

# A New Flying Range Sensor: Aerial Scan in Omini-directions

Bo Zheng, Xiangqi Huang, Ryoichi Ishikawa, Takeshi Oishi, Katsushi Ikeuchi

The University of Tokyo

{zheng, huang, Ishikawa, Oishi, ki}@cvl.iis.u-tokyo.ac.jp

## Abstract

*This paper presents a new flying sensor system to capture 3D data aerially. The hardware system, consisting of an omni-directional laser scanner and a panoramic camera, can be mounted under a mobile platform (e.g., a balloon or a crane) to achieve the aerial scanning with high resolution and accuracy. Since the laser scanner often requires several minutes to complete an omni-directional scan, the raw data is distorted seriously due to the unknown and uncontrollable movement during the scanning period. To overcome this problem, 1) we first synchronize the two sensors and spherically calibrate them together; 2) our approach then recovers the sensor motion by utilizing the spatial and temporal features extracted both from the image sequences and point clouds; and 3) finally the distorted scans can be rectified with the estimated motion and aligned together automatically. In experiments, we demonstrate that the method achieves a substantially good performance for indoor/outdoor aerial scanning in the applications such as Angkor Wat 3D preservation and manufacturing 3D survey with respect to other state-of-the-art methods.*

## 1. Introduction

Many applications in science and industry, such as heritage preservation, 3D documentation, urban modeling and manufacturing survey, rely on aerially capturing the 3D shape of scenes. Since the data missing always happens on top of the scans captured by the sensors on ground (as an example shown Fig. 1 (a)), the aerial scanning is often required to capture the top missing part. In this paper, we design a new flying range sensor extend from our previous work in [4] to achieve aerial scanning, as shown in Fig. 1 (b) and (c).

3D laser scanners using Time-of-Flight (ToF) methods which measure the laser flying time (or phase difference) that it takes for an object, provide the most accurate shape recovery compared to the passive or physical techniques [5].

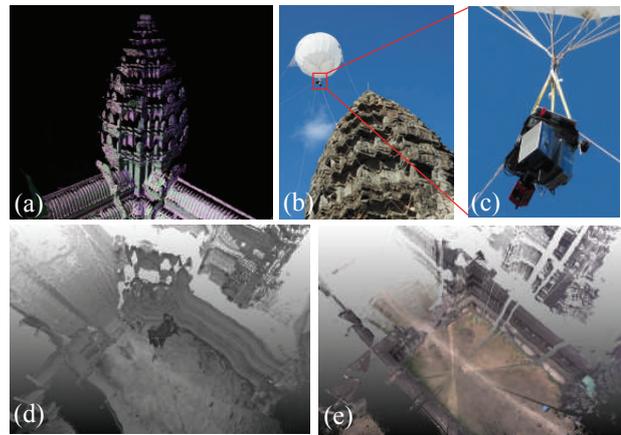


Figure 1. (a) Data missing from the top part of the tower. (b) Proposed aerial scanning system mounted under a balloon. (c) Close-up of (b). (d) Raw data with distortion obtained by balloon scanner. (e) Our method: Rectified data by removing the distortion.

Many 3D laser scanners in the market, such as Z+F Imager 5010 [2], achieve the accuracy at 0.1-5mm according to the distances and materials, and have been shown to produce the most accurate depth information over the distance from 1m to 120m. In this paper, we focus on laser scanning techniques adopted in aerial scan.

No matter how accurate and detailed data the laser scanners can capture, it usually takes several minutes for one scan under the moderate resolution. Thus, different from the scans on the ground, unfortunately the aerial scans are often distorted due to the unknown and uncontrollable sensor motion, as an example shown in Fig. 1(d) which is captured from the balloon as shown in Fig. 1 (b) and (c) where the sensor often shakes due to the physical disturbance such as wind.

We present an approach for removing the data distortion by estimating the sensor motion as shown in Fig. 1 (e). We first synchronize the two sensors by a UTC time clocker such as that each 3D point and image frame can be associ-

ated with UTC time. Second we calibrate the two sensors by assuming that each of them is associated with a spherical coordinate system and then a 6-parameters transformation is estimated for aligning coordinate systems. Third we extract image features which are tracked through 2D image sequence and scanned by the laser, and recover the sensor motion by utilizing the spacial and temporal features extracted both from the image sequences and point clouds.

This paper makes the following contributions.

- To our knowledge, we are the first to build an aerial scanning system using a high-resolution and omnidirectional laser scanner.
- We propose a simple but accurate and automatic method to spherically calibrate the two sensors.
- We develop a new approach that estimates sensor motion. Different from the previous methods which independently relies on the visual odometry, Lidar odometry or a third-party positioning sensor such as GPS or IMU, we explored the sensor-fused features extracted from both range and color images.
- After the aerial scans are rectified, our method achieves the fully automatic registration by utilizing the correspondences between scans, since each 3D point is associated with a pixel on a image frame.

## 2. Related Work

The study of 3D sensing technologies in a mobile platform can be traced back to related studies in the following streams:

**Camera system** Image based methods are common for state estimation [26]. While the stereo camera systems, such as [22, 18], help determine scale of the motion provided by the baseline reference between two cameras, the monocular camera systems (*e.g.*, [10, 15, 21, 9, 16]), generally does not solve the scale of motion without aiding from other sensors or assumptions about motion. These method have shown impressive results but the precision and resolution is still far from the level of the laser scanners.

**RGB-D camera system** The introduction of RGB-D cameras provides an efficient way to associate visual images with depth. Motion estimation with RGB-D cameras [20, 30, 8, 28] can be conducted easily with scale. A number of RGB-D SLAM methods are also proposed showing promising results [12, 11, 14]. While the real-time RGB-D sensor is good for motion tracking, the low Light Concentration Ratio (LCR) [19] leads to the low signal-to-noise ratio (SNR). On the other hand, laser scanning systems concentrate the available light source power in a smaller region, resulting in a largest SNR [19]. Also RGB-D cameras usually do not have the shooting range as long as laser scanners.

**Robotics: SLAM** The motion estimation methods with images or/and additional depth sensor (*e.g.*, 1-axis lidar) are well designed for SLAM system in robotics, such as [31, 8, 28, 12, 11, 14]. However, our method is designed to utilize depth information from a 2-axis laser scanner. It involves features both on images and 3D point clouds in solving for motion.

Laser scan gets large signal-to-noise ratio, but require long acquisition times, which leads to the motion distortion present in point clouds as the scanner continually ranges and moves. One way to remove the distortion is incorporating other sensors to recover the motion. For example, Scherer *et al.* provide navigation system [25, 7] using stereo visual odometry integrated with an IMU to estimate the motion of a micro-aerial vehicle. Distorted clouds are rectified by the estimated motion. In comparison to these method, our method only focuses on the visual information obtained from both camera and laser scanner to capture the RGB-D data without using the third-party positioning sensor.

**Laser scanner-only system** The stop-and-go scanning manner is suitable for the static laser scanner, *e.g.*, Konica Minolta Vivid 9 adopted in [13]. It has also shown that state estimation can be made with 3D lidars only. For example, Tong *et al.* match visual features in intensity images created by stacking laser scans from a 2-axis lidar to solve for the motion [27]. The motion is modeled with constant velocity and Gaussian processes. However, in most cases of the aerial scanning, the velocity is not able to assume to be constant.

**Laser scanner + Camera system** The recent work, such as [23], combing lidar with RGB camera for indoor/outdoor 3D reconstruction which enhance the robustness for sensor motion estimation, but however these techniques cannot reach the resolution and precision required for the applications such as heritage object preservation. The work such as [6, 32] adopted similar sensors to build the scanning system, however laser scanner works on 1-axis scanning and faces the problem of horizontal movement. The most related work to our method are the studies [4, 24] that consider mounting the laser scanner sensor and camera under a balloon. While the method in [4] supposes the sensor motion is in  $C^2$  smooth, the method in [24] supposes the 3D priors are known by ground acquisition in advance. In contrast, our method is not limited in such assumptions.

## 3. Method

Fig. 2 shows the overview of our method. Input data is generally captured from two hardware sensors: a laser sensor and a panoramic image sensor. While the scanner is a 2-axis rotating laser scanner which captures a 3D point cloud in omni-directions, and the camera captures panoramic image sequences during scanning. Since the range sensor works usually at a high resolution mode, it usually spends

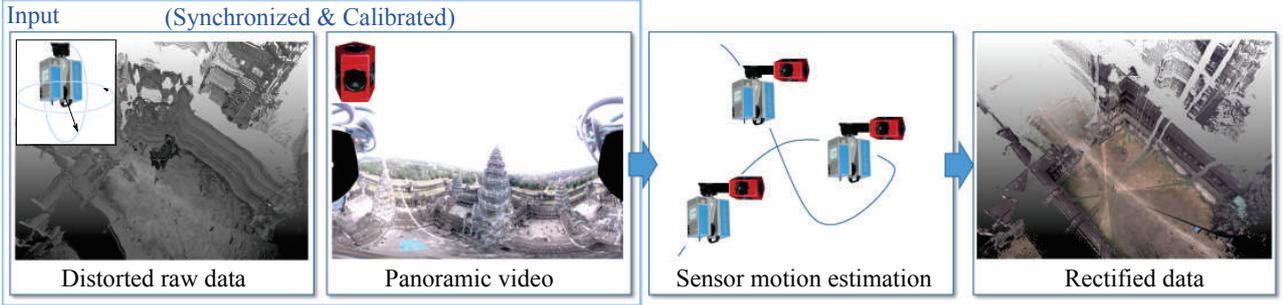


Figure 2. Method overview. From left to right: the inputs are previously synchronized and calibrated 3D point cloud and 2D image sequence; the sensor motion is estimated by using 2D/3D features; and the distorted data can be rectified according to the sensor motion.

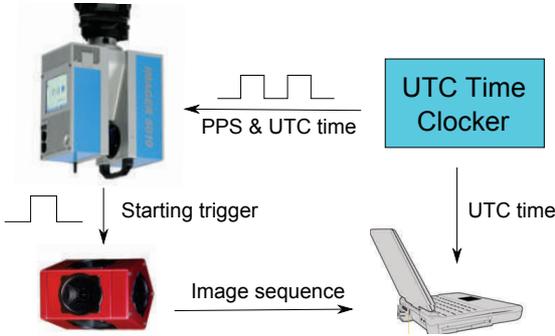


Figure 3. Sensors synchronization: UTC time clocker sends UTC time information to the range sensor and the PC recording images. The range sensor sends a pulse trigger to camera when the laser is fired to capture the point.

several minutes for a complete scan. Thus the input raw data is often distorted (as shown in Fig. 2 left most) due to the unknown sensor movement in the air.

In our hardware system, we adopted a laser scanner: Z+F Imager 5010C [2] and a panoramic camera: Point-Grey Ladybug 3 [1] which are fixed onto a board as shown in Fig. 1(c). To eliminate the occlusion between the two sensors, we fix the panoramic camera next to the scanner to guarantee both of them can capture the much more overlap in the scene.

The problem addressed in this paper is to estimate the motion of the laser scanner and camera system, and then reconstruct the 3D point cloud of the scanned environment with the estimated motion, as shown in Fig. 2 right part. In this section, we present the system design including sensor synchronization, spherical calibration and the motion estimation used for removing the distortion.

### 3.1. Sensor synchronization

As pre-process, synchronization guarantees that the laser scanner and the camera work simultaneously such as that

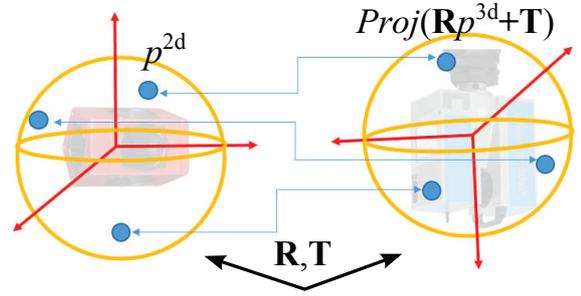


Figure 4. Calibration: the relative pose and position between camera and scanner is 6-DoF transformation including rotation  $\mathbf{R}$  and translation  $\mathbf{T}$ . Since both the panoramic image and omnidirectional scan can be projected onto a sphere, the transformation can be estimated by minimizing the differences between the correspondences on both sphere.

the capturing time of each point can be known to correspond to the frame captured at same time.

As shown in Fig. 3, a UTC time clocker consisting of a GPS receiver and a high-accuracy internal clocker sends the UTC time to both laser scanner and a PC for image storage, so as that each 3D point and image frame can be associated with UTC time. Also, once the scanner fires the first laser to start scan, it sends single pulse trigger to activate camera for capturing images. In this paper, we ignore the time delay caused by hardware which should be within 10ms.

### 3.2. Calibration

Calibration aims to find the relative pose and position between the two sensors. In this paper, we suppose this relative relation is 6-parameter transformation including a rotation matrix  $\mathbf{R}$  and a translation vector  $\mathbf{T}$ .

To figure out the 6-parameter transformation between the two sensors, we suppose a spherical coordinate system is associated with each sensor (see, Fig. 4). Suppose the spherical systems  $\mathcal{O}_l$  and  $\mathcal{O}_c$  are originated at the optical centers of laser scanner and camera respectively, once this the 6-

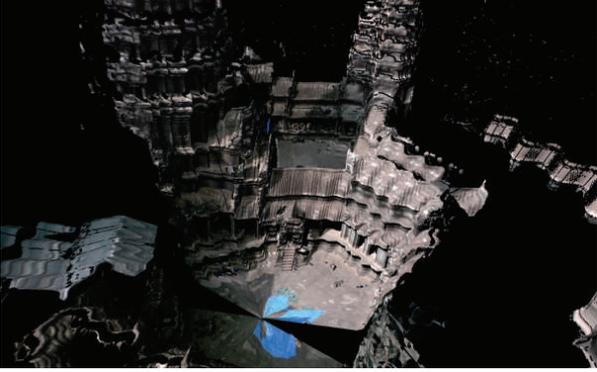


Figure 5. An example shows that after calibration, the pixels on different image frames can be mapped to the corresponding 3D points.

parameter transformation is known,  $\mathcal{O}_l$  can be aligned to  $\mathcal{O}_c$  together by the following transformation:

$$\mathbf{p}^{2d} = \text{Proj}(\mathbf{R}\mathbf{p}^{3d} + \mathbf{T}) \quad (1)$$

where  $\mathbf{p}^{2d} \in \mathcal{O}_c$  is a pixel on a unit spherical image captured by camera;  $\mathbf{p}^{3d}$  is the corresponding point in the original 3D coordinate system of the laser scanner; and function  $\text{Proj}(\cdot)$  projects the 3D point  $\mathbf{p}^{3d}$  transformed by  $\mathbf{R}$  and  $\mathbf{T}$  onto a unit sphere, *i.e.*,  $\text{Proj}(\mathbf{R}\mathbf{p}^{3d} + \mathbf{T}) \in \mathcal{O}_l$ .

Therefore the calibration problem can be viewed as solving out  $\mathbf{R}$  and  $\mathbf{T}$  through a minimization:

$$\arg \min_{\mathbf{R}, \mathbf{T}} \sum_i^N \rho_i \|\mathbf{p}_i^{2d} - \text{Proj}(\mathbf{R}\mathbf{p}_i^{3d} + \mathbf{T})\| \quad (2)$$

where  $\rho_i$  is a loss function using the Huber for robust estimation. This minimization of the spherical project error over 6 parameters can be solved by the non-linear optimization such as Levenberg-Marquardt algorithm.

Now, the problem becomes how to find the 2D-3D corresponding pairs between panoramic images and 3D point cloud. Many previous methods such as the 3D texture mapping method [3] proposed to manually select 2D/3D points on images and point clouds. However error often remains when doing the pixel or 3D point selection manually.

Fortunately, the reflectance information can be obtained from almost all the laser scanners, and thus the reflectance (intensity) associated with each 3D point can be projected onto a unit sphere to form a reflectance spherical image. In this paper, we present an automatic calibration method by matching the correspondence between two spherical images: reflectance image and RGB image. To this end, 1) we first wrap a panorama into several perspective images sampled in various directions as proposed in [29]; 2) then we extract corner points encoded with DoP descriptor [33] on each perspective images; 3) We wrap feature points back

onto the spherical images and then find the matches between two images. Note, since this calibration should rely on the number of matches in the static scene, in practice we put checker boards into the scene to achieve better performance.

However even the point-to-pixel correspondences can be solved out by calibration, data distortion often remained for aerial scan as shown in Fig. 5. Since at any time, the relative pose and position between two sensors do not change, and suppose the distortion of point cloud captured within the time taken for capturing one frame can be ignored, each 3D point can be associated with a pixel on the corresponding frame. However the colored point cloud is distorted due to the unknown sensor motion.

### 3.3. Motion estimation

To remove the distortion of data, as shown in Fig. 5, the physical motion of the sensor has to be known. In this section we present an approach for robust motion estimation. To this end, we first track temporal 2D-3D features on both images and point clouds, and then the relative geometric relation can be estimated between two consecutive frames using the features.

#### 3.3.1 Temporal 2D-3D feature tracking

Given the synchronized and calibrated data from the two sensors, a temporal 2D-3D features can be tracked using the following three steps:

**2D feature tracking:** The 2D correspondences along the image sequence are required to geometrically relate consecutive frames using KLT tracker [17]. Fig. 6 illustrates that the  $k$ -th 2D feature on frame  $i$ :  $\mathbf{x}_i^k$  can be tracked from the last frame shown in same color.

**3D feature correspondence:** Within the exposure time  $(t_i, t_i + \Delta T_e)$  for the  $i$ -th frame, we search the 3D correspondence  $\mathbf{X}(t_d)$  captured at time  $t_d$  and closest to  $\mathbf{x}_i^k$  on the frame:

$$t_d = \arg \min_{t \in (t_i, t_i + \Delta T_e)} \|\text{Proj}(\mathbf{R}\mathbf{X}(t) + \mathbf{T}) - \mathbf{x}_i^k\| \quad (3)$$

For easy description, we re-denote the 2D-3D correspondence pair  $\mathbf{x}_i^k$  and  $\mathbf{X}(t_d)$  as:  $\mathbf{x}_i^j$  and  $\mathbf{X}_i^j$ , namely the  $j$ -th feature tracked on frame  $i$ .

**Remove feature outliers :** Sometimes features on image are not “stable”, *e.g.*, the features detected from leaves of trees, which cause mismatches in the range image. In order to remove such features, we check the temporally neighbor points around  $\mathbf{X}(t_d)$ . That is, we remove the features whose neighboring 3D points in the cloud have a variance larger than a threshold:  $\frac{\partial \|\mathbf{X}(t_d)\|}{\partial t_d} > T_o$ . Also the features which are associated with low reflectance obtained from laser scanner are also removed.

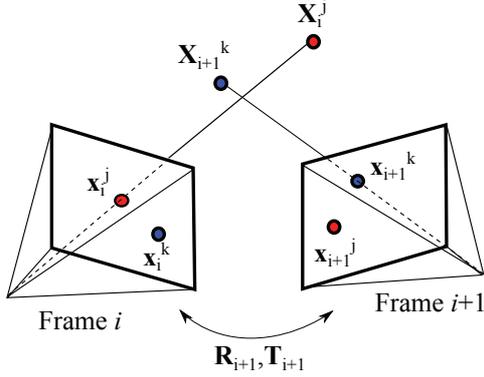


Figure 6. Motion estimation: the relative motion between two frames can be recovered by rectifying the shared features.

### 3.3.2 Consecutive pose estimation

Since the extrinsic parameters between the camera and laser scanner are calibrated, this allows us to use a single coordinate system for both sensors, namely the sensor coordinate system. For simplicity of calculation, we choose the sensor coordinate system to coincide with the camera coordinate system  $\mathcal{O}_c$  and all laser points in  $\mathcal{O}_l$  are projected into the camera spherical coordinate system upon receiving. World coordinate system  $\mathcal{O}_w$  is the coordinate system coinciding with  $\mathcal{O}_c$  at the starting position. Therefore, the pose estimation problem can be stated as: Given images and point clouds perceived in  $\mathcal{O}_c$ , determine the pose of each  $\mathcal{O}_c$  with respect to  $\mathcal{O}_w$  and map the traversed environment in  $\mathcal{O}_w$ .

**Triangulation:** To factorize the rotation and translation between two consecutive panoramic frames, our triangulation method utilizes both of the 2D-3D feature tracks obtained by the method described above.

Suppose that, as shown in Fig. 6, on two consecutive frames  $i$  and  $i + 1$ , the  $j$ -th feature track is known in both frames as:  $\mathbf{x}_i^j$  and  $\mathbf{x}_{i+1}^j$ , and the temporally 3D correspondence at frame  $i$  is found as  $\mathbf{X}_i^j$ . While the  $k$ -th feature track is same, the 3D correspondence is found at frame  $i + 1$ . The triangulation between the two frames is worked out by minimizing the projection errors as:

$$\begin{aligned} & \{\mathbf{R}_{i+1}, \mathbf{T}_{i+1}\} \\ &= \arg \min_{\{\mathbf{R}, \mathbf{T}\}} \sum_j \|\text{Proj}(\mathbf{R}^T(\mathbf{X}_i^j - \mathbf{T})) - \mathbf{x}_{i+1}^j\|_2, \\ &+ \sum_k \|\text{Proj}(\mathbf{R}\mathbf{X}_{i+1}^k + \mathbf{T}) - \mathbf{x}_i^k\|_2. \end{aligned} \quad (4)$$

We apply the Levenberg-Marquardt method for solving the non-linear optimization problem.

Now the triangulation for whole image sequence can be viewed as: suppose frames  $1, 2, \dots, n$  have been worked out (aligned to world coordinates  $\mathcal{O}_w$ ), then the  $(n + 1)$ -th

frame can be triangulated as minimization

$$\begin{aligned} & \{\mathbf{R}_{n+1}, \mathbf{T}_{n+1}\} \\ &= \arg \min_{\{\mathbf{R}, \mathbf{T}\}} \sum_{i=1}^n \sum_j \|\text{Proj}(\mathbf{R}^T(\mathbf{X}_i^j - \mathbf{T})) - \mathbf{x}_{n+1}^j\|_2, \\ &+ \sum_{i=1}^n \sum_k \|\text{Proj}(\mathbf{R}\mathbf{X}_{n+1}^k + \mathbf{T}) - \mathbf{x}_i^k\|_2. \end{aligned} \quad (5)$$

Therefore our motion estimation method can be summarized by Alg. 1.

---

#### Algorithm 1: Motion estimation

---

- Data:** 2D feature tracks  $\mathbf{x}_i^j$  and their 3D correspondences  $\mathbf{X}_i^j$
- Result:** Motion of each frame  $\{\mathbf{R}_i, \mathbf{T}_i\}$  in  $\mathcal{O}_w$
- 1 Select frame  $m$  who owns the largest number of feature pairs, and add it into the list  $\{\mathcal{L}\} \leftarrow m$ ;
  - 2 **for** frames not in  $\{\mathcal{L}\}$  **do**
  - 3     Select the frame  $n$  who shares most features in previous frames in  $\{\mathcal{L}\}$  ;
  - 4     Do triangulation between frame  $n$  and previous frames  $\{\mathcal{L}\}$  using Eq. (5);
  - 5     Add  $n$  into frame list:  $\{\mathcal{L}\} \leftarrow n$ ;
  - 6 **end**
- 

### 3.4. Automatic scan alignment

Our method benefit for the scan alignment without need the manually initial alignment before doing the interactive closest point (ICP) registration, since the 3D correspondences can be easy to find from the two sets of images. Given the 2D feature tracks  $\{\mathbf{x}_i^j\}_{i,j}$  which have been found the 3D correspondences  $\{\mathbf{X}_i^j\}_{i,j}$  in each laser scan, the problem for aligning the two scans can be solved by steps: 1) finding the matches between two sets of 2D feature tracks, where each feature can be encoded with a descriptor (e.g. [33]); and 2) calculating rigid transformation between two scans by minimizing the Euclidean distance between the two sets of 3D correspondences.

## 4. Experiments

We quantitatively and qualitatively evaluate our method based on two applications: 1) a large-scale outdoor scanning for Angkor Wat 3D preservation and 2) a indoor manufacturing scanning. While the former mounted our system under a balloon and the latter mounted it under a crane, but both of these aerial scanning face the problem caused by unknown sensor movement.

We adopted the main hardware components as: 1) laser scanner: Z+F Imager 5010C [2], 2) Panoramic camera: Ladybug 3 [1], and 3) UTC time clocker: U-blox EVK-6T



Figure 7. Comparison on data rectification: (a) raw data captured by laser scanner. (b) rectification using motion calculated by visual odometry [10]. (c) flying range sensor rectification [4] (d) our result.

evaluation took kit. The laser scanner has the field of view  $320^\circ \times 360^\circ$  working at resolution:  $2544 \times 1111$  points and 50s/scan. The panoramic camera captures the data in resolution  $5400 \times 2700$  and 16fps. The stitching mode is set

as 20m and 10m for outdoor and indoor scene respectively. UTC time closer send UTC time at 1 PPS.

The data was online recorded by the internal memory of laser scanner and PC for camera. The offline rectifica-

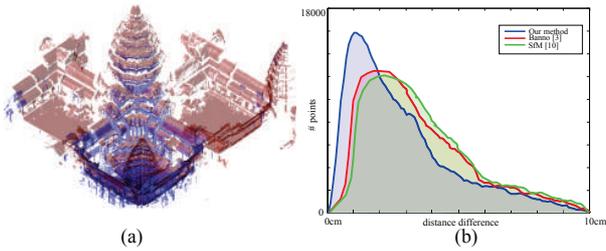


Figure 8. Accuracy comparison: (a) illustration of the aerial data in red aligned with the data captured from ground. (b) Histogram of closest-point distance error resulted from three methods.

tion process has been done with a modern PC: Intel Core i7 @ 3.4G, 16 memory. Rectifying one scan needs around 30 minutes using Matlab and C++, not including the data convention.

#### 4.1. Comparison on reconstruction accuracy

Since there is difficulty for obtaining the ground truth for sensor motion, we evaluate the accuracy of our method by comparing aerial scans with ground scans of a same scene. Then we evaluate our method against two baselines: 1) 3D reconstruction from the motion calculated by visual odometry (or structure from motion) [10] and 2) Banno *et al.*'s method [3] proposed for their flying sensor system. However the hardware system of Banno *et al.* was designed with perspective camera and scanner, in comparison we adopt their algorithm framework to our omni-directional scan.

In Fig. 7 shows the results compared with the two baselines above. Fig. 7 (a) shows the original data with distortion captured by the laser scanner; Fig. 7 (b) shows the result using the motion calculated by the structure from motion (SfM) [10]; Fig. 7 (c) shows the result obtained by the method in [4]; and Fig. 7 (d) shows our result. We can see that, while the SfM method got “high-frequency” instability on the data reconstructed, Banno *et al.*'s method shows “low-frequency” instability on the reconstruction result, which might be caused by the second order smoothness constraint in their algorithm. Our method shows better robustness than the two baseline methods.

Fig. 8 shows the comparison when the ground truth can be captured from a sensor on ground. After the aerial and ground data are aligned together, the distribution of histogram of the closest point distance error can be obtained. Fig. 8 (b) shows that our method get better accuracy than the other two method SfM in [10] and Banno *et al.*'s method [4].

#### 4.2. Angkor Wat 3D reconstruction

Angkor Wat, as one of the largest religious monuments in the world, was built in the early 12th century and located in north of the modern town, Siem Reap, Cambodia.

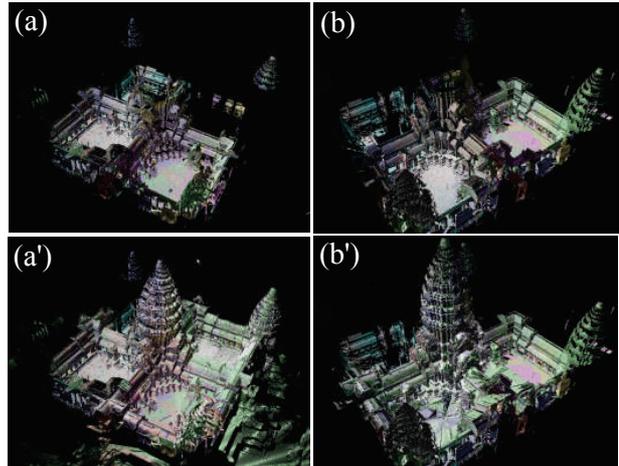


Figure 10. Comparison on with and without aerial scans. (a) and (b) are two different locations captured from the sensor on ground show data missing on the top, and which are merged with aerial scans is shown in (a') and (b') respectively.

Angkor Wat owns several tall towers including the central one in 65m height. Unfortunately Angkor Wat faces the problem of deterioration due to natural or man-made breaks and thus need preservation immediately. Fig. 9 shows the result of entire build: a) entire data including the captured from ground and balloon, b) the aerial data using balloon. However due to the images were captured at different time, the color shows unnatural at some locations.

Fig. 10 shows the aerial scanning plays an important role in Angkor Wat reconstruction that large missing data are with the scans only captured from ground as shown Fig. 10 (a) and (b). Fig. 10 (a') and (b') show that the missing parts can be filled by aerial scanning.

#### 4.3. Manufacturing 3D modeling

As shown in Fig. 11 (a), we mount our system under a crane for the indoor scanning. Since the hook is usually shaking during the scanning, the scans are distorted as shown in Fig. 11 (c) and (d). Our method shows the robustness shown in (c') and (d'). However, little distortion remains due to the motion blur happened on the image (see Fig. 11 (b)), which makes the feature tracking uncorrected in several pixels.

### 5. Discussion

This paper presented a general solution for aerial scanning system. It shows there is space for enhancing the performance, if the frame rate of camera and scanning resolution of scanner can be increased. The sensor system is possible to support various mobile platform. With the emergency of small type of laser scanner and panoramic camera, it is potentially applicable for a UAV system.

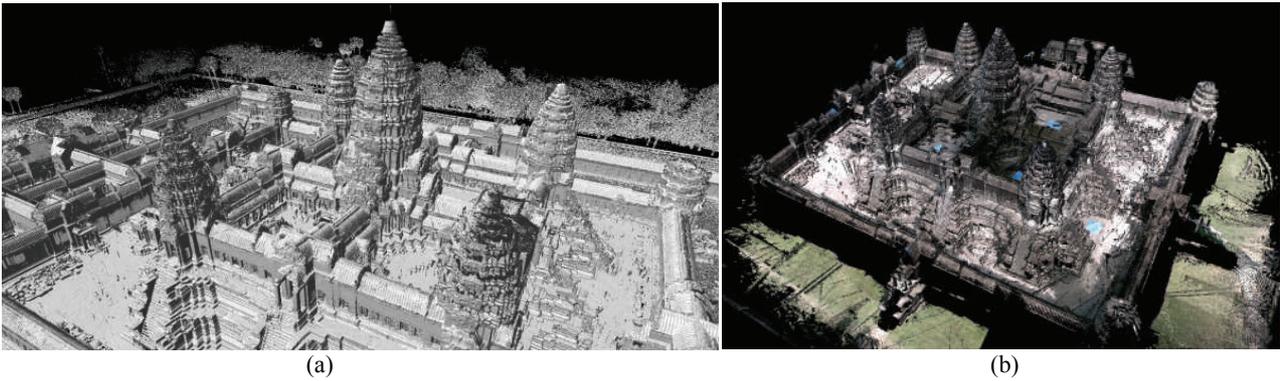


Figure 9. Entire view of 3D Angkor Wat: (a) and (b) with and without ground data included respectively. (b) shows point cloud with color captured in different time.

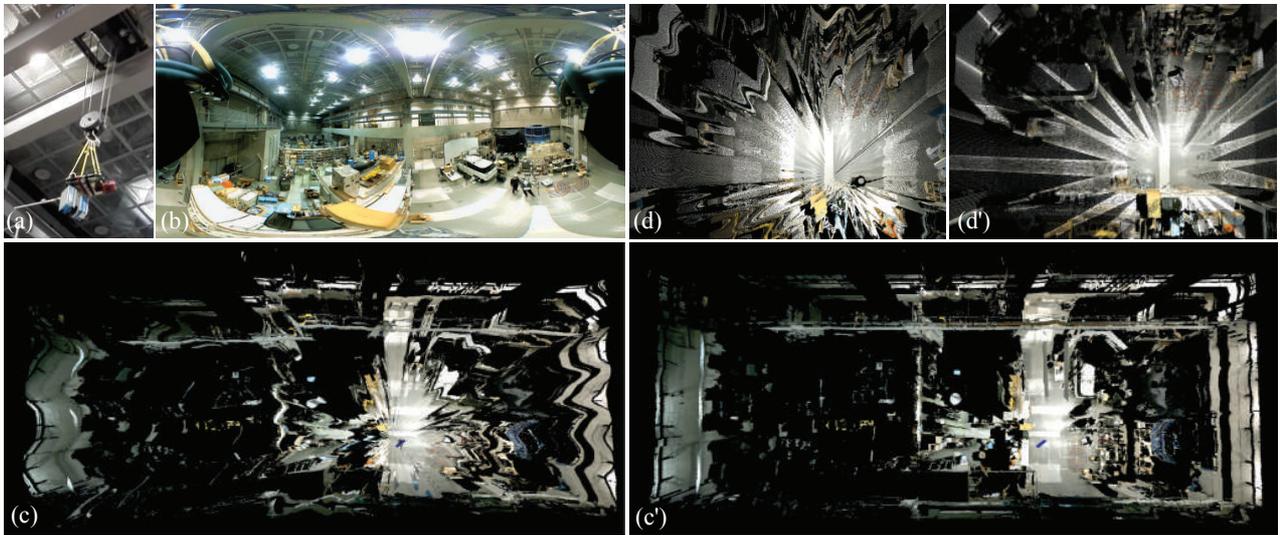


Figure 11. Manufacturing scan: (a) Our system is mounted under a crane for aerial scan, where the hook usually shakes due to the uncontrollable mechanic disturbance. (b) One panoramic frame captured by camera. (c) The original scan with distortion, and whose rectified result is shown in (c'). (d) and (d') the close-ups associated with (c) and (c') respectively.

## Acknowledgment

We thank to Min Lu, ZhiPeng Wang, Masataka Kagesawa, Yasuhide Okamoto, Shintaro Ono in Computer Vision Lab, the University of Tokyo, for the helps on data capture, data processing and sensor setting up. This work is partly supported by JSPS KAKENHI Grant Number 24254005 and 25257303.

## References

- [1] <http://www.ptgrey.com/>. 3, 5
- [2] <http://www.zf-laser.com/>. 1, 3, 5
- [3] A. Banno and K. Ikeuchi. Omnidirectional texturing based on robust 3d registration through euclidean reconstruction from two spherical images. *CVIU*, 114(4):491 – 499, 2010. 4, 7
- [4] A. Banno, T. Masuda, T. Oishi, and K. Ikeuchi. Flying laser range sensor for large-scale site-modeling and its applications in bayon digital archival project. *IJCV*, 78(2-3):207–222, 2008. 1, 2, 6, 7
- [5] P. Besl. Active, optical range imaging sensors. *Machine vision and applications*, 1(2), 1988. 1
- [6] Y. Bok, Y. Jeong, D.-G. Choi, and I. Kweon. Capturing village-level heritages with a hand-held camera-laser fusion sensor. *IJCV*, 94(1):36–53, 2011. 2
- [7] M. Bosse, R. Zlot, and P. Flick. Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28(5), 2012. 2
- [8] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard. Real-time 3d visual slam with a hand-held rgb-d camera. In *RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, 2011. 2

- [9] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *ICRA*, 2014. 2
- [10] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *TPAMI*, 32(8):1362–1376, Aug 2010. 2, 6, 7
- [11] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research (IJRR)*, 31(5), 2012. 2
- [12] A. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *International Symposium on Robotics Research (ISRR)*, 2011. 2
- [13] K. Ikeuchi and D. Miyazaki. *Digitally Archiving Cultural Objects*. Springer-Verlag, 2007. 2
- [14] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. In *ICRA*, 2013. 2
- [15] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007. 2
- [16] B. Klingner, D. Martin, and J. Roseborough. Street view motion-from-structure-from-motion. In *ICCV*, 2013. 2
- [17] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI1981*, pages 674–679, 1981. 4
- [18] M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(2), 2007. 2
- [19] N. Matsuda, O. Cossairt, and M. Gupta. Mc3d: Motion contrast 3d scanning. In *ICCP*, 2015. 2
- [20] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 2
- [21] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011. 2
- [22] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1), 2006. 2
- [23] G. Pandey, J. R. McBride, and R. M. Eustice. Ford campus vision and lidar data set. *IJRR*, 30(13):1543–1552, 2011. 2
- [24] I. Ryoichi, B. Zheng, T. Oishi, and K. Ikeuchi. Rectification of aerial 3d laser scans via line-based registration to ground model (to appear). *IPSJ Tran. on Computer Vision and Applications*, 2015. 2
- [25] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske, and S. Singh. River mapping from a flying robot: state estimation, river detection, and obstacle mapping. *Autonomous Robots*, 32(5), 2012. 2
- [26] S. Shen and N. Michael. State estimation for indoor and outdoor operation with a micro-aerial vehicle. In *International Symposium on Experimental Robotics (ISER)*, 2012. 2
- [27] C. H. Tong, S. Anderson, H. Dong, and T. Barfoot. Pose interpolation for laser-based visual odometry. *Journal of Field Robotics*, 31(5), 2014. 2
- [28] T. Whelan, H. Johannsson, M. Kaess, J. Leonard, and J. McDonald. Robust real-time visual odometry for dense rgb-d mapping. In *ICRA*, 2013. 2
- [29] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *CVPR*, 2012. 4
- [30] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013. 2
- [31] J. Zhang and S. Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *ICRA*, 2014. 2
- [32] B. Zheng, T. Oishi, and K. Ikeuchi. Rail sensor: A mobile lidar system for 3d archiving the bas-reliefs in angkor wat (to appear). *IPSJ Tran. on Computer Vision and Applications*, 2015. 2
- [33] B. Zheng, Y. Sun, T. Jun, and K. Ikeuchi. A feature descriptor by difference of polynomials. *IPSJ Tran. on Computer Vision and Applications*, 5:80–84, 2013. 4, 5