Multi-Modality Mobile Image Recognition Based on Thermal and Visual Cameras

Jui-Hsin(Larry) Lai*, Chung-Ching Lin*, Chun-Fu(Richard) Chen†, and Ching-Yung Lin*
*IBM T.J. Watson Research Center
1101 Kitchawan Road, Yorktown Heights, NY 10598, USA
Email: {larrylai, cclin, chingyung}@us.ibm.com
†Department of Electrical Engineering, National Cheng Kung University
1 University Road, Tainan City, 70101, Taiwan
Email: n28991146@mail.ncku.edu.tw, cfchen@us.ibm.com

***Abstract*—The advances of mobile computing and sensor technology have turned the mobile devices into powerful instruments. The integration of thermal and visual cameras extends the capability of computer vision, due to the fact that both images reveal different characteristics in images; however, image alignment is a challenge. This paper proposes an effective approach to align image pairs for event detection on mobile through image recognition. We leverage thermal and visual cameras as multi-modality sources for image recognition. By analyzing the heat pattern, the proposed APP can identify the heating sources and help users inspect their house heating system; on the other hand, with applying image recognition, the proposed APP furthermore can help field workers identify the asset condition and provide the guidance to solve their issues.**

***Keywords*-mobile computing; heat pattern analysis; image alignment; thermal image**

I. Introduction

With the advance in sensor technology and mobile computing capability, more and more emerging applications with real-time processing can be implemented on mobile devices. By combing thermal and visual cameras for event detection, the mobile devices can turn into powerful instruments.

In this paper, we take thermal and visual cameras as the multi-modality sources for heat source detection and asset inspection based on image recognition. Figure 1 shows the configuration of the mobile devices, including an external camera connecting to a smartphone. The integration of the multi-modality image sources is a challenge because the thermal and visible images are captured from different cameras, containing different information of the scene. The literature associated with thermal image applications and mutli-modality image pair alignment in state-of-the-art is surveyed in Section II. We propose the methods in Section III to align the visible-visible images and thermal-visible images. By leveraging the computation capability on mobile devices, we propose the application with thermal pattern analysis for heating source detection in Section IV to help users inspect the mansion's heating system. In Section V, we apply thermal and visual cameras for image recognition to help field workers identify the asset condition and provide the guidance to solve their tasks.

Figure 1: Multi-modality camera sources integrated on the mobile device.

We use FLIR ONE Personal Thermal Imager[1] as the external camera, which can capture thermal and visible images with 240×320 resolution, and connects to the iOS device equipped with a 4K resolution camera; therefore, there are two visual cameras and one thermal camera in our system.

II. Related Works

Gade and Mpeslund [1] comprehensively described the applications based on thermal cameras and visual cameras. Due to different characteristics in thermal images from visible images, several applications based on single thermal image have been proposed, such as heat distribution inspection [2], stability of heat source [3], pedestrian detection and tracking [4], face recognition [5], blood pressure measurement [6]. Nonetheless, due to the lack of texture and low spatial resolution in thermal images, its capability is confined. Therefore, researchers are towards the combination of visible and thermal images.

Gade and Mpeslund [1] and Conaire *et al.* [7] also pointed out the challenges in aligning and fusing thermal and visible images. The basis in aligning and fusing images is similar to aligning two visible images from different viewpoints [8], that is, estimate the homography between two images. Trivedi *et al.* [9] reported a heated chessboard to calibrate cameras' parameters, the corresponding corner points of heated chessboard can be easily found via both

[1]http://www.flir.com/flirone/

thermal and visible images. Sonn *et al.* [10] developed a fast image registration on thermal-visible image pairs, they aligned two images at key-points level to ensure the correspondences are matched. Irani and Anandan [11] proposed a pyramid structure to align multiple sensor sources, they roughly align images through global information and then refine the results via local details. Wu *et al.* [12] matched thermal-visible image pair via visual salient features that are extracted according to the traits of thermal and visible images, respectively.

III. Thermal and Visible Images Alignment

In this section, we propose the algorithm to compensate image misalignment due to the images captured from various cameras. As shown in Figure 1, to align the thermal image from external camera and the image captured by the smartphone, our system has two alignment methods: visible-visible image alignment and thermal-visible image alignment. The visible-visible image alignment registers the visible image captured by the external camera to the image captured by the smartphone. The thermal-visible image alignment registers the thermal image to the visible image captured by the external camera.

A. Visible-Visible Image Alignment

The scene of this application is usually not in a plane shape. If we use a global homography model to register the images, the parallax issues may cause the alignment not accurate. To provide better alignment, local homography model is used to register the visible images.

Let the target and the reference images be denoted by I and image I'. Given a pair of matching points $\mathbf{p} = [x\ y]^T$ and $\mathbf{p}' = [x'\ y']^T$, between I and I', the homographic transformation $\mathbf{p}' = \mathbf{h}(\mathbf{p})$ can be represented as

$$\mathbf{h}_x(\mathbf{p}) = \frac{h_1 x + h_2 y + h_3}{h_7 x + h_8 y + h_9}, \tag{1}$$

$$\mathbf{h}_y(\mathbf{p}) = \frac{h_4 x + h_5 y + h_6}{h_7 x + h_8 y + h_9}. \tag{2}$$

In homogeneous coordinates $\mathbf{p} = [x\ y\ 1]^T$, and $\mathbf{p}' = [x'\ y'\ 1]^T$, it can be represented up to a scaling using the homography matrix $\mathbf{H} \in \mathbb{R}^{3\times 3}$ as

$$\hat{\mathbf{p}}' \sim \mathbf{H}\hat{\mathbf{p}}. \tag{3}$$

The columns of $\mathbf{H}$ are given by $\mathbf{h}_1 = [h_1\ h_4\ h_7]^T$, $\mathbf{h}_2 = [h_2\ h_5\ h_8]^T$, and $\mathbf{h}_3 = [h_3\ h_6\ h_9]^T$. Taking a cross product on both sides of (3), we obtain

$$\mathbf{0}_{3\times 1} = \hat{\mathbf{p}}' \times \mathbf{H}\hat{\mathbf{p}} \tag{4}$$

which can be rewritten as

$$\mathbf{0}_{3\times 1} = \begin{bmatrix} \mathbf{0}_{3\times 1} & -\hat{\mathbf{p}}^T & y'\hat{\mathbf{p}}^T \\ \hat{\mathbf{p}}^T & \mathbf{0}_{3\times 1} & -x\hat{\mathbf{p}}^T \\ -y\hat{\mathbf{p}}^T & x'\hat{\mathbf{p}}^T & \mathbf{0}_{3\times 1} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix}. \tag{5}$$

We denote the 9×1 vector in (5) as $\mathbf{h}$. Since only two rows of the 3×9 matrix in (5) are linearly independent, for a set of N matching points $\{\hat{\mathbf{p}}_\mathbf{i}\}_{i=1}^N$, and $\{\hat{\mathbf{p}}'_\mathbf{i}\}_{i=1}^N$, we can estimate $\mathbf{h}$ using

$$\mathbf{h} = \underset{\mathbf{h}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \begin{bmatrix} \mathbf{a}_{i,1} \\ \mathbf{a}_{i,2} \end{bmatrix} \mathbf{h} \right\|^2 = \underset{\mathbf{h}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{h}\|^2, \tag{6}$$

where $\mathbf{a}_{i,1}$ and $\mathbf{a}_{i,2}$ correspond to the first two rows of the matrix in (5). We also incorporate the constraint $\|\mathbf{h}\|^2 = 1$ since the homographic transformation has only 8 degrees of freedom.

In [13], authors introduced moving DLT framework to estimate local homography by including locality-enforcing weights in the objective of (6). The local homography at the location $\mathbf{p}_j$ is estimated as

$$\mathbf{h}_j = \underset{\mathbf{h}_j}{\operatorname{argmin}} \sum_{i=1}^{N} \omega_{i,j} \left\| \begin{bmatrix} \mathbf{a}_{i,1} \\ \mathbf{a}_{i,2} \end{bmatrix} \mathbf{h} \right\|^2 \tag{7}$$

which can be written in matrix form as

$$\mathbf{h}_j = \underset{\mathbf{h}}{\operatorname{argmin}} \|\mathbf{W}_j\mathbf{A}\mathbf{h}\|^2, \tag{8}$$

where $\mathbf{W}_j = \operatorname{diag}([\omega_{1,j}\ \omega_{1,j}\ \ldots\ \omega_{N,j}\ \omega_{N,j}])$. In [13], the weights are generated using the offsetted Gaussian which assumes high value for pixels in the neighborhood of $\mathbf{p}_j$ and equal values for those that are very far,

$$\omega_{i,j} = \max(\exp(-\|\mathbf{p}_i - \mathbf{p}_j\|^2/\sigma^2),\ \gamma). \tag{9}$$

The parameter $\gamma \in [0\ 1]$ is the offset used to prevent numerical issues.

In our method, we use the moving DLT without offset [14] to estimate the local homography. This weighting scheme is insensitive to parameter selections.

B. Thermal-Visible Image Alignment

The registration between the thermal and visible images is presented in this subsection. Although these two cameras provide different information of the scene, both edge can provide boundary information of the same objects. To register the thermal and visible images, the edge maps of both images are extracted. Since the centers of the visible and thermal cameras are at the same horizontal line and the rotational parameters of both cameras are the same, the corresponded pixels have only horizontal movements between the images. The horizontal displacements of the pixels can be obtained by fitting the edge maps. Therefore, the finding the displacement to fit the edge maps is formulated by the following equation.

$$d^* = \underset{d}{\operatorname{argmin}} \sum_x \sum_y |E_t(x,y) - \tilde{E_v^d}(x,y)|^2, \tag{10}$$

where $E_t(x,y)$ is the edge map of thermal image, $\tilde{E_v^d}(x,y)$ is the edge map of visible image with a displacement d,

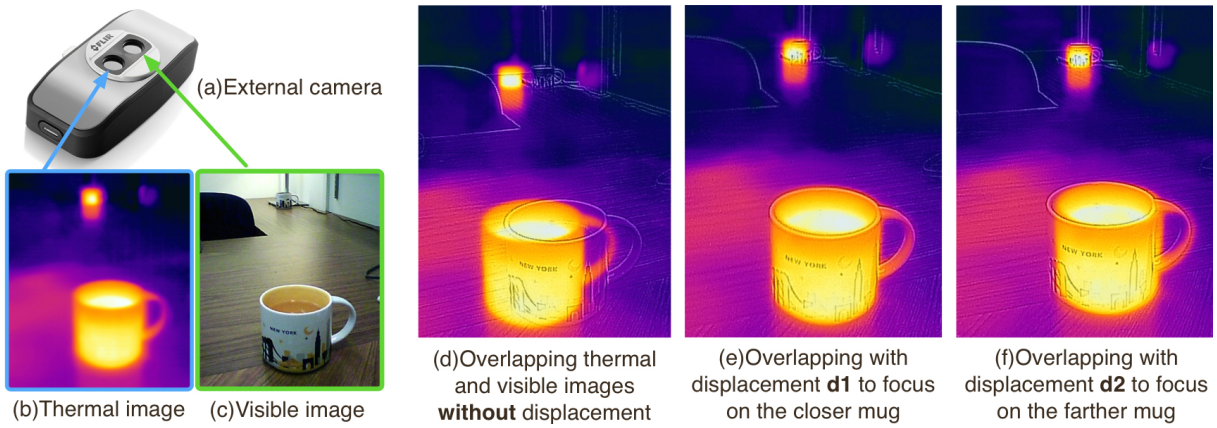


Figure 2: Thermal-visible image alignments with various displacements.

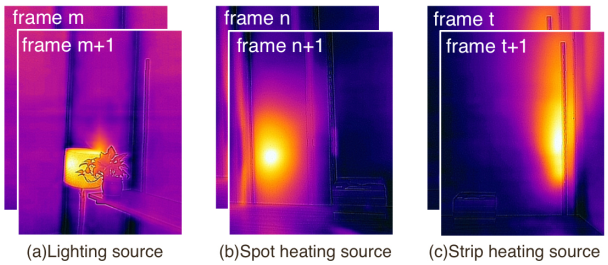


Figure 3: Thermal videos captured around a mansion, where the warmer color indicates higher temperature. (a) Lighting source, (b) spot heating source, and (c) strip heating source.

x and y are image coordinates, and d^* is displacement that has the minimum difference of edge maps. Figure 2 shows an example of the thermal-visible image alignment. By changing the displacement d, the thermal-visible image alignment can fit to objects in different depth. In addition, the horizontal displacements can be used to register the images and create a depth map.

IV. Application: Heat Source and Pattern Detection via Multi-Modality Video Analysis

In this section, we apply the proposed multi-modality video analysis, running on iOS devices with thermal camera, to detect heat source and pattern. Here is a scenario: a person would like to inspect heating system and understand the heat distribution in every corner of his/her house. This APP is capable of indicating heat source, analyzing heat pattern, visualizing heat distribution on the fly in cooperation with the installed visual and thermal camera. Figure 3 shows the examples of captured thermal videos, the warmer color indicates higher temperature.

To differentiate various heat patterns, we predefine four common patterns with distinguishable traits; and then, apply the proposed mobile framework to classify them into corresponding class. Four patterns and their detection criteria are

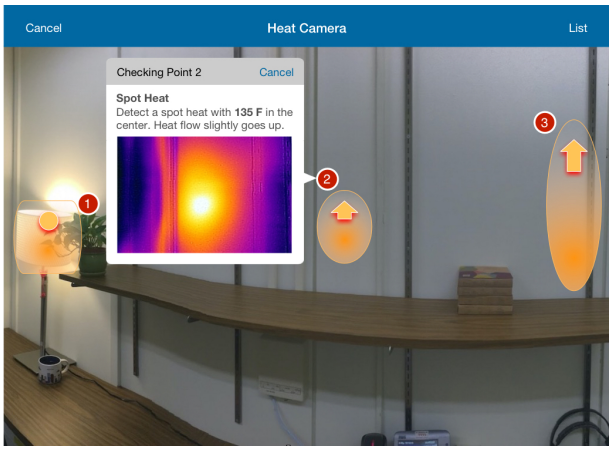


Figure 4: Results of video analysis with heat gradient and flow overlapped on the panorama. By touching the circled number, a pop-up UI describes the details.

illustrated as follows:

- **Lighting Source**: A region with higher temperature than its surroundings is detected, whose luminance distribution on visible video frames is high as well. The lighting source detection can be used to evaluate the energy conversion efficiency by analyzing the ratio of temperature and luminance values.
- **Fire Source**: A region with higher temperature than its surroundings is detected, and the heat pattern is changing quickly over a short period of time. The fire source detection is critical to safety issues, especially getting highlighted in digital home applications.
- **Spot Heating Source**: A region with higher temperature than its surroundings is detected, whose distribution like a circle and stable over a period of time, and the luminance distribution on visible video frames is not higher than its surroundings. The spot heating detection is an important feature to prevent electrical

fire.

- **Strip Heating Source**: A region with higher temperature than its surroundings is detected, whose distribution like a strip and stable over a period of time, and the luminance distribution on visible video frames is not higher than its surroundings. The strip heating source is common in house heating system like the hot water flowing in the pipe. Users can use it to monitor their home heating conditions.

In addition, if the heating source does not belong to one of above patterns, it will be classified into *other* class. Figure 3 shows three different heating sources including lighting, spot, and strip, from left to right.

Nonetheless, thermal images are usually visualized with warm-cool color palette, which is not informative for most people due to the shortage of visible texture. We build a panorama from visible videos and overlap the detected heating sources and patterns to ease the analysis.

Figure 4 shows the APP generates the panorama with heating sources and patterns overlapped on it. The shapes in orange color represent the patterns of heating source. The color gradient within shapes represents heat gradient in thermal video frames. The arrow length and direction represent the magnitude and direction of heat gradient, respectively. By touching the circled number, a pop-up UI describes the evidences of detection results, and displays the thermal image beyond the visible world. As a consequence, users can easily inspect the pattern of heat source and check the stability of heat source. A demo video is available on the website [2].

In terms of thermal imaging accuracy, the temperature measured from our thermal camera depends on a variety of factors, including the distance from the object, the ambient temperature, and the emissivity of the material being observed, etc. In our experiments, the measured temperature would be inaccurate when measuring distance is longer than 3 meters.

V. Application: Asset Inspection with Multi-Modality Image Recognition

We apply thermal and visible images as multi-modality sources for asset inspection to provide immediate assistance for field workers. Here is one scenario of applications: a field worker has no idea to fix an asset because of the shortage of experiences or the construction difficulty; however, companies might have well-documented the solutions for those issues. Hence, based on multi-modality image recognition, our application will assists them in repairing the asset right away through fast retrieving correct documents. Through automatic model identification and issue identification, field workers can retrieve precise instructions, which will be overlapped on the captured image and illustrate the steps field workers must perform to resolve the asset, to patch the device.

We design the APP user interface (UI) is as shown in Figure 5 (a) to make field workers work efficiently. The table chart on the left panel lists all to-do tasks for the field worker, and each task contains customer address, task description, location, and fixing history of this asset are displayed on the right panel. First of all, we detect the asset model to indicate the instruction manual based on QRcode or image recognition; and then, we search exact solution in the manual for asset's issues through comparing visual and thermal conditions of asset with golden references built by original manufacturers.

A. Asset Model Recognition

Field workers have two approaches in recognizing the asset model: (I) QRcode detection and (II) image recognition, and field workers can select one of them at the UI shown in Figure 5(a), where QRcode and Camera button both enable the visual camera to detect asset model. QRcode detection is intuitive and the detection result is usually convinced. We deploy the QRcode detection in CoreImage on iOS platform, which is a native iOS framework (equivalent to accelerated by GPU on mobile device[3]. Once decoding the QRcode, the APP can pop out the instruction manual for fixing the asset.

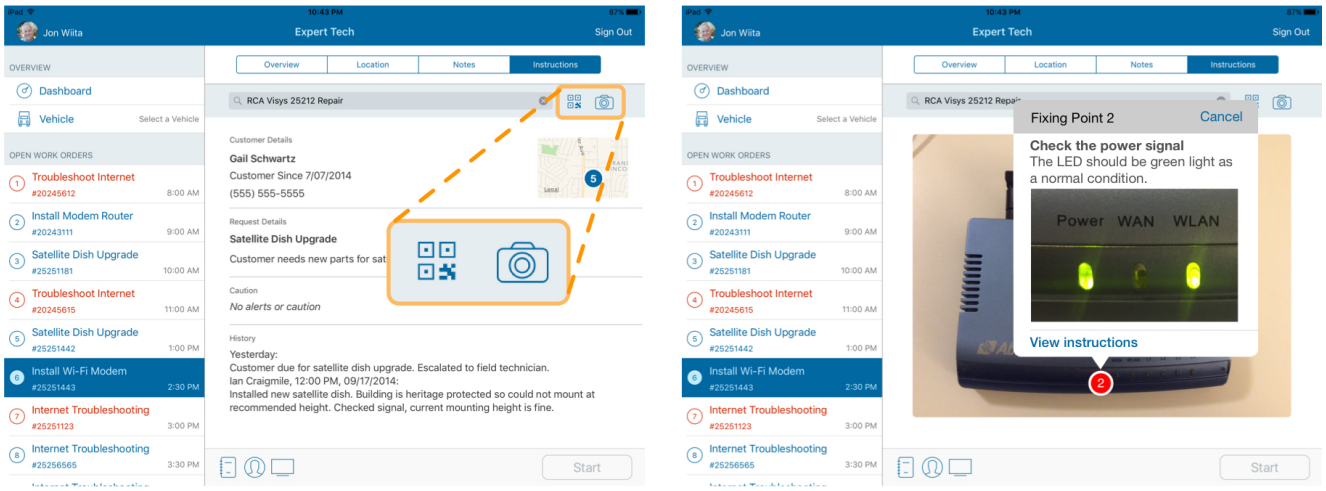
On the other hand, since QRcode label is not always attached on every asset or it might be torn to result in unsuccessful QRcode detection; therefore, using image recognition to identify the asset model is a promising solution in practical. In general, we can upload the captured image to the cloud server for image recognition; however, field workers sometimes do the job under the environment without Internet access. Therefore, image recognition running on a mobile device instead of cloud servers becomes an important demand. Now, low computation capability of mobile devices is one of major challenges for deploying image recognition; therefore, to make image recognition perform on mobile devices, we simplify the algorithm and leverage iOS frameworks to reduce computation time [15]. The detailed algorithm of image recognition on mobile is illustrated as follows.

- To accelerate image feature extraction, we replace SIFT feature extractor [15] with Harris corner detector [16]. Although the features detected be Harris corner detector are not scale-invariant, we use a bounding box in the UI to guide field workers in normalizing the asset size. Furthermore, the Harris corner detector can be modified into pixel-wise operations, and then, we can easily realize it with OpenGL Shading Language[4], which automatically accelerates the processing by mobile device's GPU.

[2]www.larrylai.tw/mobilethermal

[3]https://developer.apple.com/library/mac/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_intro/ci_intro.html

[4]https://en.wikipedia.org/wiki/GLSL



(a)QRcode and Camera buttons to enable image recognition. (b)Image recognition results with visible evidences.

Figure 5: The APP UI for image recognition and visible image analysis. A pop-up UI shows the visible evidences to the asset condition.

- After feature detection, we need to build image feature sets via SIFT descriptor; however, the SIFT descriptor takes lots of computations, especially for trigonometric functions and square root functions. To reduce computations, we adopt a look-up-table(LUT) mechanism, which stores all results of complicated function in building SIFT descriptors. Note that the designed LUT preserves the precision and improves processing speed.
- Depending on applications, there are usually tens of thousands of training images in the database, and it results in feature matching takes major computation time in whole process. To accelerate feature matching, image color histogram is used as one of classifiers to reduce matching candidates [17]. By adopting the architecture of graph traversal, we can dramatically decrease the feature matching [18]. Note that all of image feature sets are stored with CoreData[5], the iOS SQL framework, which provides an efficient data access.

After featuring matching, we can find the matching pairs of feature points between the captured image and training images in the database.

B. Asset Condition Check and Operation Inspection

With the perspective transform, we are able to get the geometric mapping between the captured image and training images. Figure 5(b) shows an example of asset condition check, which exactly indicates the power light on the cable is abnormal. To implement the function of condition check, we build training images for each asset with different capturing perspectives and manually label checkup points on each training image. Note that the perspective transform can map those checkup points on training images to the captured image. Next, the image processing for condition check is conducted on the captured image.

For the case in Figure 5(b), it analyzed the captured image and found the cable's power signal turned off. By comparing to the asset conditions in training images, the cable power signal should flash in green as a normal condition. Therefore, the pop-up UI shows an image for the asset under normal condition with green flashing light. Note that the criteria for checking asset conditions are case-dependent. The asset producer needs to define the checkup points, and then the additional analysis for condition check should be cascaded to our image recognition algorithm.

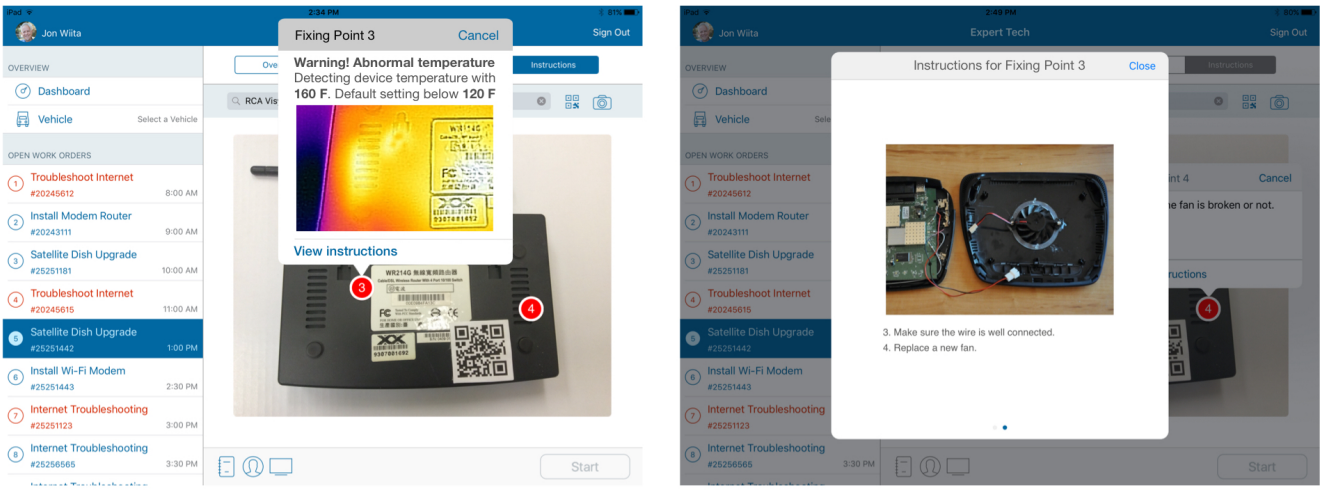
Once the asset model is recognized, the asset operation temperature in default value can look up in local database or pull down from cloud server. By inspecting the temperature distribution, the APP can immediately identify if the asset is currently running under normal condition. Figure 6(a) shows a pop-up UI will come up when the APP detects the cable temperature higher than the default settings. The pop-up UI shows the recognition results and display thermal evidences, which can help field workers quickly figure out this issue. With touching the View Instructions in the bottom of the pop-up UI, the instruction manual as shown in Figure 6(b) will pop over to guide field workers to fixing the issue step-by-step. A demo video is available on the website.[6]

VI. Conclusion

By integrating thermal and visual cameras on the mobile device, novel and practical applications are presented for event detection. To eliminate the influence of misaligned image pairs, we proposed the visible-visible and

[5]https://developer.apple.com/library/prerelease/mac/documentation/Cocoa/Reference/CoreData_ObjC/index.html

[6]www.larrylai.tw/mobilethermal



(a)Image recognition results and thermal evidences.

(b)Instructions for fixing the issue.

Figure 6: The APP UI for image recognition and thermal image analysis. A pop-up UI shows the thermal evidences to the asset in operation and exhibits the instructions to fix this issue.

thermal-visible alignment to integrate multi-modality imaging sources; therefore, we can achieve more precise event detection. By analyzing the heat pattern, the proposed APP can identify four heating sources and help users inspect their house heating system; on the other hand, with applying image recognition, the proposed APP can help field workers identify the asset condition and guide them to solve their issues.

REFERENCES

[1] R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," *Machine Vision and Applications*, vol. 25, no. 1, pp. 245–262, Nov. 2013.

[2] J. R. M.-D. Dios and A. Ollero, "Automatic Detection of Windows Thermal Heat Losses in Buildings Using UAVs," in *2006 World Automation Congress*. IEEE, jul 2006, pp. 1–6.

[3] "Diagnosis of Sheet Metal Stamping Processes Based on 3-D Thermal Energy Distribution," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 22–30, jan 2007.

[4] Fengliang Xu and K. Fujimura, "Pedestrian detection and tracking with night vision," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 1. IEEE, 2002, pp. 21–30.

[5] S. G. Kong, J. Heo, B. R. Abidi, J. Paik, and M. A. Abidi, "Recent advances in visual and infrared face recognitionâĂŤa review," *Computer Vision and Image Understanding*, vol. 97, no. 1, pp. 103–135, jan 2005.

[6] "A Non-invasive Method for Measuring Blood Flow Rate in Superficial Veins from a Single Thermal Image," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, jun 2013, pp. 354–359.

[7] C. O'Conaire, N. O'Connor, E. Cooke, and A. Smeaton, "Comparison of Fusion Methods for Thermo-Visual Surveillance Tracking," in *2006 9th International Conference on Information Fusion*. IEEE, jul 2006, pp. 1–7.

[8] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[9] M. Trivedi, "Multiperspective Thermal IR and Video Arrays for 3D Body Tracking and Driver Activity Analysis," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, vol. 3. IEEE, 2005, pp. 3–3.

[10] S. Sonn, G.-A. Bilodeau, and P. Galinier, "Fast and Accurate Registration of Visible and Infrared Videos," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, jun 2013, pp. 308–313.

[11] "Robust multi-sensor image alignment," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. Narosa Publishing House, 1998, pp. 959–966.

[12] F. Wu, B. Wang, X. Yi, M. Li, J. Hao, H. Qin, and H. Zhou, "Visible and infrared image registration based on visual salient features," *Journal of Electronic Imaging*, vol. 24, no. 5, p. 053017, sep 2015.

[13] J. Zaragoza, T.-J. Chin, Q.-H. Tran, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving dlt," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 7, pp. 1285–1298, 2014.

[14] C.-C. Lin, S. U. Pankanti, K. N. Ramamurthy, and A. Y. Aravkin, "Adaptive as-natural-as-possible image stitching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1155–1163.

[15] D. Lowe, "Distinctive image features from scale-invariant keypoint," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, pp. 63–86, 2004.

[17] J.-H. Lai and S.-Y. Chien, "Baseball and tennis video annotation with temporal structure decomposition," in *Proceedings of IEEE International Workshop on Multimedia Signal Processing, MMSP 2008*, 2008, pp. 676–679.

[18] Y. Xia, J.-H. Lai, L. Nai, and C.-Y. Lin, "Concurrent image query using local random walk with restart on large scale graphs," in *Proceedings of IEEE International Conference on Multimedia and Expo, ICME 2014*, 2014, pp. 1–6.