frontiers | Frontiers in Robotics and AI

TYPE Methods
PUBLISHED 16 March 2023
DOI 10.3389/frobt.2023.1120357

OPEN ACCESS

EDITED BY
Qing Gao,
The Chinese University of Hong Kong,
China

REVIEWED BY
JiaGang Song,
Central South University, China
Xuanyu Zhang,
Northwestern Polytechnical University,
China

*CORRESPONDENCE
Yan Zhang,
✉ yan.zhang@tu-ilmenau.de

SPECIALTY SECTION
This article was submitted to Robot Vision and Artificial Perception, a section of the journal Frontiers in Robotics and AI

RECEIVED 09 December 2022
ACCEPTED 02 March 2023
PUBLISHED 16 March 2023

CITATION
Zhang Y, Fütterer R and Notni G (2023), Interactive robot teaching based on finger trajectory using multimodal RGB-D-T-data.
Front. Robot. AI 10:1120357.
doi: 10.3389/frobt.2023.1120357

Interactive robot teaching based on finger trajectory using multimodal RGB-D-T-data

Yan Zhang[1]*, Richard Fütterer[1] and Gunther Notni[1,2]

[1]Group for Quality Assurance and Industrial Image Processing, Technische Universität Ilmenau, Ilmenau, Germany, [2]Fraunhofer Institute for Applied Optics and Precision Engineering IOF Jena, Jena, Germany

The concept of Industry 4.0 brings the change of industry manufacturing patterns that become more efficient and more flexible. In response to this tendency, an efficient robot teaching approach without complex programming has become a popular research direction. Therefore, we propose an interactive finger-touch based robot teaching schema using a multimodal 3D image (color (RGB), thermal (T) and point cloud (3D)) processing. Here, the resulting heat trace touching the object surface will be analyzed on multimodal data, in order to precisely identify the true hand/object contact points. These identified contact points are used to calculate the robot path directly. To optimize the identification of the contact points we propose a calculation scheme using a number of anchor points which are first predicted by hand/object point cloud segmentation. Subsequently a probability density function is defined to calculate the prior probability distribution of true finger trace. The temperature in the neighborhood of each anchor point is then dynamically analyzed to calculate the likelihood. Experiments show that the trajectories estimated by our multimodal method have significantly better accuracy and smoothness than only by analyzing point cloud and static temperature distribution.

KEYWORDS

multimodal image processing, RGB-D-T-data, point cloud processing, finger trajectory recognition, robot teaching, meshless finite difference solution

1 Introduction

Nowadays, robots are already capable of supporting humans for some precise or dangerous tasks in a wide variety of fields, such as assembly robots, welding robots and medical robots. In general, robots need some customization and system integration to satisfy such specialized tasks, which requires users to have some expertise in robot operating. In this respect, the most common basis task is trajectory teaching. In order to respond to evolving industrialization levels, a modern dynamic production line requires an efficient approach for robot trajectory generation. For this purpose, robot developers and manufacturers have been trying to work on teach pendant. However, by using this device, the teaching of a complex arbitrary trajectory containing an extremely high number of waypoints is very time-consuming. If it needs to be more efficiently solved, a professional programmer is required. Therefore, an easy-to-use and still effective trajectory teaching approach becomes a research hotspot in recent years to lower the employment barrier for such skill based professions.

Regarding this application, there have been many studies in recent years. For example, in the research (Braeuer-Burchardt et al. (2020)), a demonstrator system for selected quality checks of industrial work pieces with a human machine interaction was proposed, in which the check position of a work piece is determined by a finger pointer. In addition, the company Wandelbots Teaching (Wandelbots (2022)) developed a robot teaching system using a TracePen as an input device. The pen works like a tracker for recording a sequence of waypoints (rotations and translations under the robot base coordinate system) that can be further refined manually by their software.

In this article, we will propose a vision-based trajectory teaching method. In our approach, the core module is a finger trajectory recognizer, which is realized by using a multimodal vision sensor system. It consists1 of a color camera (RGB), a 3D sensor (D) and a thermal camera (T) for multimodal point cloud (RGB-D-T) recording. The touch of an object with a finger results in a slight temperature change of the object surface. If the finger is moved on the 3D-object along the robot's imaginary motion path it leaves a heat trace on the object surface resulting in a 3D-heat-trajectory, which directly represents the robot motion trajectory.

To get an accurate 3D-trajectory one has to take into account that mostly finger trajectory recognizers are based on hand detection or human skeleton detection, for example, (Halim et al. (2022); Du et al. (2018)). Such approaches are usually inaccurate because at the moment when the finger and the object are in touch, the actual contact point will definitely be blocked by the finger at the camera's perspective. In our method, introducing multimodal point cloud analysis, the finger movement process can be considered as a heat transfer process caused by a moving Gaussian point heat source. By analyzing the residual heat on the object surface, the trajectory can be predicted more accurately. In recent years an increasing number of multimodal sensor-based image processing methods have been discussed and applied to scenarios with human interaction. For example (Jeon et al. (2016)), introduced an outdoor intelligent surveillance system with a color and a thermal camera, which is capable of recognizing humans in both day and night. In most approaches, temperature is analyzed as a static feature in the same way as color. In fact, in comparison to color, even if the heat source is fixed or removed, temperature still changes relative to time and spatial variables. More attention should be paid to these characteristics in order to extract more information from multimodal point cloud to achieve more diverse human-machine interactions. Therefore, according to the heat equation, a node in a temperature field is in a heat dissipation state when it has a negative divergence. The greater its absolute value, the higher the rate of heat transfer. Thus in our method the node with a low divergence will be considered as a candidate of contact point with a high probability.

In this regard, the temperature analysis in 2D thermal images is limited. By using a 2D camera, an temperature field can only be accurately captured when the surface of the object is a plane and parallel to the sensor plane. Otherwise, the spatial independent variables (x-, y- and z-coordinates in the world coordinate system) used to calculate divergence will be non-uniformly observed by a 2D camera. This unevenness is related to the complexity of the object surface and the placement posture of the object. It leads to errors in the solution of gradient or divergence. By using 3D point cloud analysis, this problem can be avoided. However, a point cloud is a meshless and unordered point set. Common finite difference methods such as the central difference formula cannot be used directly to calculate the numerical solution of the partial differential. Therefore, in this paper we propose a fast method to find the approximated solution of the divergence for each node in a meshless 3D temperature field (a thermal point cloud).

In overall terms, our approach follows Bayesian theory. A candidate region (prior probability) is firstly determined with the help of hand/object semantic segmentation in multimodal point cloud. Then a distribution of the divergence (likelihood) is calculated in candidate regions. Finally, the by finger obscured contact points (posterior probability) will be estimated. By using these contact points, a realistic robot motion trajectory is generated through interpolation. In the experimental section, the error of the approximated divergence solution and the deviation of robot motion trajectory generation will be evaluated and discussed.

2 Related work

Currently, most industrial robot manufacturers provide a teach pendant with a manual motion mode, with which the robot can be moved manually to a set of positions they are marked as waypoints. The robot can then be simply programmed to execute a trajectory consisting of these waypoints in sequence. As mentioned above, it will be particularly time consuming when complex and arbitrary trajectories with a large number of waypoints are defined. In this regard, teaching by human body language has become a popular area of research. The previous works (Braeuer-Burchardt et al. (2020)) and (Wandelbots (2022)) demonstrate the application of modern human-machine interaction methods for straightforward robot teaching. However, they have some limitations. (Braeuer-Burchardt et al. (2020)) brings a contactless interaction, but the exact check position cannot be obtained by only a finger pointer. The method of (Wandelbots (2022)) can avoid this problem, but an expensive TracePen is required to achieve the tracking. Other than that, in the works of (Liu et al. (2022); Jen et al. (2008); Yap et al. (2014); Manou et al. (2019);Ratticò and Lamberti (2021)), Virtual Reality (VR) technology was used to improve the human-robot interface without the requirement for complicated command or programming. Such as in the studies (Su et al. (2018); Abbas et al. (2012); Stadler et al. (2016); Pettersen et al. (2003)), Augmented Reality (AR) systems were designed to allow users to govern the movement of real robots in a 3D space *via* a virtual one generated through AR technology. Whether in the application using VR or AR, the recognition of finger trace always plays the role of a bridge between realistic movements and virtual trajectories. The most intuitive solution to solve this core task is hand skeleton recognition based on color or depth image, such as (Halim et al. (2022); Baek et al. (2018); Cheng et al. (2021); Zhang et al. (2020); Osokin (2018)). However, the finger trace defined by these methods is not exactly equivalent to the robot motion trajectory on the object surface. Due to the occlusion by finger, the trajectory cannot be captured in real time by cameras. Hence, we recommend introducing multimodal sensors (RGB-D-T) to collect more diverse information, in order to improve prediction results closer to true trajectories.

Guanglong et al. (Du et al. (2018)) introduced a particle filter and neural network based gesture estimator using a multimodal sensor system containing a RGB-D camera and an inertial measurement unit, in which multimodal information including

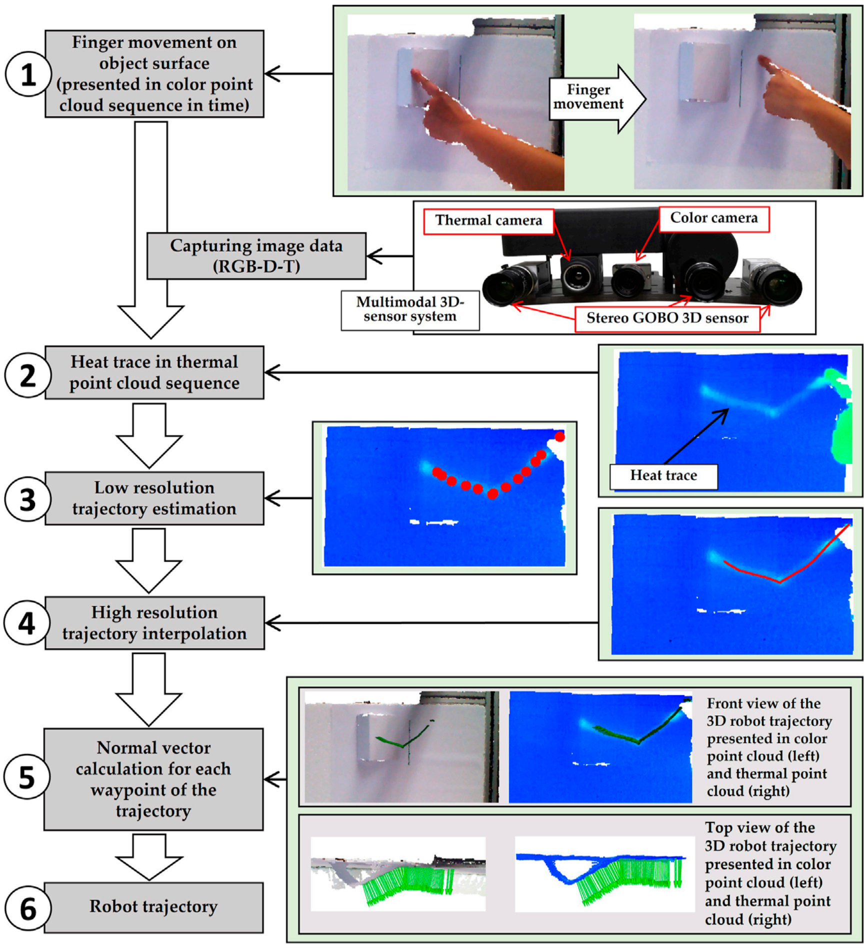
1
Finger movement on object surface (presented in color point cloud sequence in time)
Finger movement
Capturing image data (RGB-D-T)
Thermal camera
Color camera
Multimodal 3D-sensor system
Stereo GOBO 3D sensor
2
Heat trace in thermal point cloud sequence
Heat trace
3
Low resolution trajectory estimation
4
High resolution trajectory interpolation
5
Normal vector calculation for each waypoint of the trajectory
Front view of the 3D robot trajectory presented in color point cloud (left) and thermal point cloud (right)
Top view of the 3D robot trajectory presented in color point cloud (left) and thermal point cloud (right)
6
Robot trajectory

FIGURE 1
Basic concept of our approach.

RGB-D image, velocity and acceleration of hand as well as speech are fused. Then a neural network is used to encode such data to predict the finger trace. In the research of Zhang et al. (Zhang et al. (2021)), by using PointNet (Qi et al. (2017a)), PointNet++ (Qi et al. (2017b)) and RandLANet (Hu et al. (2020)), multimodal image data with color, thermal and point cloud was encoded and decoded to perform a pixel-wise hand/object semantic segmentation for application of a hand over robot. Their experimental results showed a better segmentation performance of the hand-object interaction region with the help of thermal information compared to RGB-D-based segmentation, if the object has a similar color or temperature as the hand especially. However, temperature is analyzed only as a static feature, like color. In other words it should be considered as a multispectral 3D image analysis.

3 System overview

Figure 1 shows the basic concept of our approach. At first (step 1) the finger touches the object and moves along an imaginary robot trajectory to teach in. During this movement the multimodal 3D-Sensor consisting of a 3D-sensor, a RGB-camera and a thermal camera (picture right) capturing a series of RGB-D-T data resulting in a 3D-heat trace data set (step 2). With the help of these collected information, the trajectory with low resolution is estimated (step 3) and then used to interpolate a dense smooth 3D motion trajectory for robots (step 4). By using the 3D-data of the object, the orientation of each waypoint of the 3D-trajectory will be recalculated, which is equal to the surface normal vector at its position, to ensure that the robot can always move perpendicularly to the object surface (step 5). Finally, this high-resolution trajectory is used as the input for an identical movement of the robot (step 6).

As mentioned above, the core component of our approach is a finger recognizer. A finger trajectory recognition can be regarded as a branch of the task of object (contact point) tracking. The process of such tasks is often described as a hidden discrete-time Markov chain and a number of solutions for this are theoretically based on Bayesian theory. The concept of such solutions is divided into three steps. Firstly, the prior probability is inferred from the system model. Then the likelihood is estimated based on the observation model. Finally, the posterior probability is calculated dependently on the prior probability and the likelihood, which

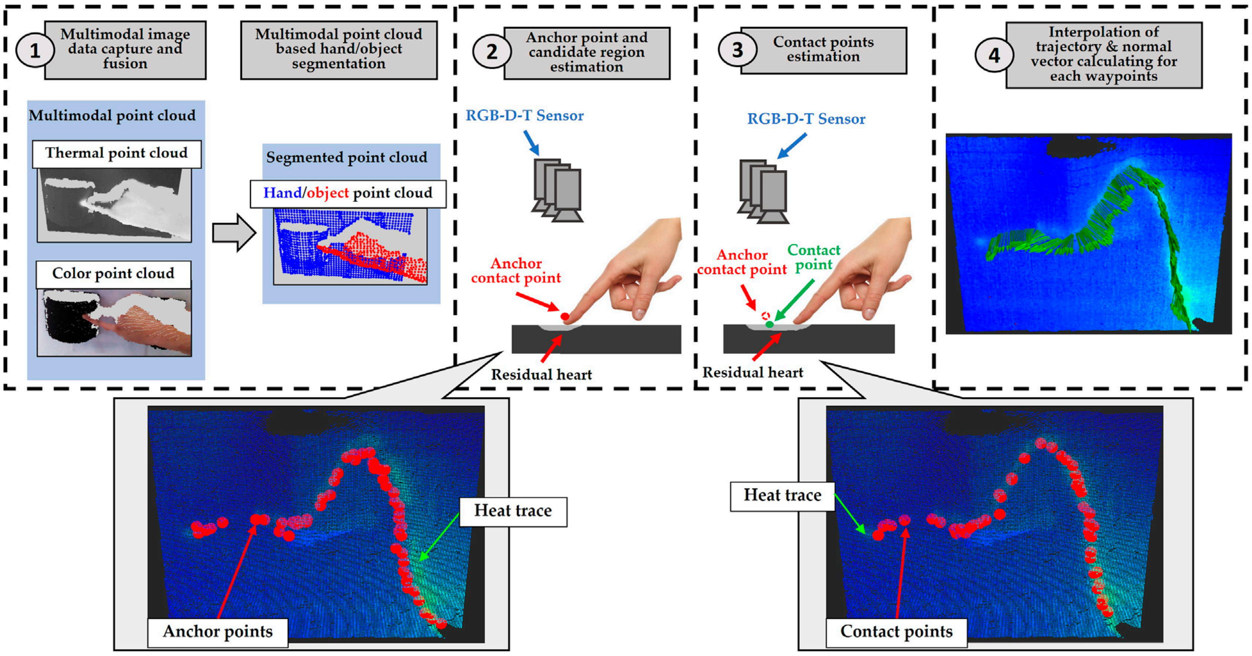

FIGURE 2
Method architecture.

will be used to produce the prediction for the target random event. Our system is also developed along this lines, in which a random event of whether a point in a candidate region is a contact point will be predicted. As shown in the Figure 2, our finger recognizer schema is divided into four modules. In module 1, see chapter 4, the multimodal sensor system and robot base need to be calibrated in order to calculate intrinsic parameters and extrinsic parameters of cameras and the robot. They will be utilized to fuse multimodal 3D image data as well as to interconvert 3D waypoints (orientation and position) between the camera and robot coordinate systems. Then the hand and objects will be semantically segmented in point cloud. In module 2, see chapter 5, by using the segmentation results, a set of anchor finger-object contact points on time series is estimated roughly, and a neighborhood searching is performed for each anchor point to determine a local candidate region. The prior probability P_{prior} is calculated for each point in the candidate region. Furthermore, based on these local candidate regions, the computational complexity of estimating the real contact points is reduced significantly. Module 3 (see chapter 6) provides thermodynamic analysis in each candidate region to obtain the likelihood distribution L. It is worth mentioning here that we introduce a divergence-based temperature field analysis, which has a noticeable advantage over the analysis based only on temperature distribution for trajectory estimation. Meanwhile it is verified in the experimental chapter 8. Finally, the true contact points are predicted by calculating posterior probability

$$P_{posterior} = w_1 P_{prior} + w_2 L, \tag{1}$$

where w_1 and w_2 are the weights for prior probability and likelihood. In our experience, setting these weights needs to consider the heat transfer coefficient of the object and the speed of finger movement. Finally, the finger-object contact points will be defined as

$$ContactPoint = argmax\left(P_{posterior}\right). \tag{2}$$

In module 4, see chapter 7, a high-resolution trajectory will be interpolated based on these contact points in order to achieve smooth robot motion.

4 Multi-modal point cloud fusion and segmentation (module 1)

4.1 Multimodal sensor system

As shown in Figure 1, our multi-modal 3D imaging system consists of a high-resolution active stereo-vision 3D sensor based on GOBO (Goes Before Optics) projection (Heist et al. (2018)), a color camera (Genie Nano C1280 (Genie (2022))) and a thermal camera (FLIR A35 (FLIR (2022))).

4.2 Multimodal image data fusion and hand object segmentation

Inspired by (Zhang et al. (2021)), the multimodal sensor system was calibrated using a copper-plastic chessboard as the calibration target. The multimodal image data was then fused using the intrinsic and extrinsic parameters as well as further hand-object segmented using RandLANet (Hu et al. (2020)). RandLANet is a lightweight neural network with a multi-level architecture designed for large-scale 3D point cloud semantic segmentation. In each level, a random

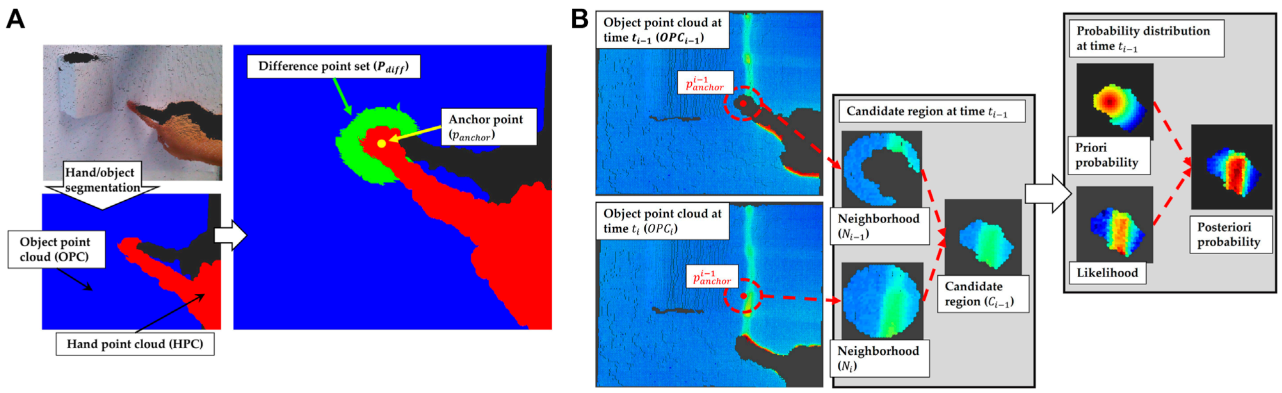
A
Hand/object segmentation
Object point cloud (OPC)
Hand point cloud (HPC)
Difference point set (P_{diff})
Anchor point (p_{anchor})
B
Object point cloud at time t_{i-1} (OPC_{i-1})
p^{i-1}_{anchor}
Object point cloud at time t_i (OPC_i)
p^{i-1}_{anchor}
Candidate region at time t_{i-1}
Neighborhood (N_{i-1})
Candidate region (C_{i-1})
Neighborhood (N_i)
Probability distribution at time t_{i-1}
Priori probability
Likelihood
Posteriori probability

FIGURE 3
(A): hand/object segmentation and the method for defining anchor points (P_{anchor}) **(B)**: the concept to define the candidate region (C_i) and a visualization of the prior probability (P_{prior}), likelihood (L) and posterior probability ($P_{posterior}$).

downsampling is used to enable that the point density of the point clouds is progressively decreased. By using a local spatial encoding module (LocSE) in each neighborhood, XYZ-coordinates of all points, Euclidean distances as well as XYZ-differences between the centroid point and all neighboring points are explicitly encoded using shared multi-layer perceptron. Additionally, in between two adjacent levels, an attentive pooling is utilized to aggregate the features. Then, multiple LocSE and attentive pooling units with a skip connection are stacked as a dilated residual block, which is repeatedly used in the RandLANet. A hierarchical propagation strategy with distance-based interpolation and a cross level skip links is adopted to upsample the point clouds to the original size.

The experimental results in (Zhang et al. (2021)) indicate that based on an XYZ-RGB-T (XYZ: spatial coordinate, RGB: color, T: thermal) point cloud, the RandLANet can learn the complex aggregation and combination of multimodal features. In this way, the information from each channel compensates for their respective weaknesses. The segmentation of the XYZ-RGB-T point cloud has better robustness than the XYZ-RGB and XYZ-T for some objects that have similar color, surface texture, or temperature as the hand. For example, the heat trace left on the object by the finger did not worsen the segmentation results. This statement is an important premise for the application of this article.

5 Calculating the candidate region (module 2)

5.1 Anchor points estimation

By using the segmented hand and object point clouds (*HPC* and *OPC*), a number of anchor points can be simply determined by the mean of the nearest point pair between them. However, such an anchor point is not optimal, because it will be identified as a point on top of the fingertip rather than a point on the object surface (contact area between finger and object). On the other hand, the 3D points within these areas usually cannot be reconstructed by a 3D sensor due to occlusion. Hence, in order to estimate an anchor point p^i_{anchor} that is closer to the true contact point at time t_i, we use two different distance thresholds d_1 and d_2 to segment two point clouds P_1 and P_2 from the object point cloud OPC^i. They consist of a number of object points whose distance to their nearest hand points is less than d_1 and d_2. Since both P_1 and P_2 are subsets of OPC^i, then a difference set $P_{diff} = P_2 \setminus P_1$ can be calculated using set operator simply. P_{diff} is approximately an annular point cloud whose centroid will be defined as the anchor point p^i_{anchor}, as shown in **Figure 3A**. Details of the algorithm will be given in **Supplementary Appendix SA**.

5.2 Candidate regions estimation

Obviously, at time t_i, the candidate region C_i should be located in the spherical neighborhood N_i of the anchor point p^i_{anchor}. As shown in the **Figure 3B**, at the time t_{i-1} in N_{i-1}, there is an area that was obscured by the finger. This area will be observed at the time t_i in N_i. It will be defined as the candidate region C_{i-1}. For calculating C_{i-1}, we need to determine the difference set of N_{i-1} and N_i. However, N_{i-1} and N_i were captured at different times, thus this difference set cannot be calculated by using set operators. We introduce a tolerance r_c. If a point in N_i has a nearest point in N_{i-1} and their distance is less than r_c, this point will be considered that it has an approximate overlapping point in N_{i-1}. Then a point cloud consisting of a number of points in N_i without overlapping points in N_{i-1} will be defined as the candidate region C_{i-1} at time t_{i-1}. In other words, the range of C_{i-1} is determined at time t_{i-1}, while the information (point position and temperature) is acquired at time t_i. The details of this algorithm will be explained in **Supplementary Appendix SB**.

5.3 Prior probability calculation

In each candidate region, a prior probability density function related to the distribution centered on the corresponding anchor point can be defined as

$$P_{prior}(x) = \int G(x)\,C(x)\,\mathrm{d}x, \tag{3}$$

where x denotes the position of random variables in a domain. In our case that is the 3D coordinates of the object points in the candidate region. $G(x)$ denotes a Gaussian probability density function and $C(x)$ denotes a function to describe the relationship between the distribution of the candidate points and the probability whether they are true contact points. In which, a point has the probability that is negatively correlated with its distance to the anchor point. In other words, a point closer to the anchor point has a higher probability, as shown in **Figure 3B**.

6 Calculating the optimized contact point (module 3)

In this section we discuss how to solve the divergence of each point in the candidate region. Based on the divergence, the likelihood is further calculated to complete the Bayesian approach. In this regard, the candidate region can be considered as a local temperature field, in which the finger can be considered as a moving Gaussian point heat source. The heat transfer state of the residual heat trace on the object surface can be described by the heat equation in a Cartesian coordinate system:

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right), \tag{4}$$

where (x, y, z) and t denotes the spatial variables and time variable of each point. In our case the object is assumed isotropic and homogeneous, thus the thermal diffusivity of the medium α will be constant. This equation indicates that the first-order derivative of temperature U related to time variable $\frac{\partial u}{\partial t}$ exhibits a linear relationship to the second-order derivative related to spatial variables $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$. Also the divergence of a 3D temperature field can be solved by

$$Divergence = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2}. \tag{5}$$

We do not have to perform heat conduction simulation, but only to find the divergence of each point in the temperature field. Then the point with a lower divergence (faster cooling) has a greater probability to be a finger-object contact point. Hence, the numerical solution of these three second-order partial derivative terms $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 u}{\partial y^2}$ and $\frac{\partial^2 u}{\partial z^2}$ at each point in the candidate region need to be calculated.

Unfortunately, for a non-grid or meshless structure such as point cloud, the common second-order central difference formula

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{U(x_0 + \Delta x, y_0, z_0) - 2U(x_0, y_0, z_0) + U(x_0 - \Delta x, y_0, z_0)}{\Delta x^2} \tag{6}$$

is not available. Because in the point cloud, which cannot be ensured, the temperature at both of the two points $(x_0 + \Delta x, y_0, z_0)$ and $(x_0 - \Delta x, y_0, z_0)$ can be observed by the cameras at the same time. To solve this problem, we propose a fast meshless finite difference method, in which a temperature difference field needs to be firstly calculated for each point in the candidate region. Then a system of differential equations based on Taylor expansion will be solved using an elimination method for calculating the divergence.

6.1 Temperature difference along each axis

Given a point P_0 in the candidate region, a further neighborhood search is performed for P_0 to obtain its neighbor point set. Then a vector set of temperature difference $\overrightarrow{\Delta U}$ between each neighbor points and P_0 is calculated. In which, $\overrightarrow{\Delta U_i}$ is a vector whose direction is the same as $\overrightarrow{P_0 P_i}$ and its norm equals the temperature difference between the points P_i and P_0. Then based on $\overrightarrow{\Delta U}$, the components of temperature difference along the X-axis, Y-axis and Y-axis directions ΔU^x, ΔU^y and ΔU^z can be calculated using

$$\Delta U_i^k = \|\overrightarrow{P_i^x}\| \frac{\|\overrightarrow{\Delta U_i}\|}{\|\overrightarrow{P_0 P_i}\|}; \quad i = [1, Num]; \quad i \in \mathbb{N}; \quad k \in \{x, y, z\}, \tag{7}$$

where $\overrightarrow{P_i^x}$ denotes the X-component of $\overrightarrow{P_0 P_i}$ along the X-axis and Num denotes the number of points in the neighborhood of P_0.

6.2 Solving second-order derivative

It is well known that the Taylor series is fundamental for solving partial differential equations. In our case, if $\Delta x \neq 0$, $\Delta y = 0$ and $\Delta z = 0$, the second-order Taylor expansion of $U(p)$ for a 3D point $p_0\ (x_0, y_0, z_0)$ and one of its neighbor point $p_i\ (x_0 + \Delta x, y_0 + \Delta y, z_0 + \Delta z)$ is

$$U(x_0 + \Delta x, y_0, z_0) = U(x_0, y_0, z_0) + \Delta x U_x' + \frac{\Delta x^2}{2} U_{xx}'' + E_2, \tag{8}$$

where E_2 denotes a second-order error term. In our case, it is considered to be approximately equal to zero. We use an elimination method to solve this system of equations. Thus a coefficient vector A will be introduced and each Taylor expansion for p_0 and each neighbor point p_i is multiplied by the coefficient $a_i \in A$ and then summed to obtain

$$\begin{aligned} A \cdot (U_0 + \Delta U^x)^T &\approx A \cdot U_0^T + C_1 U_x' + C_2 U_{xx}'' \\ C_1 &= A \cdot \Delta X^T \\ C_2 &= \frac{1}{2} A \cdot (\Delta X \circ \Delta X)^T, \end{aligned} \tag{9}$$

where ΔU^x denotes a vector consisting the x-components of temperature difference between p_0 and p_i that was obtained in the previous section. ΔX is a vector consisting of the X-components of distances between p_0 to p_i. U_0 denotes a vector consisting of the temperature at the point P_0, in which all the elements are equal. We need to find the coefficient vector A that meets the conditions $C_1 = 0$ and $C_2 \neq 0$. Then the second-order partial derivative of U at the point p_0 with respect to x can be solved using

$$\frac{\partial^2 U}{\partial x^2} \approx \frac{2A \cdot \Delta U^{xT}}{A \cdot (\Delta X \circ \Delta X)^T}. \tag{10}$$

In **Supplementary Appendix SC**, an algorithm for solving this system of differential equations will be described in detail. Similarly, $\frac{\partial^2 U}{\partial y^2}$ and $\frac{\partial^2 U}{\partial z^2}$ can be solved for and the divergence can be calculated using Eq. 5.

6.3 Likelihood calculation

A likelihood function related to the divergence field is defined as

$$L(x) = \int L(D(x))\,\mathrm{d}x, \tag{11}$$

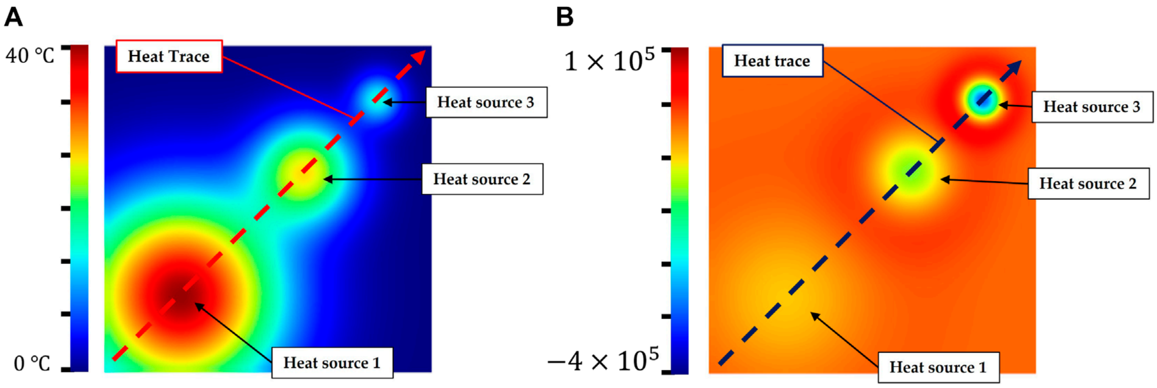

FIGURE 4
Simulated test data for the experiment to evaluate the accuracy of divergence solution. **(A)**: simulated thermal point cloud with a trajectory including three Gaussian point heat sources **(B)**: the ground truth of divergence solved by the second-order central difference formula.

TABLE 1 The parameters for three Gaussian point heat sources.

	Heat source 1 (bottom left)	Heat source 2 (middle)	Heat source 3 (top right)
Amplitude	40°C	30°C	20°C
Standard deviation	0.01 m	0.005 m	0.001 m

where x denotes the spatial variables (point position) in the candidate region. The function $D(x)$ describes the divergence distribution, i.e., Eq. 5. $L(D(x))$ refers to the likelihood distribution negatively relative to the divergence distribution, as shown in Figure 3B. Finally, by using the Eqs 1, 2, the real finger-object contact points will be calculated.

7 Robot motion trajectory calculation (module 4)

Furthermore, the normal vector of each contact point will be calculated using the 3D-data to allow that the robot moves always perpendicular to the object surface. However, it is obvious that the resolution of the trajectory obtained by this method is limited by the width of the finger. Hence, in response to this, we have to perform linear interpolation twice, the first time in 3D spatial space to achieve a smooth path (positions and orientations) for the end effector and the second time in robot joints space ensure limited angular velocity for the robot joint motion.

8 Experiments

8.1 Experiment for divergence solution

In this section we will present an experiment to evaluate the accuracy of the divergence solution. In order to avoid the influence of sensor-specific noise on evaluation results, we built a uniform regular point cloud that can be gridded, as shown in Figure 4A. This simulated data was a board with a spatial resolution of 0.5 mm as well as its length and width are 0.1 m. A temperature field was initialized with a heat trace caused by three different Gaussian point heat sources. The parameters of these heat sources are shown in Table 1.

Heat source 3 has a lower amplitude and standard deviation compared to heat sources 1 and 2. This indicates that heat source 3 is a new heat source, but it has a lower temperature than others. In contrast, heat source 1 has the highest temperature but it is the oldest heat source (Gaussian function with a lowest standard deviation). Hence, these three heat sources (hs_1, hs_2 and hs_3) at time t_1, t_2 and t_3 have temperature that consistent with $1 > 2 > 3$, and their divergence were consistent with $1 < 2 < 3$, as shown in the Figure 4. The experiment was set up in this way because the residual temperature of the finger on the object depends on the contact area between the finger and the object as well as the duration of contact. In other words, a new contact point is not certainly hotter than an old one.

By using the finite difference method (Eq. 6), the ideal divergence was calculated as ground truth from the uniform grid data, as shown in Figure 4B. The point cloud was then randomly downsampled into meshless data. Finally, our method was utilized to solve partial derivative solution in the sampled meshfree point cloud.

In order to solve the divergence, a further neighborhood search should be performed for each point in the candidate region. In this experiment, two common neighborhood search methods (k-nearest neighbor (KNN) and radius nearest neighbor (RNN)) were evaluated. Figure 5 shows the results (mean divergence deviation for each point in sampled point clouds) by using RNN with various search radius $r = (0.005\ \text{m}, 0.03\ \text{m})$ and using KNN with various number of neighbors $k = (100, 500)$. Meanwhile, the uniform regular point cloud was downsampled with various downsampling rate

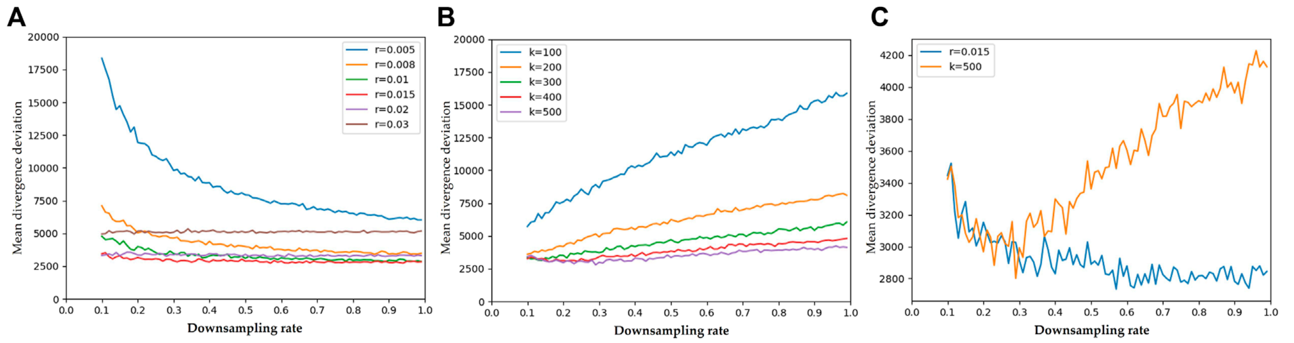

FIGURE 5
The results of the experiments to evaluate the accuracy of divergence solution. **(A)**: mean deviation of divergence using RNN neighbor search with various search radius $r = [0.005\,m, 0.03\,m]$ and downsampling rate $r = [0.1, 0.99]$ **(B)**: mean deviation of divergence using KNN neighbor search with various number of neighbors $k = [100, 500]$ and downsampling rate $r = [0.1, 0.99]$ **(C)**: comparison between the best results using RNN ($r = 0.015\,m$) and KNN ($k = 500$).

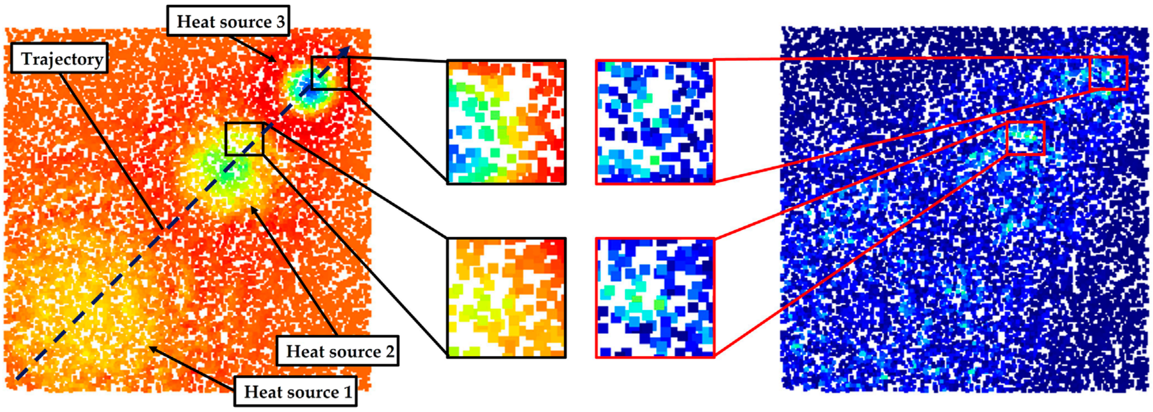

FIGURE 6
Visualization of a calculated divergence distribution (left) and the corresponding error distribution (right) using our method, where brighter points in the error distribution indicate greater errors.

$\beta = (0.1, 0.99)$. This downsampling rate is defined as

$$\beta = \frac{N_{sampled}}{N_{original}}, \tag{12}$$

where $N_{sampled}$ and $N_{original}$ denote the number of points in the sampled and original point cloud.

The results of RNN show that the radius of 0.015 m has the lowest mean error. It is also robust towards a variety of downsampling rates. The radius of 0.005 m has the worst results and there is a clear tendency for the results to be worse as the downsampling rate β is decreased. This is because the random downsampling not only leads to a reduction of the spatial resolution of point cloud, it also causes some random defects in the point cloud. These defects result in a non-uniform distribution of samples in the eight quadrants of neighborhoods for solving the divergence at a point. The non-uniformity significantly affects the accuracy of the solution for divergence. As shown in Figure 6, the left point cloud shows the predicted divergence and the absolute deviation for each point is presented by the right cloud, with the brighter points having a greater error. Obviously, the error is larger in areas where the neighbor samples are unevenly distributed and where the absolute value of the divergence is great (there is a strong positive or negative heat transfer). A straightforward solution to this problem is to dilute the non-uniformity with a large search radius. However, unfortunately a large search radius also dilutes the fine-grained information. Therefore, with a radius of 0.03 m, the error is consistently large, although there is robustness towards different downsampling rates, as shown in Figure 5A. This is also reflected in the results of KNN (as shown in Figure 5B). With a fixed number of neighbors, the perceptual field of the neighborhoods grow up as the sampling rate increases, leading to more errors. Figure 5C shows a comparison of the best results by RNN and KNN respectively. It is clear that RNN has better robustness than KNN, which is in line with our expectations. In the point clouds captured by a real 3D sensor, some defects will inevitably appear because there are always some object points that cannot be 3D reconstructed. The neighborhood determined by KNN cannot even ensure that the target point is located in the geometric center of the neighborhood. Therefore, in

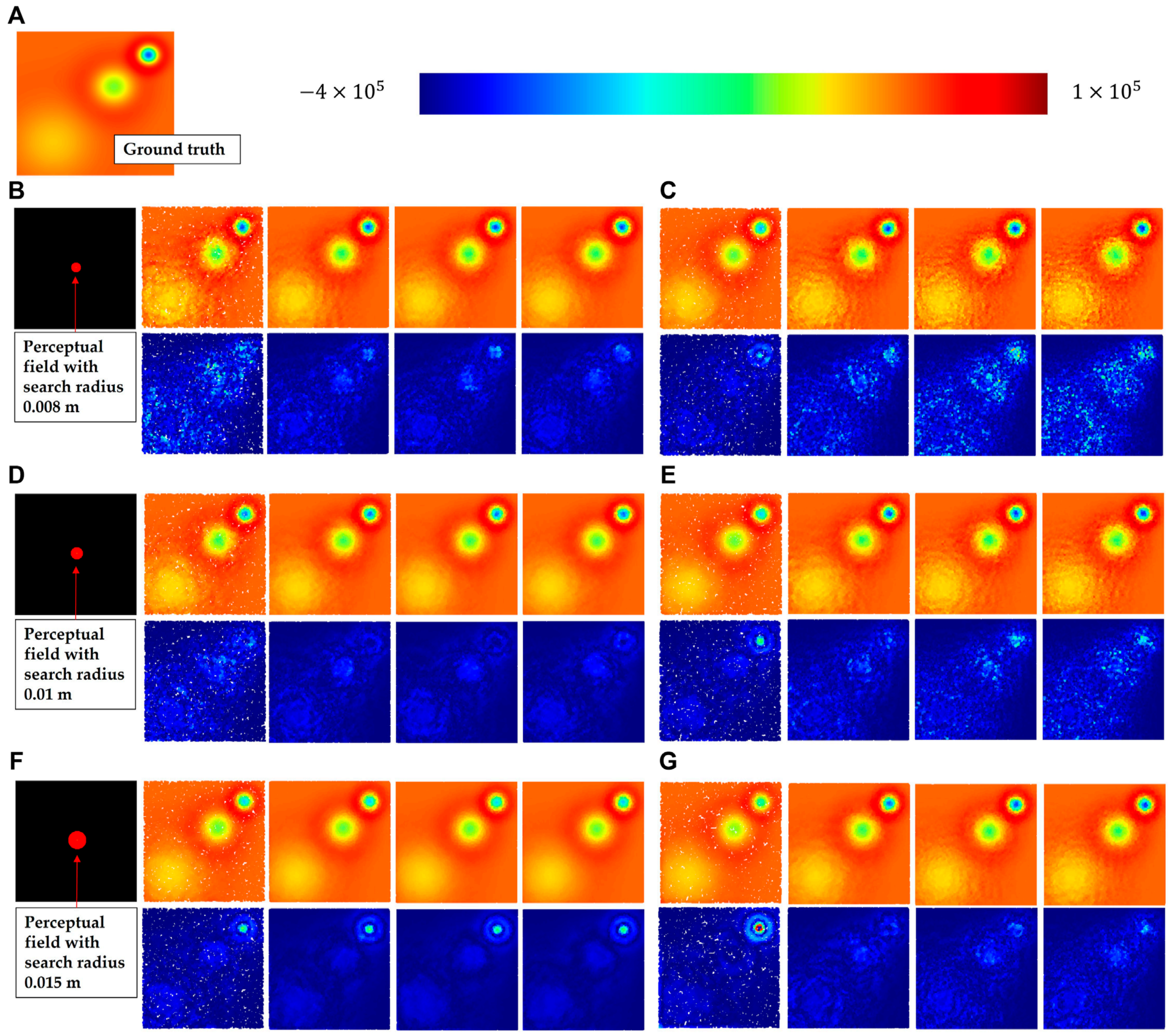

FIGURE 7
These figures show the results of the experiment to evaluate the accuracy of divergence solution. **(A)** Shows the ground truth of the divergence distribution; then, the results (top) and errors (bottom) using RNN with search radius (top to bottom: 0.008 m **(B)**, 0.01 m **(D)** and 0.015 m **(F)**) are presented; finally, the results using KNN with number of neighbors (top to bottom: 200 **(C)**, 300 **(E)** and 500 **(G)**) are presented. Each subfigure also shows the results with various same downsampling rates (left to right: 0.1, 0.4, 0.7, 0.9).

our case it is an optimal choice to adopt a suitable search radius, provided the spatial resolution of the 3D sensor is already known.

Figure 7 shows some visualization of this experiment. Figure 7A presents the ground truth of the divergence field. Then, Figures 7B, D, F exhibit the results by RNN with search radius of 0.008 m, 0.01 m and 0.015 m respectively, where the top-left graph shows the perceptual field of the search radius. The following presents in turn the results (top) and errors (bottom) for sampled point clouds with various downsampling rates of 0.1,0.4,0.7 and 0.9. Figures 7C, E, G show the results and errors of KNN with number of neighbors of 200, 300 and 500 towards the identical downsampling rates respectively. In the error plots, brighter points denote higher errors.

The results of RNN show that there is the lowest mean error at the search radius 0.015 m. However, compared to the result of the radius 0.01 m, it has a noticeably larger error in the nearby region of heat source 3 hs_3. It confirms what was aforementioned, that too large a search radius will dilute the fine-grained information and lead to deviation. In fact, we always focus more on these high frequency areas in practice, because in our case the points with the lowest divergence require more attention. Therefore, the principle for setting the search radius should be carried out by choosing the smallest radius while ensuring a uniform distribution of samples in the neighborhood. Compared to RNN, KNN performs weakly in both mean error and fine-grained error.

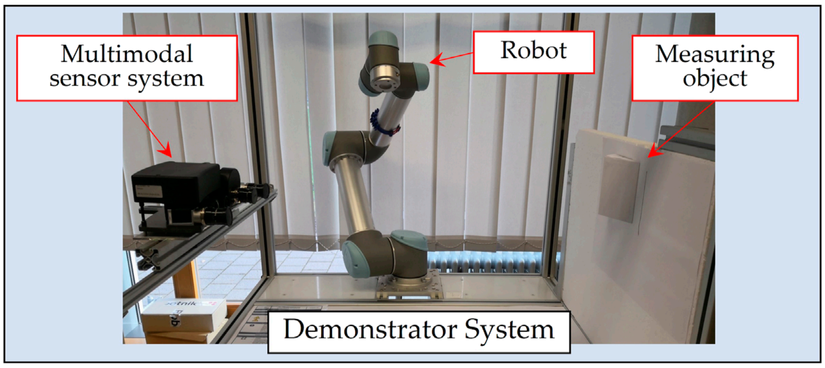

FIGURE 8
Experiment environment and our demonstrator system.

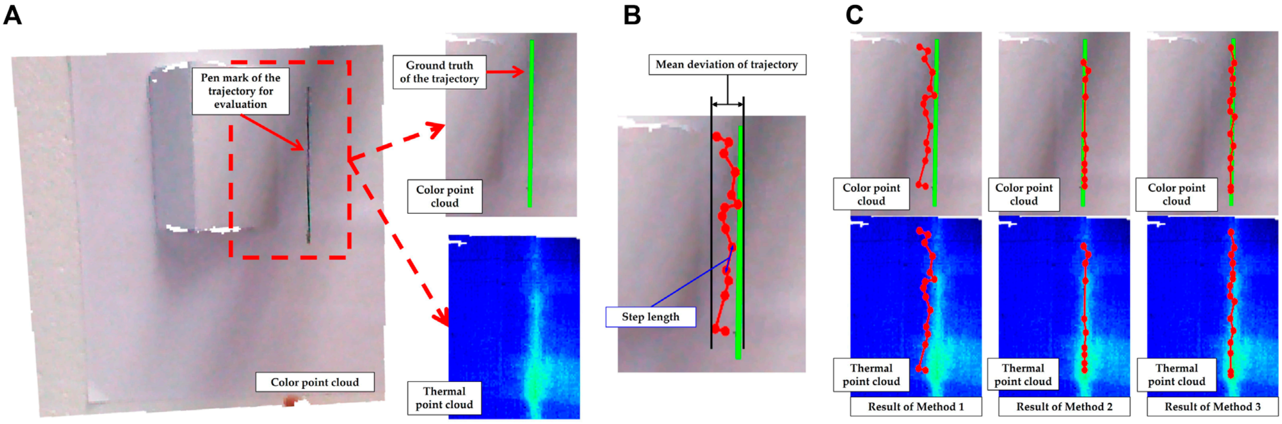

FIGURE 9
(A) Shows the straight line pen mark on the object surface (left), the ground truth of finger movement (top left) and the correspondent thermal point cloud **(B)** shows the evaluation criteria (Mean deviation of the trajectory and step length between each two adjacent contact points of the trajectory) of the experiment to evaluate the estimation for a straight line finger trajectory **(C)** shows the estimated contact points of a trajectory using method 1 (left), method 2 (middle) and method 3 (right), which are presented with color point cloud (top) and thermal point cloud (bottom). (Method 1 (left): anchor points were defined directly as contact points. This means that this method is not based on temperature analysis. Method 2 (middle): contact points were estimated using likelihoods that were positively correlated with temperature in the neighborhoods of the anchor points. In other words, in this method the temperature is analyzed as a static feature similar to color. In method 3 (right), likelihoods negatively dependent on the divergence were used for prediction (our method)).

8.2 Experiments for linear finger trace estimation

In the subsequent experiments, the performance of our finger recognition method will be validated in a real environment. As shown in Figure 8, the measuring object is placed roughly 1 m in front of the sensors and the robot is situated between them. A 15 cm long straight line mark was drawn on the object surface with a pen. In the multimodal point cloud, the points belonging to this line were found based on color and further in fitting a 3D straight line as ground truth of a finger trajectory, as shown in Figure 9A. Then the finger drew a heat trace along this line, which was repeated 30 times. Finally, 15 contact points were estimated on each heat trace by using three different methods, as shown in Figure 9C. Method 1 (left): anchor points were defined directly as contact points. This means that this method is not based on temperature analysis. Method 2 (middle): contact points were estimated using likelihoods that were positively correlated with temperature in the neighborhoods of the anchor points. In other words, in this method the temperature is analyzed as a static feature similar to color. In method 3 (right), likelihoods negatively dependent on the divergence were used for prediction (our method).

The first row in Table 2 shows the mean deviation (as shown in Figure 9B) of the results by using these three methods on the 30 heat traces relative to ground truth. It is obvious that method 2 and method 3 have significantly lower deviation than method 1. It confirms that the application of multimodal data processing provides significant enhancement for this task. In Figure 9C, by method 2, it is surprising that the number of contact points which can be observed is less than 15, due to the overlap of multiple points. In fact, when this heat trace was created, the finger pressure on the object surface was not homogeneous, leading to excessive

TABLE 2 Mean deviation and step length standard deviation of the contact point estimation using three methods.

	Method 1 (mm)	Method 2 (mm)	Method 3 (mm)
Mean deviation	9.046	2.842	2.185
Standard deviation of step length	3.989	7.943	3.506

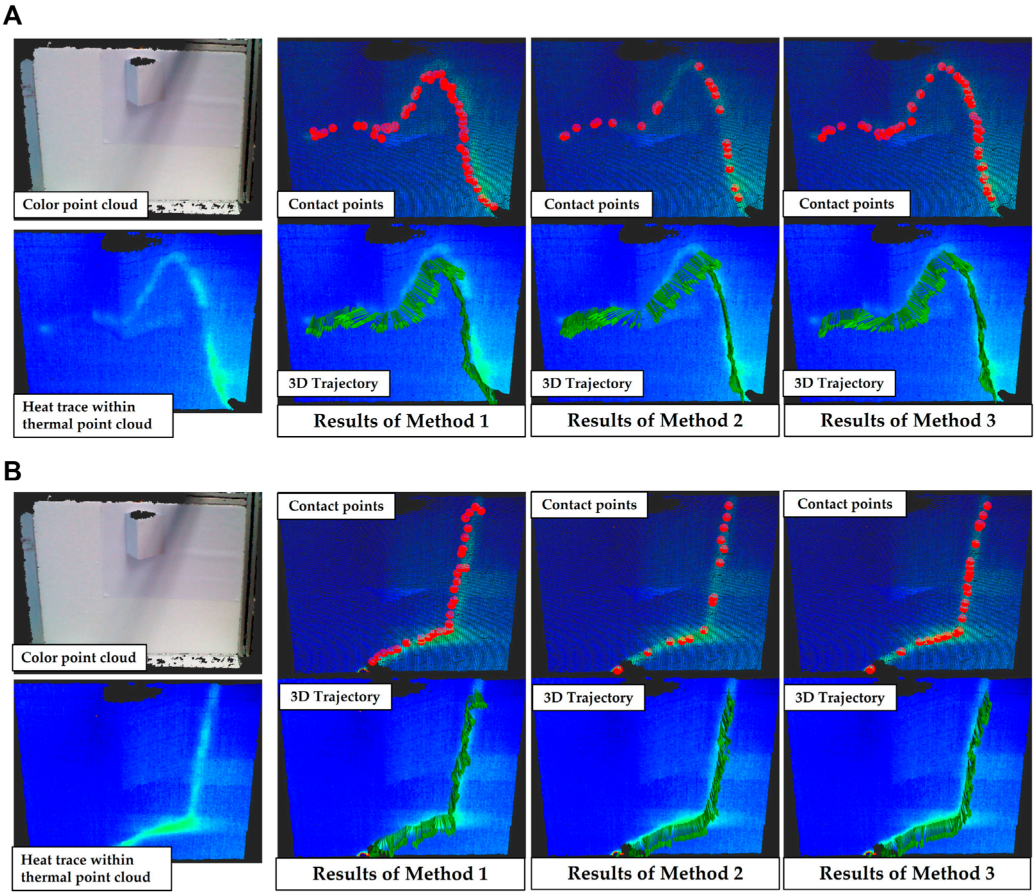

FIGURE 10
Robot motion trajectory predictions for two arbitrary finger movements are shown in figures **(A,B)**. In each subfigure, the first column shows a color object point cloud (top) and a residual heat trace presented within a thermal point cloud (bottom); the following columns show the predicted contact points (top) and interpolated 3D trajectories (bottom) obtained using various three methods. (left to right: Method 1: anchor points were defined directly as contact points. This means that this method is not based on temperature analysis. Method 2: contact points were estimated using likelihoods that were positively correlated with temperature in the neighborhoods of the anchor points. In other words, in this method the temperature is analyzed as a static feature similar to color. In method 3, likelihoods negatively dependent on the divergence were used for prediction (our method)).

temperatures in some areas than others. Around these areas, the temperature-related likelihood (method 2) provides incorrect information, guiding the contact points to a deviated position (hottest position). In this respect, we further calculated the standard deviation of the step length between each two adjacent contact points of a trajectory (as shown in Figure 9B) using

$$std = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|s_i - \bar{s}|^2}, \tag{13}$$

where $\bar{s}$ denotes the mean step length of a trajectory and N denotes the contact point number of a trajectory. The results are shown in the second row of Table 2. It exhibits that the contact points in trajectories obtained by method 2 always have non-uniform step length. In the case where the finger trajectory is no longer a straight line but an arbitrary curve, it results in the robot trajectory being interpolated not smoothly and imprecisely. In the next experiment, this hypothesis will be confirmed.

8.3 Experiments for arbitrary finger trace estimation

As shown in Figure 10, in this experiment two different arbitrary finger movements and the predicted robot motion trajectories are presented. In each subfigure, the image on the top left shows the target object, in the bottom left image the heat traces left on the object surface are displayed by a thermal point cloud. The second column of plots shows the predicted contact points by method 1 (top) and the robot motion trajectory obtained by interpolation based on these contact points (bottom). The third and last columns show respectively the results using method 2 and method 3.

It can be observed that the prediction of the trajectory calculated by method 1 are not accurate or smooth. Temperature-based (method 2) contact point prediction has improved in terms of accuracy (all of the contact points land within the hot areas). However, as mentioned previously, since the residual temperature depends on the touch area and touch duration between finger and object, the temperature of a new contact point is not definitely higher than an old one. Therefore, the temperature-based contact points will be clustered with multiple overlaps in high temperature regions, resulting in distorted and non-smooth interpolated trajectories. In contrast, the trajectory obtained by method 3 has significant advantages in terms of precision and smoothness.

9 Discussion and conclusion

This work proposed a multimodal vision-based robot teaching approach. By using RandLANet, Hand/object semantic segmentation is performed on multimodal 3D image data containing temperature, color and geometric features. Then a dynamic analysis for meshless 3D temperature field is achieved by our elimination method. Furthermore, the hand/object contact point is precisely estimated based on Bayesian theory. The experimental results show that based on our method, the multimodal information is sufficiently extracted, and the resulting robot motion trajectory has good accuracy (mean deviation: 2.185 mm) and smoothness.

We consider that for physical quantities such as temperature, for which the derivative related to time and spatial variables has a constant relationship, we should explore more deeply the useful information hidden behind them, rather than handling them as static features in the same way as color. This is a remarkable difference between multi-modal image processing and multi-channel or multi-spectral image processing.

In our method, semantic segmentation is achieved using a deep neural network technique and analysis of the temperature field is realized based on a traditional method (heat transfer equation). We believe that it is worth exploring to choose the occasion for neural network technology rationally when it is so widely applied nowadays. For example, when a validated physical model is already existent for temperature field analysis, traditional methods should be chosen. This will improve the interpretability of the entire system and reduce the strong dependence on datasets in similar deep neural networks.

The experiments proved that our schema works. It also raises a common problem for point cloud processing that the adjust of some parameters such as the search radius for neighborhood, is still based on experience. Also, the search radius in the RandLANet is actually a hyper-parameter that needs to be adjusted artificially. In the future, if an adaptively adjustable parameter mechanism can be devised, the technical barrier for robot teaching will be lowered even further.

Moreover, this article represents only the first step of our Long-Term plan. Our future goal is to develop a self-heating pen that is much cheaper than Wandelbot's TracePen, yet provides significantly higher accuracy than the finger-based method described in this article. Currently, a significant cause of trajectory errors (mean deviation: 2.185 mm) is due to the width of the finger, which is typically about 1 cm. A self-heating pen with a very fine nib can significantly improve accuracy. Additionally, a pen with adjustable temperature can help the system adapt to the effects of varying ambient temperatures. We are planning to increase the accuracy of this system to meet the low accuracy requirements of some industrial manufacturing scenarios. In such cases, implementing non-contact methods may be difficult to achieve.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Author contributions

GN contributed to Conceptualization, supervision, project administration, funding and acquisition. YZ contributed to methodology, software, validation, formal analysis, visualization and writing—original draft preparation. YZ and RF contributed to investigation, resources and data curation. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

This research was funded by Group for Quality Assurance and Industrial Image Processing, Technische Universität Ilmenau.

Acknowledgments

We thank Robin Horn and Zheng Liu for helping us revise the manuscript. We thank Zheng Liu for contributing his wisdom to construct this method. We acknowledge support for the publication costs by the Open Access Publication Fund of the Technische Universität Ilmenau.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2023.1120357/full#supplementary-material

References

Abbas, S. M., Hassan, S., and Yun, J. (2012). "Augmented reality based teaching pendant for industrial robot," in *2012 12th international conference on control, automation and systems* (IEEE), 2210–2213.

Baek, S., Kim, K. I., and Kim, T.-K. (2018). "Augmented skeleton space transfer for depth-based hand pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8330–8339.

Braeuer-Burchardt, C., Siegmund, F., Hoehne, D., Kuehmstedt, P., and Notni, G. (2020). "Finger pointer based human machine interaction for selected quality checks of industrial work pieces," in *ISR 2020; 52th international symposium on Robotics (VDE)*, 1–6.

Cheng, W., Park, J. H., and Ko, J. H. (2021). "Handfoldingnet: A 3d hand pose estimation network using multiscale-feature guided folding of a 2d hand skeleton," in *Proceedings of the IEEE/CVF international conference on computer vision*, 11260–11269.

Du, G., Chen, M., Liu, C., Zhang, B., and Zhang, P. (2018). Online robot teaching with natural human–robot interaction. *IEEE Trans. Industrial Electron.* 65, 9571–9581. doi:10.1109/tie.2018.2823667

FLIR (2022). Flir a35 product overview. available at: https://www.flir.com/products/a35/(accessed: december 05, 2022).

Genie (2022). Genie nano c1280 product overview. available at: https://www.edmundoptics.de/p/c1280-12-color-dalsa-genie-nano-poe-camera/4049/(accessed: december 05, 2022).

Halim, J., Eichler, P., Krusche, S., Bdiwi, M., and Ihlenfeldt, S. (2022). No-Code robotic programming for agile production: A new markerless-approach for multimodal natural interaction in a human-robot collaboration context. *Front. Robotics AI* 9, 1001955. doi:10.3389/frobt.2022.1001955

Heist, S., Dietrich, P., Landmann, M., Kühmstedt, P., Notni, G., and Tünnermann, A. (2018). Gobo projection for 3d measurements at highest frame rates: A performance analysis. *Light: Sci. Appl.* 7, 71–13. doi:10.1038/s41377-018-0072-3

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., et al. (2020). "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11108–11117.

Jen, Y. H., Taha, Z., and Vui, L. J. (2008). Vr-based robot programming and simulation system for an industrial robot. *Int. J. Industrial Eng.* 15, 314–322.

Jeon, E. S., Kim, J. H., Hong, H. G., Batchuluun, G., and Park, K. R. (2016). Human detection based on the generation of a background image and fuzzy system by using a thermal camera. *Sensors* 16, 453. doi:10.3390/s16040453

Liu, Y., Kukkar, A., and Shah, M. A. (2022). Study of industrial interactive design system based on virtual reality teaching technology in industrial robot. *Paladyn, J. Behav. Robotics* 13, 45–55. doi:10.1515/pjbr-2022-0004

Manou, E., Vosniakos, G.-C., and Matsas, E. (2019). Off-line programming of an industrial robot in a virtual reality environment. *Int. J. Interact. Des. Manuf. (IJIDeM)* 13, 507–519. doi:10.1007/s12008-018-0516-2

Osokin, D. (2018). *Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. arXiv preprint arXiv:1811.12004*.

Pettersen, T., Pretlove, J., Skourup, C., Engedal, T., and Lokstad, T. (2003). "Augmented reality for programming industrial robots," in *The second IEEE and ACM international symposium on mixed and augmented reality, 2003. Proceedings* (IEEE), 319–320.

Pratticò, F. G., and Lamberti, F. (2021). Towards the adoption of virtual reality training systems for the self-tuition of industrial robot operators: A case study at kuka. *Comput. Industry* 129, 103446. doi:10.1016/j.compind.2021.103446

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. neural Inf. Process. Syst.* 30.

Stadler, S., Kain, K., Giuliani, M., Mirnig, N., Stollnberger, G., and Tscheligi, M. (2016). "Augmented reality for industrial robot programmers: Workload analysis for task-based, augmented reality-supported robot control," in *2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN)* (IEEE), 179–184.

Su, Y.-H., Chen, C.-Y., Cheng, S.-L., Ko, C.-H., and Young, K.-Y. (2018). "Development of a 3d ar-based interface for industrial robot manipulators," in *2018 IEEE international conference on systems, man, and cybernetics (SMC)* (IEEE), 1809–1814.

Wandelbots (2022). Wandelbots teaching product overview. available at: https://wandelbots.com/en/(accessed: december 05, 2022).

Yap, H. J., Taha, Z., Md Dawal, S. Z., and Chang, S.-W. (2014). Virtual reality based support system for layout planning and programming of an industrial robotic work cell. *PloS one* 9, e109692. doi:10.1371/journal.pone.0109692

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., et al. (2020). *Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214*.

Zhang, Y., Müller, S., Stephan, B., Gross, H.-M., and Notni, G. (2021). Point cloud hand–object segmentation using multimodal imaging with thermal and color data for safe robotic object handover. *Sensors* 21, 5676. doi:10.3390/s21165676