


Article

# Visual Saliency Detection for Over-Temperature Regions in 3D Space via Dual-Source Images

Dawei Gong , Zhiheng He \*, Xiaolong Ye and Ziyun Fang

School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; pzhzhx@126.com (D.G.); yxl198868@foxmail.com (X.Y.); f459534668@126.com (Z.F.)

\* Correspondence: 201821040223@std.uestc.edu.cn

Received: 13 May 2020; Accepted: 14 June 2020; Published: 17 June 2020



**Abstract:** To allow mobile robots to visually observe the temperature of equipment in complex industrial environments and work on temperature anomalies in time, it is necessary to accurately find the coordinates of temperature anomalies and obtain information on the surrounding obstacles. This paper proposes a visual saliency detection method for hypertemperature in three-dimensional space through dual-source images. The key novelty of this method is that it can achieve accurate salient object detection without relying on high-performance hardware equipment. First, the redundant point clouds are removed through adaptive sampling to reduce the computational memory. Second, the original images are merged with infrared images and the dense point clouds are surface-mapped to visually display the temperature of the reconstructed surface and use infrared imaging characteristics to detect the plane coordinates of temperature anomalies. Finally, transformation mapping is coordinated according to the pose relationship to obtain the spatial position. Experimental results show that this method not only displays the temperature of the device directly but also accurately obtains the spatial coordinates of the heat source without relying on a high-performance computing platform.

**Keywords:** component; robot work; object detection; adaptive sampling; surface mapping; coordinate mapping

## 1. Introduction

In the path planning of mobile robots, it is common to construct a map using the dynamic vision fusion of cameras and multi-sensors [1–3]. In a specific industrial environment, the robot needs to monitor the temperature of the equipment and work in the area of abnormal temperature points. The existing neural network control method shows high stability [4–6], but it also needs to accurately find the location of the abnormal temperature's point. Traditionally, using a visible-light binocular camera to reconstruct the target is not possible, because it cannot accurately operate on the abnormal temperature point area [7–10]. At present, the most commonly used temperature detection methods use sensor contact measurements [11–13]. However, there are installation and use problems in engineering applications, so non-contact space measurements can be used to solve the installation problem. Visual target detection can solve this problem.

In the field of target detection, deep learning is a commonly used technology. At present, in 2D target detection, many methods of optimizing the structure of deep convolutional neural networks improve the accuracy of target detection [14–16], such as fully convolutional networks (FCN), progressive fusion [17], multi-scale depth encoding [18], and data set balancing and smearing methods [19,20]. In mobile robot navigation, precise positioning of the target often requires obtaining spatial coordinates. The depth camera can be used to obtain depth information for 2.5D target positioning. The deep network also plays an important role in this field. The variational autoencoder [21,22], the adaptive window and weight matching algorithm [23], the deep purifier,

and the feature learning unit greatly improve the accuracy of detection. However, deep learning requires more sophisticated hardware and relies on a large number of training samples [24–27].

With the development of 3D reconstruction technology, the application of 3D reconstruction technology in real life has become extensive, attracting the attention of many experts and scholars [28,29]. Commonly used 3D visual reconstruction methods include feature extraction and matching, sparse point cloud reconstruction, camera pose solution, dense point cloud reconstruction, and surface reconstruction [30–33]. Through the research of different experts and scholars, related technologies such as feature matching, depth calculation, and mesh texture reconstruction have made great breakthroughs, which have resulted in a higher degree of reduction in visual 3D reconstruction [34–36].

The method proposed in this paper mainly uses ordinary and infrared cameras to take pictures of targets and then sparse point cloud reconstruction through ordinary pictures to obtain the pose of the camera when imaging. Then, image fusion is performed on ordinary pictures and infrared pictures. The original camera's internal and external parameters do not change. The original image can be replaced with the fusion image to surface-map the dense point cloud in order to generate a three-dimensional surface. In addition, a three-dimensional reconstruction target that visually displays the surface temperature is obtained [37–39]. This paper uses an adaptive random sampling algorithm to obtain the main texture features, remove redundant point clouds, and finally, use the depth confidence to filter the wrong point clouds [40–42].

To reduce the calculation cost and dependence on training samples, this paper mainly uses the characteristics of infrared images to detect the center coordinates of the heat source. First, the infrared images are pre-processed by channel extraction and image segmentation. Then, the position of the two-dimensional plane temperature abnormal points is detected. Finally, the coordinate transformation is calculated based on the camera's imaging pose relationship in order to calculate its spatial coordinates [43–45]. Therefore, it is possible to use the reconstructed target as an obstacle to plan the movement path of the robot and to work on the temperature abnormal point area according to the obtained spatial coordinate information. The schematic diagram is shown in Figure 1. The robot rotates around the target center once to reconstruct a complete target and quickly finds the center position of the heat source that needs to be operated using the above method.

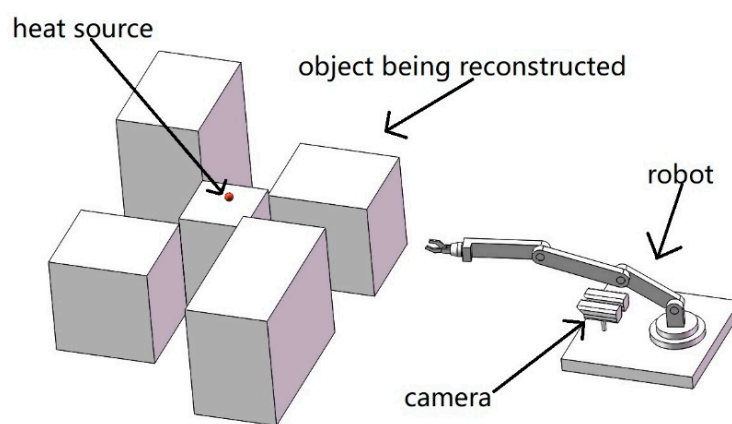


Figure 1. Robot operation diagram.

## 2. Materials and Methods

The process of sparse point cloud reconstruction is as follows: Feature extraction, feature matching, elimination of mismatched pairs, 3D point cloud initialization, and camera pose calculation. Among these steps, the image mismatch elimination and pose solution have a great impact on the sparse point cloud reconstruction effect. The text uses the random sample consensus (RANSAC) algorithm to remove false matches and the beam adjustment method to recalculate the camera pose. The visible light camera used in this article is a 200W pixel POE DS-2CD3T25-I3 with a focal length of four millimeters. The device was manufactured by HIKVISION Company in Hangzhou, China.

## 2.1. Reconstruction of the Sparse Point Cloud to Obtain the Camera Attitude

### 2.1.1. Use of the Scale-Invariant Feature Transform (SIFT) Algorithm to Find Feature Points

The process of sparse point cloud reconstruction includes feature extraction, feature matching, elimination of mismatched pairs, 3D point cloud initialization, and camera pose calculation. Among these steps, the image mismatch elimination and pose solution have the greatest impact on the sparse point cloud reconstruction effect. The text uses the RANSAC algorithm to remove false matches and the beam adjustment method to recalculate the camera pose.

To realize 3D reconstruction, the feature points of the picture first need to be extracted. The scale-invariant feature transform (SIFT) algorithm is a computer vision algorithm that is used to detect and describe local features of images, find extreme points in the interscale, and extract their position, scale, and rotation invariants. It is divided into the following four steps:

- Multi-scale spatial extreme point detection: This searches image locations on all scales and uses Gaussian differential functions to identify potential rotation invariants and scale candidate points.
- Accurate positioning of key points: After determining candidate positions, a high-precision model is fitted to determine the scale and position. The stability of key points is used as the basis for selection.
- Calculation of the main direction of key points: Based on the local gradient direction of the image, each key point obtains one or more directions. In the future, the image processing will be transformed relative to the key-point scale, direction, and position to ensure the invariance of the transformation.
- Descriptor construction: In the field of key points, the direction of local gradients is measured according to the scale selected above, and these gradients are transformed into another representation.

The effect of feature point extraction is shown in Figure 2.



**Figure 2.** Scale-invariant feature transform (SIFT) feature point extraction results.

This shows the reconstruction of a potted plant on a 3.0 GHz CPU desktop computer, selecting 30 consecutive shots at a resolution size of  $4000 \times 3000$  ppi. The maximum calculation memory required during the reconstruction process, before using the adaptive sampling algorithm, is 5.3 GB. After adapting to the sampling algorithm, it is 3.2 GB, which proves that the algorithm effectively reduces the memory required for calculation.

### 2.1.2. Error Matching Elimination Based on the RANSAC Algorithm

There will be matching errors after feature matching. RANSAC is a commonly used error elimination algorithm. The grid-based motion (GMS) [46] algorithm, recently proposed by scholars, can match features in a short time and is very robust. It can remove wrong matches to a certain extent.

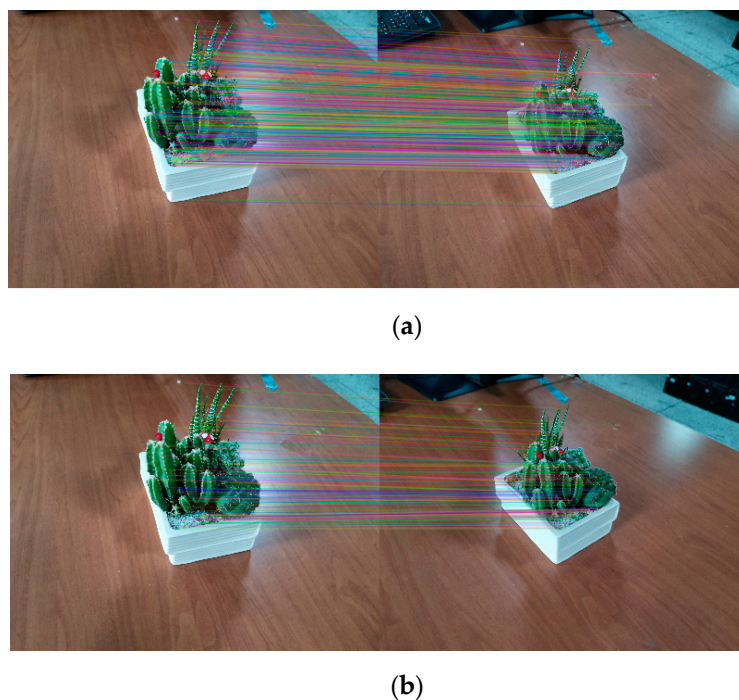
However, the original author notes that the GMS algorithm is suitable for supplementing the RANSAC algorithm but not replacing it. Therefore, this article mainly uses the RANSAC algorithm to eliminate wrong feature matching. The algorithm works by using Equation (2) as the cost function to iteratively update the sample set.

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

$$\sum_{i=1}^n \left( x_i' \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 + \left( y_i' \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 \quad (2)$$

In the above formula,  $(x, y)$  represents the corner position of the target image,  $(x', y')$  is the corner position of the scene image,  $s$  is the scale parameter, and  $H$  is a  $3 \times 3$  homography matrix.

Error matching elimination based on RANSAC is shown in Figure 3.



**Figure 3.** Comparison of algorithm effects, where (a) is the original matching effect diagram and (b) is the error matching elimination diagram of the random sample consensus (RANSAC) algorithm.

### 2.1.3. The Position Pose of the Phase Machine Is Solved by the Beam Adjustment Method

After the image alignment, the 3D point cloud and camera pose can be obtained. However, there will be interference noise when calculating the position and the 3D point, and there will be significant error in the subsequent calculation. Therefore, bundle adjustment is used to reduce the error [9], and the P matrix and F matrix of each picture after correction can be obtained. The reprojection error is defined as:

$$E = \sum_j \rho_j(\pi(P_C, X_k) - x_j^2) \quad (3)$$

where  $\pi$  is a projection matrix from three-dimensional to two-dimensional,  $\rho_j$  is a kernel function, and  $\pi(P_C, X_k) - x_j^2$  is a cost function. Figure 4 shows the sparse point cloud obtained after the bundle adjustment (BA) algorithm is used to solve the position pose. The green dot is the posture of the solved camera.

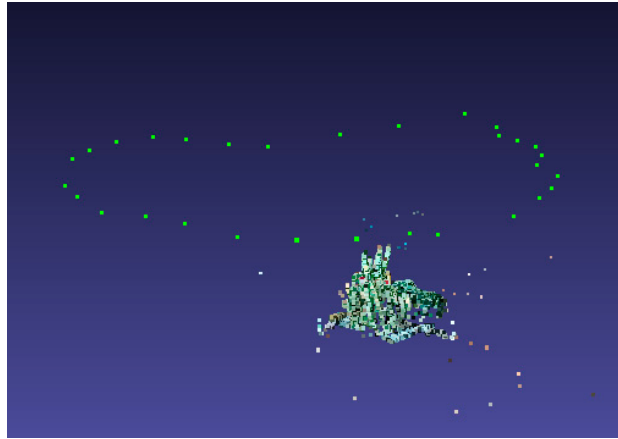


Figure 4. Schematic diagram after the camera pose calculation.

## 2.2. Three-Dimensional Surface Generation

### 2.2.1. Adaptive Random Sampling

- A pixel point  $\hat{x}_i$  is randomly selected from the obtained point cloud image.  $D_i(x_i)$  is the depth value of the pixel point and is inversely mapped into the three-dimensional space according to Equation (4). The tangent plane  $P(\hat{x}_i)$  is obtained according to the normal direction.  $K_i$  is the camera internal parameter,  $R_i$  is the rotation matrix, and  $T_i$  is the translation vector.

$$P(\hat{x}_i) = R_i^T (K_i^{-1} D_i(\hat{x}_i) [\hat{x}_i 1]^T - T_i) \quad (4)$$

Specific steps are as follows:

- Expand outwards with  $\hat{x}_i$  as the center, expand the radius  $r$  one pixel at a time, and calculate the three-dimensional coordinates  $P(x_{i'})$  of each pixel  $x_{i'}$  in the expansion range.
- Calculate the distance  $d_i$  of each pixel  $x_{i'}$  to the tangent plane within the current expansion range, and set the threshold size as  $t_d$ . If  $d_i \leq t_d$ , the pixel point can be considered to be in the smooth area, and the point can be removed.
- When the expansion radius  $r$  is larger than the maximum expansion radius  $r_{max}$ , or a point cloud of a certain proportion of  $p_i$  in the expansion range is removed, the expansion stops.  $r_{max}$  and  $p_i$  are tunable parameters. They can be determined according to the point cloud redundancy. During debugging, it is found that there are still many redundant point clouds after culling.  $r_{max}$  can be increased and  $p_i$  can be decreased. If the point cloud is over-eliminated, the parameter adjustment method is reversed.
- Then, randomly select a pixel point and repeat the above steps until all the sampling points in the current 3D point cloud image are sampled.

### 2.2.2. Deep Confidence Removes the Cloud of Error Points

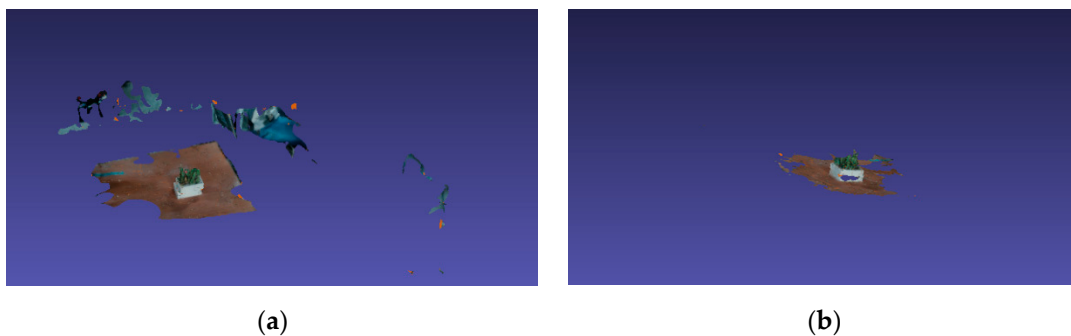
$$E_d(P(\hat{x}_i)) = \frac{\sum_{i' \in N(i)} \|D_i(\hat{x}_i) - D_i(\hat{x}_{i \rightarrow i'})\|^2}{|N(i)|} \quad (5)$$

The above formula is the depth value estimation of the point cloud, i.e., the larger the estimated value, the smaller the error value and the higher the reliability. Among these values,  $E_d(P(\hat{x}_i))$  is the depth value estimation of two adjacent frames,  $\hat{x}_{i \rightarrow i'}$  represents the projection point of the  $i'$  pixel projected by the current pixel, and  $N(i)$  represents the number of frames taken. The specific steps are as follows:

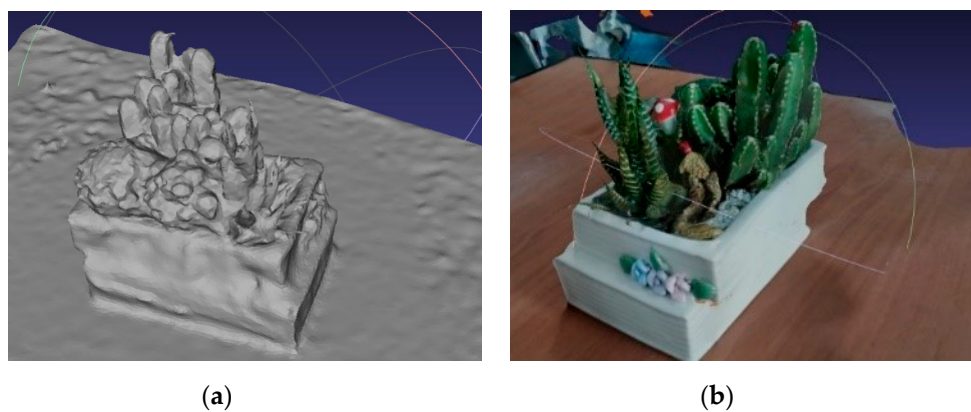


- The point cloud for the current frame  $k$  is sorted from high to low according to the estimated value, and the confidence threshold  $\varepsilon_d$  is set, starting from the point where the estimated value is the smallest. If  $E_d(P(\hat{x}_i)) < \varepsilon_d$ , the point is eliminated, the calculation continues until  $E_d(P(\hat{x}_i)) > \varepsilon_d$  stops, and the remaining point clouds are stored in the sequence  $S_k$ . Then, the same calculation is performed on the next frame point cloud image until the point cloud image is calculated and the sequence set  $S = \{S_k | k = 1, \dots, n\}$  is obtained.
- Starting from the  $k$  frame depth map, all three-dimensional points  $\hat{x}_i$  are mapped to  $\hat{x}_{i+1}$  on the  $k + 1$  frame. Compare the estimated values of the two points, the smaller three-dimensional coordinates of the larger estimated points of the estimated values, and so on, until all depth maps are completed.
- The three-dimensional sampling points of all depth maps are intersected to obtain the final three-dimensional point cloud image. Then, perform the mesh reconstruction and mesh texture generation on the filtered dense point cloud. The effect before and after filtering is shown in Figure 5.

The reconstruction details are shown in Figure 6.



**Figure 5.** Effect before and after filtering, where (a) is the picture before removing the redundant point cloud, and (b) is the picture after removing the redundant point cloud.



**Figure 6.** Surface reconstruction details, where (a) is the picture before the sticker and (b) is the picture after the sticker.

### 2.3. Image Fusion

After reconstructing the sparse point cloud, the camera parameters are obtained. The original image can be corrected for distortion. The infrared image can be calibrated and corrected by itself. The image registration error is shown in the following formula:

$$\sigma_x = \frac{f \cdot d_x}{l_{pix}} \left( \frac{1}{D_{target}} - \frac{1}{D_{optimal}} \right) \quad (6)$$

where  $f$  is the focal length,  $l_{pix}$  is the pixel size, and  $d_x$  is the baseline length.  $D_{optimal}$  is the target distance, and the alignment error of the image is zero. Only objects that are far away from the camera will be precisely aligned.

### 2.3.1. Calculate Scale Factor

As the focal length and resolution of infrared and visible images are different, the imaging size of objects in space from the two camera types is not consistent. At the same time, the optical center of the hardware systems of the two camera types deviates in the Y direction. Therefore, it is not easy to scale the image by focal length.

The method adopted in this paper calculates the pixel difference between two corner points in infrared and visible images by using the checkerboard calibration board to obtain the image scale.

$$scale = \frac{infrared_n - infrared_{n-1}}{visible_n - visible_{n-1}} \quad (7)$$

It is assumed that the checkerboard calibration board corner with  $k$  line,  $l$  column, namely  $kl$ , is accumulated.  $n$  is the corner number on the checkerboard, the upper left corner is minimum 1, and the lower right corner is the maximum  $kl$ . The values increase from left to right, and from top to bottom,  $infrared_n$  is the x or y coordinate of the corner  $n$  on the infrared image, and  $visible_n$  is the X or Y coordinate of the corner  $n$  on the visible light image.

### 2.3.2. Relative Offset of the Image

The factor scale is used to realize the unification of space objects in infrared and visible images. Then, the same corner point on the checkerboard is selected to calculate the relative offset of infrared and visible images.

$$X_{diff} = infrared_x - visible_x \quad (8)$$

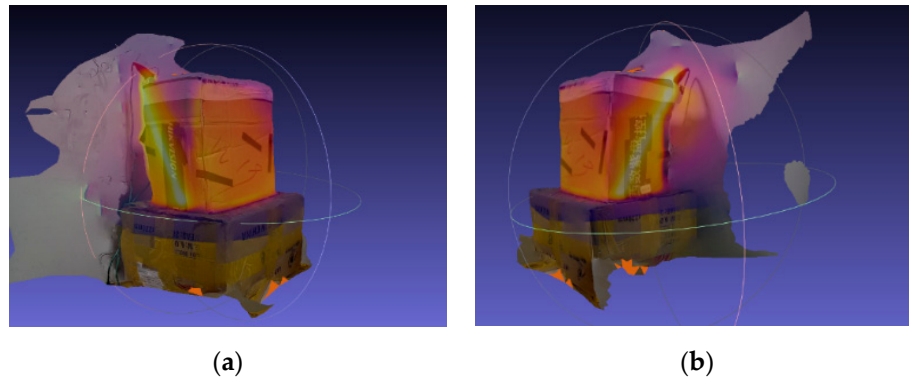
$$Y_{diff} = infrared_y - visible_y \quad (9)$$

where  $X_{diff}$  and  $Y_{diff}$  are the offsets required for each pixel in the infrared image. The RGB color model is a color standard in the industrial world. It obtains various colors by changing the three-color channels of red (R), green (G), and blue (B) and by superimposing each on others. After the completion of each pixel offset, the values of the three channels of RGB of the infrared and visible pixel pairs in the same coordinate can be fused, and the fusion effect is shown in Figure 7. Figure 7 is the heating plate placed in the carton. An infrared camera with a resolution of  $384 \times 288$  ppi is used. The infrared camera and visible light camera take pictures at the same time.



**Figure 7.** 2D fusion picture, where picture (a) is the picture before fusion and picture (b) is the picture after fusion.

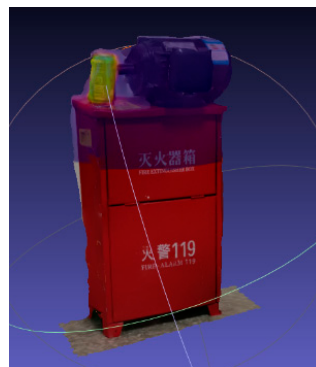
The camera pose is calculated based on the reconstructed sparse point cloud, and all the fused pictures are surface-reconstructed. The 3D reconstruction effect of the temperature display is shown in Figure 8.



**Figure 8.** Schematic representation of temperature surface reconstruction, where (a) is reconstructed position 1 and (b) is reconstructed position 2.

#### 2.4. 3D Target Detection

As shown in Figure 9, in this experiment, a high-temperature bottle is used as the temperature abnormal region of the overall device, and its spatial coordinates need to be calculated.



**Figure 9.** Detection target.

##### 2.4.1. Target Detection of the Heat Source

In the infrared picture, the pixel temperature generated by the detection is proportional to the R channel value, so the image can be preprocessed first. The R channel value size of the original image is extracted, and all pixels are sorted according to the R value. However, noise in the image is unavoidable and will interfere with the sorting results. To avoid incorrect sorting, the extracted image can be cut and divided into sub-regions. The size of the region can be determined according to the input original image size. Then, the average value of the R channel in each area is calculated, and the area is sorted according to the average value to obtain the R channel size set of each area  $R_{agg} = \{R_1, R_2, R_3, \dots, R_n\}$ , assuming  $R_{max}$  is its maximum value.

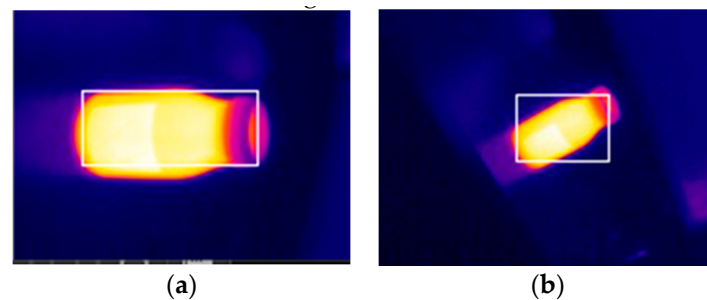
After the infrared image preprocessing is complete, the R channel value of each small area can be obtained. To allow the detection frame to be adaptively scaled, the size of the heat source needs to be calculated, so small squares (that meet the conditions) can be calculated and recorded for each small area location. The criteria are:

$$R_i > k * R_{max} \quad (10)$$

$$size_r = size_p * p_r \quad (11)$$



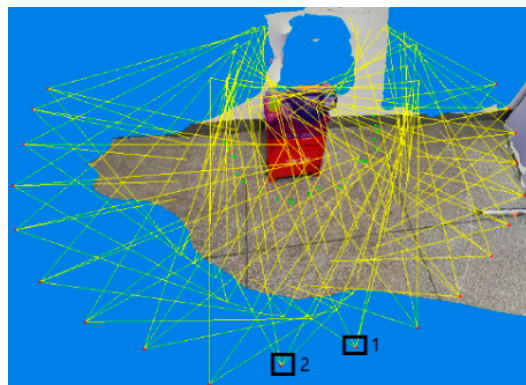
Among them,  $R_i$  represents the value of the R channel region, and  $k$  is a proportionality coefficient that needs to be adjusted according to specific conditions. After calculating the situation of each sub-region, each region can be assessed, in order from left to right and top to bottom. Each sub-region is set to be square. The size of each sub-region  $size_r$  can be determined according to Equation (11), where  $size_p$  is the size of the infrared image used for detection, the proportion of  $p_r$  sub-regions, and  $p_r$  is an adjustable parameter. If four of the eight regions around the area meet the conditions, that area is a sub-area within the heat source range, and the position coordinate is recorded and evaluated. Finally, the size of the heat source border can be obtained from the coordinate position. The effect is shown in Figure 10.



**Figure 10.** Heat source detection, where (a) is position 1 and (b) is position 2.

#### 2.4.2. Coordinate Transformation Mapping in 3D Space

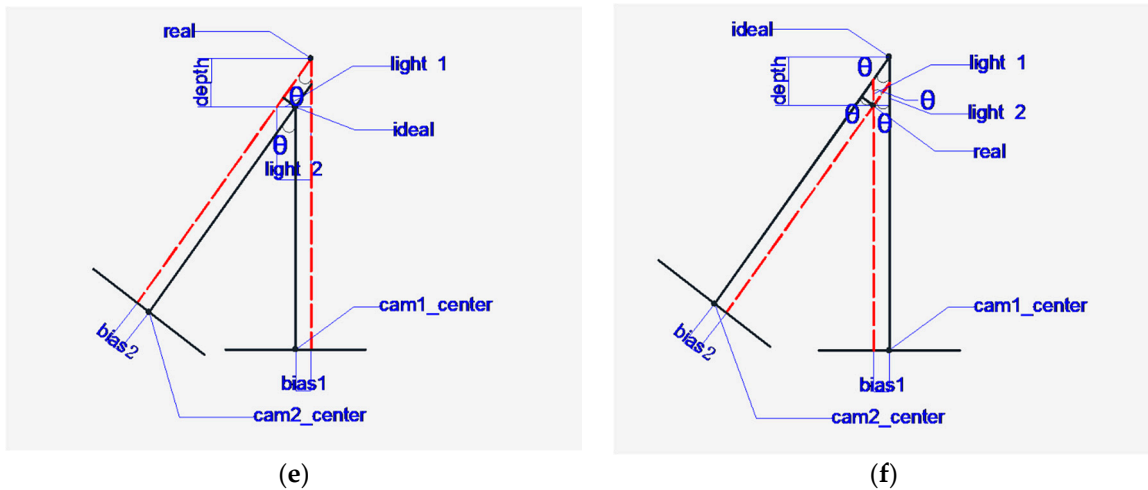
After the detection of the heat source target, the coordinates of the heat source center in each infrared picture can be obtained; because the shoot is a head-up relationship, the horizontal deviation and the height deviation can also be obtained. The steps are as follows: Take the center of the first picture as the center point of the space and choose another angle during the shoot as the second position. As shown by two positions in Figure 11, calculate the deviation between the actual heat source and the ideal heat source. The following situations can occur:



**Figure 11.** Camera imaging pose.

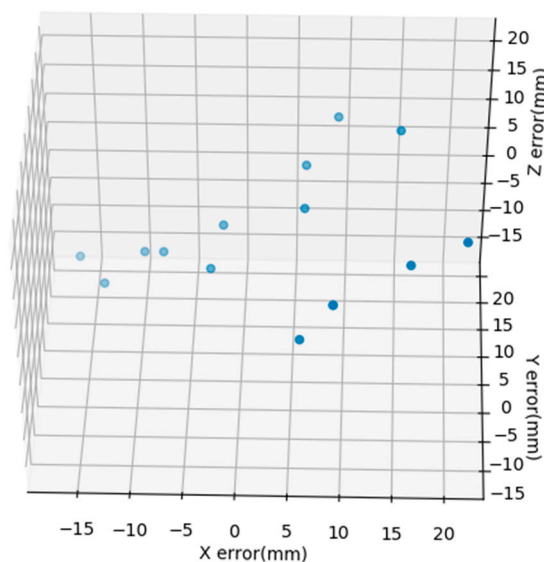
Figure 12 is a top view of various situations. Taking Figure 12a as an example,  $cam1\_center$  and  $cam2\_center$  are the imaging center points of the camera at two positions, “ideal” is the most central position of the heat source processing experiment and is the intersection of the two imaging centers, and “real” is the actual position of the heat source. When the heat source reaches the imaging plane, the distance from the center of the camera is  $bias_1$  and  $bias_2$ , where  $\alpha$  is the angle of rotation of the





**Figure 12.** Schematic diagram of the ideal position and the actual position, where (a–f) corresponds to the situation of six actual heat sources relative to the ideal heat source.

Although detection speed has been greatly improved by the enhanced convolutional neural network structure, it still cannot provide high-precision results, and relies on high-performance GPUs. The method in this paper conducted 15 experiments, only running on a 3.0 GHz desktop computer, using the thermos randomly placed in the above figure as a simulated heat source. The camera is 10 m away from the ideal heat source. The error values were obtained from the actual measured coordinates and calculated coordinates. The error results of the experiment are shown in Figure 13. It can be seen from the experimental results that the error value is within  $\pm 20$  mm, with high accuracy, and the calculation speed is 20 ms, which meets the detection requirements of industrial equipment.



**Figure 13.** Camera imaging pose.

### 3. Conclusions

The experimental results demonstrate that the method proposed in this paper can fuse target surface temperature information captured by infrared cameras into a three-dimensional point cloud while ensuring the accuracy and speed of the reconstruction and that the reconstructed object can intuitively display its surface temperature. The spatial coordinates of the heat source are calculated using the spatial transformation mapping relationship of the infrared picture. The experimental results

demonstrate that the algorithm is highly accurate and meets the requirements of robot navigation and positioning.

#### 4. Patents

A 3D reconstruction method based on point cloud optimization sampling; a 3D surface temperature display method based on infrared and visible image fusion is presented; the invention relates to a method for detecting the heat source center in three-dimensional space.

**Author Contributions:** D.G. is responsible for designing adaptive random sampling, image fusion and three-dimensional target space positioning algorithms. Z.H. and X.Y. are responsible for the algorithm and software design, as well as experimental debugging. Z.F. is responsible for literature research and paper writing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (61603076) and the National Defense Pre-Research Foundation of China (1126170104A, 1126180204B, 1126190402A, 1126190508A).

**Conflicts of Interest:** The authors declare no conflicts of interest.

#### References

1. Fan, Y.; Lv, X.; Lin, J.; Ma, J.; Zhang, G.; Zhang, L.; Correction: Zhang, G. Autonomous Operation Method of Multi-DOF Robotic Arm Based on Binocular Vision. *Appl. Sci.* **2019**, *9*, 5294. [[CrossRef](#)]
2. Kassir, M.M.; Palhang, M.; Ahmadzadeh, M.R. Qualitative vision-based navigation based on sloped funnel lane concept. *Intell. Serv. Robot.* **2018**, *13*, 235–250. [[CrossRef](#)]
3. Li, C.; Yu, L.; Fei, S. Large-Scale, Real-Time 3D Scene Reconstruction Using Visual and IMU Sensors. *IEEE Sens. J.* **2020**, *20*, 5597–5605. [[CrossRef](#)]
4. Yang, C.; Jiang, Y.; He, W.; Na, J.; Li, Z.; Xu, B. Adaptive Parameter Estimation and Control Design for Robot Manipulators with Finite-Time Convergence. *IEEE Trans. Ind. Electron.* **2018**, *65*, 8112–8123. [[CrossRef](#)]
5. Yang, C.; Peng, G.; Cheng, L.; Na, J.; Li, Z. Force Sensorless Admittance Control for Teleoperation of Uncertain Robot Manipulator Using Neural Networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**. [[CrossRef](#)]
6. Peng, G.; Yang, C.; He, W.; Chen, C.P. Force Sensorless Admittance Control with Neural Learning for Robots with Actuator Saturation. *IEEE Trans. Ind. Electron.* **2020**, *67*, 3138–3148. [[CrossRef](#)]
7. Mao, C.; Li, S.; Chen, Z.; Zhang, X.; Li, C. Robust kinematic calibration for improving collaboration accuracy of dual-arm manipulators with experimental validation. *Measurement* **2020**, *155*, 107524. [[CrossRef](#)]
8. Xu, L.; Feng, C.; Kamat, V.R.; Menassa, C.C. A scene-adaptive descriptor for visual SLAM-based locating applications in built environments. *Autom. Constr.* **2020**, *112*, 103067. [[CrossRef](#)]
9. Yang, C.; Wu, H.; Li, Z.; He, W.; Wang, N.; Su, C.Y. Mind Control of a Robotic Arm with Visual Fusion Technology. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3822–3830. [[CrossRef](#)]
10. Lin, H.; Zhang, T.; Chen, Z.; Song, H.; Yang, C. Adaptive Fuzzy Gaussian Mixture Models for Shape Approximation in Robot Grasping. *Int. J. Fuzzy Syst.* **2019**, *21*, 1026–1037. [[CrossRef](#)]
11. Shen, S. Accurate Multiple View 3D Reconstruction Using Patch-Based Stereo for Large-Scale Scenes. *IEEE Trans. Image Process.* **2013**, *22*, 1901–1914. [[CrossRef](#)] [[PubMed](#)]
12. Wang, L.; Li, R.; Sun, J.; Liu, X.; Zhao, L.; Seah, H.S.; Quah, C.K.; Tandianus, B. Multi-View Fusion-Based 3D Object Detection for Robot Indoor Scene Perception. *Sensors* **2019**, *19*, 4092. [[CrossRef](#)] [[PubMed](#)]
13. Yamazaki, T.; Sugimura, D.; Hamamoto, T. Discovering Correspondence Among Image Sets with Projection View Preservation For 3D Object Detection in Point Clouds. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 3111–3115.
14. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
15. Fu, K.; Zhao, Q.; Gu, I.Y.; Yang, J. Deepside: A general deep framework for salient object detection. *Neurocomputing* **2019**, *356*, 69–82. [[CrossRef](#)]
16. Wang, W.; Shen, J. Deep Visual Attention Prediction. *IEEE Trans. Image Process.* **2018**, *27*, 2368–2378. [[CrossRef](#)]

17. Tang, Y.; Zou, W.; Hua, Y.; Jin, Z.; Li, X. Video salient object detection via spatiotemporal attention neural networks. *Neurocomputing* **2020**, *377*, 27–37. [[CrossRef](#)]
18. Zhao, J.X.; Liu, J.J.; Fan, D.P.; Cao, Y.; Yang, J.; Cheng, M.M. EGNet: Edge Guidance Network for Salient Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019.
19. Ren, Q.; Hu, R. Multi-scale deep encoder-decoder network for salient object detection. *Neurocomputing* **2018**, *316*, 95–104. [[CrossRef](#)]
20. Fan, D.P.; Cheng, M.M.; Liu, J.J.; Gao, S.H.; Hou, Q.; Borji, A. Salient Objects in Clutter: Bringing Salient Object Detection to the Foreground. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
21. Zhang, J.; Yu, X.; Li, A.; Song, P.; Liu, B.; Dai, Y. Weakly-Supervised Salient Object Detection via Scribble Annotations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020.
22. Zhang, J.; Fan, D.P.; Dai, Y.; Anwar, S.; Saleh, F.S.; Zhang, T.; Barnes, N. UC-Net: Uncertainty Inspired RGB-D Saliency Detection via Conditional Variational Autoencoders. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020.
23. Wang, Z.; Liu, J. Research on flame location based on adaptive window and weight stereo matching algorithm. *Multimed. Tools Appl.* **2020**, *79*, 7875–7887.
24. Fan, D.P.; Ji, G.P.; Sun, G.; Cheng, M.M.; Shen, J.; Shao, L. Rethinking RGB-D Salient Object Detection: Models, Datasets, and Large-Scale Benchmarks. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**. [[CrossRef](#)]
25. Hu, Q.H.; Huang, Q.X.; Mao, Y.; Liu, X.L.; Tan, F.R.; Wang, Y.Y.; Yin, Q.; Wu, X.M.; Wang, H.Q. A near-infrared large Stokes shift probe based enhanced ICT strategy for F- detection in real samples and cell imaging. *Tetrahedron* **2019**, *75*, 130762. [[CrossRef](#)]
26. Song, W.T.; Hu, Y.; Kuang, D.B.; Gong, C.L.; Zhang, W.Q.; Huang, S. Detection of ship targets based on CFAR-DCRF in single infrared remote sensing images. *J. Infrared Millim. Waves* **2019**, *38*, 520–527.
27. Zhao, X.; Wang, W.; Ni, X.; Chu, X.; Li, Y.F.; Lu, C. Utilising near-infrared hyperspectral imaging to detect low-level peanut powder contamination of whole wheat flour. *Biosyst. Eng.* **2019**, *184*, 55–68. [[CrossRef](#)]
28. Hruda, L.; Dvořák, J.; Váša, L. On evaluating consensus in RANSAC surface registration. *Comput. Graph. Forum* **2019**, *38*, 175–186. [[CrossRef](#)]
29. Qu, Y.; Huang, J.; Zhang, X. Rapid 3D Reconstruction for Image Sequence Acquired from UAV Camera. *Sensors* **2018**, *18*, 225.
30. Aldeeb, N.H.; Hellwich, O. 3D Reconstruction Under Weak Illumination Using Visibility-Enhanced LDR Imagery. *Adv. Comput. Vis.* **2020**, *1*, 515–534.
31. Xie, Q.H.; Zhang, X.W.; Cheng, S.Y.; Lv, W.G. 3D Reconstruction Method of Image Based on Digital Microscope. *Acta Microsc.* **2019**, *28*, 1289–1300.
32. Zhang, J.; Zhang, S.X.; Chen, X.X.; Jiang, B.; Wang, L.; Li, Y.Y.; Li, H.A. A Novel Medical 3D Reconstruction Based on 3D Scale-Invariant Feature Transform Descriptor and Quaternion-Iterative Closest Point Algorithm. *J. Med. Imaging Health Inf.* **2019**, *9*, 1361–1372. [[CrossRef](#)]
33. Zhang, K.; Yan, M.; Huang, T.; Zheng, J.; Li, Z. 3D reconstruction of complex spatial weld seam for autonomous welding by laser structured light scanning. *J. Manuf. Process.* **2019**, *39*, 200–207. [[CrossRef](#)]
34. Zhang, X.; Zhao, P.; Hu, Q.; Wang, H.; Ai, M.; Li, J. A 3D Reconstruction Pipeline of Urban Drainage Pipes Based on Multiview Image Matching Using Low-Cost Panoramic Video Cameras. *Water* **2019**, *11*, 2101. [[CrossRef](#)]
35. Zheng, Y.; Liu, J.; Liu, Z.; Wang, T.; Ahmad, R. A primitive-based 3D reconstruction method for remanufacturing. *Int. J. Adv. Manuf. Technol.* **2019**, *103*, 3667–3681. [[CrossRef](#)]
36. Zhu, C.; Yu, S.; Liu, C.; Jiang, P.; Shao, X.; He, X. Error estimation of 3D reconstruction in 3D digital image correlation. *Meas. Sci. Technol.* **2019**, *30*, 10. [[CrossRef](#)]
37. Kiyasu, S.; Hoshino, H.; Yano, K.; Fujimura, S. Measurement of the 3-D shape of specular polyhedrons using an M-array coded light source. *IEEE Trans. Instrum. Meas.* **1995**, *44*, 775–778. [[CrossRef](#)]
38. Pollefeys, M.; Nistér, D.; Frahm, J.M.; Akbarzadeh, A.; Mordohai, P.; Clipp, B.; Engels, C.; Gallup, D.; Kim, S.J.; Merrell, P.; et al. Detailed Real-Time Urban 3D Reconstruction from Video. *Int. J. Comput. Vis.* **2008**, *78*, 143–167. [[CrossRef](#)]



39. Furukawa, Y.; Ponce, J. Carved Visual Hulls for Image-Based Modeling. *Int. J. Comput. Vis.* **2009**, *81*, 53–67. [[CrossRef](#)]
40. Zhan, Y.; Hong, W.; Sun, W.; Liu, J. Flexible Multi-Positional Microsensors for Cryoablation Temperature Monitoring. *IEEE Electron Device Lett.* **2019**, *40*, 1674–1677. [[CrossRef](#)]
41. Zhou, H.; Zhou, Y.; Zhao, C.; Wang, F.; Liang, Z. Feedback Design of Temperature Control Measures for Concrete Dams based on Real-Time Temperature Monitoring and Construction Process Simulation. *KSCE J. Civ. Eng.* **2018**, *22*, 1584–1592. [[CrossRef](#)]
42. Zrelli, A. Simultaneous monitoring of temperature, pressure, and strain through Brillouin sensors and a hybrid BOTDA/FBG for disasters detection systems. *IET Commun.* **2019**, *13*, 3012–3019. [[CrossRef](#)]
43. Sun, H.; Meng, Z.H.; Du, X.X.; Ang, M.H. A 3D Convolutional Neural Network towards Real-time Amodal 3D Object Detection. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 8331–8338.
44. Shen, X.L.; Dou, Y.; Mills, S.; Eysers, D.M.; Feng, H.; Huang, Z. Distributed sparse bundle adjustment algorithm based on three-dimensional point partition and asynchronous communication. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 889–904. [[CrossRef](#)]
45. Snavely, N.; Seitz, S.; Szeliski, R. Photo tourism: Exploring photo collections in 3D. *ACM Trans. Graph. (TOG)* **2006**, *25*, 835–846. [[CrossRef](#)]
46. Bian, J.-W. GMS: Grid-Based Motion Statistics for Fast, Ultra-robust Feature Correspondence. *Int. J. Comput. Vis.* **2020**, *128*, 1580–1594. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).