IEICE TRANSACTIONS

on Information and Systems

VOL. E101-D NO. 5
MAY 2018

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

IEICE TRANS. INF. & SYST., VOL.E101–D, NO.5 MAY 2018

PAPER Special Section on Machine Vision and its Applications

Superimposing Thermal-Infrared Data on 3D Structure Reconstructed by RGB Visual Odometry

Masahiro YAMAGUCHI[†], **Trong Phuc TRUONG**[†], ***Nonmembers***, **Shohei MORI**[†], ***Member***, **Vincent NOZICK**[†,††], ***Nonmember***, **Hideo SAITO**[†a)], ***Fellow***, **Shoji YACHIDA**[†††], ***and*** **Hideaki SATO**[†††], ***Nonmembers***

SUMMARY In this paper, we propose a method to generate a three-dimensional (3D) thermal map and RGB + thermal (RGB-T) images of a scene from thermal-infrared and RGB images. The scene images are acquired by moving both a RGB camera and an thermal-infrared camera mounted on a stereo rig. Before capturing the scene with those cameras, we estimate their respective intrinsic parameters and their relative pose. Then, we reconstruct the 3D structures of the scene by using Direct Sparse Odometry (DSO) using the RGB images. In order to superimpose thermal information onto each point generated from DSO, we propose a method for estimating the scale of the point cloud corresponding to the extrinsic parameters between both cameras by matching depth images recovered from the RGB camera and the thermal-infrared camera based on mutual information. We also generate RGB-T images using the 3D structure of the scene and Delaunay triangulation. We do not rely on depth cameras and, therefore, our technique is not limited to scenes within the measurement range of the depth cameras. To demonstrate this technique, we generate 3D thermal maps and RGB-T images for both indoor and outdoor scenes.

key words: *thermal-infrared camera, visual odometry, 3D thermal map, Delaunay division, calibration*

1. Introduction

Thermal-infrared cameras measure infrared rays emitted from any objects and can use this ray information to estimate the object temperature. This temperature, which can not be retrieved from RGB cameras, is a valuable information that leads to many applications in industrial domains as well as in academic research. Indeed, thermal-infrared cameras can be used to detect gas leaks, fires, abnormalities in electronic apparatus, and so on. Most of these issues are hardly detectable with visible light, thus thermal-infrared cameras constitute a precious tool in these cases. However, thermal-infrared camera also presents some drawbacks. In practice, the material used to produce the lenses of thermal-infrared cameras usually makes their field of view usually narrow compared to the lenses of standard RGB cameras. Moreover, textures perceptible in visible spectral domain (i.e. captured by RGB cameras) are likely lost in invisible light. Since the human eye deals with visible light, it is often difficult for humans to understand thermal-infrared images due to this texture alteration.

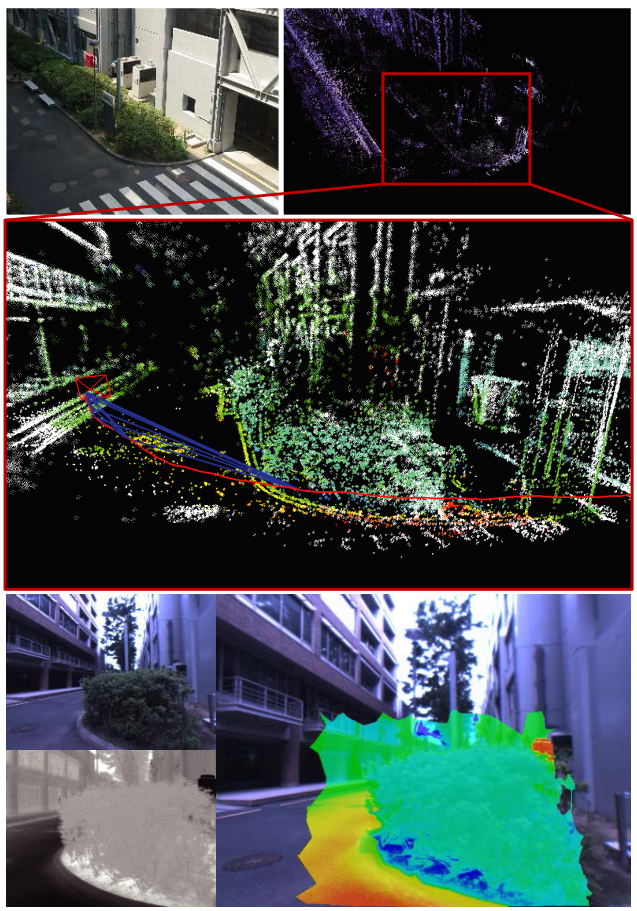

Fig. 1 Example of a generated thermal map and RGB-T image. The correspondence between RGB and thermal-infrared images is obtained using the 3D structure generated from DSO. The top row shows a reference view and a 3D reconstructed result, the middle row shows a 3D thermal model generated by the proposed method, and the bottom shows an RGB-T image and reference frames of an RGB and a thermal-infrared cameras.

These issues would be easily solved if it was possible to perfectly superpose a thermal-infrared image with its corresponding RGB image. This is feasible only if both RGB and thermal-infrared cameras share the same projection center (i.e., the same position). Some very specific cameras, such as the one used in [1], can simultaneously capture RGB and thermal-infrared images. However, in standard situations, RGB and thermal-infrared images are captured using differ-

Manuscript received September 7, 2017.
Manuscript revised November 21, 2017.
Manuscript publicized February 16, 2018.

[†]The authors are with Keio University, Yokohama-shi 223–8522 Japan.
[††]The author is with Universite Paris-Est Marne-la-Vallee, Champs-sur-Marne, France.
[†††]The authors are with NEC, Kawasaki-shi, 211–8666 Japan.
a) E-mail: hs@keio.jp

DOI: 10.1587/transinf.2017MVP0023

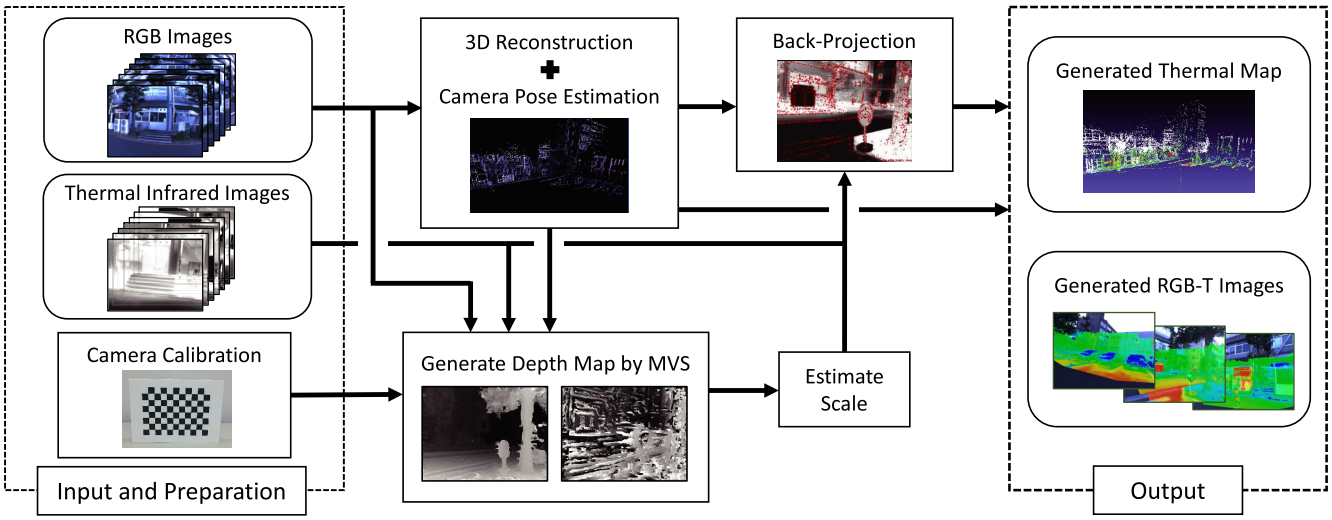

Fig. 2 Overview of the proposed method.

ent cameras. This paper addresses the problem to effectively associate thermal data to its corresponding RGB counterpart for visualization purposes.

2. Related Works

In this paper, we consider the situation where the acquisition device is composed of at least an RGB and a thermal-infrared camera. In a standard situation, the two (or more) cameras can not share the same position. Thus, the matching of these data requires some geometric computation.

A feasible way to proceed consists of adding a 3D sensor to the thermal-infrared and RGB cameras to handle these geometric constraints. Borrmann et al. [2] use a laser scanner and an RGB camera. Due to its heavy weight, the laser scanner is attached to the RGB camera and mounted on a wheeled robot. The robot moves around the scene and performs laser scans repeatedly to generate a 3D RGB map. For each scan of the scene, the robot also acquires the corresponding 3D thermal map. While this method performs well, both the laser scanner and the robot are very expensive. Moreover, the need for a wheeled robot prevents the acquisition process to be conducted in scenes composed of stairs or other low-quality paths.

A cheaper alternative consists of using an RGB-D sensor to generate the 3D model of the scene. Vidas et al. [3], as well as Matsumoto et al. [4] mount a Kinect with a thermal-infrared camera on a hand-held stereo rig. They obtain a 3D structure of the scene using KinectFusion [5]. Although, this method performs well in indoor situations, low cost depth sensors, such as Kinect, usually fail in outdoor environments where the sunlight affects the depth value acquisition [6], [7]. Moreover, the size of the acquired scene is usually limited to small areas with the use of KinectFusion.

Prakash et al. [8] combines two thermal-infrared cameras and use stereo vision to compute a 3D model of the scene. In this method, they use epipolar geometry from the thermal-infrared images to constrain the computation of correspondence. However, the thermal stereo method is relevant only when there are significant temperature gradients on the object surface. Up to now, finding stereo correspondences between 2 thermal-infrared images is still very challenging, as stated in [9]. Indeed, since the thermal-infrared images are low textured, standard matching method often find very few correspondences.

Finally, Ham et al. [10] compute the geometry of the scene using the Structure from Motion (SfM) technique [11] and superimpose the thermal information on the resulting 3D structure. This system uses only an RGB camera and a thermal-infrared camera, and thus handles both indoor scenes and outdoor scenes. However, the calculation cost of the 3D reconstruction from the SfM is significant. Indeed, SfM combines all the images of the considered sequence in order to match them. Thus, the generated 3D point cloud is accurate, however this process leads to a significant computational cost. On the other hand, SLAM uses selective frames (e.g., keyframes) or the last consecutive frames to improve its real-time performance during the video capture. Thus, this approach results in a low computational cost for a 3D reconstruction not as accurate as for SfM [12].

In this work, we propose a method combining an RGB camera with a thermal-infrared camera. As discussed in [10], this setup does not include any depth sensor and thus also supports outdoor scenes. We use a simultaneous localization and mapping (SLAM) method instead of a SfM approach, so the calculation cost is lower than in [10]. Moreover, we propose a robust calibration process for thermal-infrared cameras as well as an accurate camera relative pose for a more efficient thermal to RGB mapping.

3. Overview of the Proposed Method

In this paper, we present a method to generate RGB-T images (RGB plus Thermal) by superimposing thermal information obtained from the thermal-infrared camera onto the RGB images.

More precisely, our system consists of four stages: first, we attach an RGB camera and a thermal-infrared camera on a stereo rig, then calibrate them using a common calibration board that can be simultaneously detected by both cameras, as described in Sect. 4. Second, we acquire a set of images of the scene with the RGB camera and generate at runtime a 3D structure of the environment, as explained in Sect. 5. In the third step, we estimate the projective scale between the 3D structure and the calibration board using depth images generated from both RGB and thermal-infrared camera sequences, as described in Sect. 6. Finally, the thermal map and the RGB-T images are generated using the 3D structure computed from the RGB images and the estimated scale. An overview of the whole process is depicted in Fig. 2.

Our main contribution is this paper consists in both the accurate thermal camera calibration process (Sect. 4) and the overall process to superimpose Thermal-infrared data on 3D reconstruction from the RGB images, including the specific problem of finding the unknown projective scale between RGB and Thermal-infrared cameras (Sect. 6).

4. Camera Calibration

4.1 Calibration Board Issues

Camera calibration is a well known process for pinhole cameras. This process, that usually requires point correspondences between images, has been wildly studied for RGB images but appears to be more challenging for thermal-infrared cameras, due to the lack of textures. Assuming that the calibration process is performed using a calibration board, the thermal-infrared camera hardly detects the calibration pattern. Indeed, a thermal-infrared camera measures the temperature from the infrared rays emitted by objects. The locally emitted infrared light intensity depends more on the object temperature than on the color of the object. Even though it is true that black objects absorb and emit a larger amount of infrared energy than brighter objects, in practice, a thermal-infrared camera cannot detect a checker pattern if the temperature of the calibration board is uniform across the black and white parts.

A general way to make the board detectable by the thermal-infrared camera is to heat the calibration board in order to make the re-emitted light significantly hotter than the intrinsic object temperature. Matsumoto et al. [4] uses a hair dryer to heat the calibration board while Ham et al. and Weinmann et al. [10], [13] prefer to use a set of lamps. Some other methods proposed by Prakash et al. and Saponaro et al. [14], [15] use a flood lamp to heat the calibration board. However, Vidas et al. [16] suggest that the use of a flood lamp is comparatively inaccurate.

In our method, we designed our calibration board from a printed chessboard pattern and placed black plastic tape on the black parts. Then, we used a freezer to cool the calibration board. Heating the board with a lamp is also a good way to proceed, however using a freezer is more convenient in our case.

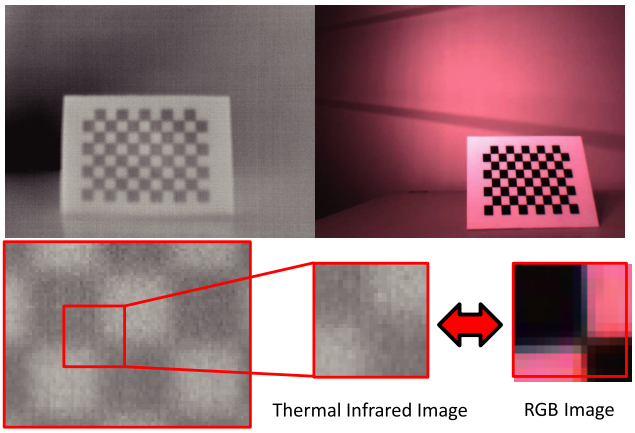

Fig. 3 Blur on the thermal-infrared camera image. The top row shows images of a calibration board captured by both a thermal-infrared camera and an RGB camera. At the bottom row, a zoomed corner on a thermal-infrared image and an RGB image are compared. It is more difficult to detect a thermal-infrared image than an RGB image because of the blur.

The resulting image obtained after heating or cooling still presents some very blurred textures, as shown in Fig. 3. This blur is caused by the fact that the board temperature quickly tends to diffuse from one cell to the next. Thus, after the heating or cooling process, the measured temperature gradually diffuse from a black part to a white one.

This blur is a serious issue for the calibration process, in term of accuracy but also for any automatic chessboard detection. In our method, we first estimate the camera lens distortion parameter using the plumb line method (more details in Sect. 4.3). Next, we perform an initial estimation of the checker pattern corners. These estimated points are significantly refined using a non-linear process involving constrains such as cross-ratio and vanishing point consistency.

4.2 Camera Model

In this paper, we use the pinhole camera model [17] for both the RGB and thermal-infrared cameras. Let $\mathbf{x} = (u, v, 1)^\top$ and $\mathbf{X} = (X, Y, Z, 1)^\top$ be the image coordinates and the world coordinates, respectively. A camera projection matrix is defined as follows:

$$\mathrm{P} = \mathrm{K}\begin{bmatrix}\mathrm{R} \mid \mathbf{t}\end{bmatrix}, \quad \text{with } \mathrm{K} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\mathrm{R} \in SO(3), \quad \mathbf{t} \in \mathbb{R}^3. \tag{1}$$

where K is the camera's intrinsic matrix, defined by the camera principal point (x_0, y_0) and the camera focal length $\left(f_x, f_y\right)$ expressed in pixel units. R is a 3×3 rotation matrix defining the camera orientation and $\mathbf{t}$ is a transformation vector representing the camera position. A 3D world coordinate point $\mathbf{X} = (X, Y, Z, 1)^\top$ projects to a 2D image point $\mathbf{x} = (u, v, 1)^\top$ by $s\mathbf{x} = \mathrm{P}\mathbf{X}$, with $s \in \mathbb{R}^*$.

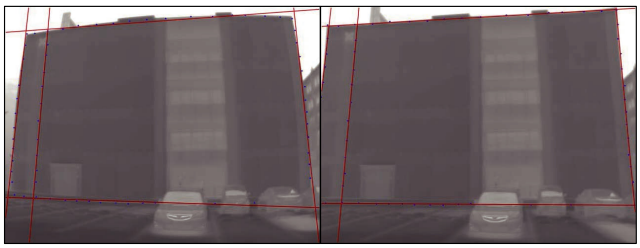

Fig. 4 Estimation of distortion parameter. (Left) A set of points supposed to be aligned in the real scene and their least-square fitted line. (Right) Corresponding undistorted image.

4.3 Distortion Correction

The camera lens distortion correction is performed only once per camera unless the lenses are changed or manipulated (zoom or strong refocus). We use the plumb line method described by Devernay and Faugeras [18] in order to correct the lens distortion by making straight lines of the scene also straight in the image. This method requires a set of points assumed to be aligned in the real world. In our case, we manually extract these points from buildings images that contain good straight lines with strong thermal contrasts with the sky. The method defined in [18] non-linearly computes the fine radial distortion parameters such that the selected points become aligned. The resulting images present a very good correction, as depicted in Fig. 4. This radial distortion correction procedure is strongly required for the next steps of the camera calibration process.

4.4 Corner Refinement

This section presents our chessboard corner points refinement. This process highly improve the accuracy of the camera calibration.

4.4.1 Initial Detection

A very common way to detect a calibration chessboard is to use the automatic tools provided by OpenCV[†]. This tool automatically detects the corner of the chessboard and is wildly used to calibrate RGB cameras. However, in the thermal-infrared camera case, the inherent blur (shown in Fig. 3) makes the point detection accuracy drastically fall. The point detection is clearly inaccurate and messy for thermal images. Since a good accuracy in camera calibration is required for the thermal to RGB mapping, we propose a chessboard detection refinement process.

4.4.2 Corner Refinement based on Line Intersection

This process starts from an undistorted image of the calibration chessboard obtained from Sect. 4.3. Thus, every corner of the checkerboard pattern should be detected such that each point lying on the same row (respectively column) should be aligned. We refine the corner points using line intersection optimization inspired by the method proposed by De la Escalera and Armingol [19]. In our method, we perform a least square fitting of the lines from the "inaccurate" detected corners instead of using the Hough transform on the image.

Naturally, these first estimated lines are not accurate due to the blur on the thermal-infrared images, see Fig. 5 (left). However, they can be considered as a good starting estimation for the first step of our refinement process. These lines are transformed to fit the neighboring maximum gradient of the image intensity, computed only in the orthogonal direction of the lines. In other words, the lines are transformed to lie on the middle of the blur between the black and the white cells of the chessboard. Since the maximum gradient point set may include outliers, the line fitting is performed with RANSAC line fitting. Finally, the corner points are estimated based on the intersection between each rows and columns.

4.4.3 Corner Refinement Based on Geometric Constraints

Firstly, let $\mathcal{P}$ denote the set of the corners initially detected on the calibration board and refined in Sect. 4.4.2. The next step of our refinement consists in a non-linear process. Indeed, we also optimize the corners' position according to both the cross-ratio extracted from each chessboard cell and the vanishing points of the chessboard rows and columns respectively.

Cross ratio

Corner points are placed at constant intervals on the chessboard and thus should satisfy some constraints of cross-ratio. The cross-ratio is a perspective invariant defined in the invariance theory of cross-ratio [17], [20]. If the points A, B, C and D are collinear in space, the four corresponding projected image points A', B', C' and D' are also collinear. The cross-ratio of two sets of any such consecutive four points from any row (or column) of the chessboard are identical and satisfies the following equation:

$$\hat{\rho} = \frac{AC}{CB} / \frac{AD}{DB} = \frac{A'C'}{C'B'} / \frac{A'D'}{D'B'} \tag{2}$$

In practice, it is possible to measure how the detected points satisfy this cross-ratio constraint by a cost function $C_{\text{cross}}(\mathcal{P})$ inspired by a method proposed by Ricolfe-Viala et al. [21]. This cross-ratio cost is equally divided into two costs dedicated respectively to rows and columns constrains:

$$C_{\text{cross}}(\mathcal{P}) = C^{\text{rows}}_{\text{cross}}(\mathcal{P}) + C^{\text{cols}}_{\text{cross}}(\mathcal{P})$$

where $C^{\text{rows}}_{\text{cross}}(\mathcal{P})$ and $C^{\text{cols}}_{\text{cross}}(\mathcal{P})$ are computed in the same way. For clarity purpose, we will detail only the computation of $C^{\text{rows}}_{\text{cross}}(\mathcal{P})$. According to the cross-ratio constraint, each four successive points of a row should lead to the theoretical cross-ratio $\hat{\rho}$ defined by any consecutive points on the

[†]Open Source Computer Vision Library (OpenCV) http://opencv.org/

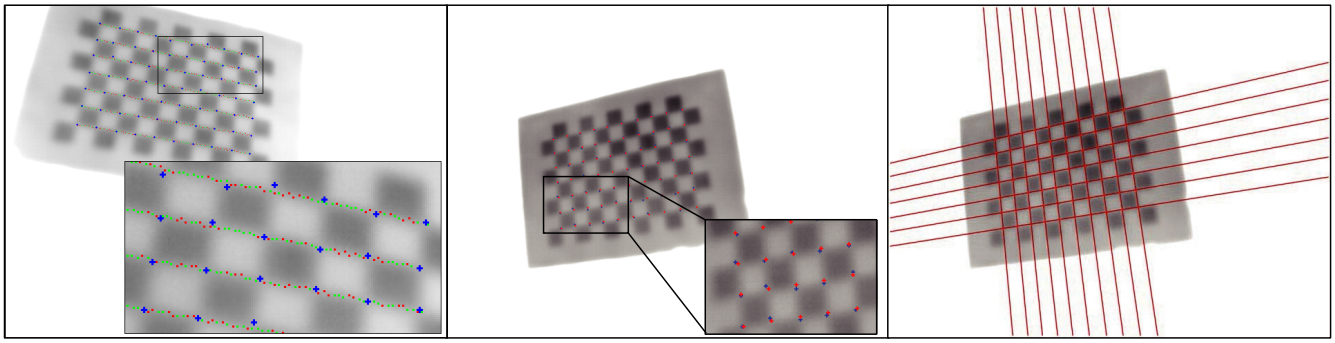

Fig. 5 Initial detection and refining the points. Refined corner points should lie along straight lines in the left image. The center image shows the refinement of the points. Blue points are points detected by OpenCV and Red Points are the refined points, and the right image shows lines drawn connecting the corners.

chessboard. Let Q_{rows} be the set of all combination of four consecutive points $q = A, B, C, D$ of $\mathcal{P}$, extracted only from the rows of the calibration board. The cross-ratio $\rho(q_i)$ of each $q_i \in Q_{\text{rows}}$ is defined as:

$$\rho(q_i) = \frac{A_iC_i}{C_iB_i} / \frac{A_iD_i}{D_iB_i}$$

Then, the cross-ratio cost function dedicated to the rows points can be defined as:

$$C_{\text{cross}}^{\text{rows}}(\mathcal{P}) = \frac{1}{\text{card}(Q_{\text{rows}})} \sum_{q_i \in Q_{\text{rows}}} \left(1 - \frac{\rho(q_i)}{\hat{\rho}}\right) \tag{3}$$

The same procedure holds for Q_{cols} with:

$$C_{\text{cross}}^{\text{cols}}(\mathcal{P}) = \frac{1}{\text{card}(Q_{\text{cols}})} \sum_{q_i \in Q_{\text{cols}}} \left(1 - \frac{\rho(q_i)}{\hat{\rho}}\right) \tag{4}$$

Vanishing points

As for the cross-ratio cost function, the vanishing point cost function is divided into two costs since a calibration board contains two main vanishing points:

$$C_{\text{vanish}}(\mathcal{P}) = C_{\text{vanish}}^{\text{rows}}(\mathcal{P}) + C_{\text{vanish}}^{\text{cols}}(\mathcal{P})$$

Again, let's just consider the chessboard row vanishing point. The cost function procedure starts by an estimation of the vanishing point (x_v, y_v) computed as the least square intersection of all the line extending each rows. Then, the vanishing point cost $C_{\text{vanish}}^{\text{rows}}(\mathcal{P})$ is defined as the average distance from this vanishing point to these lines. Since each of those lines should pass throw the vanishing point, the average distance from the lines to the vanishing point should tend to zero if the corner points are accurately positioned. More formally, let $\mathbf{r}_i = (a_i, b_i, c_i)^\top$ be the Hessian form of the i^{st} row, and $\mathcal{V}$ the set of rows of $\mathcal{P}$, then the cost to minimize can be expressed as:

$$C_{\text{vanish}}(\mathcal{P}) = \frac{1}{\text{card}(\mathcal{V})} \sum_{\mathbf{r}_i \in \mathcal{V}} \frac{|a_i x_v + b_i y_v + c_i|}{\sqrt{a_i^2 + b_i^2}} \tag{5}$$

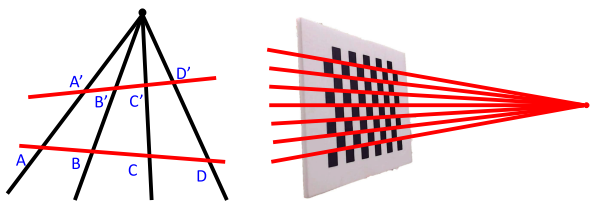

Fig. 6 Constraints of cost function. The left figure describes the cross-ratio constraint, which is satisfied by Eq. (2). The right figure shows vanishing point of the calibration board.

The denominator transforms the line equations in their normalized Hessian form. Moreover, note that we don't use this formulation when the calibration board is orthogonal to the principal ray of the camera since in this situation, the vanishing points would lie at infinity which is not compatible with our vanishing point scoring method.

Figure 6 depicts both the cross-ratio and the vanishing points constraints.

Data fidelity term

Using only the two first terms would lead to a set of points perfectly aligned and with the perfect distances from one to the next, but not necessary consistent with the calibration board since the refined points could freely move anywhere. Thus, to maintain a certain consistency between the points and the calibration board, we add a data fidelity term $C_{\text{dist}}(\mathcal{P})$. This cost simply defines the ℓ_2 distance $d(\mathrm{x}_i, \hat{\mathrm{x}}_i)$ from any original point $\hat{\mathrm{x}}_i$ computed in Sect. 4.4.2 to its corresponding refined points $\mathrm{x}_i \in \mathcal{P}$:

$$C_{\text{dist}}(\mathcal{P}) = \frac{1}{\text{card}(\mathcal{P})} \sum_{\mathrm{x} \in \mathcal{P}} d(\mathrm{x}_i, \hat{\mathrm{x}}_i) \tag{6}$$

global cost function

The final cost function to optimize the chessboard corner position estimation can be described as the weighted sum of the three first cost functions:

$$C(\mathcal{P}) = \arg\min_{\mathcal{P}} (C_{\text{vanish}}(\mathcal{P}) + \alpha C_{\text{cross}}(\mathcal{P}) + \beta C_{\text{dist}}(\mathcal{P})) \tag{7}$$

where α, β are given parameters. Once the corners are refined by minimizing the weighted sum of Eq. (7), these refined points are used to calibrate the camera parameters using Zhang's method [22].

4.4.4 Deciding the Weights α and β

The weights α and β of Eq. (7) are decided experimentally. We generate a virtual chessboard consisting in a set of points regularly arranged and transform them using a random but realistic perspective transformation (homography). Then, we added a Gaussian noise on these points. For a large set of values for α and β, we generate many noised virtual chessboards and ran our algorithm. We selected the couple (α, β) that reconstruct the best the original virtual chessboard in average over all the virtual noised chessboards.

5. Reconstructing the 3D Structures

Assuming the cameras to be calibrated, the next step consists in the computation of a 3D structure of the scene and in the camera pose estimation of each frame in the 3D structure coordinate system. This 3D structure will be the support to projected thermal data for the final visualization. Note that we only need the camera pose of the RGB camera since we already know the relative pose between the RGB camera and the thermal-infrared camera, from the camera stereo rig calibration.

Many different techniques exist to compute a 3D reconstruction. Table 1 shows a comparison between state-of-the-art methods. *Density* refers to how dense the 3D structure is, *cost* corresponds to the calculation cost, and *scale* specifies whether the method can handle large scenes.

Among these methods, techniques using depth sensors based on time of flight or structured light in the infrared domain (e.g., Microsoft Kinect sensor) are immediately discarded since such low-cost depth sensors do not perform well in outdoor scenes due to interference with sunlight. Moreover, low-cost depth sensors cannot handle large scenes due to their intrinsic limited range of action. These constrains exclude the use of KinectFusion. Better sensors are expensive and thus do not match our purpose of keeping the whole process affordable. Moreover, our method should be easy to set up and to implement, so we also discard Lidar 3D scanners. The remaining solutions suggest to use an RGB camera to perform the 3D reconstruction.

There are two main approaches to generate 3D structures from RGB images: direct methods, like Engel et al. [23], referring directly to image intensity, and feature-based methods, such as Klein et al. [24], that generates and matches feature points to construct the 3D map. Since our goal is to generate a large thermal map with real-time performance, possibly on outdoor scenes, the best candidates are direct sparse odometry (DSO) [25], large-scale direct monocular simultaneous localization and mapping (LSD-SLAM) [26] and ORB-SLAM [27]. In these three 3D reconstruction systems, DSO and ORB-SLAM are more accurate than LSD-SLAM [25], [27]. Moreover, the 3D structures generated from DSO have a better density than those generated from ORB-SLAM. Thus, we selected DSO for our 3D reconstruction.

Table 1 Comparison of 3D reconstruction methods.

	method	density	cost	scale
DSO [25]	direct	sparse	small	large
LSD-SLAM [26]	direct	semi-dense	small	large
PTAM [24]	feature	sparse	small	small
ORB-SLAM [27]	feature	sparse	small	large
SfM [11]	feature	sparse	large	large
KinectFusion [5]	RGB-D	dense	small	small
Lidar	3D scanner	dense	-	large

6. Relative Scale between the Point Clouds

Any 3D models generated by a monocular RGB camera is build and defined up to an unknown scale. Since we want to back-project the thermal-infrared images on the 3D model obtained by DSO, this unknown scale should be estimated. In practice, this scale is also the scale that relates a depth map generated from the RGB image to the corresponding depth map generated from the thermal-infrared images. This Section defines how to estimate the scale of the DSO point cloud in the stereo rig coordinate system, and how to back-project the thermal-infrared images on this point cloud.

DSO divides frames into two types: key frames for which a depth map is computed, and the other frames just used to refine the depth map generated by the key frames. We use these depth maps to superimpose thermal information on the 3D structure. For a given key frame, the depth value of each pixel $\mathbf{x}$ refers to a 3D point that can be projected on the infrared thermal image on $\mathbf{x}'$. This projected point $\mathbf{x}'$ corresponds to a thermal value that can be associated to the pixel $\mathbf{x}$ in the RGB image.

More precisely, let $\mathbf{I}_{rgb}$ express an RGB image and $\mathbf{I}_{ir}$ a thermal-infrared image. The depth maps of DSO are sparse so some points have depth value and some others do not. Image coordinates with depth value d are denoted by $\mathbf{x} = (x, y, 1)^\top$. A general back-projection function [17] to converts the RGB image coordinate to a 3D coordinate according to the depth d in the DSO depth map is given by:

$$\mathbf{X}(d) = \begin{pmatrix} (\mathrm{KR})^{-1}(d\mathbf{x} - \mathrm{Kt}) \\ 1 \end{pmatrix} \tag{8}$$

If we consider that P_{rgb} defines the coordinate system of the stereo rig, then $\mathrm{P}_{rgb} = \mathrm{K}_{rgb}[\mathrm{Id}|\mathbf{0}]$ and $\mathrm{P}_{ir} = \mathrm{K}_{ir}[\mathrm{R}_{ir}|\mathbf{t}_{ir}]$, where $[\mathrm{R}_{ir}|\mathbf{t}_{ir}]$ is the relative pose between the two cameras, usually with the arbitrary constraint $\|\mathbf{t}_{ir}\|^2 = 1$. Since we are back projecting from $\mathrm{P}_{rgb} = \mathrm{K}_{rgb}[\mathrm{Id}|\mathbf{0}]$, the back-projection function simplifies to:

$$\mathrm{P}'_{rgb}(\mathbf{x}, d) = \begin{pmatrix} d\mathrm{K}_{rgb}^{-1}\mathbf{x} \\ 1 \end{pmatrix} \tag{9}$$

where K_{rgb} is the intrinsic matrix of RGB camera. Then, the following equation is used to project the DSO depth map

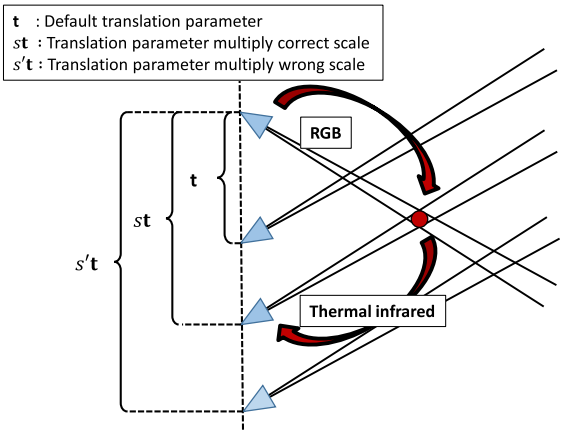

Fig. 7 Variation of back-projection by different scale. The red circle represent a 3D object of the scene, with a certain arbitrary scale. The translation $\mathbf{t}$ between the two cameras is initially subject to $\|\mathbf{t}_{ir}\|^2 = 1$ can be scaled with the fine scale factor s to fit the 3D reconstruction from the first camera.

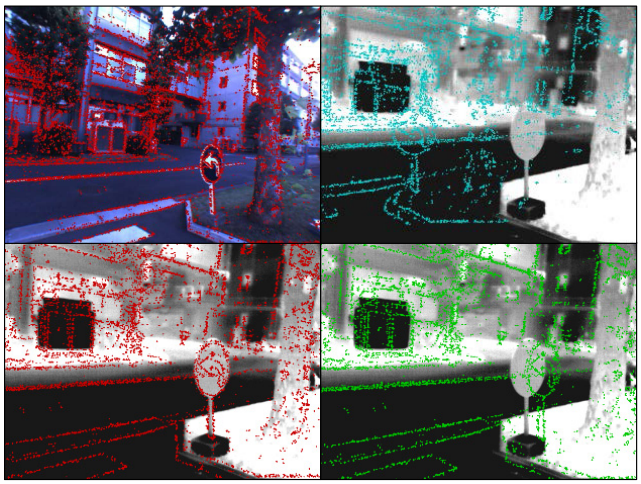

Fig. 8 Result of back-projection to thermal-infrared images using different scales. The top left is a reference RGB camera image. The other images are generated by back-projection of the DSO point cloud using different scales. The bottom left image shows the accurate scale.

onto the thermal-infrared image to read its thermal intensity value:

$$\mathbf{x}' = \mathrm{M}_{rgb\_ir}\mathrm{P}'_{rgb}(\mathbf{x}, d) \tag{10}$$

such that the pixels of the RGB image can be associated to the corresponding pixel in the thermal-infrared image:

$$\mathbf{I}_{rgb}(\mathbf{x}) \leftrightarrow \mathbf{I}_{ir}(\mathbf{x}') \tag{11}$$

However, since the relative pose of P_{ir} in P_{rgb} coordinate system is defined up to scale, the transformation matrix M_{rgb_ir} from P_{rgb} in P_{ir} is also defined up to scale s:

$$\mathrm{M}_{rgb\_ir} = \mathrm{K}_{\mathrm{ir}}\left[\mathrm{R}_{\mathrm{ir}} \mid s\mathbf{t}_{ir}\right] \tag{12}$$

Here, $\mathbf{t}_{ir}$ is a translation vector corresponding to the position of the thermal infrared camera in the stereo rig coordinate system. Since $\mathbf{t}_{ir}$ is defined up to an arbitrary scale (here $\|\mathbf{t}_{ir}\|^2 = 1$), it can be also defined up to any other scale factor s. Moreover, the 3D structure and thus the depth values provided by any monocular RGB SLAM are also defined up to scale, thus we cannot project the DSO point cloud to thermal-infrared camera images unless we can determine a common scale factor between the 3D structure system coordinate and the stereo rig system coordinate. For this process, we can fix the 3D structure scale factor to its default value and search for the fine the stereo rig scale factor s of that fits the best to the 3D structure, as depicted in Fig. 7.

Figure 8 shows projection result from DSO point cloud to thermal-infrared images using difference scale s in Eq. (12), varying the parameter s. In our method, we estimate the scale of the point cloud generated by a monocular RGB camera and project it to thermal-infrared images accurately.

Thermal values and RGB values have different modalities, and thus cannot be compared directly. We use instead the RGB and Thermal depth value to compare both images, since depth is a common modality. In our method, we generate a depth map from both the RGB and thermal-infrared images using a multi view stereo (MVS) algorithm [28], [29]. A depth map is generated using several images near focus frames and the trajectory of the camera obtained from DSO. We generate a patch for each pixel, then get the score by comparing the patch and other frame patches using zero-mean normalized cross correlation (ZNCC).

$$d_i = \arg\min_{d \in D} C(\boldsymbol{i}, d) \tag{13}$$

Equation (14) describes the score of the ZNCC where C expresses the ZNCC score of the patch centered in pixel i with a certain depth value d.

$$C(\mathbf{i}, d) = -\sum_{\mathbf{j} \in \mathbf{I}_p} \frac{(\mathbf{I}_p(\mathbf{j}) - \overline{\mathbf{I}_p})\,(\mathbf{I}'_p(\mathbf{j}) - \overline{\mathbf{I}'_p})}{\sigma(\mathbf{I}_p)\,\sigma(\mathbf{I}'_p)} \tag{14}$$

This score is calculated comparing a square window $\mathbf{I}_p$ in depth image $\mathbf{I}$ and a square window $\mathbf{I}'_p$ in depth image $\mathbf{I}'$. $\mathbf{I}_p(\mathbf{j})$ and $\mathbf{I}'_p(\mathbf{j})$ describe the intensity value on pixel $\mathbf{j}$. Finally, $\overline{\mathbf{I}_p}$ and $\sigma(\mathbf{I}_p)$ refer to the means and the standard deviation of the window $\mathbf{I}_p$. Figure 9 shows some results of depth maps generated from both RGB and thermal-infrared images.

The depth map generated from the RGB image sequence is successively translated to thermal-infrared camera coordinates by Eq. (12) by iterating on $\mathbf{t}$. The translated RGB depth map and the thermal-infrared depth map are compared using the mutual information defined in [30], [31]. This mutual information is often used to compare images of different modalities. In our case, this modality variety can be expressed as the difference of accuracy between the RGB and thermal camera depth map computation. Indeed, thermal-infrared depth maps are always significantly worse than the depth maps build from RGB images. The translation $\mathbf{t}$ leading to the best depth map overlap defines

Fig. 9 Generating a depth map using a multi-view stereo (MVS) algorithm. The top left shows a reference RGB camera image and the bottom left shows a reference thermal-infrared camera image. Right images show the result of MVS using each image.

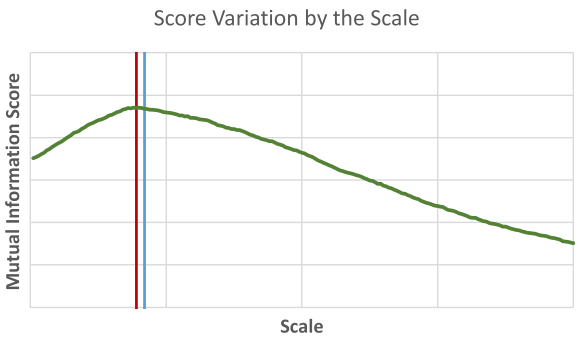

Fig. 10 Variation of the MI score with scale. The green curve is the average score for each scale. We use some samples to estimate the scale and get the average of the score. The red vertical line is the place where the score is maximum, and the blue vertical line is the ground truth obtained by the experiment in Sect. 8.4.

the scale s. Equation (15) describes the mutual information score.

$$MI(\mathbf{I}_{rgb}, \mathbf{I}_{ir}) = \eta\Big(H(\mathbf{I}_{rgb}) + H(\mathbf{I}_{ir}) - H(\mathbf{I}_{rgb}, \mathbf{I}_{ir})\Big) \quad (15)$$

In Eq. (15), $\mathbf{I}_{rgb}$ is the patch converted by Eq. (11) using given scale s, and η describes the ratio of appear the RGB depth map on the thermal-infrared camera coordinate after converting. $H(\mathbf{I})$ describes the appearance ratio of depth i on the depth image $\mathbf{I}$ and $H(\mathbf{I}, \mathbf{I}')$ describes the two dimensional appearance ratio of depth i on the depth image $\mathbf{I}$ and depth j on the depth image $\mathbf{I}'$.

$$H(\mathbf{I}) = -\sum_{i=0} p_{\mathbf{I}}(i) \log\left(p_{\mathbf{I}}(i)\right) \quad (16)$$

$$H(\mathbf{I}, \mathbf{I}') = -\sum_{i=0}\sum_{j=0} p_{\mathbf{II}'}(i, j) \log\left(p_{\mathbf{II}'}(i, j)\right) \quad (17)$$

Figure 10 shows the variation of the mutual information scores with different scales s. The green curve is the average score for each scale. The optimal scale can be estimated uniquely as a global maximum from this the graph. This scale is a valuable information to correctly superimpose the temperature data on the point cloud.

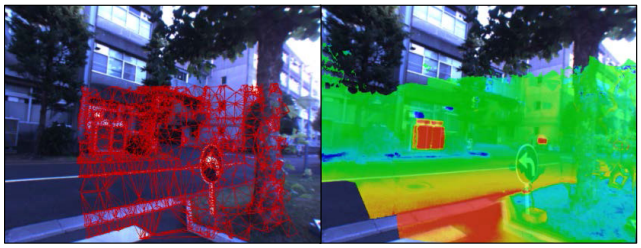

Fig. 11 RGB-T image. The left shows the result of overlaying a 2D mesh on the image, and the right shows the resulting generated an RGB-T image.

7. Generating RGB-T Images from the Point Cloud

At that stage, we have a point cloud computed by the RGB camera and the correct scale between this point cloud and the stereo rig referential. Thus, it is basically possible to project associate to each point a RGB value as well as a thermal value. In practice, the point cloud generated from DSO is sparse, so we sometimes have to interpolate the thermal information where the point cloud is not dense enough. Thus, the 3D point cloud is first back-projected on the targeted RGB image using the camera pose provided by DSO. Second, we generate triangle meshes from the projected point cloud using Delaunay triangle division[†] [32]. Then each triangle vertex is back projected on the thermal image using Eq. (10), in order to compute the right texture coordinates of each triangle in the thermal image. The final rendering consists in the superimposition of the dense thermal textured mesh back projected onto the corresponding RGB image. Figure 11 shows an example of an RGB-T image build with this process.

8. Experiments

8.1 Experimental Setups

In our experiments, we fixed a Flea3 in Point Grey monocular RGB camera and a PI640 Optris thermal-infrared camera into a hand-held stereo rig, as depicted in Fig. 12. This device can be easily manipulated to capture two videos of the scene.

8.2 Calibration Refinement

The calibration refinement process described in Sect. 4 is evaluated as follows. We first calibrate the cameras with the non-optimized chessboard corners detected by OpenCV, then compute the reprojection error on each camera, i.e. the average distance from each detected corner in the image and

[†]Fade2D Delaunay Triangulation
http://www.geom.at/products/fade2d/

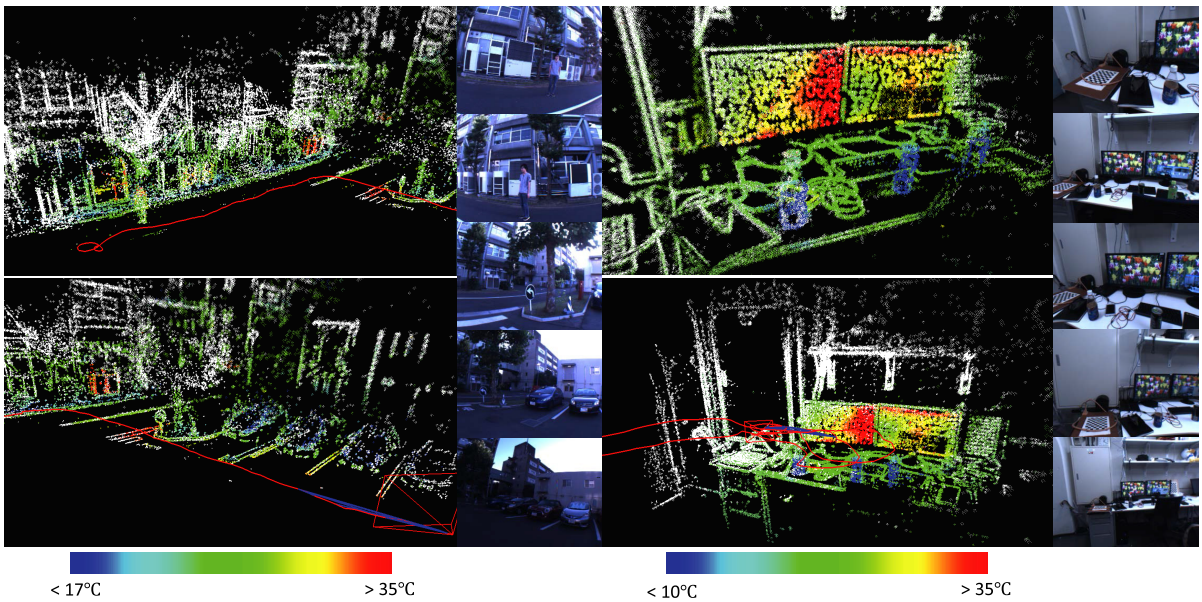

Fig. 13 Thermal maps. Left: outdoor sequence, made out of 1330 pairs of RGB and thermal-infrared images. Right: indoor sequence with 1264 pairs of RGB and thermal-infrared images.

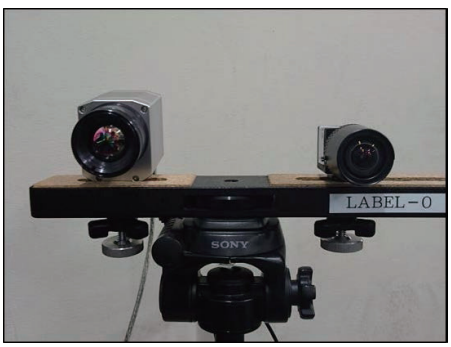

Fig. 12 Camera rig. Left: thermal-infrared camera. Right: RGB camera.

Table 2 Calibration RMS in pixel.

	Initial	Refined
Thermal	1.4980	0.6311
Stereo	1.2345	0.7594
RGB	0.1723	-

the projected corned with the projection matrix. Second, we repeated this process with the refined chessboard corners. These reprojection errors are shown in Table 2. Even through the reprojection error of the thermal camera is still not as accurate as for the RGB camera, we can note a noticeable improvement, roughly by a factor of 2, in the reprojection error of the thermal camera calibration.

8.3 Thermal Maps and RGB-T Images

As detailed in Sect. 7, the final RGB-T images are computed by superimposing thermal information on the RGB-images. The thermal data is rendered with a colormap with red, green, blue gradation, where red indicates high temperature and blue indicates lower temperature. For our experiments, we generated thermal maps from both an outdoor sequence and an indoor sequence. Figure 13 shows the resulting generated thermal maps.

In the outdoor sequence, the building equipments and the asphalt exposed to the sun light get a high temperature, and thus appear in red. The cars parked in the shadows of buildings have a lower temperature and are rendered in blue. In the indoor sequence, we can notice the high temperature of the displays in red, and low temperature of the beverage rendered in blue.

We also generated RGB-T images from various scenes. Figure 14 shows the resulting RGB-T images and reference RGB images.

8.4 Evaluation the Estimated Scale

In this section, we describe the evaluation of our scale estimation method and its accuracy. Figure 15 shows the environment we used to evaluate the scale. In this experiment, we reconstruct a calibration board of known length. Using the scale estimated by our method, we compared the estimated size of the calibration board with the actual size.

First, we reconstruct a thermal map as usual, as shown on the left of Fig. 15, then we reconstruct the calibration board in the right part of the figure. Second, the point cloud is reprojected to the image capturing the calibration board and we pick up the points on the corners of each cell of the calibration board. The points are very noisy, so we remove the outliers that are not on the plane of the calibration board by RANSAC and compute a estimated length $\hat{L}$ of each cell side:

$$\hat{L} = \frac{\|\mathbf{t}\|}{s} \tag{18}$$

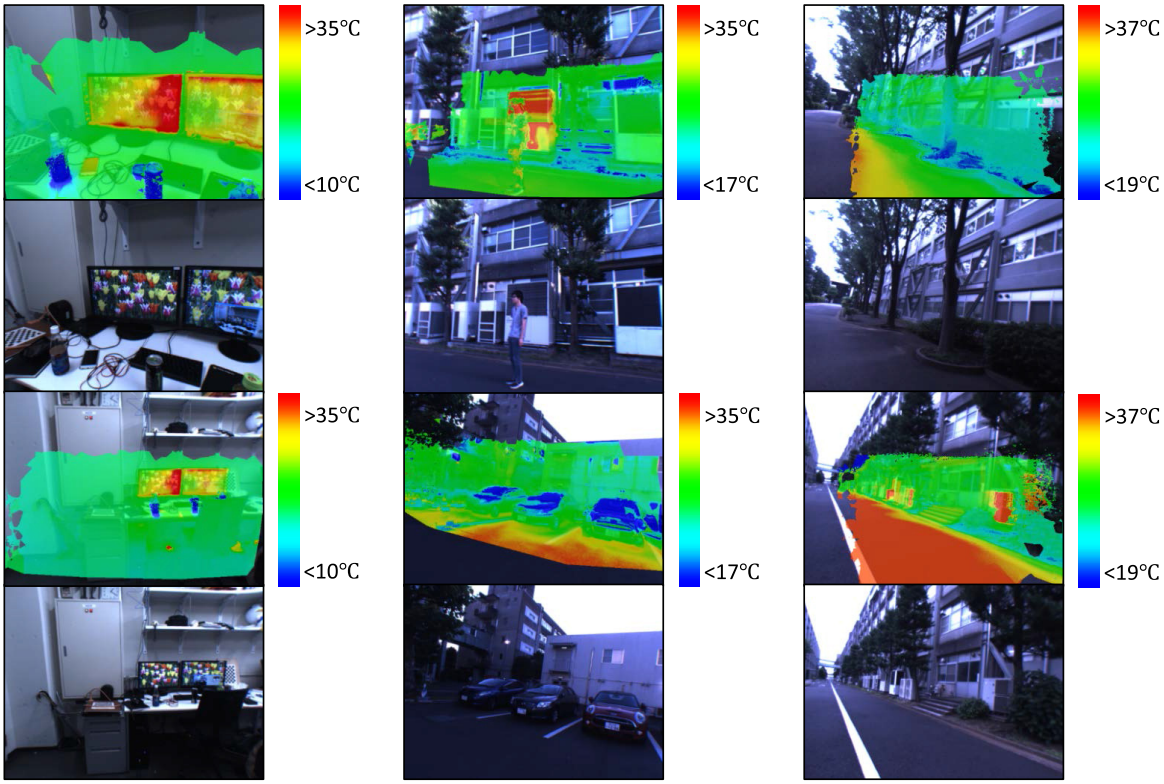

Fig. 14 RGB-T images from various scenes. This images are generated from 3 different sequences.

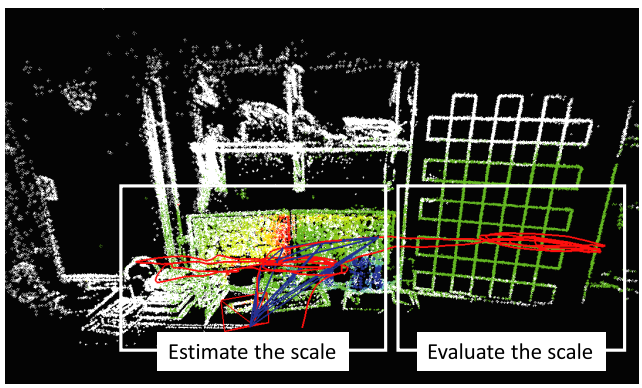

Fig. 15 Evaluation of the estimated scale. The reconstructed environment includes a calibration board. First, estimate the scale of the point cloud. Second, compare the estimated scale and the actual scale by using the side of the calibration board.

We then estimate the average length of the cells, which is the length of the DSO point cloud scale. Using Eq. (18), we converted the scale to the actual size. We calculated the mean relative error against the actual size L using Eq. (19):

$$\epsilon = \frac{\hat{L} - L}{L} \tag{19}$$

Considering the randomness of RANSAC, we repeated the process 100 times. Table 3 describes the mean and the standard deviation of the relative error. We evaluated the indoor and outdoor scenes in two sequences.

Table 3 Mean and standard deviation of the relative error between the estimated scale and the ground truth.

	Average	Standard deviation
Scene1	−0.1026	0.0048
Scene2	0.0719	0.0386

9. Conclusion

In this paper, we visualize temperature more effectively by generating a thermal map and RGB-T images using 3D structures obtained from DSO. We further demonstrate the results of our method using indoor and outdoor scenes.

In the proposed method, first, we calibrate an RGB and a thermal-infrared camera using a calibration board that can be detected by the thermal-infrared camera. Then, temperature information is superimposed onto the generated 3D structure using the extrinsic parameter between both cameras. At that time, we have to obtain the scale of the 3D point cloud. Thus, we estimate the scale using depth maps generated from MVS. Moreover, we generate RGB-T images that can superimpose temperature on RGB images where sufficient 3D points are not obtained from DSO using Delaunay triangulation in order to create triangle mesh.

We can generate thermal maps and RGB-T images by using the proposed method. We plan to expand our system to generate automatic alerts for abnormalities in electricity in plants based on variations in the temperature.

References

[1] K. Yasuda, T. Naemura, and H. Harashima, "Thermo-key: Human region segmentation from video," IEEE Computer Graphics and Applications, vol.24, no.1, pp.26–30, 2004.
[2] D. Borrmann, J. Elseberg, and A. Nüchter, "Thermal 3D mapping of building façades," Intelligent Autonomous Systems 12, vol.193, pp.173–182, 2013.
[3] S. Vidas, P. Moghadam, and M. Bosse, "3D thermal mapping of building interiors using an RGB-D and thermal camera," Robotics and Automation (ICRA), 2013 IEEE International Conference on, pp.2311–2318, IEEE, 2013.
[4] K. Matsumoto, W. Nakagawa, H. Saito, M. Sugimoto, T. Shibata, and S. Yachida, "AR visualization of thermal 3D model by hand-held cameras," VISAPP (3), pp.480–487, 2015.
[5] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on, pp.127–136, IEEE, 2011.
[6] R. Horaud, M. Hansard, G. Evangelidis, and C. Ménier, "An overview of depth cameras and range scanners based on time-of-flight technologies," Machine Vision and Applications, vol.27, no.7, pp.1005–1020, 2016.
[7] G. Alenyà, S. Foix, and C. Torras, "Using ToF and RGBD cameras for 3D robot perception and manipulation in human environments," Intelligent Service Robotics, vol.7, no.4, pp.211–220, 2014.
[8] S. Prakash, P.Y. Lee, and T. Caelli, "3D mapping of surface temperature using thermal stereo," Control, Automation, Robotics and Vision, 2006, ICARCV'06, 9th International Conference on, pp.1–4, IEEE, 2006.
[9] W. Treible, P. Saponaro, S. Sorensen, A. Kolagunda, M. O'Neal, B. Phelan, K. Sherbondy, and C. Kambhamettu, "CATS: A Color and Thermal Stereo Benchmark," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[10] Y. Ham and M. Golparvar-Fard, "An automated vision-based method for rapid 3D energy performance modeling of existing buildings using thermal and digital imagery," Advanced Engineering Informatics, vol.27, no.3, pp.395–409, 2013.
[11] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S.M. Seitz, and R. Szeliski, "Building rome in a day," Communications of the ACM, vol.54, no.10, pp.105–112, 2011.
[12] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.6, pp.1052–1067, 2007.
[13] M. Weinmann, J. Leitloff, L. Hoegner, B. Jutzi, U. Stilla, and S. Hinz, "Thermal 3D mapping for object detection in dynamic scenes," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol.II-1, pp.53–60, 2014.
[14] S. Prakash, P.Y. Lee, T. Caelli, and T. Raupach, "Robust thermal camera calibration and 3D mapping of object surface temperatures," SPIE Proceedings: ThermoSense XXVIII, vol.6205, p.62050J, 2006.
[15] P. Saponaro, S. Sorensen, S. Rhein, and C. Kambhamettu, "Improving calibration of thermal stereo cameras using heated calibration board," Image Processing (ICIP), 2015 IEEE International Conference on, pp.4718–4722, IEEE, 2015.
[16] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan, and T. Wark, "A mask-based approach for the geometric calibration of thermal-infrared cameras," IEEE Trans. Instrum. Meas., vol.61, no.6, pp.1625–1635, 2012.
[17] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, Cambridge university press, 2003.
[18] F. Devernay and O. Faugeras, "Straight lines have to be straight," Machine Vision and Applications, vol.13, no.1, pp.14–24, 2001.
[19] A. De la Escalera and J.M. Armingol, "Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration," Sensors, vol.10, no.3, pp.2027–2044, 2010.
[20] G. Zhang, J. He, and X. Yang, "Calibrating camera radial distortion with cross-ratio invariability," Optics & Laser Technology, vol.35, no.6, pp.457–461, 2003.
[21] C. Ricolfe-Viala and A.-J. Sánchez-Salmerón, "Robust metric calibration of non-linear camera lens distortion," Pattern Recognition, vol.43, no.4, pp.1688–1699, 2010.
[22] Z. Zhang, "A flexible new technique for camera calibration," IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no.11, pp.1330–1334, 2000.
[23] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," Proceedings of the IEEE International Conference on Computer Vision, pp.1449–1456, 2013.
[24] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," Mixed and Augmented Reality, 2007, ISMAR 2007, 6th IEEE and ACM International Symposium on, pp.225–234, IEEE, 2007.
[25] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," IEEE Trans. Pattern Anal. Mach. Intell., vol.40, no.3, pp.611–625, 2017.
[26] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," European Conference on Computer Vision, vol.8690, pp.834–849, Springer, 2014.
[27] R. Mur-Artal, J.M.M. Montiel, and J.D. Tardos, "ORB-SLAM: a versatile and accurate monocular slam system," IEEE Trans. Robot., vol.31, no.5, pp.1147–1163, 2015.
[28] M. Okutomi and T. Kanade, "A multiple-baseline stereo," IEEE Trans. Pattern Anal. Mach. Intell., vol.15, no.4, pp.353–363, 1993.
[29] V. Pradeep, C. Rhemann, S. Izadi, C. Zach, M. Bleyer, and S. Bathiche, "MonoFusion: Real-time 3D reconstruction of small scenes with a single web camera," Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on, pp.83–88, IEEE, 2013.
[30] C. Studholme, D.L.G. Hill, and D.J. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment," Pattern recognition, vol.32, no.1, pp.71–86, 1999.
[31] P. Viola and W.M. Wells III, "Alignment by maximization of mutual information," International Journal of Computer Vision, vol.24, no.2, pp.137–154, 1997.
[32] D.T. Lee and B.J. Schachter, "Two algorithms for constructing a Delaunay triangulation," International Journal of Computer & Information Sciences, vol.9, no.3, pp.219–242, 1980.

Masahiro Yamaguchi received his B.E. degree in information and computer science from Keio University, Japan, in 2016. Since 2016, he has been a master student in the Department of Science and Technology at Keio University, Japan. His research interests include multi modal sensor, SLAM, 3D reconstruction, and computer vision.

Trong Phuc Truong received his B.E. degree in electrical engineering from École Polytechnique de Bruxelles, Belgium, 2015. He continued his specialization in electronics and information technology with Bruface, a jointly-organized master program between École Polytechnique de Bruxelles and Vrije Universiteit Brussel. After a year, he enrolled in a double degree program in Japan. Since 2016, he has been a master student in the Department of Science and Technology at Keio University, Japan.

Shohei Mori received a B.S., M.S., and Ph.D. degrees in engineering from Ritsumeikan University, Japan, in 2011, 2013, and 2016, respectively. He was a Research Fellowship for Young Scientists (DC-1) from the Japan Society for the Promotion of Science until 2016. He is currently a Research Fellowship for Young Scientists (PD) at Keio University and a guest researcher at Graz University of Technology.

Vincent Nozick received his PhD degree in 2006 from the University Paris-Est Marne-la-Vallee, France. In 2006, he is laureate of a Lavoisier fellowship for a post-doc position at Keio University, Japan. From 2008, he is hired as a “maitre de conferences” at the University Paris-Est Marne-la-Vallee, France.

Hideo Saito received his Ph.D. degree in electrical engineering from Keio University, Japan, in 1992. Since then, he has been on the Faculty of Science and Technology, Keio University. From 1997 to 1999, he joined the Virtualized Reality Project in the Robotics Institute, Carnegie Mellon University as a visiting researcher. Since 2006, he has been a full professor in the Department of Information and Computer Science, Keio University. His recent activities for academic conferences include being Program Chair of ACCV2014, a General Chair of ISMAR2015, and a Program Chair of ISMAR2016. His research interests include computer vision and pattern recognition, and their applications to augmented reality, virtual reality, and human robotics interaction.

Shoji Yachida received his B.E. degree in electrical engineering from Hiroshima Institute of Technology, Japan, in 1988. He joined NEC Home Electronics in the same year. Since 2000, he has worked in NEC Central Research Laboratories and engaged in developing computer vision devices.

Hideaki Sato received his Ph.D. degrees in engineering from Tsukuba University, Japan, in 2013. He researched the augmented reality about a mirror there. He joined NEC in the same year. After working in NEC Common Carrier Solutions Division, since 2015, he has worked in NEC Central Research Laboratories and engaged in researching about a way of working and an operations efficiency of system engineers and object recognition.