ray through the perspective panoramic camera center $\boldsymbol{C}$, an edge pixel p on panoramic image, intersects $\boldsymbol{AB}$ on point $\boldsymbol{P}$, as illustrated in **Figure 7**. Let the line be represented by the two endpoints $\boldsymbol{X_A}$ and $\boldsymbol{X_B}$, an arbitrary point $\boldsymbol{P}$ is defined by Equation (10),

$$\boldsymbol{P} = \boldsymbol{X_A} + t(\boldsymbol{X_B} - \boldsymbol{X_A}) \tag{10}$$

with $\boldsymbol{t}$ a scale factor along the line.

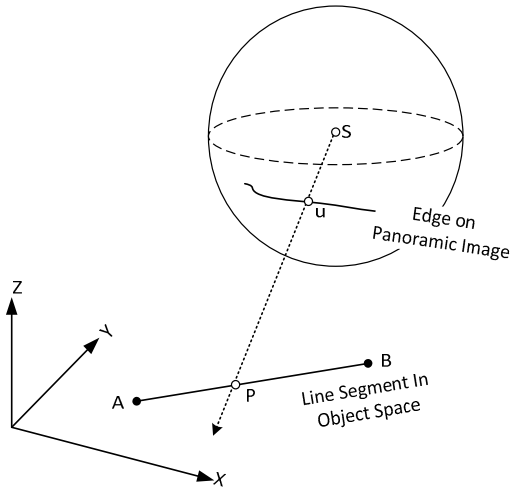


Figure 7. Line-based transformation model on panoramic image

Substituting the object point $\boldsymbol{P}$ in Equation (10) to (4) yields the line-based sphere camera model

$$\lambda \boldsymbol{X} = \boldsymbol{R}^{-1} \left[(\boldsymbol{X}_A - \boldsymbol{T}) + t(\boldsymbol{X_B} - \boldsymbol{X_A}) \right] \tag{11}$$

Further, we combine Equation (2) and (3) and substitute $\boldsymbol{P}$ in Equation (10) to (8), yielding the line-based panoramic camera model

$$\boldsymbol{T_r} + \lambda(\mathbf{X}' - \boldsymbol{T_r}) = \boldsymbol{R}^{-1} \left[(\boldsymbol{X}_A - \boldsymbol{T}) + t(\boldsymbol{X_B} - \boldsymbol{X_A}) \right] \tag{12}$$

where $\boldsymbol{X}'$ can be obtained from Equation (6). The scalar parameter λ and the line parameter t are unknown. What we try to resolve is the rotation matrix $\boldsymbol{R}$ and translation $\boldsymbol{T}$.

3.2. Solution

To get the best alignment between the two datasets, we use the non-linear least squares method to solve the unknowns iteratively. Euclidean distance in the panoramic image coordinate system is used as the similarity metric. Denoting the right hand term in Equation (11) as $[\bar{X}\ \bar{Y}\ \bar{Z}]^{\mathrm{T}}$ and combining Equation (2) and (4), we get Equation (13) as follows:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

$$\begin{cases} u=\left[\tan^{-1}\left(\frac{\bar{X}}{\bar{Y}}\right)\cdot\frac{W}{\pi}+W\right]\cdot\frac{1}{2} \\ v=\left[1-\sin^{-1}\left(\frac{\bar{Z}}{\sqrt{\bar{X}^2+\bar{Y}^2+\bar{Z}^2}}\right)\cdot\frac{2}{\pi}\right]\cdot\frac{H}{2} \end{cases} \tag{13}$$

where (u,v) is the coordinates in panoramic image coordinate system, (W,H) is the panoramic image size.

In addition to the six orientation and translation values $(X,Y,Z,\varphi,\omega,\kappa)$, we also have unknown line parameter t to be estimated. Here the right terms are multivariate composite functions $f_u(\boldsymbol{R}, \boldsymbol{T}, t)$ and $f_v(\boldsymbol{R}, \boldsymbol{T}, t)$. Given one pixel on the corresponding lines, two equations in (11) is formed with one line-parameter t introduced. In order to solve the six unknowns, we need at least six points. If one point per line is used, we need six pairs of corresponding lines. And if two points per line are used, three pairs of corresponding lines are needed. More than two points on a line does not reduce the rank deficient but only increase the redundancy.

The equations of i-th pair of corresponding lines can be termed by $\begin{cases} u_i=f_u(\boldsymbol{R},\boldsymbol{T},t) \\ v_i=f_v(\boldsymbol{R},\boldsymbol{T},t) \end{cases}$. Defining a parameter vector $\boldsymbol{X}=(X,Y,Z,\varphi,\omega,\kappa,t_1,\cdots,t_n)^T$ for n pairs of corresponding lines, Equation (13) is then expanded as Equation (14) after linearization by Taylor series:

$$\begin{cases} u=u^0+\frac{\partial f_u}{\partial X}\cdot\Delta X+\frac{\partial f_u}{\partial Y}\cdot\Delta Y+\cdots+\frac{\partial f_u}{\partial t_n}\cdot\Delta t_n \\ v=v^0+\frac{\partial f_v}{\partial X}\cdot\Delta X+\frac{\partial f_u}{\partial Y}\cdot\Delta Y+\cdots+\frac{\partial f_v}{\partial t_n}\cdot\Delta t_n \end{cases} \tag{14}$$

The above equation could be expressed in matrix form as Equation (15):

$$\boldsymbol{V}=\boldsymbol{A}\Delta-\boldsymbol{L} \tag{15}$$

where $\Delta=(\Delta X_T,\Delta Y_T,\Delta Z_T,\Delta\varphi,\Delta\omega,\Delta\kappa,\Delta t_1,\cdots,\Delta t_n)^T$, and $\boldsymbol{L}=(L_1;\cdots;L_n)$ with $\boldsymbol{L_i}=(u_i-u_i^0,v_i-v_i^0)^T$. The coefficient matrix $\mathbf{A}(\boldsymbol{A_1};\cdots;\boldsymbol{A_n})$ are defined as partial derivative of function f_u and f_v

$$\boldsymbol{A_i}=\begin{Bmatrix} \frac{\partial f_u}{\partial X},\frac{\partial f_u}{\partial Y},\frac{\partial f_u}{\partial Z},\frac{\partial f_u}{\partial\varphi},\frac{\partial f_u}{\partial\omega},\frac{\partial f_u}{\partial\kappa},\frac{\partial f_u}{\partial t_1},\cdots,\frac{\partial f_u}{\partial t_n} \\ \frac{\partial f_v}{\partial X},\frac{\partial f_v}{\partial Y},\frac{\partial f_v}{\partial Z},\frac{\partial f_v}{\partial\varphi},\frac{\partial f_v}{\partial\omega},\frac{\partial f_v}{\partial\kappa},\frac{\partial f_v}{\partial t_1},\cdots,\frac{\partial f_v}{\partial t_n} \end{Bmatrix} \tag{16}$$

The results can be obtained by solving the normal equation $\Delta=(\boldsymbol{A^TA})^{-1}\boldsymbol{A^TL}$. The unknowns $\boldsymbol{X}$ are updated through $\boldsymbol{X}\leftarrow\boldsymbol{X}+\Delta$ iteratively, until the elements of Δ are less than a given threshold. In order to assess the accuracy of the results, the standard deviation is calculated by Equation (17):

$$m_0=\pm\sqrt{\frac{\boldsymbol{V^TV}}{r}} \tag{17}$$

Here r is the number of redundant observations, $r=(2\times n)-(n+6)$. n is the number of pixels involved in the transformation, and usually $n=2m$ with m pairs of corresponding lines.

To handle the mismatch between LiDAR cloud lines and image lines as well as the occasional large biases in GPS/IMU records, RANSAC paradigm [29] is applied in iteration to remove the outliers in the corresponding line segments from LiDAR and camera.

4. Line Feature Extraction from LiDAR

The insufficient density of LiDAR points is always a problem in a low-cost configuration, which makes fitting linear features from the point cloud a challenge. The previous works about line extraction from LiDAR points tend to use the intersection of two neighboring unparalleled planar patches. However, the method cannot work well in our case because there is barely any intersection of planar patches in the dataset except at turns since the linear-array laser scanners are installed perpendicular to the driving direction. Hence we fit linear features directly from cloud points.

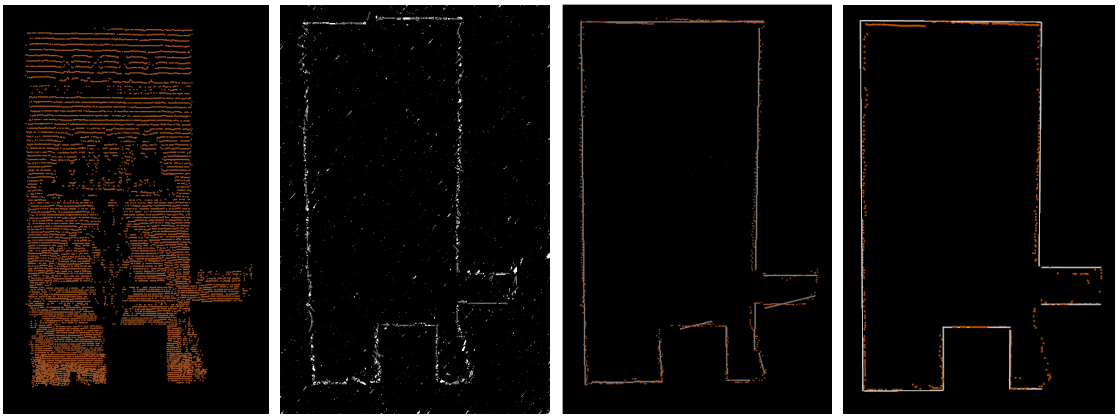
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

There are three types of objects containing abundant linear features in the common street-view scenes: buildings, pole-like objects, and curbs. This section introduces the methods to fit 3D straight lines from points belonging to the three objects. It is noted that the line-fitting is based on the well classified 3D point cloud achieved by existing automatic methods [34,35].

4.1. Buildings

Buildings provide the most reliable straight line features in street view. Given a mobile LiDAR point cloud dataset $P\{p_i|p_i(x_i, y_i, z_i)\}$ of buildings, we would like to get a group of 3D line segments $L\{l_j|l_j(x_{0j}, y_{0j}, z_{0j}; x_{1j}, y_{1j}, z_{1j})\}$. The detection procedure consists of three steps: (1) apply a region growing segmentation[36] on P, and get a set of planar segments $S\{s_k|s_k(i_{k1}, i_{k2}, \cdots, i_{kn})\}$; (2) project points onto the 3D plane model of segments s_k (**Figure 8(a)**) and detect boundary points in each segment s_k (**Figure 8(b)**); (3) fit the boundary points into 3D straight border lines with RANSAC (**Figure 8(c)**).

In order to overcome the weakness of the traditional least-square regression method[37], we introduce certain constraints: (1) lines must be through the outermost point, instead of the centroid; (2) only the lines close to being vertical or horizontal are considered; (3) only the lines consisting of enough points are considered (**Figure 8**(d)). Compared with line segments detected by [33] (**Figure 8**(c)), the fitting lines with constraints are more reliable.



(a) **(b)** **(c)** **(d)**

Figure 8. Line segments fitting for building patch: (a) projected points; (b) boundary points; (c) fitting lines using conventional least square method. (d) fitting lines using regularity constrains.

4.2. Street Light Poles

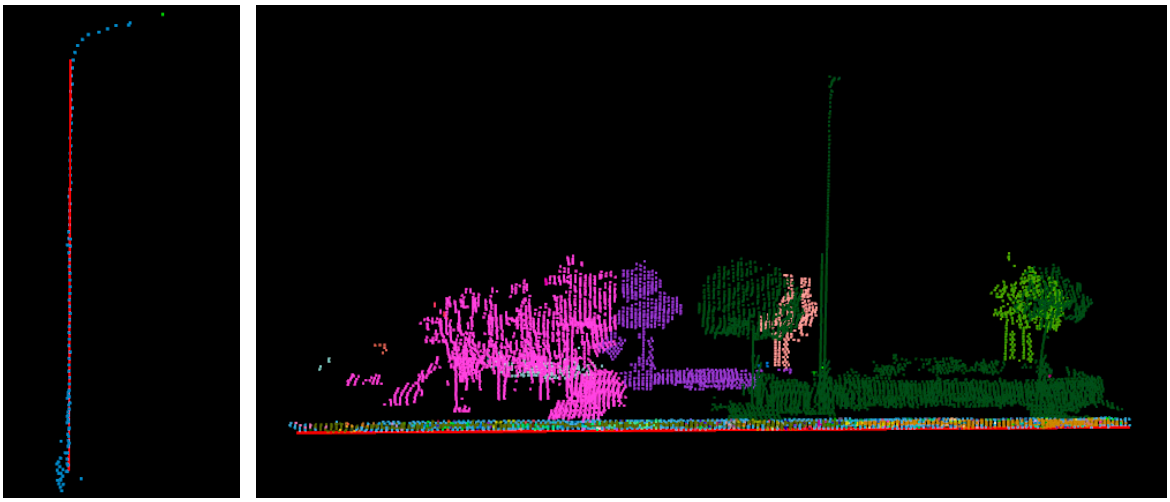
The pole-like segments are labeled and divided into separate objects by spatial connectivity and presented by one or two arrays of points. We adopted a percentile-based pole recognition algorithm[38] to extract the pole part, which could exclude the disturbing structures, such as flowerbed at the bottom of a light pole and non-pole elements such as lamps. The main steps of the algorithm are as follows: (1) the segment is first sliced into subparts, for which 2D enclosing rectangles and centroids are derived; (2) check the deviation of the centroids between neighboring subparts; (3) check the diagonal length of the rectangle; (4) keep the neighboring subparts with the maximum length. The final fitting line segment is defined by the 2D centroids and minimum and maximum Z values. The fitting result is shown in **Figure 9**(a).

4.3. Curbs

Curbs are usually located at a height of 10-20 cm above the road surface, designed to separate roads from sidewalks [39,40]. The density of the points on the ground is relatively high and points on curbs present narrow stripes which are vertical to road surface[41]. A curb-line can be approximated as the intersection of the vertical curb with the ground surface. In this work, we fit the curb-lines in following steps: (1) the points assigned as curbs are fitted into plane parallel to Z-axis

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

under RANSAC method with direction constrain, and the noises are filtered as outliers; (2) fit the points into the 2D line segment in OXY plane, (3) use the height of ground as Z value of 2D line segment. The fitting result sees **Figure 9**(b).



(a) **(b)**

Figure 9. Line segments fitting for (a) pole-like objects and (b) curbs

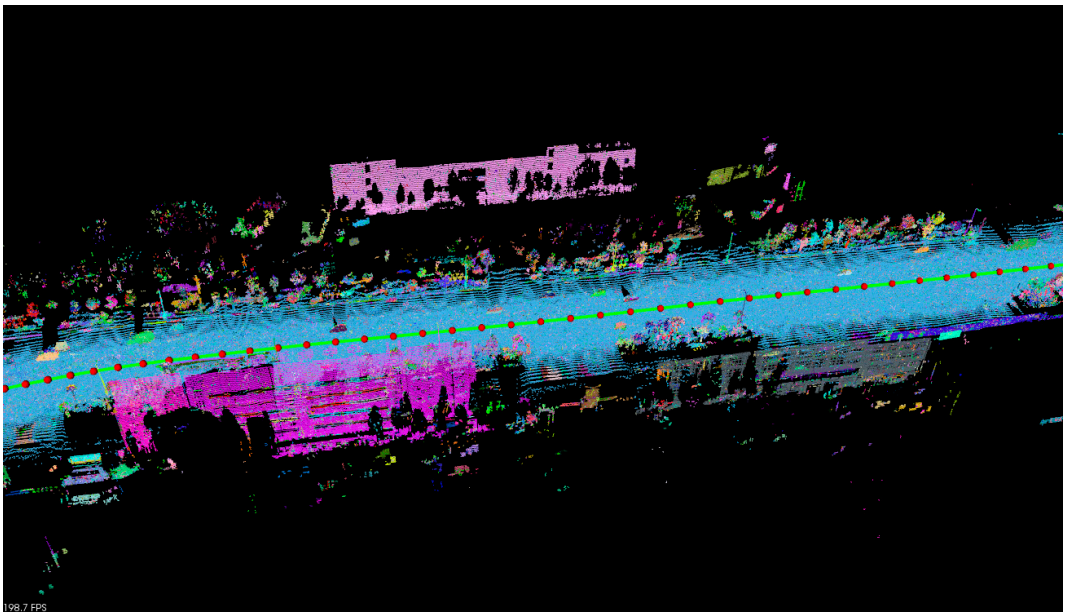
5. Experiments and Results

5.1. Datasets

Our test data were collected in Hankou North Street in the north of Wuhan City, including buildings, trees, poles, streets and moving cars. **Figure 10** (a) shows our test area in Google Earth, and **Figure 10**(b) shows the 3D point cloud of the test area containing about 1.2 million points, which have been previously classified and rendered by classification. In both subfigures the red dots are the driving path, and each of the dots is the location where the panoramic camera exposed at an approximate spacing of 7.5 m. The GPS observations were post-processed with RTK [42] technology and can reach an accuracy of up to 0.1 m.

(a)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1



(b)

Figure 10. Overview of the test data: (a) the test area in Google Earth; (b)3D point cloud of the test area

We first extract a number of three typical line features from the LiDAR dataset. Second, we project the lines to the rectified mono-images to obtain the corresponding 2D lines, followed by a manual check for eliminating possible one-to-many uncertainty. Then, the proposed registration approach based on the panoramic camera model is applied. A linear feature from LiDAR is defined by two 3D endpoints; the linear feature from images is a sequence of pixels, in which only two pixels are used in the transformation. Finally, the registration results are assessed based on 2D and 3D visual comparison before and after registration, quantitative evaluation of check points, and statistical evaluation of edge pixels and 3D boundary points.

As mentioned in Section 2.1, the MMS records the POS data of vehicle when it captures image, while the exterior orientation parameters (EOP) of the camera relative to the vehicle platform coordinate system have been acquired in advance through system calibration. The EOP defines the position and rotation of the camera at the instant of exposure with six parameters: three Euclidean coordinates *(X, Y, Z)* of the projection center and the three angles of rotation $(\varphi, \omega, \kappa)$. Table 1 shows an example of POS data and EOP. Here, POS and EOP correspond to $\boldsymbol{M}_1(\boldsymbol{R}_1, \boldsymbol{T}_1)$ and $\boldsymbol{M}_2(\boldsymbol{R}_2, \boldsymbol{T}_2)$ respectively in Equation (9).

Table 1. POS of the vehicle platform and EOP of the camera aboard

	POS	EOP
***X*(m)**	38535802.519	-0.3350
***Y*(m)**	3400240.762	-0.8870
***Z*(m)**	762,11.089	0.4390
φ (°)	0.2483	-1.3489
ω (°)	0.4344	0.6250
κ (°)	87.5076	1.2000

Table 2 shows the known parameters of the six cameras in the panoramic camera model. R_x, R_y, R_z are the rotation angles about X, Y, Z axes, and T_x, T_y, T_z are translation along X, Y, Z axes. x_0, y_0 tell the pixel location of camera center, and f is the focal length. For more details about these parameters and definition of the coordinate systems one may refer to Section 2. These parameters are used in the line-based panoramic camera model (12).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

Table 2. Parameters of mono-cameras in the panoramic camera model (image size is 1616 x 1232)

Lens ID	R_x (radians)	R_y (radians)	R_z (radians)	T_x (meters)	T_y (meters)	T_z (meters)	x_0 (pixels)	y_0 (pixels)	f (pixels)
0	2.1625	1.5675	2.1581	0.0416	-0.0020	-0.0002	806.484	639.546	400.038
1	1.0490	1.5620	-0.2572	0.0114	-0.0400	0.0002	794.553	614.885	402.208
2	0.6134	1.5625	-1.9058	-0.0350	-0.0229	0.0006	783.593	630.813	401.557
3	1.7005	1.5633	-2.0733	-0.0328	0.0261	-0.0003	790.296	625.776	400.521
4	-2.2253	1.5625	-0.9974	0.0148	0.0388	-0.0003	806.926	621.216	406.115
5	-0.0028	0.0052	0.0043	0.0010	-0.0006	0.06202	776.909	589.499	394.588

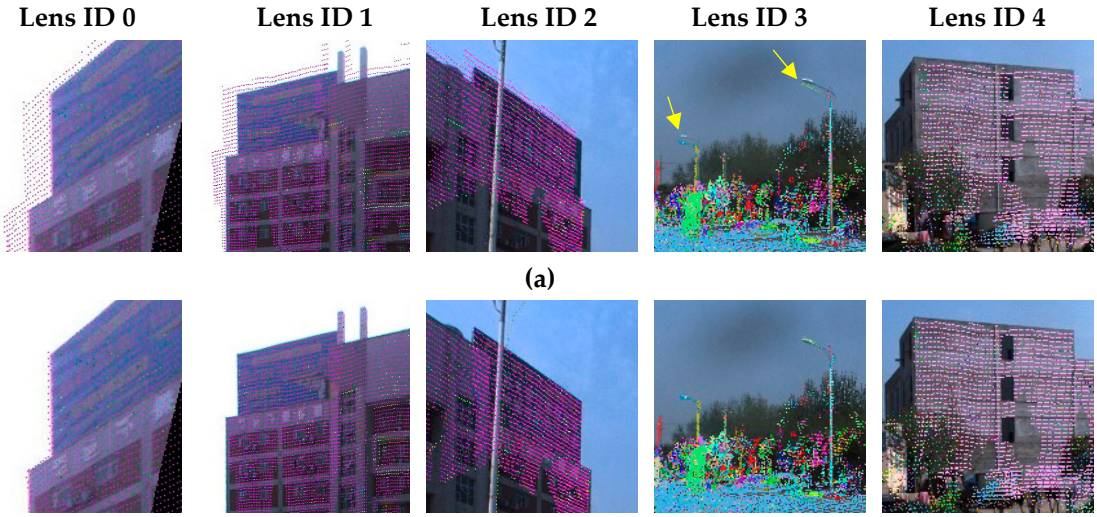
5.2. Registration Results

This section analyzes the registration results based on the panoramic camera model as the following routines: registration, visual inspection, quantitative and statistical evaluation. For comparison purpose, we also provide the results from the spherical camera model. **Table 3** lists the registration results based on the spherical and panoramic camera models. Here the deltas are the corrections after registration, which are the correction term in Equation (15). The RMSE (root mean square error) is the assessment of registration accuracy, which is defined in Equation (17). Both of RMSEs are below 5 pixels. Given an object point 20 meters away from the camera center, the error in object space is about 6 centimeters.

Table 3. Registration results based on the spherical and panoramic camera models

Model	Spherical		Panoramic	
	Deltas	Errors	Deltas	Errors
X (m)	-3.4372e-2	1.1369e-3	3.4328e-2	1.0373e-3
Y(m)	1.0653	1.2142e-3	1.0929	1.0579e-3
Z(m)	1.9511e-1	9.9237e-4	2.2075e-1	8.0585e-4
φ (°)	-1.2852e-2	1.4211e-3	-1.4731e-2	1.0920e-3
ω (°)	5.8824e-4	1.4489e-4	1.5866e-3	1.2430e-3
κ (°)	-7.9019e-3	8.4789e-4	-6.7691e-3	7.7509e-4
RMSE(pixels)	4.718		4.244	

Figure 11 (a) and (c) are the panoramic images and the labeled LiDAR point cloud before registration, and Figure (b) and (d) are those after registration with the panoramic camera model, both of which visually prove that the proposed method effectively removes the displacement between the panoramic image and LiDAR point cloud (see the borders of buildings in Lens ID 0-2, the two poles marked with yellow pointers in Lens ID 3, and the windows in Lens ID 4).



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

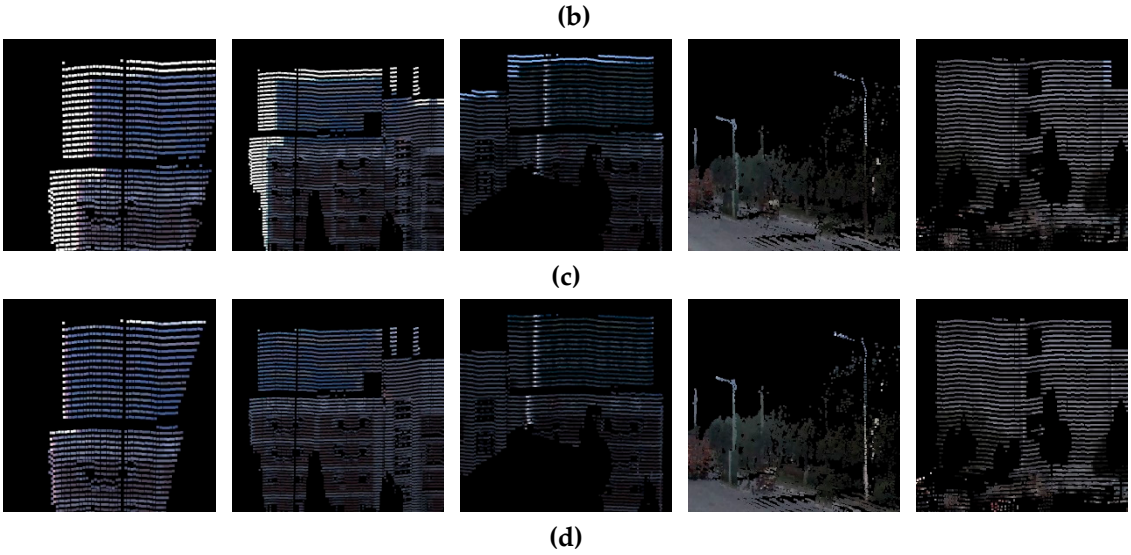


Figure 11. Alignments of two datasets before and after registration based on the panoramic camera model with lens id 0-4. (a) and (b) are LiDAR points projected to a panoramic image before and after registration respectively; (c) and (d) are 3d point cloud rendered by corresponding panoramic image pixels respectively.

To quantitatively evaluate the registration results based on the panoramic camera model, we manually selected check points both in LiDAR points and images. The selection of check points follows several rules: check points must be correspondent and recognizable in both datasets; check points should be selected from stationary objects with dense enough LiDAR points; check points should be evenly distributed horizontally. As a result, 20 check points are selected (see **Figure 12**) in 3D LiDAR point cloud and panoramic image, while 28 corresponding points are selected on the rectified mono-camera images. Please note that the number of check points for images are more than for LiDAR points because for the panoramic model the check points in overlapping areas appear twice in adjacent cameras.

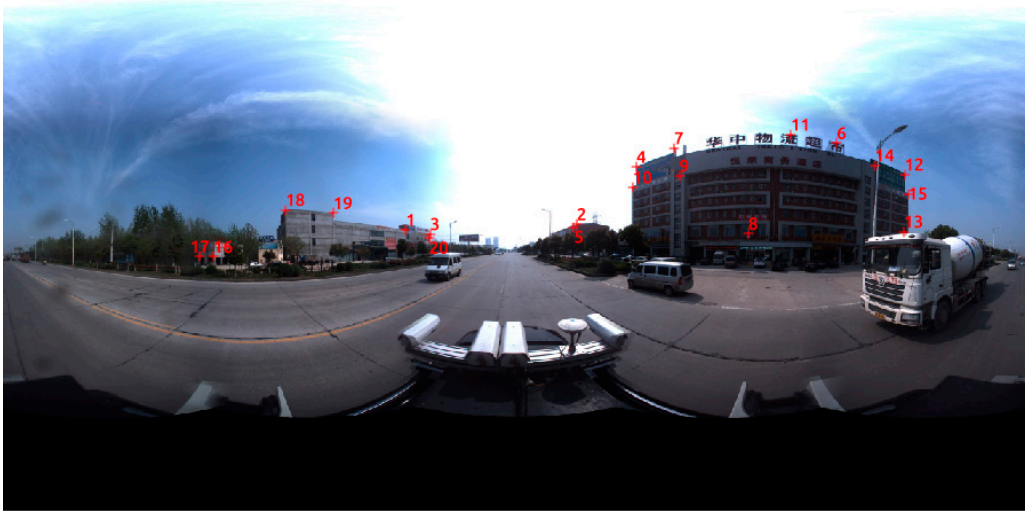
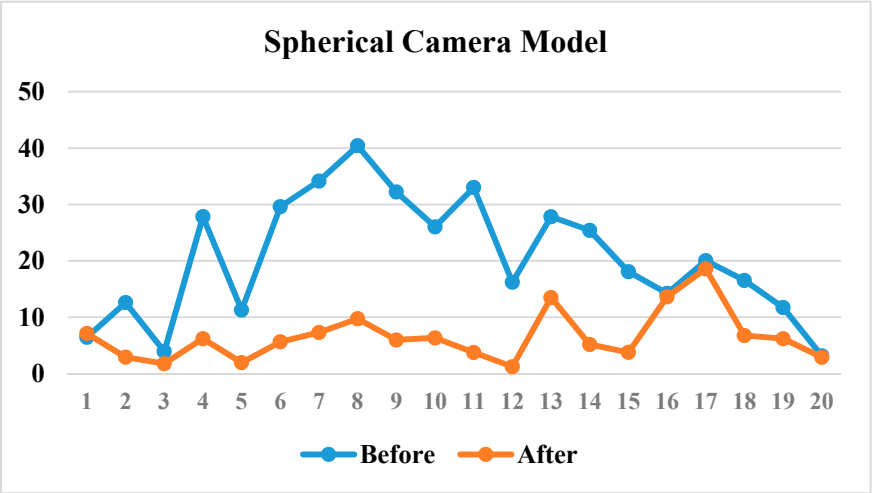


Figure 12. Check points distribution shown on panoramic image

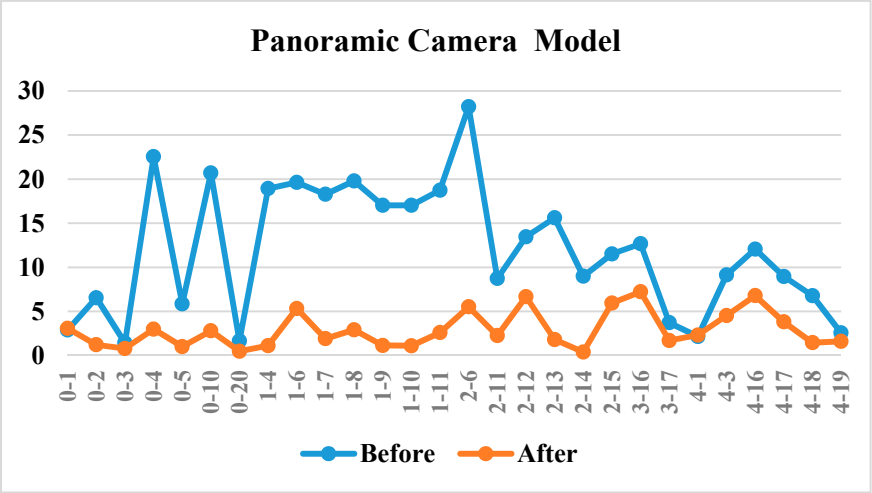
We projected all the check points to images to get their 2D coordinates before and after registration separately. Then we calculate the Euclidean distances between the projected 2D points and the image check points as residuals. According to **Figure 13**, both the spherical and panoramic camera models can reduce the residuals significantly while the latter shows a little advantage. The average residual decreases from 12.0 to 2.9 pixels with panoramic camera model (**Figure 13**(b)), and from 20.5 to 6.5 pixels with spherical model (**Figure 13**(a)) after registration. residuals on most check

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

points decrease much after registration, while a few of them changed little. The latter happens on the checks points with small enough initial residuals before registration, like No. 1, 3, 20.



(a)

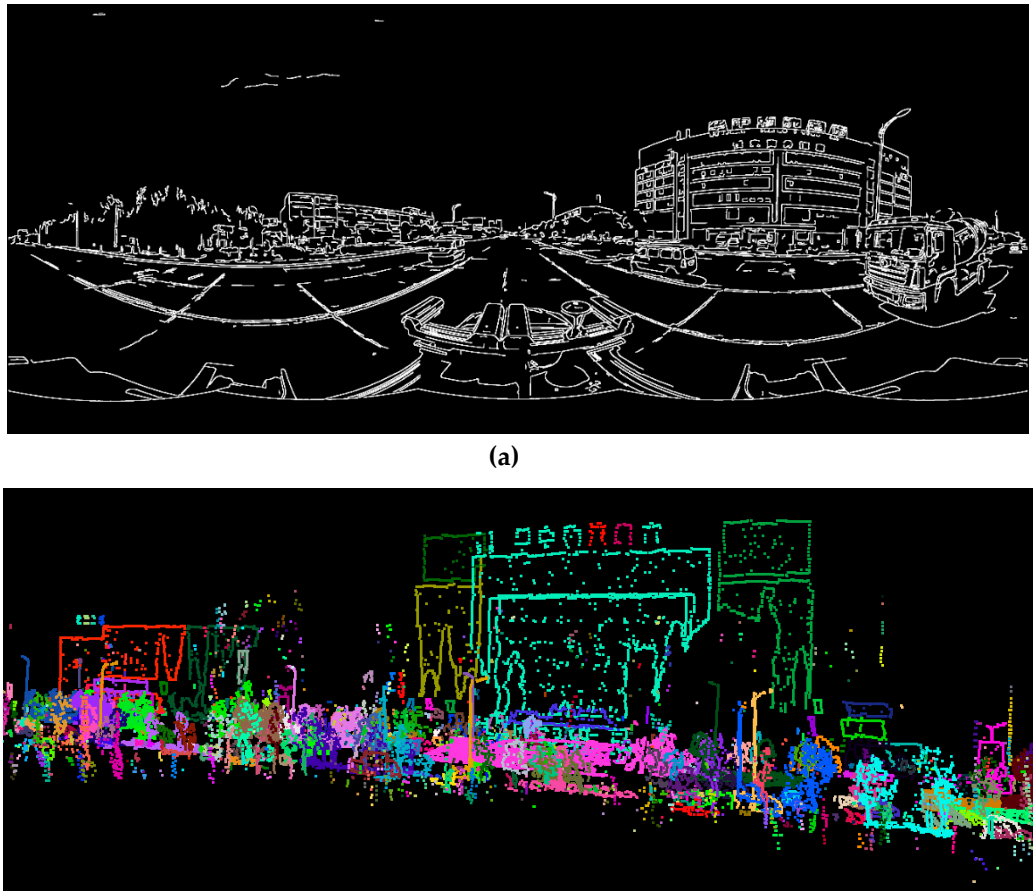


(b)

Figure 13. The residuals of check points before and after registration based on the panoramic camera model. The vertical axis is the residual in pixels; the horizontal axis is (a) ID of check points and (b) ID of lens and check points (lens ID - check point ID).

In order to further evaluate the overall effect of registration, we also calculate the statistical value *"overlap rate"* of linear features from the two datasets. For the panoramic image, we adopt the EDISON edge detector[43] to extract the edge pixels, as is shown in **Figure 14**(a). In EDISON edge detection, we use 8×8 windows and 20 pixels as the min length of lines to remove noise. For LiDAR points, we extract the boundary points using 30 k-nearest neighborhoods, and the result is shown in **Figure 14**(b). From the figures, we could notice that most of the geometric linear features in the two datasets are corresponding. Meanwhile, there are some disturbing elements such as the edges due to color difference in image and the boundary points from missing data in LiDAR points.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1



(a)

(b)

Figure 14. Linear features of the two datasets. (a)EDISON edge pixels in the panoramic image; (b) boundary points in LiDAR point cloud.

Figure 15 illustrates the *"overlap rate"* of linear features from the two datasets. Firstly, we project the 3D boundary points onto an image and get the binary image (see **Figure 15**(b)). Secondly, we perform the **Overlap** and **Union** operation on the two binary images (see **Figure 15**(a) and (b)) to get the overlap binary image (see **Figure 15**(c)) and the union binary image (see **Figure 15**(d)). We then count the numbers of non-zero pixels in both binary images: the number in overlap binary image is n_o, and the number in union binary image is n_u. Finally, we define the overlap rate as follows:

$$r = n_o/n_u \tag{18}$$

If the alignment of the image and LiDAR point cloud improves, there will be more overlapping linear features. This means the n_o will increase while n_u will decrease. Thus, the overlap rate will increase after the efficient registration. The results are shown in **Table 4**, where all of the overlap rates slightly increases up to 2% after registration. Among the five lenses, Len ID 1 has the most significant improvement. The main reason is that the main part of the image captured by Len ID 1 is a façade containing plenty of linear elements (see **Figure 5**).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 December 2016 doi:10.20944/preprints201612.0016.v1

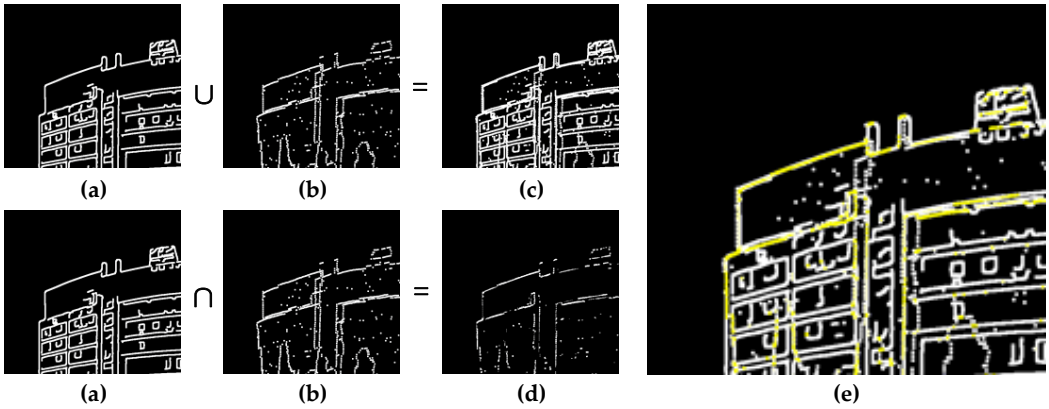


Figure 15. Definition of overlap rate: (a)image edge pixels; (b) LiDAR projected boundary points; (c) union of (a) and (b); (d) overlap of (a) and (b); (e) composition of (c) and highlighted (d).

Table 4. Overlap rate based on the panoramic camera model before and after registration

Lens ID	Before (%)	After (%)
0	7.80	8.29
1	8.31	10.30
2	11.32	11.83
3	9.84	9.90
4	7.42	7.54

6. Conclusions

This paper proposed a line-based registration approaches for panoramic image and LiDAR point cloud collected by a MMS. We first established the transformation model between primitives from the two datasets in the camera-centered coordinate system. Then we extract primitives, three typical linear features in street view, from LiDAR automatically and from panoramic images through a semi-automation process. Using the extracted features, we resolved the relative orientations and translations between the camera and LiDAR.

Compared with other related work, the main contribution of this work is that we focus on the registration between LiDAR and the panoramic camera, widely used in MMSs, instead of the conventional frame camera. Two types of camera model, the spherical and panoramic camera models, are utilized in our registration. The experiments results show both can remove obvious misalignment between the LiDAR point cloud and the panoramic image. However, the panoramic model could achieve a better registration accuracy. It is suggested one may need to choose a suitable camera model on certain data fusion task. For rendering LiDAR point cloud with acceptable misalignment, the spherical camera model is adequate; while for high level fusion task such as façade modeling, the panoramic camera model is needed.

There are still some ways to further improve the registration accuracy and automation in the future work. First, the errors from LiDAR point cloud itself could not be overlooked. In our case, the LiDAR points were collected by three laser scanners, whose calibration errors also influence the registration accuracy. Moreover, finding reliable correspondence from different datasets may need to use local statistical similarity such as mutual information. In addition to geometrical features, the physical attributes of LiDAR such as intensity may also be utilized in future research.

Acknowledgments: The authors would like to thank Professor Yuchun Huang and other staff in Mobile Mapping Laboratory, School of Remote Sensing and Information Engineering, Wuhan University, for their assistance with data collection and preprocessing. This work was partially supported by the National Natural Science Foundation of China (41271431, 41471288 and 61403285), the National Science and Technology Major Project of China (01-Y30B16-9001-14/16), and the National High-Tech R&D Program of China (2015AA124001).

Author Contributions: Tingting Cui performed the experiments and wrote the draft manuscript; Shunping Ji conceived and designed the experiments; Jie Shan advised the study and contributed to writing in all phases of the work; Jianya Gong and Kejian Liu revised the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cornelis, N.; Leibe, B.; Cornelis, K.; Van Gool, L. 3d urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision* **2008**, *78*, 121-141.
2. Wonka, P.; Muller, P.; Watson, B.; Fuller, A. Urban design and procedural modeling. In *ACM SIGGRAPH 2007 courses*, ACM: San Diego, California, 2007; pp 229-229.
3. Zhuang, Y.; He, G.; Hu, H.; Wu, Z. A novel outdoor scene-understanding framework for unmanned ground vehicles with 3d laser scanners. *Transactions of the Institute of Measurement and Control* **2015**, *37*, 435-445.
4. Li, D. Mobile mapping technology and its applications. *Geospatial Information* **2006**, *4*, 125.
5. Pu, S.; Vosselman, G. Building facade reconstruction by fusing terrestrial laser points and images. *Sensors* **2009**, *9*, 4525-4542.
6. Brenner, C. Building reconstruction from images and laser scanning. *International Journal of Applied Earth Observation and Geoinformation* **2005**, *6*, 187-198.
7. Pu, S. Knowledge based building facade reconstruction from laser point clouds and images. University of Twente, Enschede, the Netherlands, 2010.
8. Wang, R. Towards urban 3d modeling using mobile lidar and images. McGill University, Montreal, Canada, 2011.
9. Park, Y.; Yun, S.; Won, C.S.; Cho, K.; Um, K.; Sim, S. Calibration between color camera and 3d lidar instruments with a polygonal planar board. *Sensors* **2014**, *14*, 5333-5353.
10. Naroditsky, O.; Patterson, A.; Daniilidis, K. In *Automatic alignment of a camera with a line scan lidar system*, Robotics and Automation (ICRA), 2011 IEEE International Conference on, 9-13 May 2011, 2011; pp 3429-3434.
11. Gong, X.; Lin, Y.; Liu, J. 3d lidar-camera extrinsic calibration using an arbitrary trihedron. *Sensors* **2013**, *13*, 1902-1918.
12. Zhuang, Y.; Yan, F.; Hu, H. Automatic extrinsic self-calibration for fusing data from monocular vision and 3-d laser scanner. *IEEE Transactions on Instrumentation and Measurement* **2014**, *63*, 1874-1876.
13. Levinson, J.; Thrun, S. In *Automatic online calibration of cameras and lasers*, Robotics: Science and Systems, 2013.
14. Mishra, R.; Zhang, Y. A review of optical imagery and airborne lidar data registration methods. *The Open Remote Sensing Journal* **2012**, *5*, 54-63.
15. Ayman, H.; Mwafag, G.; Michel, M.; Rami, A.-R. Photogrammetric and lidar data registration using linear features. *Photogrammetric engineering and remote sensing* **2005**, *71*, 699-707.
16. Brown, L. A survey of image registration techniques. *ACM Computing Surveys* **1992**.
17. Rönnholm, P. *Registration quality-towards integration of laser scanning and photogrammetry*. Bundesamt für Kartographie und Geodäsie: 2011.
18. Liu, L.; Stamos, I. In *Automatic 3d to 2d registration for the photorealistic rendering of urban scenes*, Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 20-25 June 2005, 2005; pp 137-143 vol. 132.
19. Liu, L.; Stamos, I. A systematic approach for 2d-image to 3d-range registration in urban environments. *Computer Vision and Image Understanding* **2012**, *116*, 25-37.
20. Patias, P.; Petsa, E.; Streilein, A. *Digital line photogrammetry: Concepts, formulations, degeneracies, simulations, algorithms, practical examples*. 1995; p 54.
21. Schenk, T. From point-based to feature-based aerial triangulation. *ISPRS Journal of Photogrammetry and Remote Sensing* **2004**, *58*, 315-329.

22. Zuxun, Z.; Yongjun, Z.; Jianging, Z.; Hongwei, Z. Photogrammetric modeling of linear features with generalized point photogrammetry. *Photogrammetric engineering and remote sensing* **2008**, *74*, 1119-1127.
23. Mastin, A.; Kepner, J.; Fisher, J. In *Automatic registration of lidar and optical images of urban scenes*, Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 20-25 June 2009, 2009; pp 2639-2646.
24. Parmehr, E.G.; Fraser, C.S.; Zhang, C.; Leach, J. Automatic registration of optical imagery with 3d lidar data using statistical similarity. *ISPRS Journal of Photogrammetry and Remote Sensing* **2014**, *88*, 28-40.
25. Wang, R.; Ferrie, F.P.; Macfarlane, J. Automatic registration of mobile lidar and spherical panoramas. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, IEEE: 2012; pp 33-40.
26. Torii, A.; Havlena, M.; Pajdla, T. In *From google street view to 3d city models*, Computer vision workshops (ICCV Workshops), 2009 IEEE 12th international conference on, 2009; IEEE: pp 2188-2195.
27. Micusik, B.; Kosecka, J. In *Piecewise planar city 3d modeling from street view panoramic sequences*, Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009; IEEE: pp 2906-2912.
28. Shi, Y.; Ji, S.; Shi, Z.; Duan, Y.; Shibasaki, R. Gps-supported visual slam with a rigorous sensor model for a panoramic camera in outdoor environments. *Sensors* **2012**, *13*, Sensors.
29. Ji, S.; Shi, Y.; Shi, Z.; Bao, A.; Li, J.; Yuan, X.; Duan, Y.; Shibasaki, R. Comparison of two panoramic sensor models for precise 3d measurements. *Photogrammetric Engineering & Remote Sensing* **2014**, *80*, 229-238.
30. PointGrey. Ladybug 3. https://www.ptgrey.com/ladybug3-360-degree-firewire-spherical-camera-systems
31. SICK. Lms5xx. https://www.sick.com/de/en/product-portfolio/detection-and-ranging-solutions/2d-laser-scanners/lms5xx/c/g179651
32. Sairam, N.; Nagarajan, S.; Ornitz, S. Development of mobile mapping system for 3d road asset inventory. *Sensors* **2016**, *16*, 367.
33. Point Grey Research, I. Geometric vision using ladybug cameras. http://www.ptgrey.com/tan/10621
34. Yang, B.; Wei, Z.; Li, Q.; Li, J. Automated extraction of street-scene objects from mobile lidar point clouds. *International Journal of Remote Sensing* **2012**, *33*, 5839-5861.
35. Arachchige, N.H.; Perera, S.N.; Maas, H.G. Automatic processing of mobile laser scanner point clouds for building fa ade detection. In *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, Copernicus Publications: 2012; Vol. XXXIX-B5, pp 187-192.
36. Rabbani, T.; van Den Heuvel, F.; Vosselmann, G. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **2006**, *36*, 248-253.
37. Schnabel, R.; Wahl, R.; Klein, R. In *Efficient ransac for point-cloud shape detection*, Computer graphics forum, 2007; Wiley Online Library: pp 214-226.
38. Pu, S.; Rutzinger, M.; Vosselman, G.; Oude Elberink, S. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing* **2011**, *66*, S28-S39.
39. Yu, Y.; Li, J.; Guan, H.; Wang, C.; Yu, J. Semiautomated extraction of street light poles from mobile lidar point-clouds. *IEEE Transactions on Geoscience and Remote Sensing* **2015**, *53*, 1374-1386.
40. Kumar, P.; McElhinney, C.P.; Lewis, P.; McCarthy, T. An automated algorithm for extracting road edges from terrestrial mobile lidar data. *ISPRS Journal of Photogrammetry and Remote Sensing* **2013**, *85*, 44-55.
41. Tan, J.; Li, J.; An, X.; He, H. Robust curb detection with fusion of 3d-lidar and camera data. *Sensors* **2014**, *14*, 9046-9073.
42. J., M.; T., H.; J., T.; R., K. In *Development of an autonomous mobile surveillance system using a network-based rtk-gps*, International Conference on Robotics and Automation, 2005.
43. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2002**, *24*, 603-619.