J Supercomput (2013) 64:383–408
DOI 10.1007/s11227-012-0781-y

Thermal camera networks for large datacenters using real-time thermal monitoring mechanism

Hang Liu · Eun Kyung Lee · Dario Pompili · Xiangwei Kong

Published online: 8 June 2012

Abstract Thermal cameras provide fine-grained thermal information that enables monitoring and autonomic thermal management in large datacenters. The real-time thermal monitor network employing thermal cameras is proposed to cooperatively localize hotspots and extract their characteristics (i.e., temperature, size, and shape). These characteristics are adopted to classify the causes of hotspots and make energy-efficient thermal management decisions such as job migration. Specifically, a sculpturing algorithm for extracting and reconstructing shape characteristics of hotspots is proposed to minimize the network overhead. Experimental results show the validity of all the algorithms proposed in this paper.

Keywords Data center · Thermal management · Thermal camera networks · Energy consumption · Fine-grained data collection

This work was performed while H. Liu was visiting the CPS Lab, Rutgers University.

H. Liu (✉)
College of Electronic Science and Technology, Dalian University of Technology, 2 Linggong Road, Dalian, Liaoning, 116023, China
e-mail: liuhang@dlut.edu.cn

E.K. Lee · D. Pompili
NSF Center for Autonomic Computing, Department of Electrical and Computer Engineering, Rutgers University, 94 Brett Road, Piscataway, NJ, 08854, USA

E.K. Lee
e-mail: eunkyung_lee@cac.rutgers.edu

D. Pompili
e-mail: pompili@cac.rutgers.edu

X. Kong
School of Information and Communication Engineering, Dalian University of Technology, 2 Linggong Road, Dalian, Liaoning, 116023, China
e-mail: kongxw@dlut.edu.cn

1 Introduction

The last few years have seen a dramatic increase in the number, size, and use of datacenters. Power consumption and heat management have emerged as key design challenges in creating new datacenter architectures. Based on previous investigations [14], it can be assumed that datacenters today account for almost 2 % of all electricity consumed in the US growing at a rate of more then 12 % each year. A significant fraction of the power consumption, i.e., up to 50 %, is directed to cooling the heat generating equipment [10]. Although recently built datacenters exhibit better cooling efficiency, cooling energy consumption is still significant [32]. One of the causes of energy cost for cooling is that many current datacenters are not equipped with elaborate sensing infrastructure that monitor the phenomenon of "heat transfer" to make cooling decisions. Since heat transfers in various forms (i.e., convection, conduction, and radiation), it is complicated to model and predict the future thermal behavior of a datacenter. Most of the time, datacenters only have sparsely-deployed global sensing devices to aid in adjusting their cooling solutions. This lack of knowledge impairs the cooling cycle of datacenters because heat imbalances and localized hotspots [16] may not be detected using only few global sensors. These hotspots eventually interfere with cold air that constantly extracts heat [19], resulting in increasing cooling costs.

This paper addresses the autonomic thermal monitoring mechanism for datacenters, which is based on the real-time heat distribution in datacenters to optimize performance in terms of energy consumption, and throughput. Specifically, this paper focuses on characteristics of hotspots which can be obtained by the thermal monitor network to improve the overall efficiency of the cooling system. The main tasks of the thermal monitor network are to detect and localize hotspots, and extract their characteristics for thermal management.

In Fig. 1, we introduce the architecture for the thermal monitor network, where two wireless sensor sub-networks with different sensors are shown. The first sub-

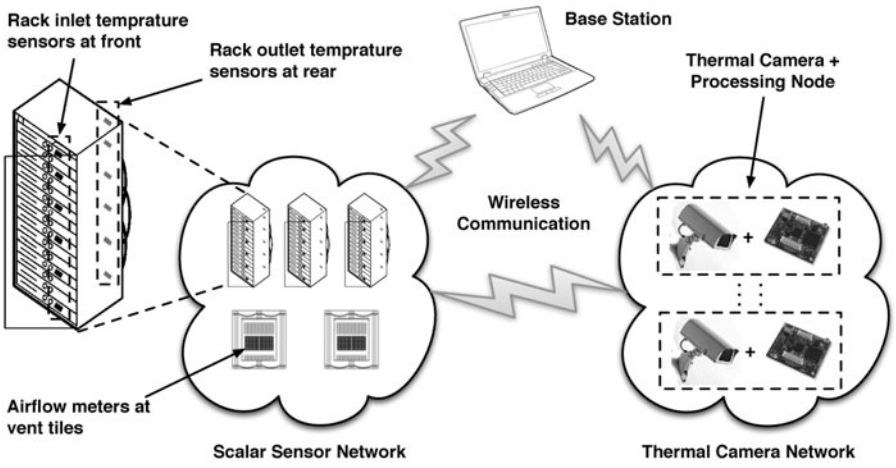


Fig. 1 The architecture of the thermal monitor network proposed in this paper

network on the left shows a scalar sensor network in which the nodes are deployed at significant spots to monitor temperature, humidity, airflow, etc. The scalar sensor network completes the same functions as that in [1]. For example, temperature sensors are deployed in the front and back of the chassis to measure the inlet and outlet temperature; airflow meters are deployed at the vent tiles to measure the airflow into the datacenter. Most of thermal-aware scheduling algorithms are based on these measured values, such as Uniform Outlet Profile and Minimal Computing Energy [20]. To further analyze the temperature environment in datacenters, the other subnetwork on the right is proposed. It shows a thermal camera network (TCN) that is composed of thermal camera nodes (multiple thermal cameras + processing nodes) on the ceiling in order to thermally "see" the datacenter in high granularity and process thermal images so to aid in efficient cooling of the datacenter. Thermal cameras in the TCN capture these hotspots and convert the data to corresponding images of temperature, so that each pixel in the image represents temperature data, providing detailed information about heat distribution. The TCN collects this *raw data*, extracts *characteristics* through in-network processing, and then enables acquisition of *knowledge* about hotspots (such as the causes for hotspots) inside the datacenter. This solution can be applied to not only datacenters, but also other applications that utilize multiple thermal cameras.

When a hotspot is detected, an event is triggered, for example, to locate a resource with the lowest temperature for job reallocation. A part of the workloads (e.g., jobs at the hotspot) may be migrated to keep the operating conditions within the desired constraints for a longer time without requiring expensive global allocation across the whole datacenter. The initial cost for multiple thermal camera nodes may be expensive as more than one thousand dollars each, but total operational cost for cooling the datacenter can be reduced more by deploying a TCN. Contributions of this paper are summarized as follows:

1. A thermal monitor network that is a new paradigm using multiple thermal camera nodes for datacenters is proposed.
2. Cooperative detection and localization algorithm for hotspots using the TCN is proposed.
3. A novel algorithm (sculpturing algorithm) is proposed for compression and reconstruction of hotspots' shape.

The remainder of this paper is organized as follows. In Sect. 2, we summarize related work. In Sect. 3, we propose algorithms to detect and localize hotspots using multiple camera nodes cooperatively. In Sect. 4, we propose extraction and reconstruction of characteristics of hotspots using the TCN. Specially, we propose the sculpturing algorithm to minimize shape data of hotspots. We also discuss simulation and experimental results in both Sects. 3 and 4, and finally summarize and conclude our work in Sect. 5.

2 Related work

In this section, we review previous topics that are relevant to our work: thermal mapping for datacenters, cooperative detection algorithms, cooperative localization algorithms, and extraction and reconstruction algorithms for shape.

Thermal mapping for datacenters A thermal map that is the temperature field in the datacenter space can enable holistic IT-facilities design, increase hardware reliability, decrease cooling costs, increase compaction, and improve operational efficiencies. A datacenter monitoring system, for example, either a temperature sensor-based monitoring service [31] or by using compute fluid dynamics (CFD) software [2, 5], can generate a thermal map for a datacenter. Some research [19, 31] claims that the CFD-based model is too complex and is not suitable for real-time scheduling in a datacenter.

Measurement-based modeling for a datacenter is used to reduce the complexity in the specifications of the model [12]. Based on the integration of embedded sensors with computational models and workload schedulers, it is used to watch for hotspots to improve the overall efficiency of the cooling system. Workload schedulers use online information about the datacenter operating conditions obtained from the sensors to generate appropriate management policies. Furthermore, local processing within the sensor network is used to enable timely responses to changes in operating conditions and determine job migration strategies, such as Uniform Outlet Profile (UOP), Minimal Computing Energy (MCE), and Uniform Task (UT) [20]. Patel et al. present some of the early work in datacenter monitoring and management [23, 24, 30] identifying some key inefficiencies in cooling and CRAC configurations. Several studies explore techniques for efficient energy and thermal provisioning in data centers [3, 6]. Bash et al. [1], also consider efficient datacenter monitoring by selective sensor sampling and determining the best set of fixed sensor locations, while our technique is based on fixed sensor and pan-tilt sensing, which can complement static instrumentation. Some researchers have explored robotic monitoring approaches for different environments. Patel et al. [22, 24] argue for robotic monitoring and management for data centers, but do not describe a design or provide any implementation details. Pon et al. [25] describe a cable-based robot for monitoring environmental data in rivers and forests. Lenchner et al. [17] describe a robot that serves as a physical autonomic element, capable of navigating, mapping, and monitoring data centers with little or no human involvement. To navigate, the robot takes advantage of the square grid formed by industry standard $20' \times 20'$ datacenter floor tiles to move one floor tile at a time, continuously feeling its way around the datacenter and keeping track of where it has already been, eventually visiting all unobstructed tiles. Whenever it visits a tile, the robot stops at its center, and takes a set of sensor readings at various heights. The total scan time is about tens of minutes [17], thus the robot can only be used to measure stable temperature environment in datacenters, while more attention is paid on hotspots formed in a short time in this paper.

Cooperative detection and localization algorithms Although a few recent papers are specifically concerned with cooperative detecting methods, they focus on a 2D detecting problem of how to cover a planar region which is typical of a building's floor plan. In [7], they focus on the camera placement problem, where the goal is to determine the optimal positioning and number of cameras for a floor plan region to be observed. The method in [7] can find targets on the floor such as a human, however, it is incapable of detecting levitating hotspots because it lacks a 3D detection capability. In this paper, the TCN needs to scan the entire 3D space to detect hotspots.

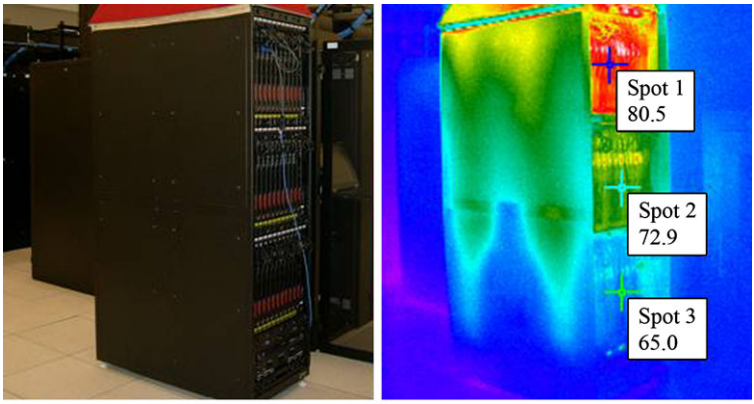


Fig. 2 The visible image and the thermal image of the same object

As one thermal camera cannot range (find the distance to) an object, two or more cameras are needed to find a hotspot's location using the principle of computational stereo vision. This principle has been known for more than 20 years and involves two main problems which have been extensively studied, namely the correspondence problem (i.e., match points on one image to their corresponding points on the other) and the reconstruction problem (i.e., construct 3D map from matched points in 2D images). Among these two problems, the correspondence problem is more challenging. If several cameras find a hotspot, they must compare its features, to make sure they have found the same one before they localize it. A common strategy is to match distinct features that are extracted from these images. The features should be robust enough to avoid distortion from noise and geometric transformation. Many robust local feature detectors for visible images exist (e.g., Harris corner detector [11], the scale invariant feature transform [18], Kadir's feature detector [13]). However, visible images (such as Fig. 2(a)) include many distinct corners and edges, while thermal images (such as Fig. 2(b)) do not. Thus, the above detectors are not capable of accurately detecting good features in thermal images because hotspots do not include any distinct corners or edges [8].

A few recent papers are specifically concerned with computational stereo vision problems in thermal images. The literature on the subject is extremely limited. The technique proposed in [26] recovers a 3D temperature map from a pair of calibrated thermal cameras using epipolar lines. However, estimating an epipolar line is difficult because it relies on correspondence in blurry thermal images which lack robust features. The localization system used in [8] and [9] is a pair of adjacent thermal cameras. However, the initial cost of deploying pairs of adjacent thermal cameras is double when compared to implementing the TCN that uses one camera to cover the same area. Their approach is too expensive considering a large datacenter and the cost of thermal cameras. Thus, the TCN is more suitable to detect and localize hotspots in a wide area.

Extraction and reconstruction algorithms for shape Hotspots can be stored using common image compression algorithms, such as the Joint Photographic Experts Group (JPEG) [34], Embedded Zero-tree Wavelet (EZW) [29], Set-Partitioning in Hierarchical Trees (SPIHT) [28], Embedded Block Coding with Optimized Truncation (EBCOT) [33], pyramidal coding [15], directional decomposition based coding [35], segmentation based coding [27], and vector quantization [21]. Based on the results of the evaluation conducted with the above eight popular image compression algorithms, it is found in [4] that SPIHT wavelet-based image compression is the most suitable image compression algorithm to implement for a wireless sensor network in a hardware constrained environment. However, a series of shape data at different times are required to monitor the state change of each hotspot. If these shape data are saved as an image compression algorithm, the size of compressed data is still large.

Also, a hotspot can be represented by a similar geometrical figure with fewer parameters, such as a corresponding ellipse or the smallest convex polygon. However, the similar geometrical figure cannot well represent the details of the edge.

3 Cooperative detection and localization algorithm

The TCN consists of many thermal camera nodes with pan and tilt capability to increase the range of sight, such as C_i, C_j, and C_k shown in Fig. 3. Even though cameras can pan and tilt, they cannot see the full extent of the monitored space because of their small viewing angle and the shading effect. Moreover, hotspots may levitate in the air making "phantom" hotspots such as in Fig. 4. If camera C_i finds a hotspot H, C_j finds another hotspot H_j, C_k finds another hotspot H_k, and $\updownarrow_{C_i H}$ intersects $\updownarrow_{C_j H_j}$ or $\updownarrow_{C_k H_k}$; a "phantom" hotspot appears at that intersection. Thus, in order to detect hotspots rapidly and accurately, the thermal camera nodes must cooperatively detect all possible hotspots in a 3D space.

3.1 Cooperative detection of hotspots

According to the principle of computational stereo, the absolute coordinates of a point can be calculated by at least two cameras. The precondition for cooperatively

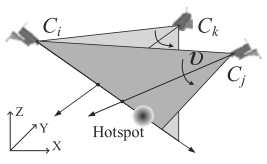


Fig. 3 Cooperative detection and localization of a hotspot

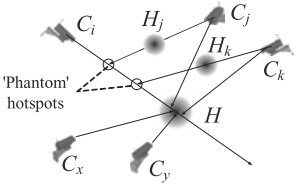


Fig. 4 The real hotspots and 'phantom' hotspots

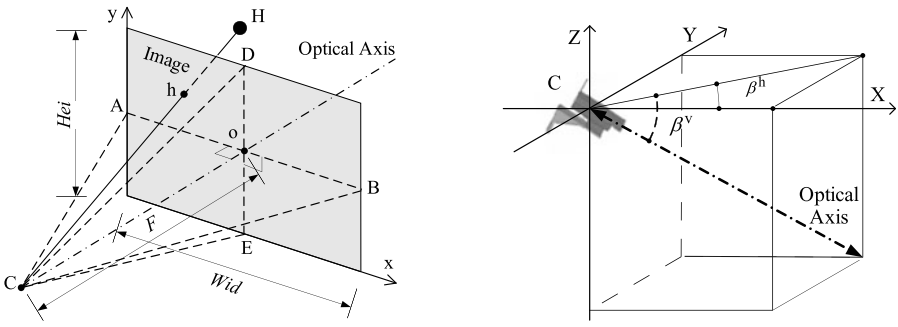


Fig. 5 (**a**): Parameters of the thermal cameras, where $\angle ACB = \alpha^h$, $\angle DCE = \alpha^v$, $p(h) = [x_h, y_h]^T$ and $P(H) = [X_H, Y_H, Z_H]^T$. (**b**): The horizontal and vertical angle of optical axis of the thermal camera

localizing a hotspot in 3D is that some thermal camera nodes find the same hotspot. For example, if a thermal camera node finds a hotspot, H, from a thermal image, it will transmit the information of H to other cameras, which will detect H according to the information. Because of the restriction of a small viewing angle, one thermal camera cannot see the full extent of the monitored space. It must take a long time for a camera to scan the whole 3D space and then compare all detected hotspots to H and choose the corresponding spot of H. Thus, we propose a method to simplify cooperative detection from 3D to 2D. In this paper, H represents the center of the real hotspot (3D), and h represents the projection of that hotspot in the thermal image (2D).

Parameters used for cooperative detection are explained in this paragraph. As shown in Fig. 5(a), focal length of the thermal cameras is denoted as F, horizontal angle of view is α^h, vertical angle of view is α^v, and principal point in the image is denoted o. $p(q)$ denotes the 2D coordinates of the point q in the thermal image, namely $p(q) = [x_q, y_q]^T$, and $P(Q)$ denotes the absolute coordinates (3D) of the point Q, namely $P(Q) = [X_Q, Y_Q, Z_Q]^T$. Thus, the coordinates of o in the thermal image is $p(o) = [Wid/2, Hei/2]^T$ where the resolution of the thermal cameras' detectors is $Wid \times Hei$ pixels. All above parameters can be obtained from the specification manual of the thermal camera. Moreover, we assume that the horizontal and vertical angle of optical axis are β_i^h and β_i^v (Fig. 5(b)), and all the locations of the center of the thermal camera nodes, $P(C_i) = [X_{C_i}, Y_{C_i}, Z_{C_i}]^T$, where $i = 1, 2, \ldots, N$, are known.

Once C_i finds a hotspot H, it calculates the straight line $\updownarrow_{C_i H}$ that passes C_i and H. Then it estimates the 2D coordinates of the hotspot center in the thermal image, $p(h)$. An isotherm line is used to estimate the hotspot center in the thermal images. The isotherm line is a line connecting points of equal temperature. The edge of the hotspot in the thermal image is defined as the isotherm line at a specific temperature. Let ξ denote the area of the hotspot in the image, and $T(x, y)$ denote the temperature

of pixel $[x, y]^T$ in the image. The coordinates of the hotspot center $p(h)$ in the image are defined as following:

$$p(h) = \begin{bmatrix} x_h \\ y_h \end{bmatrix} = \frac{1}{\sum_{(x,y)\in\xi} T(x,y)} \begin{bmatrix} \sum_{(x,y)\in\xi} (T(x,y)\cdot x) \\ \sum_{(x,y)\in\xi} (T(x,y)\cdot y) \end{bmatrix}. \tag{1}$$

Since the intersection of $\updownarrow_{C_iH}$ and the image plane is h, $\updownarrow_{C_iH}$ and $\updownarrow_{C_ih}$ are the same straight line. Then the horizontal and vertical angles between $\updownarrow_{C_iH}$ and the optical axis $\updownarrow_{C_io}$, γ^h and γ^v are

$$\begin{cases} \gamma^h = \arctan((x_h - Wid/2)/F), \\ \gamma^v = \arctan((y_h - Hei/2)/F). \end{cases} \tag{2}$$

So the horizontal and vertical angle of $\updownarrow_{C_iH}$ are

$$\begin{cases} \beta^h_{\updownarrow_{C_iH}} = \beta^h - \gamma^h, \\ \beta^v_{\updownarrow_{C_iH}} = \beta^v - \gamma^v \end{cases} \tag{3}$$

respectively. Thus, $\updownarrow_{C_iH}$ can be described as

$$\updownarrow_{C_iH} : \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\beta^v_{\updownarrow_{C_iH}} \cdot \cos\beta^h_{\updownarrow_{C_iH}} \\ \cos\beta^v_{\updownarrow_{C_iH}} \cdot \sin\beta^h_{\updownarrow_{C_iH}} \\ -\sin\beta^v_{\updownarrow_{C_iH}} \end{bmatrix} \cdot \boldsymbol{t} + \begin{bmatrix} X_{C_i} \\ Y_{C_i} \\ Z_{C_i} \end{bmatrix} \tag{4}$$

where $\boldsymbol{t}$ is an independent variable. Let $\mathbf{l}_i$ be the direction vector of $\updownarrow_{C_iH}$, namely

$$\mathbf{l}_i = \begin{bmatrix} \cos\beta^v_{\updownarrow_{C_iH}} \cdot \cos\beta^h_{\updownarrow_{C_iH}} \\ \cos\beta^v_{\updownarrow_{C_iH}} \cdot \sin\beta^h_{\updownarrow_{C_iH}} \\ -\sin\beta^v_{\updownarrow_{C_iH}} \end{bmatrix}, \tag{5}$$

so

$$\updownarrow_{C_iH} = \mathbf{l}_i \cdot \boldsymbol{t} + P(C_i). \tag{6}$$

Let the set $\tilde{C}$ consist of the cameras that can connect to C_i directly (in single hop). Then C_i will transmit relevant data to the cameras in $\tilde{C}$. The data includes the location of C_i and the direction vector of the straight line $\updownarrow_{C_iH}$, namely $P(C_i)$ and $\mathbf{l}_i$.

If a camera C_j, $(C_j \in \tilde{C})$, receives the data from C_i, it will calculate the distance between itself and $\updownarrow_{C_iH}$, denoted as $\|C_j, \updownarrow_{C_iH}\|$. The threshold TH_1 is used to avoid selecting the cameras which are too close to $\updownarrow_{C_iH}$. If $\|C_j, \updownarrow_{C_iH}\| > TH_1$, the lines $\updownarrow_{C_iH}$ and $\updownarrow_{C_jH}$ can form a certain angle to cooperatively localize, otherwise too small of an angle will lead to more error in localization. The reason is similar to that of the pair of adjacent cameras which will be explained in Sect. 3.3.1.

We define $P(C_j) = [X_{C_j}, Y_{C_j}, Z_{C_j}]^T$ and $\updownarrow_{C_iH} = \mathbf{l}_i x + P(C_i)$, and let $\mathbf{l}_i = [X_{\mathbf{l}_i}, Y_{\mathbf{l}_i}, Z_{\mathbf{l}_i}]^T$. So there is an exclusive plane υ passing C_j and $\updownarrow_{C_iH}$. The normal vector of υ is $\mathbf{w} = \mathbf{l}_i \times \updownarrow_{C_iC_j}$, namely

$$\mathbf{w} = \begin{bmatrix} X_{\mathbf{w}} \\ Y_{\mathbf{w}} \\ Z_{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} Y_{\mathbf{l}_i}(Z_{C_i} - Z_{C_j}) - Z_{\mathbf{l}_i}(Y_{C_i} - Y_{C_j}) \\ Z_{\mathbf{l}_i}(X_{C_i} - X_{C_j}) - X_{\mathbf{l}_i}(Z_{C_i} - Z_{C_j}) \\ X_{\mathbf{l}_i}(Y_{C_i} - Y_{C_j}) - Y_{\mathbf{l}_i}(X_{C_i} - X_{C_j}) \end{bmatrix}. \tag{7}$$

Because υ passes through C_j and H, C_j detecting H in υ becomes a 2D problem (Fig. 3). In order to achieve the purpose mentioned above, the camera C_j must control the horizontal and vertical angle of the optical axis $\updownarrow_{C_j o}$, namely β_j^h and β_j^v. The direction vector of $\updownarrow_{C_j o}$ is $\mathbf{l}_j$. If $\updownarrow_{C_i o}$ is in υ, then $\mathbf{l}_j \perp \mathbf{w}$, namely

$$\mathbf{l}_j \cdot \mathbf{w} = 0. \tag{8}$$

So from (5) and (8), it is obtained that

$$X_{\mathbf{w}} \cos\beta_j^v \cos\beta_j^h + Y_{\mathbf{w}} \cos\beta_j^v \sin\beta_j^h - Z_{\mathbf{w}} \sin\beta_j^v = 0. \tag{9}$$

So the relationship between β_j^h and β_j^v is as follows:

$$\beta_j^v = \arctan \frac{X_{\mathbf{w}} \cos\beta_j^h + Y_{\mathbf{w}} \sin\beta_j^h}{Z_{\mathbf{w}}}. \tag{10}$$

C_j will control β_j^h and β_j^v according to above relationship to detect H in plane υ.

3.2 Cooperative localization of hotspots

The next is to calculate the hotspot center's absolute coordinates which can be represented as the following function $\boldsymbol{f}(.)$ of relevant parameters:

$$P(H) = [X_H, Y_H, Z_H]^T = \boldsymbol{f}\big(Wid, Hei, F, p(o), \alpha^h, \alpha^v, P(C_n), \beta_n^h, \beta_n^v, p(h_n), N\big) \tag{11}$$

where N is the number of the thermal cameras used to localize a hotspot, and n is the index of the camera, $n = 1, 2, \ldots, N$. Since there are multiple thermal camera nodes in the TCN, we propose a method which can localize hotspots accurately with three cameras, namely $N = 3$.

The thermal camera C_i transmits data to cameras in $\tilde{C}$. Then these cameras detect the hotspot with the above method. To avoid "phantom" hotspot, the following process is implemented. If C_j ($C_j \in \tilde{C}$) finds a hotspot H'', it will calculate the straight line, $\updownarrow_{C_j H''}$, which passes C_j and the hotspot using Eqs. (1)–(6), and then transmits a set of data to C_i. The data includes $P(C_j)$, the direction vector of $\updownarrow_{C_j H''}$ and the intersection of $\updownarrow_{C_j H''}$ and $\updownarrow_{C_i H}$. If $\updownarrow_{C_j H''}$ and $\updownarrow_{C_i H}$ don't intersect, the intersection is defined as the point that is the intersection of their common perpendicular and $\updownarrow_{C_i H}$. If C_j finds two or more hotspots such as C_j or C_k in Fig. 4, it will transmit all the data of these hotspots to C_i. However, only one of them is H, and others can generate "phantom" hotspots. C_i can receive sets of data from some cameras and some intersections are generated on $\updownarrow_{C_i H}$. In normal cases, the straight lines intersecting $\updownarrow_{C_i H}$ at H are more than the ones intersecting at a "phantom" hotspot. On other words, the intersections related to H are more, and they are closer apart. So two intersections between which the distance is shortest are the most possible points related to H. We assume they are the intersections of $\updownarrow_{C_i H}$ and two lines ($\updownarrow_{C_j H}$ and $\updownarrow_{C_k H}$). Then the node C_i starts to calculate the absolute coordinates of H with the three lines, namely $\updownarrow_{C_i H}$, $\updownarrow_{C_j H}$, and $\updownarrow_{C_k H}$, and the other cameras will go on searching other hotspots.

The method of calculating the coordinates with three cameras is provided as following. If 3 cameras obtain a particular hotspot, 3 straight lines can be obtained using

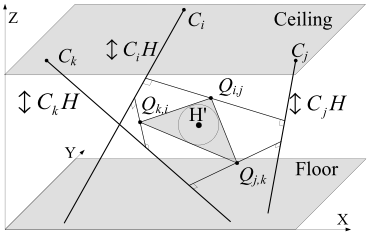


Fig. 6 Estimating the center of detected hotspot

Eqs. (1)–(6), namely $\updownarrow_{C_iH}$, $\updownarrow_{C_jH}$ and $\updownarrow_{C_kH}$. The intersecting point of these lines is the center of the hotspot, namely H. However, in practice, these lines cannot intersect at a point because of the effect of error and noise. Thus, the next step is to estimate $P(H)$. Let $\|Q, \updownarrow_{AB}\|$ denote the distance from a point Q to a straight line $\updownarrow_{AB}$. H is approximated to the point H' which minimizes $\sum_{n=i,j,k} \|H', \updownarrow_{C_nH}\|$. For convenience in calculation, $P(H')$ is estimated by the following procedures.

We assume two points, Q_i and Q_j, are a pair of closest points which are on $\updownarrow_{C_iH}$ and $\updownarrow_{C_jH}$, respectively. In other words, Q_i and Q_j are the intersections of the common perpendicular of $\updownarrow_{C_iH}$ and $\updownarrow_{C_jH}$. So

$$\begin{cases} (P(Q_i) - P(Q_j)) \cdot \mathbf{v}_i = 0, \\ (P(Q_i) - P(Q_j)) \cdot \mathbf{v}_j = 0, \end{cases} \tag{12}$$

where $P(Q_i) = P(C_i) + \|C_i, Q_i\| \cdot \mathbf{v}_i$, $P(Q_j) = P(C_j) + \|C_j, Q_j\| \cdot \mathbf{v}_j$, and $\mathbf{v}_i$ and $\mathbf{v}_j$ are the direction vectors of $\updownarrow_{C_iH}$ and $\updownarrow_{C_jH}$. The coordinates of Q_i and Q_j can be obtained by solving (12). They are

$$\begin{cases} P(Q_i) = \frac{(P(C_j) - P(C_i)) \cdot [(\mathbf{v}_j \cdot \mathbf{v}_i)\mathbf{v}_j - (\mathbf{v}_j \cdot \mathbf{v}_j)\mathbf{v}_i]}{(\mathbf{v}_i \cdot \mathbf{v}_j)^2 - (\mathbf{v}_i \cdot \mathbf{v}_i) \cdot (\mathbf{v}_j \cdot \mathbf{v}_j)} \cdot \mathbf{v}_i + P(C_i), \\ P(Q_j) = \frac{(P(C_i) - P(C_j)) \cdot [(\mathbf{v}_i \cdot \mathbf{v}_j)\mathbf{v}_i - (\mathbf{v}_i \cdot \mathbf{v}_i)\mathbf{v}_j]}{(\mathbf{v}_j \cdot \mathbf{v}_i)^2 - (\mathbf{v}_j \cdot \mathbf{v}_j) \cdot (\mathbf{v}_i \cdot \mathbf{v}_i)} \cdot \mathbf{v}_j + P(C_j). \end{cases} \tag{13}$$

The coordinates of the midpoint between Q_i and Q_j, denoted as $Q_{i,j}$ are

$$P(Q_{i,j}) = \big(P(Q_i) + P(Q_j)\big)/2. \tag{14}$$

If there are only two thermal camera nodes, $Q_{i,j}$ can be considered as the hotspot center. However, two cameras bring significant error to the coordinates. If error is too large, the thermal management system cannot cool the hottest region. Since there are multiple thermal camera nodes in the TCN, three cameras are used to improve the precision of measurement which has been proved by experiments below. If three cameras find the hotspot, 3 points are obtained, $Q_{i,j}$, $Q_{j,k,}$ and $Q_{k,i}$, from lines, $\updownarrow_{C_iH}$, $\updownarrow_{C_jH}$ and $\updownarrow_{C_kH}$. The incenter, namely the center of the inscribed circle, of the triangle $\triangle Q_{i,j} Q_{j,k} Q_{k,i}$ is considered as the point H' (Fig. 6). So that

$$\begin{aligned} P(H') &= \begin{bmatrix} X_{H'} \\ Y_{H'} \\ Z_{H'} \end{bmatrix} \\ &= \frac{[P(Q_{i,j}), P(Q_{j,k}), P(Q_{k,i})] \times [\|Q_{j,k}, Q_{k,i}\|, \|Q_{k,i}, Q_{i,j}\|, \|Q_{i,j}, Q_{j,k}\|]^T}{\|Q_{j,k}, Q_{k,i}\| + \|Q_{k,i}, Q_{i,j}\| + \|Q_{i,j}, Q_{j,k}\|} \end{aligned} \tag{15}$$

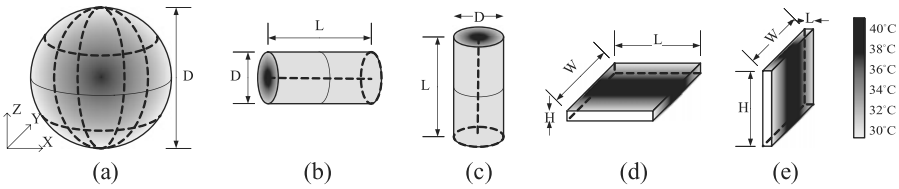


Fig. 7 Types of hotspot. (**a**) "Sphere"; (**b**) "horizontal cylinder"; (**c**) "vertical cylinder"; (**d**) "horizontal plan"; (**e**) "vertical plane"

So far, the absolute coordinates of the hotspot center have been detected and localized in a 3D space without assuming any a priori information about the layout of datacenter. However, the performance of the proposed solution (in terms of minimizing detection time and error distance) for hotspot detection and localization can be improved if we have a priori information. We are currently working on evaluating the proposed techniques on a real testbed at the NSF Center for Autonomic Computing, Rutgers University, with a priori information about our machine room layout.

3.3 Experimental evaluation

In this section, we present detailed simulations and experiments of TCNs. We use MATLAB to simulate cooperative detection and localization algorithms. Each simulation and experiment is performed case by case to show the performance of TCNs.

The system parameters in simulation are as follows. The size of the datacenter is 20 m $\times$ 20 m $\times$ 2 m. There are 4 thermal camera nodes at the ceiling corners. All the cameras can pan and tilt automatically, and the rotation angles are known. The maximum error of rotation angles is 0.5°. In practice, a thermal camera node contains a FLIR PathFindIR thermal camera and an advanced wireless sensor node platform, Intelmote2. The thermal camera has a temperature measurement range of -40 to 80 °C with temperature resolution less than ± 1 °C. It has a low resolution uncooled microbolometer detector, and can display thermal images of up to $Wid \times Hei$ pixels ($Wid = 320$ and $Hei = 240$). Each image pixel provides temperature data, and the accuracy is ± 2 °C. Focal length of the thermal cameras is $F = 50$ mm, horizontal angle of view is $\alpha^h = 36°$, and vertical angle of view is $\alpha^v = 27°$.

3.3.1 Performance of the TCN compared to other systems [simulation]

To compare the performance of the proposed TCN in terms of localization accuracy, other localization systems are used for our simulation. The first system localize hotspots using 2 thermal camera nodes deployed sparsely, and the second system using a pair of adjacent thermal camera nodes at the spacing of 20 cm.

Hotspots are randomly generated for testing. Since the shape of the hotspot can affect estimation of hotspot center in thermal image, 6 types of hotspots are tested, which are "sphere," "horizontal cylinder," "vertical cylinder," "horizontal plane," "vertical plane," and "random." All their volumes are the same and their shapes are illustrated in Fig. 7. In total, 600 hotspots are generated with 100 hotspots of each type.

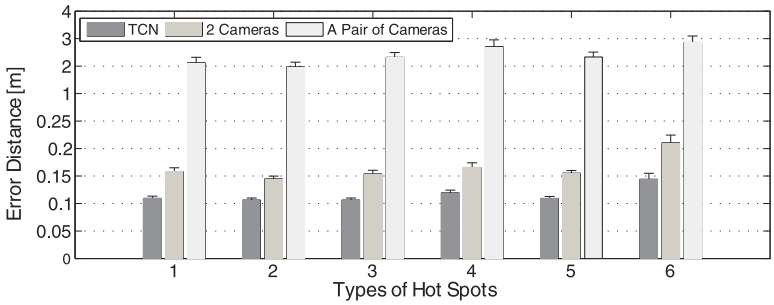


Fig. 8 Localization accuracies of different systems via types of hotspots

"Error distance" is used as the metric to evaluate the accuracy, which is the distance between the real location and estimated location of the hotspot's center. Figure 8 illustrates the average error distance of using the TCN, two cameras deployed sparsely, and a pair of adjacent cameras. It is shown that the TCN has the best accuracy because it utilizes multiple cameras to reduce error distance, and the performance does not vary much, depending upon their shapes. We find that the error is from rotating camera angles, quality of temperature detectors, and estimation of the hotspot center. The most significant cause is estimating the hotspot center in the thermal image. Since the hotspot center in the image is not a clear point like a corner in the visible image, calculating and estimating location of the centers in the thermal image cause an inevitable error.

From the results above, it is also found that the system using a pair of adjacent cameras shows much worse accuracy than the other two systems. Other than aforementioned causes of errors, the most important error in this case is the position of cameras. If the angle between two optical axis of cameras to the far hotspot center is close to zero, the estimated distance from the cameras to the hotspot center is too sensitive to detect it correctly. For example, if the distance between two cameras is 0.2 m and the distance between a hotspot and the center of two cameras is 10 m, the angle is 1.15°. If error in estimating this angle is −0.1°, the error of distance is 0.91 m, which is not acceptable because the room size is comparably small. Thus, a pair of adjacent cameras cannot estimate the locations of far hotspot centers accurately.

3.3.2 Effectiveness of using multiple cameras [simulation]

This simulation is designed to evaluate the accuracy of localization of hotspots which appear at different locations. Since the distance between the camera and the hotspot can affect the resolution of the hotspot in the thermal image, this simulation is conducted to show the effectiveness of cooperative localization.

At first, various hotspots are generated and their centers are set in the floor, middle, and ceiling plane of the datacenter. Then three cameras, C_1, C_2, and C_3, are deployed at the corners of the ceiling plane, and used to detect and localize these hotspots. The mean error distances in these planes are 0.27, 0.25, and 0.73 [m], respectively. Every plane is divided into 11×11 regions and the average error distance in each region is calculated. Table 1 illustrates the error distances in these three planes for each region,

Table 1 Error distances [m] of using 3 cameras to locate hotspots in the three planes

(a) Ceiling plane

C_1										C_2
0.20	**0.35**	**0.47**	**0.47**	**0.51**	**0.44**	**0.49**	**0.37**	**0.31**	
0.29	0.11	0.11	0.12	0.13	0.13	0.14	**2.95**	0.15	0.19	0.16
0.38	0.13	0.12	0.10	0.09	0.09	0.10	0.13	**0.22**	0.15	0.14
0.50	0.12	0.11	0.12	0.11	0.10	0.14	**0.42**	0.13	0.13	0.14
0.56	0.12	0.12	0.13	0.17	0.18	**0.72**	0.18	0.12	0.14	0.14
0.52	0.11	0.12	0.14	**0.25**	**1.15**	**0.25**	0.15	0.13	0.13	0.14
0.64	0.12	0.15	**0.27**	**1.17**	**0.32**	0.18	0.13	0.14	0.13	**0.21**
0.49	0.12	**0.23**	**1.28**	**0.30**	0.19	0.14	0.16	0.14	0.15	0.16
0.49	0.18	**0.95**	**0.25**	0.16	0.14	0.15	0.15	0.15	0.14	0.15
0.31	**0.65**	**0.20**	**1.09**	0.14	0.14	0.17	0.16	0.16	0.17	0.17
C_3	0.15	0.13	0.14	0.13	0.15	0.16	0.15	0.17	0.16	C_4

(b) Middle plane

0.14	0.11	0.12	0.16	0.15	0.16	**2.56**	**1.85**	**0.70**	**0.44**	**0.48**
0.13	0.11	0.10	0.09	0.10	0.13	0.12	**1.31**	0.14	0.15	0.15
0.18	0.11	0.12	0.10	0.09	0.09	0.11	0.12	0.13	0.14	0.15
0.17	0.11	0.12	0.13	0.11	0.11	0.13	0.15	0.13	0.12	0.14
0.18	0.10	0.11	0.13	0.17	0.17	**0.23**	0.15	0.14	0.13	0.15
0.19	0.12	0.11	0.13	**0.22**	**0.33**	**0.22**	0.12	0.12	0.13	0.16
3.89	0.11	0.14	**0.20**	**0.31**	**0.23**	0.19	0.15	0.12	0.12	0.14
1.47	0.13	0.17	**0.33**	**0.23**	0.19	0.18	0.14	0.13	0.14	0.15
0.67	0.14	**0.26**	**0.25**	0.16	0.15	0.16	0.15	0.16	0.16	0.17
0.22	**0.21**	0.18	0.15	0.14	0.16	0.13	0.15	0.15	0.17	0.16
0.19	0.16	0.14	0.13	0.13	0.14	0.15	0.15	0.17	0.19	0.18

(c) Floor plane

0.12	**2.77**	**5.78**	**5.57**	**4.57**	**3.79**	**3.13**	**1.91**	**1.22**	**0.62**	**0.94**
1.04	0.12	0.09	0.09	0.08	0.11	0.11	**0.93**	0.13	0.14	0.15
8.83	0.11	0.11	0.10	0.08	0.10	0.10	0.11	0.12	0.14	0.15
9.89	0.12	0.11	0.11	0.11	0.10	0.10	0.13	0.12	0.12	0.14
8.82	0.12	0.11	0.13	0.15	0.16	0.14	0.13	0.13	0.13	0.14
6.96	0.11	0.11	0.15	0.19	**0.20**	0.15	0.13	0.11	0.13	0.14
4.83	0.11	0.15	0.17	0.18	**0.20**	0.16	0.14	0.14	0.13	0.13
1.30	0.11	0.16	0.18	0.17	0.15	0.16	0.14	0.14	0.14	0.15
0.49	0.12	0.16	0.17	0.17	0.15	0.13	0.15	0.15	0.16	0.15
0.42	0.12	0.16	0.13	0.13	0.14	0.15	0.15	0.15	0.16	0.18
0.19	0.13	0.14	0.12	0.13	0.16	0.16	0.15	0.15	0.19	0.18

as well as the location of the three cameras. Bold numbers mark the regions where the error distances are too large (>0.2 m) and the shaded regions indicate the distribution of these regions. The ellipses indicate the large regions with relative high error. It is found that accuracy changes with location of the hotspot and the regions with high error distance distribute according to the locations of the cameras.

Usually, 2U or 3U (1U = 4.445 [cm]) chassis are used in datacenters, and the distance between two adjacent servers is about 0.09 or 0.13 [m]. Error distance should be less than these values to identify the location of hotspots. Hence, we design another simulation that can satisfy this required error distance using one more additional camera. Four cameras are used in the following experiment, and simulation conditions are the same as the previous simulation. After three random cameras localize a

Table 2 Error distances [m] of using 4 cameras to locate hotspots in the three planes

(a) Ceiling plane

C_1									C_2
0.11	0.10	0.10	0.11	0.12	0.11	0.13	0.12	0.11	
0.12	0.11	0.09	0.07	0.07	0.08	0.10	0.09	0.10	0.12	0.11
0.12	0.10	0.11	0.08	0.08	0.07	0.07	0.09	0.10	0.11	0.10
0.13	0.09	0.09	0.10	0.08	0.07	0.08	0.10	0.07	0.08	0.11
0.12	0.09	0.09	0.10	0.11	0.09	0.10	0.08	0.08	0.09	0.11
0.11	0.08	0.08	0.08	0.11	0.12	0.09	0.08	0.07	0.08	0.11
0.12	0.07	0.07	0.08	0.10	0.11	0.10	0.08	0.08	0.08	0.12
0.10	0.07	0.08	0.09	0.08	0.09	0.09	0.10	0.09	0.09	0.12
0.12	0.08	0.08	0.09	0.08	0.07	0.09	0.10	0.11	0.08	0.11
0.09	0.08	0.07	0.07	0.08	0.09	0.09	0.09	0.09	0.11	0.12
C_3	0.10	0.12	0.12	0.10	0.11	0.12	0.10	0.10	0.10	C_4

(b) Middle plane

0.14	0.08	0.07	0.09	0.08	0.09	0.11	0.14	0.13	0.13	0.14
0.09	0.10	0.08	0.07	0.07	0.08	0.09	0.09	0.09	0.10	0.10
0.12	0.09	0.11	0.08	0.07	0.07	0.08	0.09	0.09	0.10	0.09
0.11	0.09	0.10	0.11	0.09	0.08	0.08	0.08	0.08	0.08	0.09
0.10	0.08	0.08	0.10	0.10	0.08	0.09	0.08	0.08	0.08	0.09
0.10	0.08	0.08	0.09	0.11	0.12	0.09	0.07	0.07	0.08	0.09
0.14	0.08	0.08	0.08	0.09	0.09	0.10	0.08	0.07	0.08	0.10
0.11	0.08	0.08	0.08	0.08	0.09	0.10	0.12	0.08	0.08	0.10
0.10	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.13	0.10	0.10
0.08	0.07	0.08	0.08	0.08	0.09	0.07	0.08	0.08	0.11	0.09
0.14	0.10	0.09	0.09	0.09	0.09	0.09	0.10	0.10	0.10	0.13

(c) Floor plane

0.12	0.08	0.11	0.11	0.11	0.14	0.14	0.14	0.14	0.12	0.14
0.12	0.11	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	0.10
0.18	0.09	0.11	0.09	0.06	0.07	0.08	0.08	0.08	0.09	0.10
0.20	0.10	0.09	0.10	0.08	0.07	0.07	0.08	0.08	0.09	0.10
0.16	0.09	0.08	0.10	0.12	0.08	0.07	0.08	0.07	0.08	0.09
0.15	0.08	0.07	0.08	0.10	0.18	0.08	0.07	0.07	0.08	0.09
0.13	0.07	0.08	0.08	0.09	0.09	0.16	0.08	0.08	0.08	0.08
0.10	0.07	0.08	0.08	0.09	0.08	0.09	0.13	0.08	0.08	0.09
0.07	0.07	0.07	0.08	0.07	0.08	0.08	0.09	0.12	0.09	0.08
0.08	0.06	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.12	0.10
0.13	0.07	0.08	0.08	0.08	0.10	0.09	0.08	0.08	0.09	0.11

hotspot, a set of three cameras that generate minimal error distance near the hotspot are selected, according to the relation between error distribute and the locations of the cameras. And then these cameras localize it again. The average error distance using four cameras in every region is illustrated in Table 2. The mean error distances in the three planes are 0.10, 0.09, and 0.09 [m], showing approximately 4.5 times improved accuracy in distance compared to the mean error distances using three cameras.

3.3.3 Performance in case of different size and shapes of hotspots [simulation]

Because the thermal cameras used in the TCN cannot zoom, this simulation is designed to examine the effects of the volume of hotspots on accuracy. The simulations

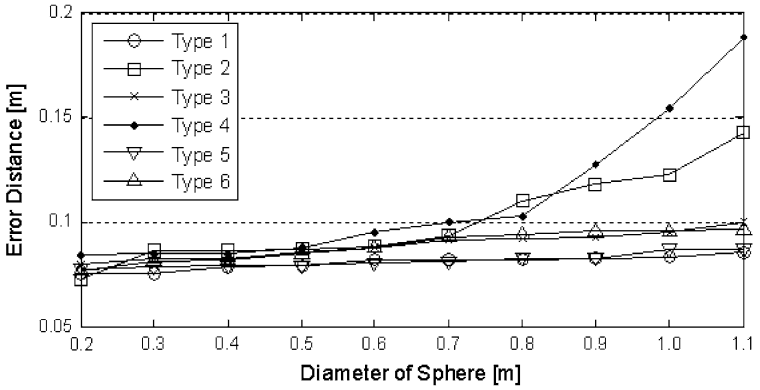


Fig. 9 Error distances of cooperative localization with different volumes of hotspots

are performed to localize sets of hotspots that are generated at various volumes with different types. Every set consists of 6 types of 100 hotspots, and the volume of each type is equivalent to the volume of a sphere with the diameters ranging from 0.2 m to 1.1 m at intervals of 0.1 m.

Figure 9 illustrates the error distance of hotspots of different volumes using a TCN. It shows that error distance increases as the volume increases. This is because cameras cannot obtain a whole hotspot in the thermal image if it is too big, and so the coordinates of the hotspot center in the image is estimated inaccurately. This problem can be solved by adaptively controlling threshold temperature to redefine hotspots.

4 Extraction of characteristics of hotspots

As mentioned above, hotspots can cause a number of problems. If the thermal monitor network knows where and how is the hotspot, and based on this to speculate why it appears, the most effective way can be selected to remove it. Thus, this section introduces the extraction of some characteristics of the hotspot, including one-dimensional scalar characteristics and a two-dimensional shape characteristic. Shape characteristic can be saved as image formats, however, the low data rate of IEEE 802.15.4, wireless sensor network protocol, limits the capacity of sending multimedia data. We thus propose an efficient compression algorithm for the hotspot shape, sculpturing algorithm, to minimize communication cost. This algorithm only extracts isotherm lines at different temperatures to compress, and shows better performance compared to other methods.

4.1 Extraction of scalar characteristics

Scalar characteristics contain aforementioned absolute coordinates of the hotspot center obtained by multiple cooperative thermal cameras nodes, and some characteristics calculated by a single node with the thermal image after localization. Some characteristics are selected according to computational complexity and importance.

Relative and real area of hotspots Area reflects the size of influenced region by the hotspot. Relative area, namely area of the hotspot in the thermal image, is expressed by the number of pixels. It is

$$A = \sum_{[x,y]^T \in \xi} 1 \tag{16}$$

where ξ is the region of the hotspot in the thermal image.

Because of distance and angle of the thermal camera, real area of the hotspot cannot be directly obtained by the processing node from the thermal image. After the base station receives other characteristics, it will estimate the real area based on the map of the datacenter:

$$S = \frac{AL^2}{s^2 F^2 \cos\langle \mathbf{v}_1, \mathbf{v}_2 \rangle} \tag{17}$$

where L is the distance between the camera and the hotspot, s is resolution of the image (pixels/cm), F is the focal length of the camera, $\mathbf{v}_1$ is the direction vector from the camera to the hotspot center, $\mathbf{v}_2$ is the unit normal vector of the surface of the equipment on which the hotspot appears, and $\cos\langle \mathbf{v}_1, \mathbf{v}_2 > \rangle = \mathbf{v}_1 \cdot \mathbf{v}_2 / (|\mathbf{v}_1| \cdot |\mathbf{v}_2|)$. $\mathbf{v}_2$ is obtained based on coordinates of the hotspot and the map of the datacenter.

Maximal and average temperature Maximal and average temperature of the hotspot are the fundamental characteristics, which roughly reflect its influence on the electronic equipments and the order to be eliminated. Each pixel value in the thermal image is corresponding to a temperature data, so the maximal temperature is that

$$T_{\max} = \boldsymbol{f}'\big(\max\big(I(x, y)\big)\big), \quad [x, y]^T \in \xi, \tag{18}$$

and the average temperature is that

$$\bar{T} = \frac{1}{A} \sum_{[x,y]^T \in \xi} \boldsymbol{f}'\big(I(x, y)\big) \tag{19}$$

where $I(x, y)$ is the pixel value at coordinates $[x, y]^T$ in the thermal image, and $\boldsymbol{f}'(.)$ is the index function from pixel value to temperature value.

Variance Variance indicates temperature change in the hotspot region. It is

$$\sigma^2 = \frac{1}{A} \sum_{[x,y]^T \in \xi} \big(\boldsymbol{f}'\big(I(x, y)\big) - \bar{T}\big)^2 \tag{20}$$

Tests show that the variances of hotspots in thermal images vary with different location of hotspots. For example, the variance of the hotspot suspending in air, such as the hotspot produced by hot air ejected from outlet, is less than that of the hotspot on the surface of the computing node. Because air convection can even temperature field, and, contrariwise, irregular surface of the computing node can complicate temperature field. Thus, medium of the hotspot can be speculated from variance.

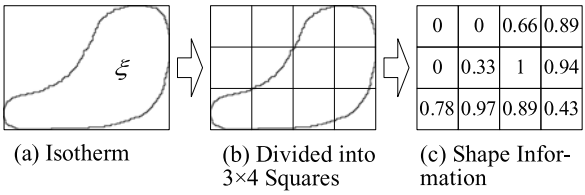


Fig. 10 Compression of an isotherm of hotspot

4.2 Extraction and reconstruction of shape characteristic

For the sake of further realizing the characteristics of the hotspot, its shape and gradient in different time need to be transmitted to the base station to save. Since multimedia data (thermal images) cause huge overhead in terms of communication and storage, we propose a novel domain-specific compression algorithm "sculpturing algorithm" to minimize this overhead in this section. The sculpturing algorithm cannot compress images, but can compress the hotspot's shape data and represent its gradient. We use real thermal image obtained in the datacenter to evaluate the performance of sculpturing algorithm in the following section. Plus, considering the low calculating ability of sensor nodes, the method of extraction should be low complexity. Thus "sculpture algorithm" is proposed to implement asymmetrical extraction and reconstruction.

4.2.1 Compression of an isotherm in sculpturing algorithm

In this section, compressing an isotherm of the hotspot is explained, where a hotspot region ξ is surrounded by an isotherm line l. At first, a binary image I of an isotherm line from a thermal image is obtained (Fig. 10(a)). Then I of size is $Hei \times Wid$ pixels is divided into $N \times M$ squares. Figure 10(b) illustrates an example of division of an image showing that I is divided into 3×4 squares. Let the side length of the square is m pixels, N is $\lceil Hei/m \rceil$ and M is $\lceil Wid/m \rceil$. In every square, the percentage of the region belonging to ξ is calculated. The percentage in ith square is denoted as p_i. It is considered as the information of the isotherm of ξ (Fig. 10(c)).

The parameters of m and quantization length affect the granularity of the hotspot. In turn, they define the degree of accuracy when reconstructing hotspots. A small number of m can divide I into more squares and the more bits of quantization length can represent an more accurate value. Thus, finer hotspots can be reconstructed with a small number of m and high number of quantization length, but more bytes are needed to transmit and store. This tradeoff can be adjusted by considering the required granularity of the hotspot.

4.2.2 Reconstruction of an isotherm in sculpturing algorithm

The reconstruction process is more complex than the compression process. At first, $N \times M$ squares are used to form an empty image I'. Now the region of ξ can be roughly defined in I'. If the ith square region contains part of ξ, namely $p_i > 0$, all the pixels in it are set as black ($p'_i = 1$); or white ($p'_i = 0$), otherwise. The rough region is shown on Fig. 11(b).

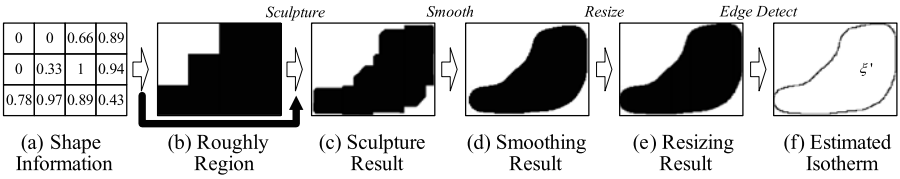


Fig. 11 Reconstruction of an isotherm of hotspot

Then a finer isotherm will be estimated. These square regions are divided into 3 groups. Group 1 consists of the square regions in which $p = 0$. It means that these square regions don't contain any part of ξ. Group 2 consists of the square regions in which $0 < p < 1$. This means that part of the square region belong to ξ. Namely, the edge of ξ is in these squares. Group 3 consists of the square regions in which $p = 1$. It means that the entire square region belongs to ξ. Thus, it is unnecessary to process the regions in Group 1 and 3. Only the regions in Group 2 need to be processed to obtain a finer isotherm of ξ. Now all the p' in these regions are 1, namely all the regions are black. These black regions will be continually "sculptured" until $p' = p$ in every region.

"Sculpturing" once includes two steps. Since the edge of ξ is curved line, the angular pixels are chipped away in the first step. An angular pixel is defined as a black pixel in whose 4-connected neighborhood at least 2 pixels are white. In the second step, an eroding process is implemented on the edge of the black region. Namely, the black pixels at the edge of ξ are changed into white. In the process of "sculpturing," the change of p' is monitored in every square. If $p'_i = p_i$, the sculpturing process is stopped in the ith square region. Through multiple sculpturing, the percentage of black area in each square region is equal to the one in corresponding square in compression, namely $p' = p$. All the black regions are considered as ξ' (Fig. 11(c)).

After the sculpturing process, the edge of ξ' in I' will be smoothed. A blur process (such as Gaussian blur $g_\gamma(x, y) = exp[-\gamma(x^2 + y^2)]$) is performed on I' and then I' is binarized (Fig. 11(d)). I' is redefined as the rectangular region containing ξ'. At last, the I' is resized to the size of I, so that aspect ratio of ξ' is as same as original ratio (Fig. 11(e)). The isotherm has been reconstructed. The black region in I' is ξ' and the edge of ξ' is the isotherm line l' (Fig. 11(f)).

4.2.3 Compression and reconstruction of multiple isotherms of the hotspot

In this section, compression and reconstruction of the hotspot represented by multiple isotherm lines are explained. Figure 12 illustrates the procedures of compressing multiple isotherm lines in an actual thermal image obtained in a datacenter. At first, the point at the highest temperature T_{top} in a hotspot region is found. Then several surrounding isotherm lines denoted as l_i $(i = 0, 1, \ldots)$ are extracted. The isotherm line at the threshold temperature T_{th} is denoted as l_0. For multiple isotherm lines with temperature gradient ΔT, higher isotherm lines than l_0 are denoted as l_i where $i = 0, 1, \ldots$, and the temperature of l_i is $T_{\text{th}} + \Delta T \cdot i$.

As ΔT decreases, more isotherms exist and more details can be represented, but this extra data must be transmitted. The region surrounded by l_i is denoted as ξ_i. The

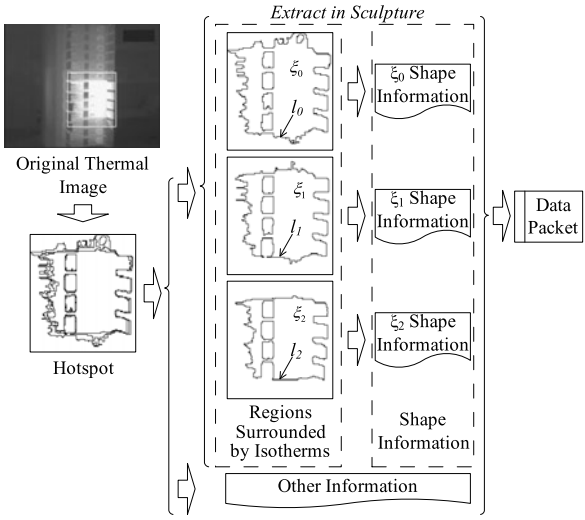


Fig. 12 Compression of multiple isotherms

rectangular region containing ξ_i is represented by its height, width, and coordinates of its top-left corner which are denoted as h_i, w_i, and $[x_i, y_i]$, respectively. The 2D shape characteristic of ξ_i is compressed separately by the sculpturing algorithm as in Sect. 4.2.1. For the communication, related parameters are added to the data packet that will be transmitted. The parameters include T_{top}, T_{th}, ΔT, and every isotherm's parameters which include original size and location of the isotherm l_i, namely h_i, w_i and $[x_i, y_i]$, the numbers of squares, namely N_i and M_i, and the information of ξ_i.

As soon as the base station receives the information ξ_i, the hotspot will be reconstructed in an image (Fig. 13). At first, all the isotherms l_i' are reconstructed by extracting N_i, M_i, and the information of ξ_i, based on the sculpturing algorithm. Then, according to h_i and w_i, the rectangular region containing l_i' is resized. All the isotherms l_i' are overlaid based on $[x_i, y_i]$. Then, the hotspot with multiple isotherms is reconstructed. Finally, the temperature at every isotherm is calculated by T_{th} and ΔT, so that the temperature gradient around the hotspot is represented.

4.3 Experimental evaluation

In this section, we use real thermal images to present the performance of our sculpturing algorithm. To evaluate the sculpturing algorithm, the error, compression gain, and complexity are compared with that of other algorithms.

4.3.1 Granularity and error [experiment]

In this simulation, the influence of parameters on the sculpturing algorithm is tested. First, the error of reconstruction is defined. If the region surrounded by the original isotherm is ξ and the reconstructed region is ξ', the error is the percentage of difference between ξ and ξ' in the area of ξ. The parameters including the side length of the square, namely m and the number of quantization bits can affect the granularity and error.

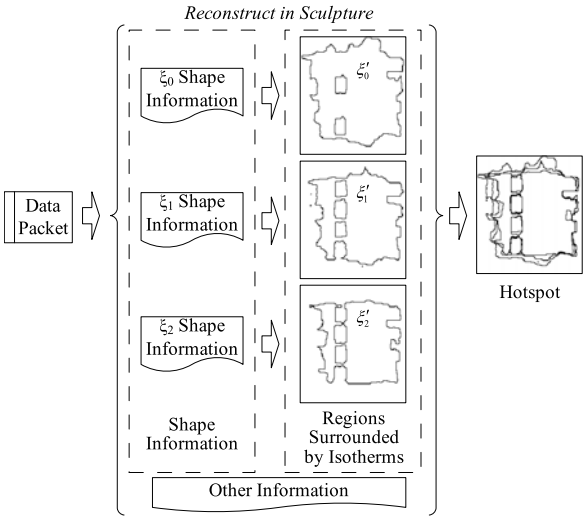


Fig. 13 Reconstruction of multiple isotherms

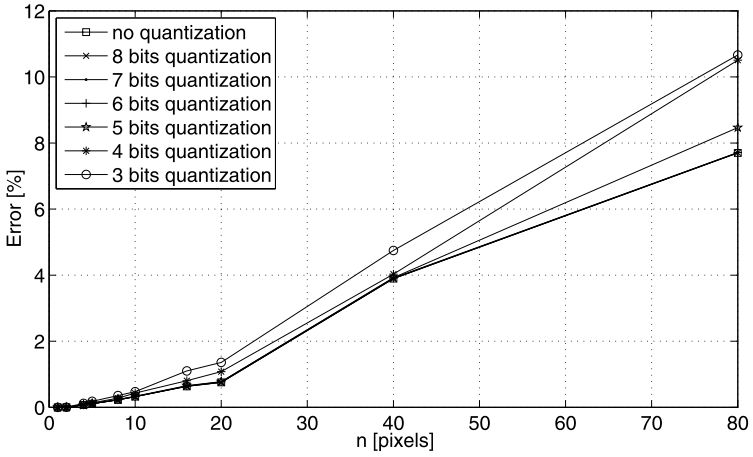


Fig. 14 The curves of the error

Five hundred thermal images are randomly chosen, each of which contains a hotspot. To decrease the interference of other parameters, the regions of hotspots are resized to the same size, $Hei = 240$ and $Wid = 320$. The shapes of hotspots are compressed and reconstructed with the different m and number of quantization bits. Figure 14 shows the results pertaining to error. It is seen that the performance using 6 bits of quantization is very close to the ideal case and error increases proportionally with m.

(a) Original hotspot (b) Corresponding ellipse (c) The Smallest Convex Polygon

Fig. 15 A hotspot is represented by similar geometrical figures. In (b) and (c), the *gray regions* are the represented hotspots, and the *black lines* are the edges of the original hotspot for comparison

4.3.2 Compression gain [experiment]

The sculpturing algorithm is proposed to minimize data used in transmitting shape characteristic of the hotspot. The shape characteristic also can be saved as a compressed image with low resolution, a corresponding ellipse, or the smallest convex polygon. Thus, in this section, the compression gain required by sculpturing algorithm and other algorithms are evaluated. At first, the hotspots which contain 3 isotherms are saved by the sculpturing algorithm. The numbers of quantization bits are 8, 6, and 4 and the numbers of squares are 3×4, 6×8, and 12×16, respectively. The size of files and the errors are listed in Table 3(a). It is found that the highest quality representation uses 594 Bytes, with 8 quantization bits and 16×12 squares. Then these hotspots are saved by some common image algorithms, such as BMP, GIF, and JPG. The parameters and size of files are shown in Table 3(b). It is found that the GIF format in black and white produces a minimal file size, but is still larger than the one produced by the sculpturing algorithm.

These hotspots are also represented by corresponding ellipses and the smallest convex polygon such as Fig. 15(b) and 15(c). Table 3(c) and 3(d) illustrate compression gain by showing the required data size (bytes) and the reconstructing error. It can be seen that the required data size by corresponding ellipses is the least, but the error is largest.

Our proposed sculpturing technique outperforms common image algorithms, when the number of hotspots, and hence the number isotherms to be represented are less (which is usually the case as we are interested only in high temperature regions). When the number of isotherms is increased (for higher level of detail), sculpturing loses its advantage. However, this problem can be overcome by using image segmentation so that each segment in turn has fewer number of hotspots, and hence isotherms.

4.3.3 Algorithm complexity [experiment]

Because of power requirement and processing ability, the complexity of compression in the sculpturing algorithm should be low. Running time in this experiment is used to measure all the algorithms' complexity. One hundred hotspot images are used, and every one is saved 100 times by an algorithm. All the algorithms run under the same hardware condition. Then the average time consumed by an algorithm is calculated

Table 3 The size of files saved by different image compression methods

(a) Sculpture Algorithm

Squares	Quantization	8 bits	6 bits	4 bits
3×4	Bytes	54	45	36
	Error	7.70 %	7.71 %	10.66 %
6×8	Bytes	162	126	90
	Error	3.89 %	3.92 %	4.03 %
12×16	Bytes	594	550	306
	Error	0.76 %	0.77 %	1.08 %

(b) Image Algorithm

Type	Color Depth	RGB 8 bits	Grayscale 4 bits	B & W 1 bits
BMP	Byte	20,278	9,718	2,462
GIF	Bytes	13,509	4,288	727
JPG	Compress	100 %	60 %	0 %
	Bytes	10,034	3,431	2,122

(c) Corresponding Ellipse Algorithm

Quantization	8 bits	6 bits	4 bits
Bytes	12	9	6
Error	30.38 %	30.71 %	34.40 %

(d) Convex Hull Algorithm

Quantization	8 bits	6 bits	4 bits
Bytes	179.76	134.82	89.88
Error	20.84 %	21.84 %	24.98 %

Table 4 Relative running time of all the algorithm

Algorithm	Sculpturing			BMP	JPEG	GIF	Convex hull	Corresponding ellipse
	3×4	6×8	12×16					
Relative Time	1	1.3367	2.8545	3.3372	3.682	5.5315	11.2781	25.9584

(Table 4). For the sake of contrast, relative time is used. It can be seen that the sculpturing algorithm takes the shortest time. This implies that its complexity is lowest.

In the above experiments, the error, compression gain, and the complexity of all the algorithms are compared. Although the hotspots can be represented by image algorithms with no errors, the data size are the greatest. Although the required data size by the corresponding ellipse algorithm is least, the error is the largest and the

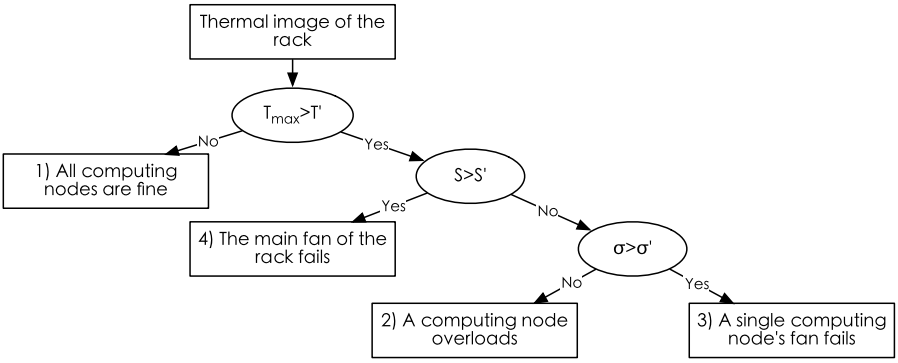


Fig. 16 Classification tree

Table 5 Accuracy of recognize four scenarios

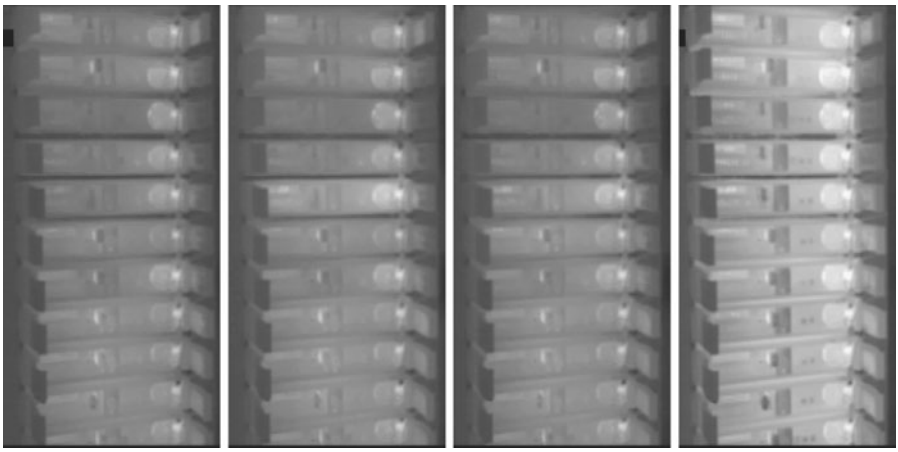
Scenario	1	2	3	4
Accuracy [%]	98.8	98.1	97.5	96.3

complexity is too high. The results of the convex hull are not satisfactory. The experimental results show that the sculpturing algorithm is most suitable for the TCN. Its complexity is the lowest, while error and data size meet the requirements.

4.3.4 Preliminary attempt on application of hotspots' characteristics

Characteristics of hotspots are employed to understand the specific thermal behavior of datacenters, and hence to cool hotspots effectively. If the causes of the hotspots are known in addition to their characteristics, selectively choosing and cooling hotspots could be readily done autonomously. In this section, we give a brief idea to classify causes of hotspots using different temperature thresholds for different causes. For example, we can use a classification tree (Fig. 16) with three different thresholds (T', S', and σ') to classify four different thermal status of our server rack, which are (1) all computing nodes are fine (Fig. 17(a)), (2) a computing node overloads (Fig. 17(b)), (3) a single computing node's fan fails (Fig. 17(c)), and (4) the main fan of the rack fails (Fig. 17(d)). If status 2, 3, or 4 turns out to be true, different types of cooling decisions can be made such as executing job migration, increasing compressor cycle, or increasing fan speed. If the hotspot appears due to the local or main fain failure of the node, the central management unit can request manual repair. As our focus is not the classification of causes of hotspots, we only show the feasibility of our framework in this section.

In the experiment, about 100 real thermal images in each scenario are used. Table 5 shows the accuracy of recognition. We can consider that the cause relates these characteristics and can be automatically recognized. Certainly, there are many other causes for hotspots, thus in future work, we will study related characteristics of hotspots to recognize more causes in a real datacenter.



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4

Fig. 17 Thermal images of the rack in four scenarios. The fifth computing node on the top overloads in Scenario 2, and its fan fails in Scenario 3

5 Conclusion

This paper proposes a thermal monitor network for datacenters which includes the base station, a scalar sensor network, and a thermal camera network. The nodes in the scalar sensor network are deployed at significant spots, such as inlets, outlets, and vent tiles, to monitor temperature, humidity, or airflow. Every node in the thermal camera network contains a thermal camera that is used to thermally monitor the datacenter in high granularity. The main focus of this paper is to extract hotspots' characteristics using the thermal camera network, including the absolute coordinates, temperature, size, gradient, and shape. This paper especially proposed a sculpturing algorithm to compress the data of the hotspot's shape characteristic in transmission and store. These characteristics can be used to know more about hotspots. They are supposed to optimize thermal management in the future research. The experiments prove that our thermal monitor network can accurately know where and how the hotspots are.

Moreover, this paper indicates experimentally that hotspots' characteristics extracted by the TCN have certain relation to the causes for generating them. Now, four causes have been able to automatically recognized. In the future, we will study on recognizing more causes from hotspots' characteristics in a real datacenter.

Acknowledgement We acknowledge the contribution of Alex Weiner served as a technical reviewer.

References

1. Bash CE, Patel CD, Sharma RK (2006) Dynamic thermal management of air cooled data centers. In: Proceedings of the 10th international conference on thermal and thermomechanical phenomena in electronics systems (ITHERM), San Diego, CA, USA, May 2006, pp 445–452

2. Beitelmal A, Patel C (2007) Thermo-fluids provisioning of a high performance high density data center. Distrib Parallel Databases 21:227–238
3. Chase JS, Anderson DC, Thakar PN, Vahdat AN, Doyle RP (2001) Managing energy and server resources in hosting centers. In: Proceedings of the 18th symposium on operating systems principles (SOSP), Chateau Lake Louise, Banff, Canada, October 2001
4. Chew LW, Ang L, Seng KP (2008) Survey of image compression algorithms in wireless sensor networks. Inf Technol 4:1–9
5. Choi J, Kim Y, Sivasubramaniam A, Srebric J, Wang Q, Lee J (2008) A cfd-based tool for studying temperature in rack-mounted servers. IEEE Trans Comput 57(8):1129–1142
6. Das R, Kephart J, Lefurgy C, Tesauro G, Levine D, Chan H (2008) Autonomic multi-agent management of power and performance in data centers. In: Proceedings of the 7th international conference on autonomous agents and multiagent systems (AAMAS), Estoril, Portugal, May 2008
7. Erdem UM, Sclaroff S (2006) Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. Comput Vis Image Underst 103(3):156–169
8. Hajebi K, Zelek JS (2006) Sparse disparity map from uncalibrated infrared stereo images. In: Proceedings of the Canadian conference on computer and robot vision, Waterloo, Canada, June 2006, pp 17
9. Hajebi K, Zelek JS (2008) Structure from infrared stereo images. In: Proceedings of the computer and robot vision (CRV), Ontario, Canada, May 2008, pp 105–112
10. Hamann H, Lùandpez V, Stepanchuk A (2010) Thermal zones for more efficient data center energy management. In: Thermal and thermomechanical phenomena in electronic systems (ITherm), June 2010, pp 1–6
11. Harris C, Stephens M (1988) A combined corner and edge detection. In: Proceedings of the alvey vision conference, Manchester, UK, August 1988, pp 147–151
12. Jiang N, Parashar M (2009) Enabling autonomic power-aware management of instrumented data centers. In International symposium on parallel and distributed processing, pp 1–8
13. Kadir T, Brady M (2001) Saliency, scale and image description. Int J Comput Vis 45(2):83–105
14. Koomey JG (2007) Estimating total power consumption by servers in the US and the world. In: A report by the Lawrence Berkeley National Laboratory
15. Kunt M, Ikonomopoulos A, Kocher M (1985) Second-generation image-coding techniques. Proc IEEE 73(4):549–574
16. Lee EK, Kulkarni I, Pompili D, Parashar M (2010) Proactive thermal management in Green datacenter. J Supercomput, 1–31
17. Lenchner J, Isci C, Kephart J, Mansley C, Connell J, McIntosh S (2011) Toward data center self-diagnosis using a mobile robot. In: International conference on autonomic computing (ICAC), Karlsruhe, Germany, June 2011
18. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
19. Moore J, Chase J, Ranganathan P, Sharma R (2005) Making scheduling "cool": temperature-aware workload placement in data centers. In: Proceedings of the USENIX annual technical conference (ATEC), Anaheim, CA, April 2005
20. Mukherjee T, Tang Q, Ziesman C, Gupta S, Cayton P (2007) Software architecture for dynamic thermal management in datacenters. In: Communication systems software and middleware (COMSWARE), January 2007, pp 1–11
21. Nasrabadi N, King R (1988) Image coding using vector quantization: a review. IEEE Trans Commun 36(8):957–971
22. Patel C (2003) A vision of energy aware computing from chips to data centers. In: Proceedings of the international symposium on micro-mechanical engineering (ISMME), Tsuchiura, Ibaraki, Japan, December 2003
23. Patel C, Bash C, Belady C (2001) Computational fluid dynamics modeling of high compute density data centers to assure system inlet air specifications. In: Proceedings of the ASME international electronic packaging technical conference and exhibition, Kauai, HI, USA, July 2001
24. Patel C, Bash C, Sharma R, Beitelmal A, Friedrich R (2003) Smart cooling of datacenters. In: Proceedings of the PacificRim/ASME international electronics packaging technology conference and exhibition (IPACK), Kauai, HI, USA, July 2003
25. Pon R, Batalin MA, Gordon J, Kansal A, Liu D, Rahimi M, Shirachi L, Yu Y, Hansen M, Kaiser W, Srivastava M, Sukhat G, Estrin D (2005) Networked infomechanical systems: a mobile embedded networked sensor platform. In: Proceedings of the 4th international symposium on information processing in sensor networks (IPSN), Los Angeles, CA, USA

26. Prakash S, Lee PY, Caelli T (2006) 3D mapping of surface temperature using thermal stereo. In: Proceedings of control, automation, robotics and vision (ICARCV), Singapore, December 2006, pp 1–4
27. Reid MM, Millar RJ, Black ND (1997) Second-generation image coding: an overview. ACM Comput Surv 29:3–29
28. Said A, Pearlman WA (1996) A new fast/efficient image codec based on set partitioning in hierarchical trees. IEEE Trans Circuits Syst Video Technol 6(12):243–250
29. Shapiro JM (1993) Embedded image coding suing zerotrees of wavelet coefficients. IEEE Trans Signal Process 41(12):3445–3462
30. Sharma R, Bash C, Patel C, Friedrich R, Chase J (2005) Balance of power: dynamic thermal management for internet data centers. IEEE Internet Comput 9(1):42–49
31. Tang Q, Gupta S, Varsamopoulos G (2008) Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach. IEEE Trans Parallel Distrib Syst 19(11):1458–1472
32. Tang Q, Gupta SKS, Varsamopoulos G (2008) Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach. IEEE Trans Parallel Distrib Syst 19(11):1458–1472
33. Taubman D (2000) High performance scalable image compression with EBCOT. IEEE Trans Image Process 9(7):1158–1170
34. Wallace GK (1991) The JPEG still picture compression standard. Commun ACM 34(4):30–44
35. Zhang C-N, Wu X (1999) A hybrid approach of wavelet packet and directional decomposition for image compression. In: IEEE Canadian conference on electrical and computer engineering, vol 2, pp 755–760