Visualization of Temperature Change using RGB-D Camera and Thermal Camera

Anonymous ECCV submission

Paper ID ***

Abstract. In this paper, we present a system for visualizing temperature changes in a scene using an RGB-D camera coupled with a thermal camera. This system has applications in the context of maintenance of power equipments where anomalies are detected with temperature changes. We propose a two-stage approach made of an with offline and an online phases. During the first stage, after the calibration, we generate a 3D reconstruction of the scene with the color and the thermal data. We then apply the Viewpoint Generative Learning (VGL) method on the color model for creating a database of descriptors obtained from features robust to strong viewpoint changes. During the second online phase we compare the descriptors extracted from the current view against the ones in the database for estimating the pose of the camera. In this situation, we are also able to display the current thermal data and compare it with the data saved during the offline phase. This system makes it possible to visualize temperature change by hand-held camera.

Keywords: thermal camera, 3D model reconstruction, camera tracking, visualization, RGB-D camera

1 Introduction

Usually, anomalies in power equipments or building structures are detected by looking for variations in the temperature which are difficult to be directly visualized. Such strong changes will often imply a malfunction or a future problem. A common way to evaluate the changes in the temperature state is to fix a camera and to compare the temperature at two different times. The resolution and the field of view of the thermal cameras is, however, quite small which makes difficult to monitor big size objects or large areas. Since the cost of such device is also high, it makes it hard to use several cameras to cover a large surface.

We then propose a system for detecting abnormalities from temperature changes in wide areas with a thermal camera coupled with an RGB-D camera. Our approach is based on two precomputed 3D reconstructions of the target scene achieved with a RGB-D camera coupled with the thermal camera as shown in Fig. 1. The first reconstruction holds the color information, while the second one holds the thermal information. The colored 3D reconstruction is used with the Viewpoint Generative Learning (VGL) [1] algorithm to detect feature points robust to strong viewpoint changes. We then generate a database with the corresponding 3D positions and descriptors of these features. For comparing the status of the temperature between the reconstruction and the current time, we accurately estimate the pose of the camera by finding keypoint correspondences between the current view and the database. Knowing the pose of the camera, we are then able to compare the thermal 3D reconstruction with the current status of the temperature from any viewpoint only by hand-held camera.

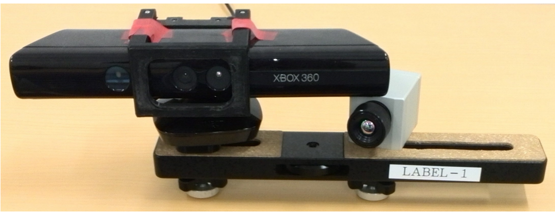


Fig. 1. Our capture system is made of the Microsoft's KINECT and the optris PI160.

Since the RGB-D camera and the thermal camera are two distinct devices, we need to estimate their relative pose. Here, we propose our own calibration board that makes easier the pose estimation of the thermal camera in reference to the RGB-D camera.

The remainder of the paper is organized as follow: After an overview of the components of our system, we describe our calibration process with the introduction of our calibration board. Section 4 will detail our reconstruction process based on Kinect Fusion, and in section 5 we will give a reminder about the VGL algorithm. After describing the online phase of our method, we finish by presenting the experiments.

In our system, Thermal information is projected onto the current color image, because that would makes easier to understand where we are looking at, and enhances our system [2]. In our knowledge, this is the first work to propose the visualization of temperature changes over middle sized areas.

2 Proposed Method

Our proposed method consists of two stages. During the first one, we precompute two 3D models of a scene, one with corresponding temperature distribution at the capture time and another with the color information. We will refer to this temperature map as the reference temperature. Kinect Fusion[3] is used to generate uncolored 3D Model. This offline phase thus requires a calibration that estimates the relative pose of the thermal camera in reference to the RGB-D camera. The thermal map will be used later for the comparison of the reference temperature with the current state. The colored model is the source of keypoints robust to strong viewpoint changes that will be used to create a database of descriptors and 3D points with the Viewpoint Generative Learning algorithm. The database will then be available for the online phase for accurately estimating the pose camera.

During the second stage, we estimate the pose of the camera and use it to compare the current temperature state with the reference one. The pose of the camera is estimated

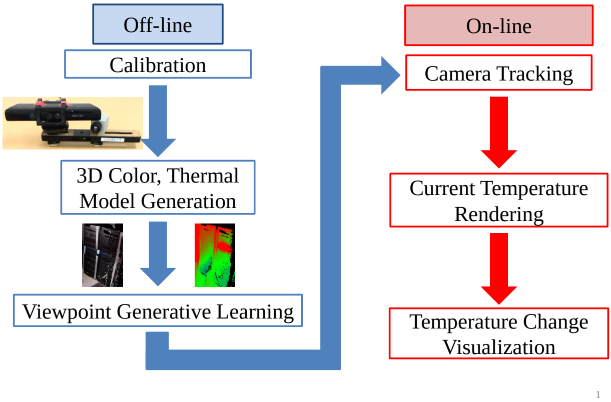

Off-line
Calibration
3D Color, Thermal Model Generation
Viewpoint Generative Learning
On-line
Camera Tracking
Current Temperature Rendering
Temperature Change Visualization
1

Fig. 2. System Overview

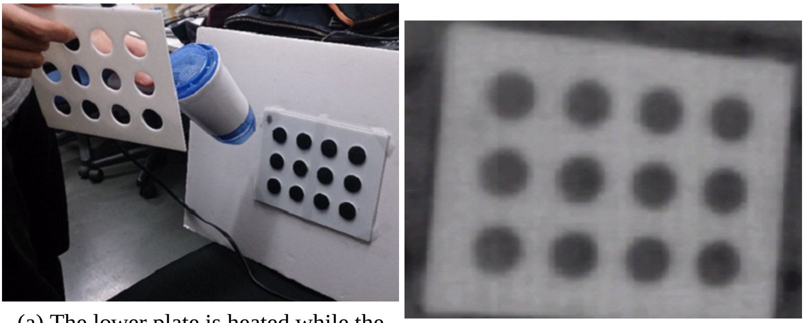
by comparing the descriptors stored in the database with the ones extracted from the current view. At this time, we know the pose of the camera, the current state of the temperature, and the reference one stored in a 3D model. With the pose estimation we can then align the thermal 3D model with the current viewpoint. By combining these data, users are able to visualize thermal changes and can freely move the camera around the scene, but in the limits of the 3D model. An overview of our system is depicted in Fig. 2.

3 Calibration

3.1 Our Calibration Board

A traditional approach for calibration is to use a planar pattern like a chessboard that can be easily detected and matched from multiple cameras [4]. If this approach works well with standard color cameras, it remains difficult to directly apply it with images from a thermal camera since the temperature on the calibration board is uniform. A common solution is to heat the board using, for instance, a flood lamp as described by [5] or [6].

We extended this idea by proposing a special calibration board that is visible from both color and thermal cameras. Our board is made of two plastic plates generated with a 3D printer. The first one, the lower plate, is made of a planar surface covered of regular bumps corresponding to the black parts of the calibration pattern. The second plate is designed to plug onto the first one, it is thus made of a planar surface with holes where the black parts of the calibration pattern should appear. At the end, combining both plates creates a flat calibration pattern like the ones commonly used. The plates can be observed in Fig. 3(a). To make it visible from the thermal camera, we simply heat the lower plate while the upper one remains at ambient temperature. This will provides



(a) The lower plate is heated while the upper one remains at ambient temperature

(b) The calibration board captured from the thermal camera

Fig. 3. The Board use for calibration the RGB-D and the thermal cameras

enough contrasts in the resulting image to detect the calibration pattern as presented in Fig. 3(b). For our board, we preferred the use of a pattern made of black circles rather than a checkerboard for two reasons. First, a circle shape makes the plug of the two plates easier. Second, the detection of the center of the circles remains robust even if the captured images are blurred or if the heat from the lower plate propagates uniformly on the upper plate.

3.2 Estimation of Intrinsic Parameters

The intrinsic parameters of the thermal camera are evaluated using the Zhang's method [4]. We capture several views of our calibration board and evaluate the corresponding focal lengths, principal point and aspect ratio. The skew parameter is considered null. For better evaluation of the parameters and since the sensor is slightly different from the pinhole camera model, we start by fixing the principal point at the center of the image plane and refined it during the calibration process.

3.3 Estimation of Extrinsic Parameters

The goal of this calibration is to estimate the pose (rigid transformation) of the thermal camera with reference to the RGB-D camera. For this process, we take advantage of the 3D coordinates provided by the depth camera. For each circle's center from the calibration board, we can obtain the corresponding 3D position. By finding the corresponding pixels in the thermal image, we create a set of 3D/2D correspondences. We then apply the Efficient Perspective-n-Point algorithm to estimate the extrinsic parameters [7].

However, the depth map generated by the RGB-D camera suffers of noise. We then propose to fit the 3D coordinates extracted from the calibration board to a planar surface. The equation of this surface is found by first gathering several samples ($\sim$400 3D points) from the target surface around the detected pixels. We then apply a singular value decomposition $U\Sigma V^*$ on the data and extract the singular vector from U describing the normal to the plane we are looking for. Finally, each of the 3D coordinates previously obtained from the center of the circles are projected onto the computed plane to accurately estimate their position. The resulting points are then used for the calibration of external parameters. Benefits of this approach will be demonstrated later in the experiment section.

4 Creation of the 3D Models

4.1 The 3D Models

Kinect Fusion[3] is used to generate uncolored 3D Model. This method estimates the pose of an RGB-D camera for each frame using a dense version of the Iterative Closest Point (ICP) algorithm on GPU [8], and integrated the depth information from each frame into a voxel grid using a Truncated Signed Distance Function(TSDF) [9].

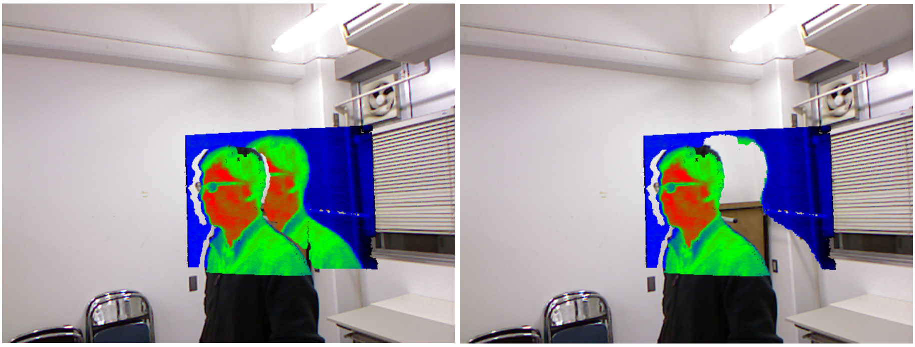
While running Kinect Fusion for generating the uncolored 3D reconstruction of the scene, we also save the pose estimation of the camera and the corresponding color information. After finishing the reconstruction, we convert the voxel grid into two meshes. Using the pose estimation of the camera, we then map the color images on one mesh, and the thermal images on the other one. The field of view of the thermal camera is however smaller than the RGB-D camera' one and thermal data will not cover all the surface of the 3D model.

4.2 Resolution of occlusions with the thermal images

As described in the previous subsection, we generate the colored 3D model and the thermal 3D model in similar ways. But, the RGB-D camera and the thermal camera are located at two slightly separate positions which implies that we need to apply the rigid transformation computed in Sec.3.3) to compensate this difference and correctly performing the mapping. Also, since the viewpoints are different, we need to deal with occlusions on the 3D reconstruction during the mapping stage as observed in Fig. 4(a).

Our solution is to perform a depth test by projecting depth and color pixels from the RGB-D camera onto the thermal image plane. First, the 3D points corresponding to the pixels from the depth image are projected onto the thermal image, and are discarded if the projection is outside of the thermal image plane. Since we are dealing with occlusions, a pixel of the thermal image can correspond to multiple depth/color values from the RGB-D image. Our goal is then to conserve the candidate with the smallest depth in the thermal image. This pixel will finally represent the surfaces visible from the thermal camera viewpoint.

At this point, several pixels of the thermal camera can still be incorrectly displayed in the color image, especially if the field of view of the cameras are strongly different. In our case, the RGB-D camera has a vertical fov of 45° while the thermal camera's field of view is 31°. So, when projecting a RGB-D pixel onto the thermal image, it will overlap multiple thermal pixels and not a single one. We resolved this issue by computing the average of the values inside of a 3×3 pixel area (empirically estimated)



(a) Before the processing

(b) After the occlusion removal

Fig. 4. Example of occlusion processing, almost all of the occlusion areas are removed by our depth test approach.

centered on the projected color pixel and by replacing the neighbors pixels with a strong absolute difference with average of this area.

Finally, for each pixel of the RGB-D image, we can find or not (in case of occlusions or if the projection is outside of the thermal image) a correspondence in the thermal image. An example of our occlusion removal process is presented in Fig. 4 (b).

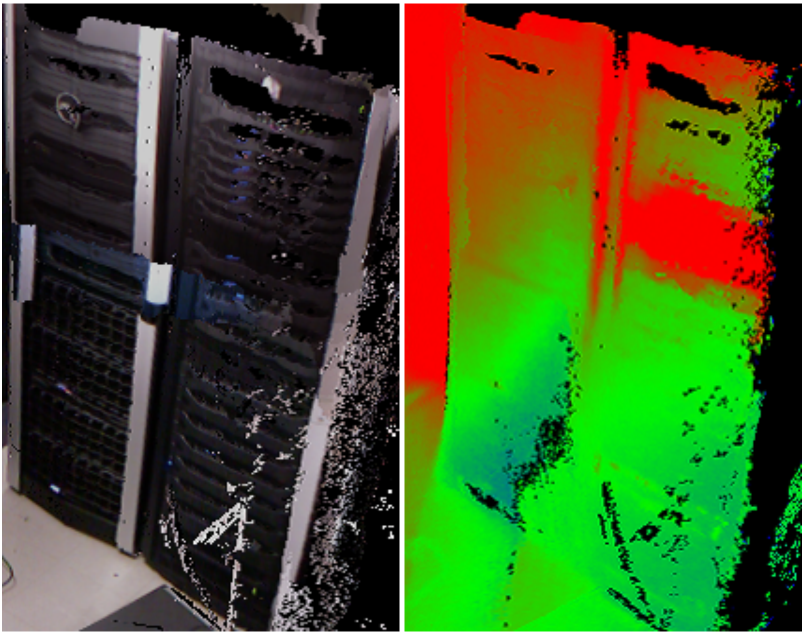
5 Viewpoint Generative Learning for tracking

During the online phase, in order to estimate the pose of the RGB-D camera with the scene captured for the 3D model, we need a tracking algorithm that can be robust against strong viewpoint changes and occlusions. Our solution is to use the Viewpoint Generative Learning (VGL) [1]. The first step requires, during the offline phase, to generate a database of descriptors from visual features with high repeatability. The idea is then to capture the reconstructed 3D model of the scene from several different views using the OpenGL rendering process as illustrated in Fig. 6. For each image obtained, we detect the features with SIFT [10]. We aggregate these features in the 3D space and conserve only the ones that can be detected over multiple views. We define these features with high repeatability as stable keypoints and extract the corresponding descriptors. At this state, however, the amount of data is too high for expecting a fast traversal of the database. We then decided to cluster the descriptors of a stable keypoints by applying k-means++ [11] on them. Finally, we store in the database the clustered descriptors and the 3D position of each stable keypoint.

6 Online phase

6.1 Camera Tracking

During the online phase, we want to display the actual temperatures of the scene and make comparisons with the reference temperature mapped on the 3D thermal model.



(a) The colored model (b) The thermal model

Fig. 5. 3D models of a scene occupied by server machines

This means that we need to find correspondences between the pixels of the current view and the 3D coordinates and descriptors of the 3D model stored in the VGL database.

The tracking algorithm consists of two phase, the first one consists in initializing the pose of the camera by comparing the current view with the database. The second phase uses the initialization for performing a frame to frame tracking. This approach appears to be faster since requesting the database is slow. Also, we can only use descriptors stored in database, so if good features are detected in current frame, we ends up discarding those if we don't have corresponding stable keypoints in database.

In the first frame, we start by detecting features in the current image captured by the RGB-D image and extract their descriptors. We look for the two most similar descriptors inside of the database using the Fast Library for Approximate Nearest Neighbors (FLANN) algorithm. We then evaluate the Euclidean distance ratio between the descriptors from the current view and these two nearest neighbors from the database. If the ratio is under a given threshold, we then verify the established correspondence, otherwise the correspondence is considered as incorrect. Using these results, we are able to generate a set of 3D/3D correspondences with the 3D position stored in the database and RGB-D current view. The pose of the RGB-D camera is finally deduced with a singular value decomposition associated to RANSAC for excluding wrong correspondences.

In the frame-to-frame tracking, we also extract descriptors from current RGB-D frame. It then searches in local neighborhood for correspondences with the feature from

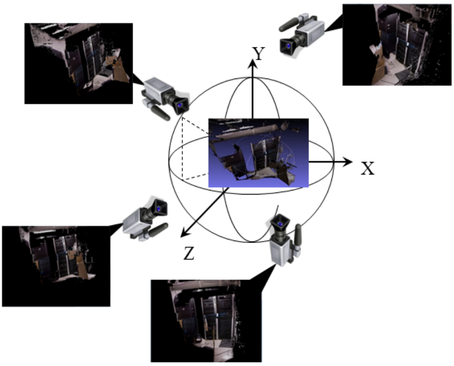
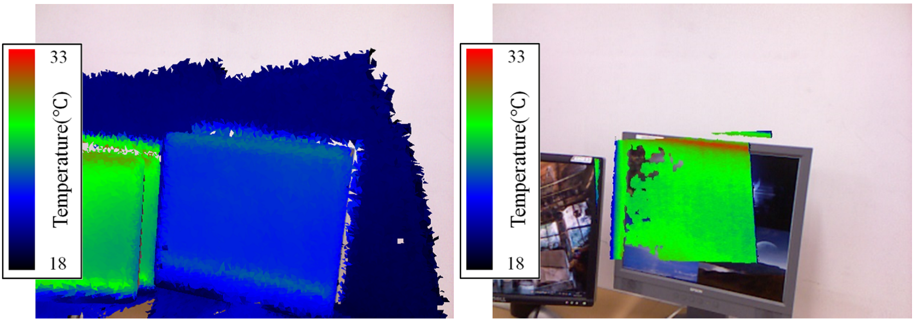


Fig. 6. Illustration of the multiple view rendering used in VGL



(a) Reference temperature

(b) Actual temperature

Fig. 7. Examples showing two different states of the temperature distribution of the scene

the previous frame assuming a small displacement. The matching pairs are evaluated based on Euclidean distance, and keep the closest one as matching pair. The pose is finally deduced with a singular value decomposition.

Fig. 7 shows an example of visualization of the reference temperature on the current captured view and of the current temperature.

6.2 Online Thermal Image Rendering

During the online processing, we project the thermal information from thermal camera onto the color image of the RGB-D camera using previously estimated intrinsic and extrinsic parameters of the camera. Occlusions are resolved in the same manner than the algorithm we mentioned in Sec.4.2, and applied on GPU with CUDA. The processing time will be presented in the experiment section.

7 Experiment

7.1 Calibration Accuracy

In order to evaluate our calibration accuracy, we estimated the field of view of the thermal camera, which is calculated using intrinsic parameter from our calibration method, and compare it with the one written in technical description of thermal camera. We used two kinds of thermal camera in the experiment. One is NEC Avio ' s Thermal Shot F30 with a resolution of 160×120 and a framerate 8 img/s. The other one is optris's PI160 with a resolution of 160×120 and a framerate 120 img/s. Vertical/Horizontal values of the field of view of the f30 is 31°/41°. Vertical/Horizontal values of the field of view of the PI160 is 21°/28°. We estimated those parameter of the F30 to 20.18°/27.56°, and PI160 to 41.6396°/30.9459°. We can safely say that our intrinsic parameters are correct while assuming that the principal point is close from the center of the image.

The accuracy of the extrinsic parameters are evaluated based on a re-projection error computation. In this experiment, we compare the average of re-projection error with the planar approximation and without it. By using the extrinsic matrix and the intrinsic matrices of the RGB-D and thermal cameras, we projected the centers of the circle from our calibration pattern from the color image onto the thermal image that we define as the ”projected point”. We then compute the re-projection error as the sum of the distances between the projected points and the detected centers of the circles in thermal image. Table.1 depicts the accuracy of our calibration process with and without the planar fitting approach. this result demonstrates that the calibration process is more accurate when we use planar approximation for reducing the noise from the depth image. The Thermal Camera is the Thermal Shot F30 with a resolution of 160×120 and a framerate 8 img/s.

Table 1. Extrinsic Calibration Accuracy Comparison

Thermal Camera	Planer Approximation	Reprojection Error(pixel)
F30	Use	5.05
	Don't Use	5.46
PI160	Use	2.84
	Don't Use	2.92

7.2 Overall System Evaluation

In this experiment, we precomputed a two 3D models as shown in Fig. 8. In Scene1/Scene2, we demonstrate that proposed system is effectiveness against small/big objects. In scene1, we also compute the processing time. The system was executed on a PC with 16.0GB of Ram, a Core i7-4800MQ CPU and a Nvidia Geforce GTX 780M graphic card. The RGB-D camera is Microsoft Kinect with a resolution 640×480 and a framerate of 30 img/s.

In scene1 we used the Thermo Shot F30, and scene2 we used optris PI160.

Processing time The processing time of our system is presented in Table.2. We computed the processing time on an average of 100 frames. We can observe 50% of the time is dedicated to the tracking. The reason is that we use SIFT [10] as local features, which is computationally expensive to extract. This might be improved by defining new descriptors which is a combination of local feature(computationally cheap one such as FAST [12]) and the depth information.

Table 2. Evaluation of Processing Time

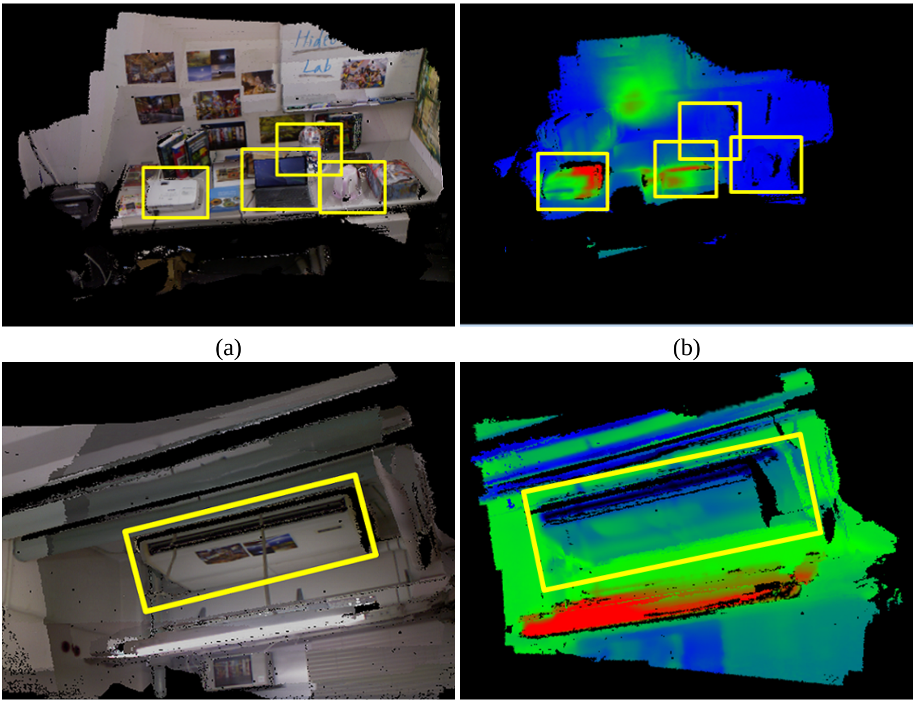
	processing time(sec)
Tracking	0.110
Render on-line thermal image	0.008
Visualization	0.084
Total	0.202

Experiment with small objects For this experiment, we used different small target objects such as a mannequin's head (manually heated), a laptop, a projector and an electric kettle. The objects can be seen in Fig. 9 with also the reference temperature and the current temperature states. We can visually notice that the thermal data match the 3D corresponding objects. For evaluating our system, we computed the average error between the depth values from the 3D model and the current captured depth map. We compared only pixels located in the area covered by the thermal data in the current view.

In the left side of Table.3, we present the average error in terms of depth for each of our target objects. For objects with a relatively simple shape such as the projector, the error becomes less than 1cm. On the other hand, with more complex objects like the mannequin's head and the electric kettle, the error varies from 1cm to 3cm. However, with the laptop PC even if its shape is simple, the error is the largest one, because its material properties increase the noise in the depth map. By observing the results, we can then conclude that our system is stable to many kinds of small objects, and that the tracking and calibration are accurate enough for our purpose.

Experiment with large object For this experiment, we used an air-conditioner as a large target object. We evaluated in the same manner as for small objects. Accuracy is evaluated from different view points(front, side, under, behind). The result is shown in Fig. 10, and the right side of Table.3. We can visually notice that the thermal data match the 3D corresponding objects. Average of depth error from ”Side”, ”Under”, ”Behind” viewpoints is under 3cm. We can then assume that the current view of the temperature is correctly projected on the reference model. Average of depth error from and ”Front” viewpoint is over 4cm and is larger compared to the one from other viewpoints.

For ”Front” viewpoint, images were captured from far, that is why camera tracking by matching descriptors would be a difficult task, and also depth accuracy with RGB-D camera would become low.



(a) (b)

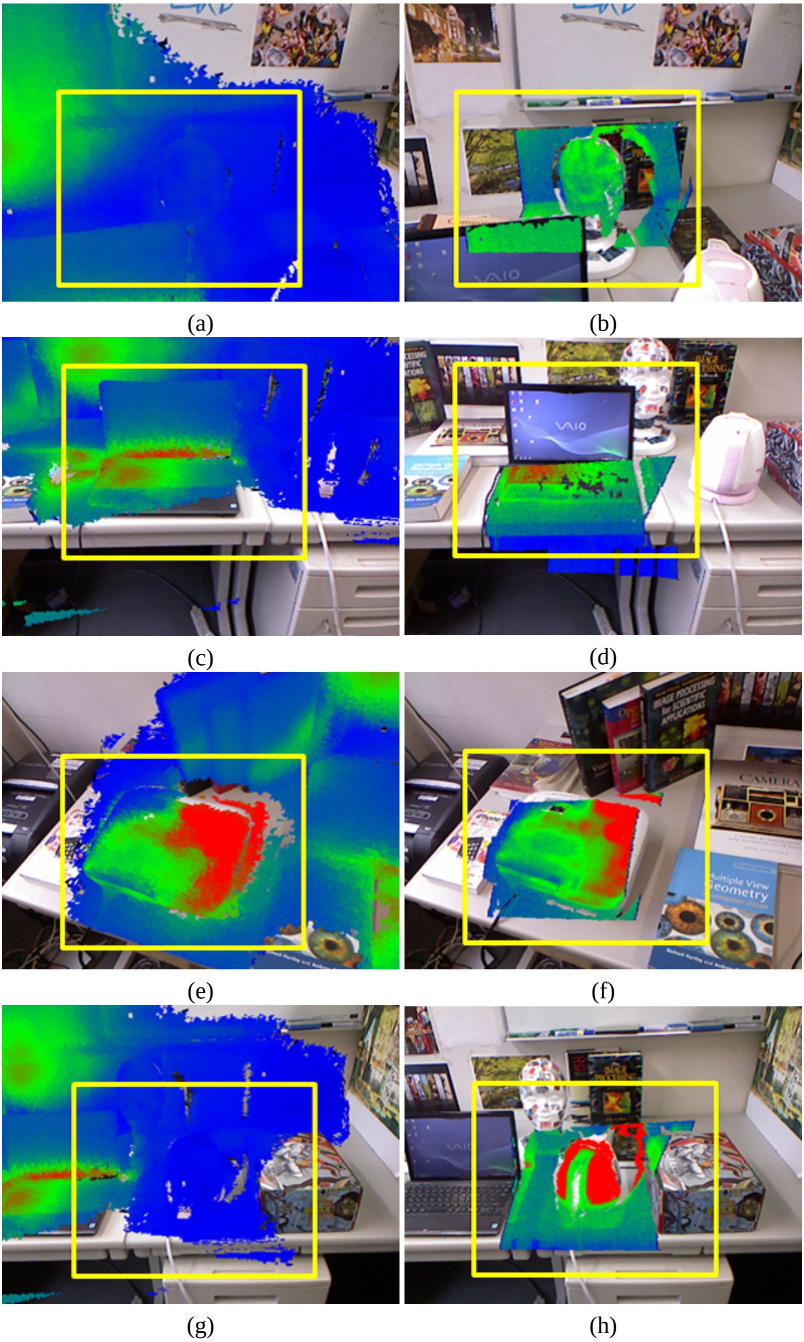
(c) (d)

Fig. 8. 3D Color and Thermal Models used in experiment. The top row is a relatively small scene filled with several small target objects captured with NEC Avio's Thermo Shot F30. The bottom row is large scene which target object is air-conditioner captured with Optris PI 160. Target objects are emphasized with yellow lines.

For these reasons, about result from ”Front” viewpoint, we can say result is acceptable. We can then conclude that our system is robust to strong viewpoint changes and works for large object which we need to see temperature changes from many viewpoints to detect abnormalities. (For example, temperature change of outlet of cold air can only be seen from front viewpoint.)

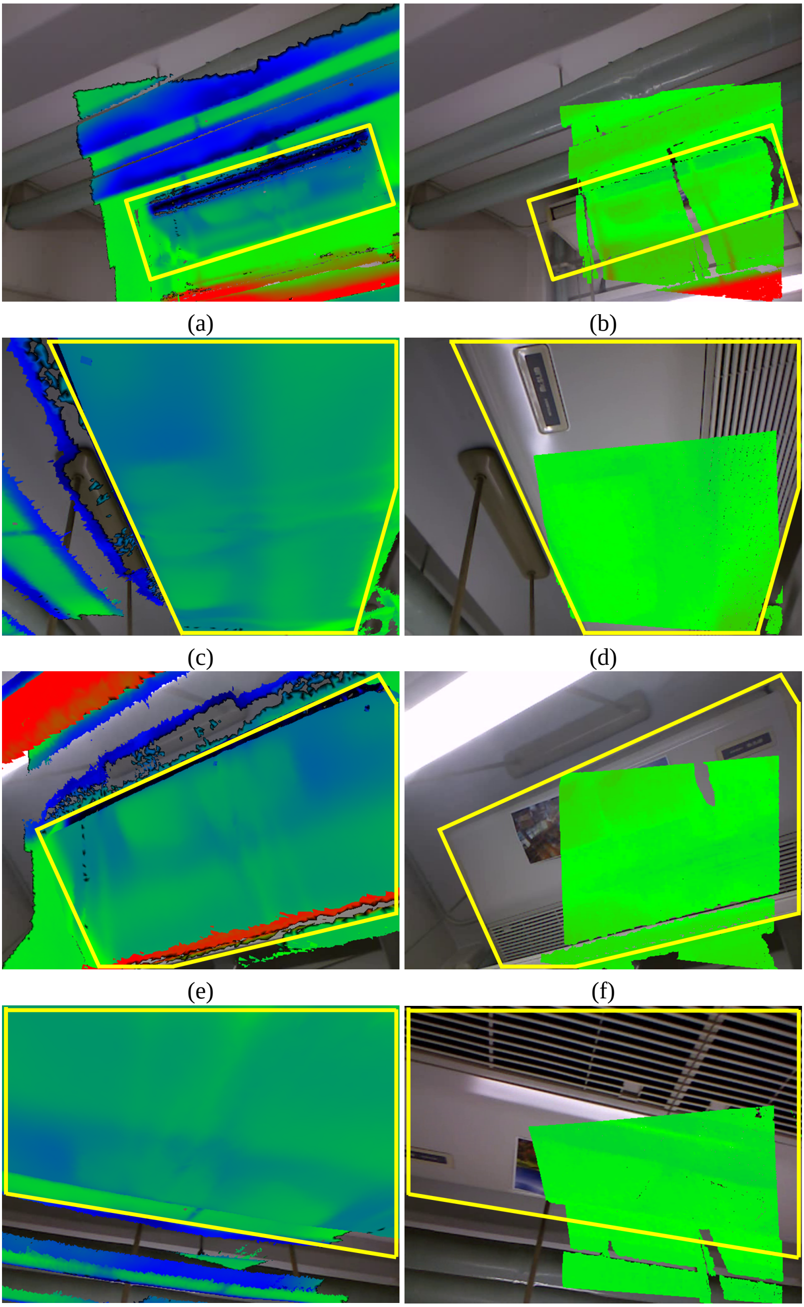
8 Conclusion

In this paper, we proposed a system for visualizing temperature changes of a given scene using a RGB-D camera coupled with a thermal camera. During an offline phase, we reconstruct a 3D model of the scene and save the poses of the camera with the corresponding color and thermal images. During the online phase, using the Viewpoint Generative Learning method applied on the 3D reconstruction of the scene, we are able to know the pose of the camera and compare the current status of the temperature compared with the reference one. With our experiments, we have shown that we can ac-



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 9. Thermal image with the reference temperature on the left column and the current temperature state in the right column. Mannequin, notebook PC, projector, electric kettle from top to bottom The size of the thermal information is smaller in the right column because left one is generated by rendering precomputed 3D thermal model from the estimated camera pose.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 10. Thermal image of normal state and abnormalities detecting time. Target object is air-conditioner and images are captured from front, side, under, behind against target object from top to bottom

Table 3. Overall System Evaluation

Scene1		Scene2	
Target Object	Average Error of Depth(mm)	Viewpoint	Average Error of Depth(mm)
Mannequin	13.63	Front	48.5208
Note-PC	43.88	Side	9.08713
Projector	6.39	Under	25.7105
Electric Kettle	23.48	Behind	26.9239

curately calibrate our capture system and visualize the differences between current and reference temperatures. In future works, we would like to optimize the tracking by using new descriptors that could be a combination of local feature and depth information, and focus on single objects tracking rather than a whole scene.

References

1. Thachasongtham, D., Yoshida, T., de Sorbier, F., Saito, H.: 3d object pose estimation using viewpoint generative learning. In: SCIA13. (2013) 512–521
2. Rosten, E., Drummond, T.: Machine learning for high speed corner detection. In: 9th Euproean Conference on Computer Vision. (2006) 430–443
3. Izadi, S., Newcombe, R.A., Kim, D., Hilliges, O., Molyneaux, D., Hodges, S., Kohli, P., Shotton, J., Davison, A.J., Fitzgibbon, A.: Kinectfusion: Real-time dynamic 3d surface reconstruction and interaction. In: ACM SIGGRAPH 2011 Talks. SIGGRAPH '11 (2011) 23:1–23:1
4. Zhang, Z., Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (1998) 1330–1334
5. Cheng, S., Park, S., Trivedi, M.: Multiperspective thermal ir and video arrays for 3d body tracking and driver activity analysis. In: Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on. (June 2005) 3–3
6. Prakash, S., Lee, P.Y., Caelli, T., Raupach, T.: Robust thermal camera calibration and 3d mapping of object surface temperatures. In: Proc. SPIE. Volume 6205. (2006) 62050J–62050J–8
7. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o(n) solution to the pnp problem. Int. J. Comput. Vision **81**(2) (feb 2009) 155–166
8. Besl, P., McKay, N.D.: A method for registration of 3-d shapes. Pattern Analysis and Machine Intelligence, IEEE Transactions on **14**(2) (1992) 239–256
9. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '96 (1996) 303–312
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. In: International Journal of Computer Vision. (November 2004) 91–110
11. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. (2007) 1027–1035
12. Szabo, Z., Berg, S., Sjokvist, S., Gustafsson, T., Carleberg, P., Uppsall, M., Wren, J., Ahn, H., Smedby, O.: Real-time intraoperative visualization of myocardial circulation using augmented reality temperature display. In: The International Journal of Cardiovascular Imaging. (2012) 521–528