Research Article

Efficient On-Demand Image Transmission in Visual Sensor Networks

Kit-Yee Chow, King-Shan Lui, and Edmund Y. Lam

Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong

In a tracking system, an object of interest is monitored continuously in a sensor network. Information about the object is kept in the sensors and sensors transmit the information upon request. In this paper, we consider the scenario where all sensors around a targeted object capture images of it and these pictures will be sent to a mobile agent upon request. Due to the size and energy limitations in sensors, images kept in sensors are often small and highly compressed. We describe a framework to facilitate a mobile agent in the sensor network to request images of the object of interest. As sensors are limited in energy, it is desirable to reduce the energy used in transmitting the images. We observe that, in a sensor network that is sufficiently dense, images from neighbor cameras would likely overlap, and therefore intermediate sensors can process and combine overlapping portions so as to reduce the energy spent on image transmission. We develop a protocol for involved sensors to determine how to transmit the images they have kept to the mobile agent in an energy efficient manner. Our protocol is truly distributed and does not require any global information. We evaluate our protocol through extensive simulations.

1. INTRODUCTION

A wireless sensor network consists of thousands of sensors that span a large geographical region. Research and development in wireless sensor networks are becoming increasingly widespread due to their low cost and low maintenance in deployment [1]. These sensors are able to communicate with each other to collaboratively detect objects, collect information, and transmit messages. Sensor networks have become an important technology especially for environmental monitoring, target tracking, military applications, disaster management, and so forth [2, 3]. A sensor is a very small device and the battery inside is not likely to be rechargeable. This limitation in energy puts extra constraints in the operations of a sensor. In order to prolong its lifetime, a sensor should carefully utilize its energy. Message transmission has been shown to be the major source of energy dissipation. To save energy used in transmission, the size of the messages and the number of messages to be transmitted should be minimized. In this paper, we study how images can be processed and transmitted efficiently in terms of energy in a visual sensor network.

We consider a tracking system where objects of interest are traced. All sensors are equipped with cameras and they can take images of a target while it moves inside the sensor network. The images captured are kept in the local memory of the sensors. As sensors are limited in size and therefore have only a small field of view, a camera often can only take a snapshot of a portion of a targeted object. In other words, to get a full view of an object, images captured by several sensors have to be combined before the images can be analyzed. In some applications, images of objects in different locations have to be examined at the same time. To reduce the energy consumption in sensors, the analysis will be carried out in some powerful nodes instead of in the sensors locally. Traditionally, these powerful nodes are assumed to be fixed servers wired with power cable sitting at the rim of the sensor network. An image is sent to these fixed servers in a hop-by-hop manner. As a server is at a peripheral position, the paths from the sensors in the central area of a network can be very long, especially when the sensors span a large geographical area. This is not energy efficient since images are large in size and several images are needed for a single object.

Recently, the benefits of using *mobile relays* have been studied [4]. It is reported that using a mobile node as a sink to collect information can prolong the lifetime of a sensor network. We adopt this approach in our studies. Instead of

sending images to a fixed server, sensors relay their pictures to this mobile sink. There are several issues that have to be addressed when data are sent to a mobile node instead of a fixed one. First, since the sink is now mobile, it also has energy concern. That is, if we keep on sending pictures to the sink, the sink may get overwhelmed and some pictures will be lost due to insufficient energy or memory. Second, when the sink is fixed, sensors can be preconfigured with its location and every sensor will know where to send its images. This does not apply when the sink becomes mobile. A mechanism that allows a sensor to locate the sink is needed.

To solve the two problems, we take an on-demand request approach. The mobile sink sends out requests to sensors when it wants to collect the images. In this way, the mobile sink can estimate how much information it can handle and makes requests accordingly. The possibility of overwhelming is highly reduced. On the other hand, the mobile sink should also put its location in the request messages. Sensors can then know where to send their pictures without much extra overhead.

After receiving a request, sensors that have images of the same target will send their images to the mobile sink. Since they may overlap with each other, intermediate sensors on the path from the cameras to the mobile sink can combine images of overlapping regions so as to reduce the size of the images. Different paths may have different hop counts and images can be combined in different ways. In this paper, we investigate how to optimize the energy resources in transmitting images on demand to a mobile node through selecting appropriate paths. We develop a distributed protocol that helps to reduce the total transmission energy. We evaluate our protocol through extensive simulations and the simulation results show that our protocol, when compared to sending images individually, can achieve a significant reduction in energy usage without any additional computational cost.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 describes the network model. Section 4 describes the protocol, and the simulation results are shown in Section 5. We finally conclude our paper in Section 6.

2.   RELATED WORK

A sensor node can reduce the energy spent in transmission by combining the data it receives from neighbors together before transmitting it out. For example, if the sensors are transmitting the average temperature of the area they reside in, when a sensor receives two averages $A1$ and $A2$ from two neighbors, it can calculate $(A1 + A2)/2$ and send it out. Two or more messages can be "combined" into one message and transmission energy is saved. The process of combining several messages into one is called *data aggregation*. The problem of finding optimal data aggregation has been proved to be NP-hard [5, 6]. Some mechanisms have been developed to aggregate simple scalars [7–10], but only a few of them study the employment of aggregation of more complex data types.

Data aggregation can also be applied to visual sensor networks, but the multimedia nature of the data presents extra challenges in designing a proper mechanism for aggregation. Several recent studies tackle various aspects of this problem. Reference [11] shows that applying maximum compression before transmission does not necessarily minimize the total energy used. The authors then develop a heuristic for selecting a good compression level based on global information. Yu et al. [12] develop a mechanism for transmitting JPEG2000 images in one hop. Given an expected image quality, the algorithm's objective is to minimize the energy spent in processing and transmission. Reference [13] studies distributed image compression, where the computation in wavelet transform and quantization of subbands in JPEG2000 are shared among different groups of sensor nodes. This approach does not aim at decreasing the total energy needed, but because each sensor only accounts for a part of the computation, the maximum energy needed in a sensor is reduced. Reference [14] also studies distributed image compression, but from a different perspective. In this setting, each sensor captures a low-resolution image of the object, and assuming that overlapping areas across images from different sensors can be identified, super-resolution of these low-resolution images is performed at the receiver. The approach in [15] uses geometrical information to estimate the correlation of visual images of the same object and to determine how to compress the images. Reference [16] describes a camera network where sensors are organized hierarchically into several tiers. Lower power cameras are at the bottom and they are capable of taking low-resolution images. When an object of interest is identified, these sensors can trigger cameras in a higher tier on demand to get better images.

None of the work mentioned above considers the effect of using different paths in transmitting the images. SPIN-IT [17] is a routing protocol for retrieving an image based on metadata of images. It focuses on how a node identifies the sources of an image requested but not the image transmission issues. In our earlier work [18, 19], we demonstrate that when there is overlap in the scene captured by the cameras, and intermediate processing is possible to combine the images, the image transmission strategy is nontrivial. Under different ratios of transmission and compression energy, we develop guidelines that stipulate different routes for transmitting images in order to minimize the overall energy used in the system. We assume that the server is fixed and only a few sensors can aggregate images in [18]. In this paper, we consider a more general problem that images are sent to a mobile sink and all sensors are equal in that they all can capture, aggregate, and transmit images.

3.   NETWORK MODEL

We consider a sensor network where sensors are uniformly distributed in a large area. Sensors are static. Each sensor knows its physical location by means of GPS or a localization algorithm [20]. Each sensor n can communicate with the sensors within its transmission range. These sensors are the neighbors of n, and n has the location information of these
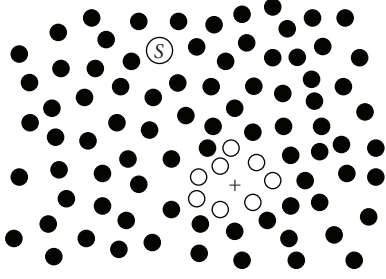
FIGURE 1: A simple network.

neighbors. All sensors are identical in terms of their capabilities in handling the images.

3.1. Tracking and image capture

In our system, the main task of the sensors is to take images of a certain targeted object and transmit them to a mobile sink upon request. The object of interest moves around in the sensor network. When a sensor detects that the object is nearby, it determines whether to take a picture of it or not. Since a sensor network is dense, many sensors can sense the same object simultaneously. We let these sensors coordinate with each other to take pictures at about the same time. Time synchronization can be done through GPS or a clock synchronization algorithm [21]. On the other hand, as these sensors are collaborating to track the object, a sensor should also realize whether its image overlaps with its neighbor's image and to what extent. It is possible that images captured do not overlap. It is also possible that some images contain overlaps but some do not. Whether the sensors want to capture overlapping images depends on the application requirement. Our protocol works for situations that may contain overlapping and nonoverlapping images.

Referring to the network in Figure 1 where the object of interest is represented as "+," the white nodes are the sensors that will capture images of the object. In a typical tracking system, the sensors should also record the time that the pictures are taken and the object involved. The images captured are kept in the local memory of sensors. Upon request, the images are then sent to the mobile sink. As the mobile sink may not request images frequently, a sensor may have taken pictures of several different targets before the mobile sink makes a request. To reduce the size of the images, images are compressed and saved in the local memory of a sensor. In our system, we use JPEG compression, but we also note that JPEG2000 and other proprietary schemes may also be used in similar settings.

3.2. Mobile sink

A mobile sink is a mobile node in the network that collects images on demand. It can appear in different locations to collect information. When it decides to collect images, it sends out a request that specifies what images it wants. For example, in a wild life surveillance application, the request can be

"Please give me the pictures of a tiger captured at 1:00 AM today." This request also contains the information about the mobile node and a hop count (hc) field to keep track of how far this request has traveled. hc is initialized to 1. Since the mobile sink does not know which sensors have the information it wants, this request is flooded to all the nodes in the network. To simplify our discussion, we assume that the mobile sink will remain static until the requested images are received or the mobile sink foresees that there will be no reply. Since the mobile sink should not be a fast moving node, if it moves after sending out the request, the nodes near its original location can collect the images and send them to the new location of the mobile sink, which should not be very far away.

3.3. Request handling

A request is flooded to all the nodes in the network. Therefore, every sensor would receive the request at least once. Nodes that possess the required images should therefore send their stored images to the mobile sink. The image transmission will be described in Section 4 in detail. In this section, we focus on discussing how nodes handle the request messages for developing forwarding paths towards the mobile sink.

When a node n receives a request for the first time, it should record how far away it is from the mobile sink in terms of hop count. hc carried in the request specifies how many hops n is from the mobile sink. n also keeps track that the hop count from the mobile sink to the neighbor is hc. n then increments hc by one and sends the request to its neighbor. It is possible that n receives the same request more than once since more than one of its neighbors send out the request. If n receives another copy of the request from neighbor b and the hop count in the request is $hc(b)$, it records this information in the database. It does not resend the request again. After all the copies are handled, n should know

(i) the location of the mobile sink,
(ii) its distance, in terms of hop count, to the mobile sink,
(iii) for each neighbor b, the distance between the mobile sink and b.

As the mobile sink does not station in a location forever, the record about a request should be cleared after some time.

4. PROTOCOL

Apart from updating the hop count and neighbor information, a node should also note whether it possesses the image requested. In this section, we will describe how sensors transmit the requested images to the sink. We first go over some notations.

4.1. Notations

To facilitate our discussion, we adopt the following notations and Figure 2 illustrates some of them.

I_i JPEG compressed image kept at camera i.

$I_{i\cap j}^{i}$ The portion of I_i that overlaps with I_j.

$I_{i\cap j}^{j}$ The portion of I_j that overlaps with I_i.

$I_i - I_{i\cap j}^i$	The portion of I_i that does not overlap with I_j.
$I_i \cup I_j$	Resultant image after stitching I_i and I_j if they overlap.
$\|I\|$	Size of image I.
E_t	Energy needed in transmitting unit byte to a neighbor.
E_c	Energy needed in compressing/decompressing unit byte.
E_a	Energy needed in performing image alignment per byte.
$hc(n)$	Hop count of node n.
$NB(n)$	Neighbor set of node n.

4.2. Determination of overlapping portions

By making use of an image-based localization algorithm such as in [22, 23], the physical locations and orientations of camera nodes can be known. We take the viewpoint of these papers that the overlapping portions among images can be deduced geometrically [24], and the energy involved in identifying the overlapping regions is small compared to the other processes in the sensor network. If the two images differ in scale, angle, or other parameters, we would need further image alignment on these visual data [25], that is, establishing the mathematical relationships that map pixel coordinates from one image to another, before performing data aggregation.

In sensor networks, the camera nodes are reasonably close to each other. We may assume that the magnifications and points of view among neighbor images are similar. The number of parameters involved in image alignment is thus small, and the energy consumption needed in performing image alignment would be manageable for a sensor node [26]. In truly random distribution of sensor nodes in 3D, the image alignment would be much more complicated. To facilitate our discussion, we consider the simpler scenario.

4.3. Transmission of two neighboring images

Consider that cameras i and j are neighbors and they both want to send their images, I_i and I_j, to the mobile sink, S. If I_i is transmitted individually to S in a hop-by-hop manner, the transmission load needed is $hc(i) \times |I_i|$. If both I_i and I_j are sent independently, the total transmission load (number of bytes to be transmitted) T_{dum} required is

$$T_{\text{dum}} = hc(i) \times |I_i| + hc(j) \times |I_j|. \tag{1}$$

In a dense network, it may happen that I_i overlaps with several other images captured in other cameras. Since the cameras take images in a collaborative manner, we can assume that each portion of I_i overlaps with at most one more neighbor image. That is, each portion can aggregate with at most one other image. In other words, we can partition I_i into several regions where some regions can be aggregated with exactly one neighbor image while some regions do not overlap with any image. For example, in Figure 2, $I_{i\cap j}^i$ is the portion of I_i that overlaps with I_j and $(I_i - I_{i\cap j}^i)$ is the
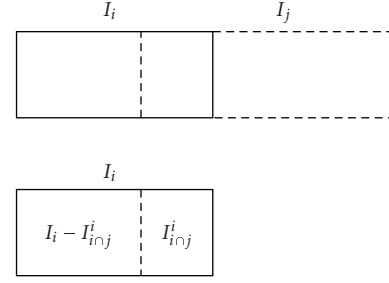

FIGURE 2: Illustration of $I_{i\cap j}^i$ and $I_i - I_{i\cap j}^i$.

nonoverlapping region of I_i. These regions will be sent independently.

Suppose that I_i and I_j overlap with each other, so $I_{i\cap j}^i \neq \varnothing$ and $I_{i\cap j}^j \neq \varnothing$. After receiving the images, it could be advantageous for S to decompress the overlapping region in both images, average them, and recompress the region, in order to save on the amount of image data needed for further transmission. The averaging is performed to reduce the resulting amount of noise; if the captured images already possess good signal-to-noise ratios (SNR), we may not need the averaging but only choose to send one of the two copies of the same scene. We denote the raw image of an JPEG image I to be $R(I)$. Let C be the decompression/compression load and let A be the image alignment load of performing data aggregation. We have

$$C = |R(I_{i\cap j}^i)| + |R(I_i^J \cap j)| + |R(I_i \cap j)|,$$
$$A = |R(I_{i\cap j}^i)|, \tag{2}$$

where $I_{i\cap j}$ is the overlapping region after averaging. Note that $|R(I_{i\cap j}^i)| = |R(I_{i\cap j}^j)| = |R(I_{i\cap j})|$.

Since all sensors are able to perform image processing functions, transmission energy can in fact be reduced by allowing intermediate sensors on the path to combine the overlapping region. For example, if i and j both send their images to neighbor node k towards S as shown in Figure 3, k can also decompress $I_{i\cap j}^i$ and $I_{i\cap j}^j$ to form $I_{i\cap j}$. The energy spent is exactly the same as the operation done in S. Nevertheless, we save transmission energy. Suppose that $hc(i) = hc(j) = hc(k) + 1$, the total transmission load T now is

$$T = |I_i| + |I_j| + hc(k) \times |I_i \cup I_j|. \tag{3}$$

We can see that

$$T_{\text{dum}} = hc(k) \times |I_i| + |I_i| + hc(k) \times |I_j| + |I_j|$$
$$> hc(k) \times |I_i \cup I_j| + |I_i| + |I_j| = T. \tag{4}$$

In other words, energy is saved as we apply image aggregation in intermediate sensor nodes when images overlap.

Due to the randomness of the network topology, $hc(i)$ may not always be the same as $hc(j)$ and i and j may not have a common neighbor leading towards S. There are altogether three possible configurations.
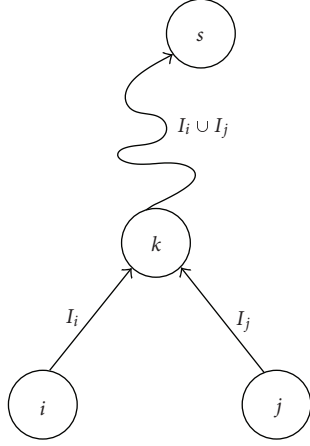
FIGURE 3: Aggregating images at a common neighbor.

(1) $hc(i) = hc(j)$

i records $hc(i)$ when it receives a request sending from its neighbor k where $hc(k) = hc(i) - 1$. k is also a neighbor lying on the shortest path towards S. There are two subcases.

(a) There exists a node k which is the common neighbor of i and j; in other words, $k \in NB(i) \cap NB(j)$ and $hc(k) = hc(i) - 1$.

(b) There does not exist a node k such that $k \in NB(i) \cap NB(j)$ and $hc(k) = hc(i) - 1$. In this case, the next hop neighbors lying on the shortest paths to S for i and j are different. We label them k_i and k_j.

(2) $hc(i) \neq hc(j)$

In this case, $hc(i)$ and $hc(j)$ can differ by at most 1 since i and j are neighbors. Without loss of generality, we assume that $hc(j) = hc(i) - 1$. Note that j can be a next hop neighbor of i leading to S.

We now describe the most energy efficient way to aggregate images that take into account the 3 scenarios. An important feature of our mechanism is that it does not require global information and both i and j can make decisions based on its own information in a distributed manner.

Case 1. This is the situation depicted in Figure 3. i and j should send their images to a common neighbor k and k stitches them to form $I_i \cup I_j$. To identify k, i should check for each $b \in NB(i)$ where $hc(b) = hc(i) - 1$, whether b is in $NB(j)$ as well. Since i knows its neighbors and their hop counts from S, i can find out $\{b \mid b \in NB(i) \text{ and } hc(b) = hc(i) - 1\}$ easily. To determine whether $b \in NB(j)$, i computes the physical distance between b and j using their locations. If the distance is smaller than the range of transmission, b is also a neighbor of j. It is possible that there are several neighbors satisfying the conditions but i and j should select the same one to send their images. A simple tie breaking mechanism is to use energy level. The neighbors with highest energy level are selected as k. Node k is then responsible for performing image alignment and data aggregation.
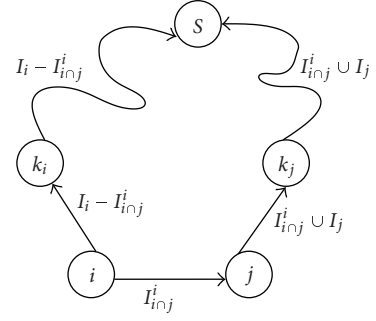
FIGURE 4: Aggregating images at node j.

```
MAIN

 1: if (hc(i) = hc(j))
 2:    { if k ∈ NB(i) ∩ NB(j) AND hc(k) = hc(i) − 1
 3:       {
 4:           I_i → k from i;
 5:           I_j → k from j;
 6:           I_i ∪ I_j → S from k;
 7:       }
 8:    else {
 9:           SEND2(i, j);
10:       }
11:   endif;
12:   }
13: else
14:   { if (hc(i) > hc(j))
15:       {
16:           SEND2(i, j);
17:       }
18:       else SEND2(j, i);
19:   }
20: endif;
```

ALGORITHM 1

Case 2. i and j know whether they have any common neighbor k where $hc(k) = hc(i) - 1$ by using the same mechanism as described in Case 1. Since k does not exist, if i and j simply send their images to a next hop neighbor leading to S, it is not guaranteed that I_i and I_j can be aggregated. Therefore, our protocol requires either i or j to send its overlapping region to the other node. That is, if i is of lower energy level, i sends $I_{i \cap j}^i$ to j and sends $I_i - I_{i \cap j}^i$ to k_i. This is shown in Figure 4. Node j is responsible for performing image alignment and data aggregation.

Case 3. Without loss of generality, we let $hc(j) = hc(i) - 1$. i should send $I_{i \cap j}^i$ to j and send $I_i - I_{i \cap j}^i$ to another neighbor k_i if $hc(k_i) = hc(j)$. In fact, the energy spent here is the same as sending the whole I_i to j. We break I_i into two portions and use different paths because we can reduce the transmission energy used in j. This is also depicted in Figure 4.

The pseudocode in Algorithms 1 and 2 summarizes the overall protocol.

```
SEND2(i, j)

1: I^i_{i∩j} → j from i;
2: I_i − I^I_{i∩j} → S from i;
3: I_{i∩j} + (I_j − I^j_{i∩j}) → S from j;
```

ALGORITHM 2

4.4. Transmission energy and computational energy

The transmission load T, compression/decompression load C, and image alignment load A of different cases are the following.

Case 1

$$T = |I_i| + |I_j| + |I_i \cup I_j| \times hc(k),$$
$$C = 3 \times |R(I_{i∩j})|, \tag{5}$$
$$A = |R(I_{i∩j})|.$$

Cases 2 and 3

$$T = |I^i_{i∩j}| + |I_i − I^i_{i∩j}| \times hc(i) + |I^i_{i∩j} \cup I_j| \times hc(j),$$
$$C = 3 \times |R(I_{i∩j})|,$$
$$A = |R(I_{i∩j})|. \tag{6}$$

4.5. Example

Suppose the area covered by the network is divided into $m \times m$ grids and the location of each grid is represented as (row, col). The camera nodes are distributed randomly in the network and each grid contains at most one node. For example, in Figure 5, the "+" represents the object of interest and the numbers indicate the hop count of each node. Only those inside the dashed rectangle will capture the images of the object. In this scenario, the object of interest is located at (11, 13) and there are 13 nodes that are responsible for capturing images. They are divided into 8 groups as circled in Figure 5. For the group of nodes at (11, 11) and (10, 11), the hop counts of both nodes are 4. We want to find a node k such that $k \in NB(i) \cap NB(j)$ and $hc(k) = hc(i) − 1$. By searching in the network, we can identify several neighbors that satisfy this condition and the neighbor with the highest energy level is selected as k. In this case, k is located at (9, 10) with $hc(k) = 3$. Let the nodes at (11, 11) and (10, 11) be i and j, respectively, and the corresponding images be I_i and I_j, i and j will send their images to k and then k will perform image alignment and stitch them to form $I_i \cup I_j$. The total energy consumption $E^{\text{Total}}_{\text{pro}}$ for this group will be

$$T = |I_i| + |I_j| + 3 \times |I_i \cup I_j|,$$
$$C = 3 \times |R(I_{i∩j})|,$$
$$A = |R(I_{i∩j})|, \tag{7}$$
$$E^{\text{Total}}_{\text{pro}} = T \times E_t + C \times E_c + A \times E_a.$$
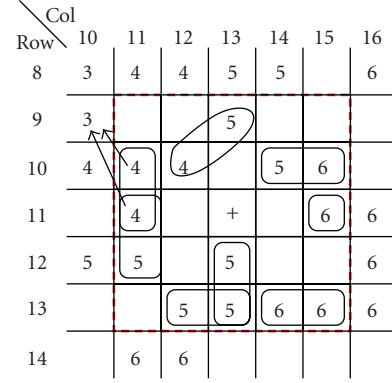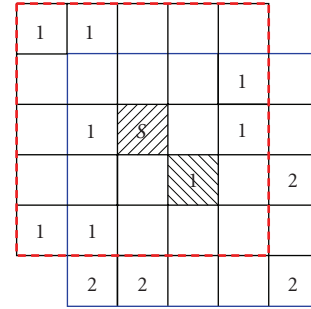

FIGURE 5: Object of interest.

FIGURE 6: Neighbor nodes of S.

5. SIMULATION

In this section, we present our simulation results that show that the protocol is beneficial in reducing the total transmission energy. The simulation results are generated using MATLAB. Network topologies are generated randomly. The whole area is divided into $N \times N$ grids. There is at most one sensor in each grid and the probability that a grid has a sensor depends on the density and is generated randomly. The mobile sink S can appear anywhere in the network. The transmission range is 3.535, which covers the neighboring 5×5 grids. For example, any nodes inside the window with S as center are the neighbors of S, and the hop count of those neighbor nodes are one, $hc(n) = 1$. Similarly, the neighbor nodes of node n with $hc(n) = 1$ will have hop counts equal to two. The hop counts of all the nodes in the network are assigned in this manner. In the example shown in Figure 6, S indicates the mobile sink and the numbers indicate the hop count of each node.

The object of interest can be anywhere within the network. Suppose any nodes inside a certain $a \times a$ window with the object of interest as the center are the sensors that will capture images of that object. These sensor nodes will be divided into groups according to their overlapping with the adjacent nodes. We assume that the sensor knows whether its image overlaps with its neighbor's image and to what extent.

TABLE 1

Density	Probability of a sensor found in a grid
Sparse	0.3
Moderate	0.6
Dense	1

It is also possible that the image does not overlap with any other image. In this case, the sensor will send its image directly back to the mobile sink along the shortest path. For the grouped nodes, they will send their images according to the protocol proposed in the previous section.

In our simulation, the network size is set to be a 50×50 grid, the mobile sink and the object of interest can be at any grid. Let D be the distance between the object of interest and the mobile sink. We consider five different cases:

(1) $0 \leq D < 5$,
(2) $5 \leq D < 10$,
(3) $10 \leq D < 15$,
(4) $15 \leq D < 20$,
(5) $20 \leq D < 25$.

Each case is simulated with three different densities of nodes: dense, moderate, and sparse. The relationship between density and probability that a grid has sensor is as shown in Table 1. There are 3 different densities and 5 different ranges of distance, so 15 combinations can be formed. Each combination is simulated 20 times to calculate the average reduction in transmission energy.

The total energy consumed in each case is compared with the dummy case, that is, each node sends the image independently along its shortest path. Let $E_{\text{pro}}^{\text{Total}}$ be the total energy consumed in tested case, let $E_{\text{dum}}^{\text{Total}}$ be the total energy consumed in the dummy case, and let E^{diff} be the energy difference.

If we assume the mobile sink will perform data aggregation on the received images, the energy equations will be

$$E_{\text{pro}}^{\text{Total}} = \sum T \times E_t + \sum C \times E_c + \sum A \times E_a, \tag{8}$$

$$E_{\text{dum}}^{\text{Total}} = \sum T_{\text{dum}} \times E_t + \sum C \times E_c + \sum A \times E_a, \tag{9}$$

$$E^{\text{diff}} = \left\{ \sum T_{\text{dum}} - \sum T \right\} \times E_t.$$

Since the computational load of both cases is the same, the total energy consumption difference is equal to the transmission load difference. Figure 7 shows the percentage change of transmission load which is calculated by

$$\text{change (\%)} = \frac{\sum T_{\text{dum}} - \sum T}{\sum T_{\text{dum}}} \times 100\%. \tag{10}$$

In our simulation, the percentage change is always positive which shows that the total transmission energy con-
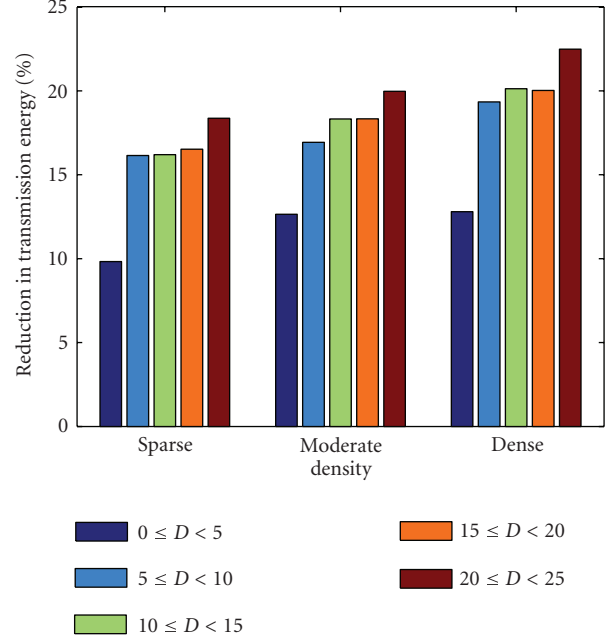

FIGURE 7: Reduction in transmission energy.

sumed by using the protocol is always less than that consumed in the dummy case. The results with different combination of object distance and sensor density are shown in Figure 7. From the result, we can observe that the reduction in transmission energy is greater when the object is farther away from S. Additionally, the reduction will be more significant as the density increases. When the network is denser, the camera nodes are closer to each other and thus the image overlaps with its neighbor's image to a greater extent. The greater the overlapping portion is, the more the energy is saved. The energy saved by applying Case 1 or is Case 2 of the protocol is $|I_{i\cap j}^i| \times hc(k) \times E_t$ and the energy saved in Case 3 is $|I_{i\cap j}^j| \times hc(j) \times E_t$. When $|I_{i\cap j}^i|$ and $|I_{i\cap j}^j|$ increase, the energy saved will also increase.

If we assume the mobile sink will not perform data aggregation on the received images, $E_{\text{pro}}^{\text{Total}}$ will be the same as (8) but $E_{\text{dum}}^{\text{Total}}$ will be different. The energy equations will be

$$E_{\text{dum}}^{\text{Total}} = \sum T_{\text{dum}} \times E_t,$$

$$E^{\text{diff}} = \left\{ \sum T_{\text{dum}} - \sum T \right\} \times E_t - \sum C \times E_c - \sum A \times E_a. \tag{11}$$

In Figures 8–10, the x-axis represents E_a/E_t and the y-axis represents the normalized energy difference between our protocol and the dummy case, that is, E^{diff}/E_t. For the normalized energy difference,

$$\frac{E^{\text{diff}}}{E_t} = \left\{ \sum T_{\text{dum}} - \sum T \right\} - \sum C \times \frac{E_c}{E_t} - \sum A \times \frac{E_a}{E_t}. \tag{12}$$
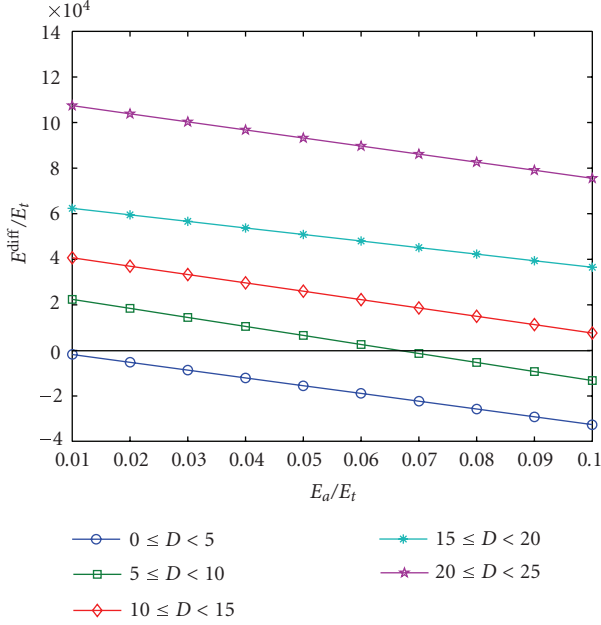
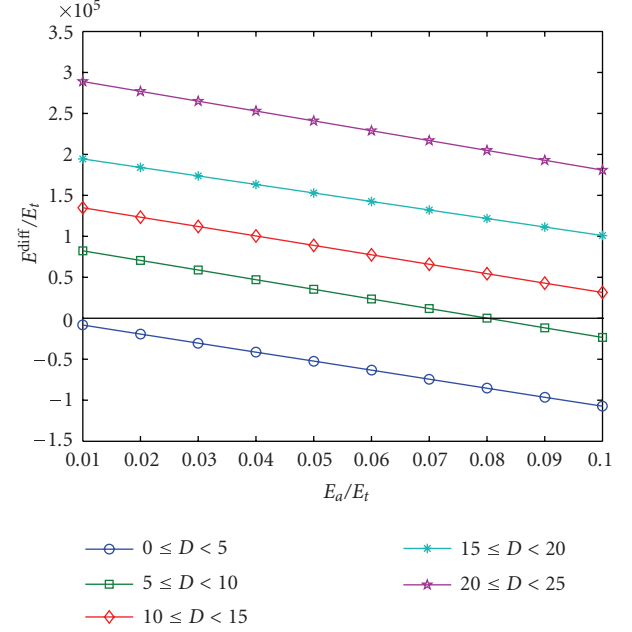FIGURE 8: Sparsely populated network.
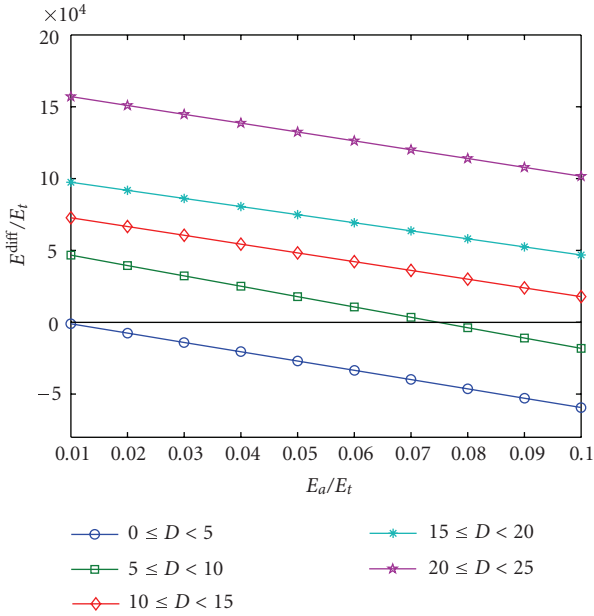

FIGURE 10: Densely populated network.

FIGURE 9: Moderately populated network.

In addition to E_a/E_t, E_c/E_t is also a factor in the above equation. Reference [27] claims that communication/computation power usage ratio can be higher than 1000, and thus E_c/E_t can be as small as 0.001. To allow for different applications where this number can vary, we take a more conservative estimate and set E_c/E_t equal to 0.005. With this assumption, we can estimate the relationship between normalized total energy difference and normalized energy needed in performing image alignment.

When the normalized energy difference is positive that means the total energy consumption in dummy case is greater than that in our protocol, and vice versa. The crossing points on the zero-axis are the critical values of E_a/E_t. If the ratio is smaller than the critical value, our protocol will be more energy efficient than sending the images independently along their shortest paths.

When the density increases, the critical value increases. A higher critical value means the protocol can tolerate a larger computational energy for image alignment. As the density increases, the amount of overlapping portions also increases. Since transmission energy is far greater than computational energy, the reduction in total transmission energy consumption is much more significant than the increase of total computational energy dissipation.

On the other hand, the critical value increases as the object of interest moves away from the sink under the same density. This indicates that the protocol can tolerate a higher processing cost for image alignment. This is reasonable because more transmission energy is saved as the object is farther away, and therefore the protocol can bear a larger computational cost.

Figure 11 shows the result of sparsely populated density network with size of 20×20 grids. By comparing Figure 11 with Figure 8, it can be observed that the two plots are very similar. The only difference is for the curve $20 \leq D < 25$, because for a 20×20 grid, the object of interest would be located at the margin of the network. Thus, we can see that the advantage of our protocol holds for different network sizes.

It is easy to verify that the transmission energy saved by applying Case 1 or Case 2 of the protocol is $|I_{i \cap j}^i| \times hc(k) \times E_t$ and the energy saved in Case 3 is $|I_{i \cap j}^j| \times hc(j) \times E_t$. In a real situation, the compression level of images is known, and thus
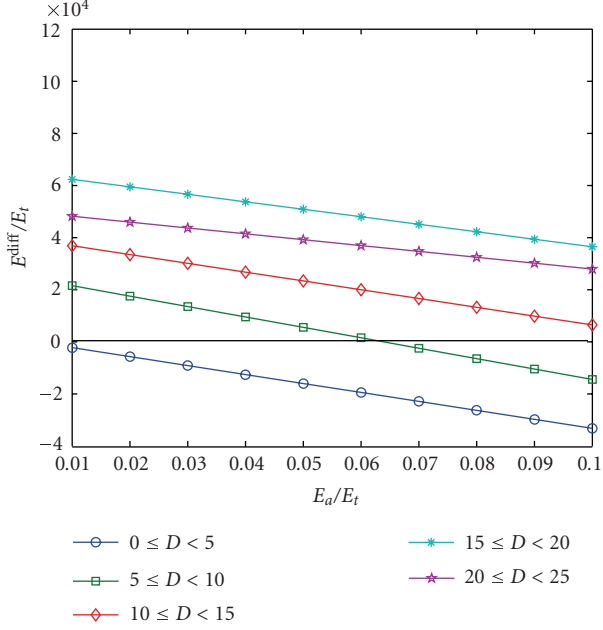
FIGURE 11: Sparsely populated network, network size $= 20 \times 20$.

we can approximate the ratio between the size of raw image data and the size of compressed image. Let M be the ratio between $|R(I_{i \cap j}^i)|$ and $|I_{i \cap j}^i|$. The computation load for each case of the protocol becomes

$$
\begin{aligned}
C &= 3M \times |I_{i \cap j}^i|, \\
A &= M \times |I_{i \cap j}^i|.
\end{aligned}
\tag{13}
$$

The normalized energy difference for Case 1 or Case 2 will be

$$
\begin{aligned}
\frac{E^{\text{diff}}}{E_t} &= |I_{i \cap j}^i| \times hc(k) - 3M \times |I_{i \cap j}^i| \times \frac{E_c}{E_t} \\
&\quad - M \times |I_{i \cap j}^i| \times \frac{E_a}{E_t}.
\end{aligned}
\tag{14}
$$

The normalized energy difference for Case 3 will be

$$
\begin{aligned}
\frac{E^{\text{diff}}}{E_t} &= |I_{i \cap j}^i| \times hc(j) - 3M \times |I_{i \cap j}^i| \times \frac{E_c}{E_t} \\
&\quad - M \times |I_{i \cap j}^i| \times \frac{E_a}{E_t}.
\end{aligned}
\tag{15}
$$

Since E_t, E_c, E_a, and hop count can also be known, each pair of nodes can determine whether to send their images according to the protocol or not by calculating the approximate value of energy difference. If $E^{\text{diff}} > 0$, the images should be sent according to the protocol or else the images should be

sent independently along their shortest paths:

$$
\begin{aligned}
0 &< \frac{E^{\text{diff}}}{E_t} \\
&= |I_{i \cap j}^i| \times \left\{ hc(\kappa) - 3M \times \frac{E_c}{E_t} - M \times \frac{E_a}{E_t} \right\} \\
&= hc(\kappa) - 3M \times \frac{E_c}{E_t} - M \times \frac{E_a}{E_t},
\end{aligned}
\tag{16}
$$

where

$$
\kappa = \begin{cases} k & \text{Case 1 or 2,} \\ j & \text{Case 3.} \end{cases}
\tag{17}
$$

Reference [28] claims that the image quality of a compressed image with compression ratio of 32-to-1 is considered to be good. In our simulation, the ratio between the size of raw image data and the size of compressed image, M, is about 35. The hop count of a node which captured the image of the object of interest in the range 0–5 unit away from the sink in a sparsely populated network is in the range of 1–3, and we assume E_c/E_t to be 0.005. By plugging the parameters into (16), we can find that the estimated critical value of E_a/E_t is around 0.0405. This value is different from the crossing point on Figure 8, since that is the overall energy difference plot which contains several pairs of nodes and the value of M is image dependent. Although we cannot predict the energy difference precisely, the approximation gives a good indication for the node to determine the path to send the image.

6. CONCLUSION

In this paper, we investigate how to optimize the energy resources in transmitting images on demand to a mobile node through selecting appropriate paths. A distributed protocol was proposed and evaluated through extensive simulations. Depending on the ratios of transmission and computational energy, network density, compression level, and object distance, the sensors should choose to send the images according to the protocol or not.

It should be noted that the assumptions used in this work have room for refinement, especially when the methodology is applied to applications where the overlapping among images may not be easily determined. Examples include tracking objects without strongly identifiable features while requiring strictly image-based localization. Similar to most of the related works, in this paper we assume that the overlapping regions among images can be deduced by the geometric positions of the sensors. One direction for our future work is to develop an explicit solution in identifying the overlapping among images using advanced vision-based approaches.

REFERENCES

[1] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufmann, San Francisco, Calif, USA, 2004.

[2] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, 2002.

[3] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[4] W. Wang, V. Srinivasan, and K.-C. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom '05)*, pp. 270–283, Cologne, Germany, August-September 2005.

[5] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the IEEE International Workshop on Distributed Event-Based Systems (DEBS '02)*, pp. 575–578, Vienna, Austria, July 2002.

[6] C. Buragohain, D. Agrawal, and S. Suri, "Power aware routing for sensor databases," in *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 1747–1757, Miami, Fla, USA, March 2005.

[7] M. Ding, X. Cheng, and G. Xue, "Aggregation tree construction in sensor networks," in *Proceedings of the IEEE 58th Vehicular Technology Conference (VTC '03)*, vol. 4, pp. 2168–2172, Orlando, Fla, USA, October 2003.

[8] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, "Medians and beyond: new aggregation techniques for sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 239–249, Baltimore, Md, USA, November 2004.

[9] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 250–262, Baltimore, Md, USA, November 2004.

[10] J.-Y. Chen, G. Pandurangan, and D. Xu, "Robust computation of aggregates in wireless sensor networks: distributed randomized algorithms and analysis," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 348–355, Los Angeles, Calif, USA, April 2005.

[11] H. Wu and A. A. Abouzeid, "Power aware image transmission in energy constrained wireless networks," in *Proceedings of the International Symposium on Computers and Communications (ISCC '04)*, vol. 1, pp. 202–207, Alexandria, Egypt, June-July 2004.

[12] W. Yu, Z. Sahinoglu, and A. Vetro, "Energy efficient JPEG 2000 image transmission over wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 5, pp. 2738–2743, Dallas, Tex, USA, November-December 2004.

[13] H. Wu and A. A. Abouzeid, "Energy efficient distributed JPEG2000 image compression in multihop wireless networks,"

in *Proceedings of the 4th Workshop on Applications and Services in Wireless Networks (ASWN '04)*, pp. 152–160, Boston, Mass, USA, August 2004.

[14] R. Wagner, R. Nowak, and R. Baraniuk, "Distributed image compression for sensor networks using correspondence analysis and super-resolution," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '03)*, vol. 1, pp. 597–600, Barcelona, Spain, September 2003.

[15] N. Gehrig and P. L. Dragotti, "Distributed compression in camera sensor networks," in *Proceedings of the IEEE 6th Workshop on Multimedia Signal Processing*, pp. 311–314, Siena, Italy, September-October 2004.

[16] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, "SensEye: a multi-tier camera sensor network," in *Proceedings of the 13th ACM International Conference on Multimedia (ACM Multimedia '05)*, pp. 229–238, Singapore, November 2005.

[17] E. Woodrow and W. Heinzelman, "SPIN-IT: a data centric routing protocol for image retrieval in wireless networks," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '02)*, vol. 3, pp. 913–916, Rochester, NY, USA, September 2002.

[18] K.-S. Lui and E. Y. Lam, "Image transmission in sensor networks," in *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS '05)*, pp. 726–730, Athens, Greece, November 2005.

[19] K.-Y. Chow, K.-S. Lui, and E. Y. Lam, "Balancing image quality and energy consumption in visual sensor networks," in *Proceedings of the 1st International Symposium on Wireless Pervasive Computing (ISWPC '06)*, pp. 1–5, Phuket, Thailand, January 2006.

[20] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proceedings of the 23rd IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2652–2661, Hong Kong, March 2004.

[21] Q. Li and D. Rus, "Global clock synchronization in sensor networks," in *Proceedings of the 23rd Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 1, pp. 564–574, Hong Kong, March 2004.

[22] H. Lee and H. Aghajan, "Vision-enabled node localization in wireless sensor networks," in *COGnitive Systems with Interactive Sensors (COGIS '06)*, Paris, France, March 2006.

[23] C. McCormick, P.-Y. Laligand, H. Lee, and H. Aghajan, "Distributed agent control with self-localizing wireless image sensor networks," in *Proceedings of the Conference on COGnitive Systems with Interactive Sensors (COGIS '06)*, Paris, France, March 2006.

[24] H. Ma and Y. Liu, "Correlation based video processing in video sensor networks," in *Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing (ICWN '05)*, vol. 2, pp. 987–992, Las Vegas, Nev, USA, June 2005.

[25] H. S. Sawhney and R. Kumar, "True multi-image alignment and its application to mosaicing and lens distortion correction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 3, pp. 235–243, 1999.

[26] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 1090–1097, Kauai, Hawaii, USA, December 2001.

[27] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.

[28] B. G. Haskell, P. G. Howard, Y. A. LeCun, et al., "Image and video coding-emerging standards and beyond," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 814–837, 1998.

Kit-Yee Chow received her B.Eng. (first class honors) degree from the University of Hong Kong in 2005. She is currently pursuing the M.Phil. degree at the Department of Electrical and Electronic Engineering, The University of Hong Kong. Her research interests are in the areas of sensor networks and image processing.

King-Shan Lui obtained her B.Eng. (first class honors) and M.Phil. degrees in computer science from The Hong Kong University of Science and Technology. She then received her Ph.D. degree, also in computer science, from the University of Illinois at Urbana-Champaign, USA, in 2002. She joined the Department of Electrical and Electronic Engineering, The University of Hong Kong, as an Assistant Professor in August 2002. Her research interests include computer network quality of service issues, protocol and algorithm design in the Internet, ad hoc networks, and sensor networks.
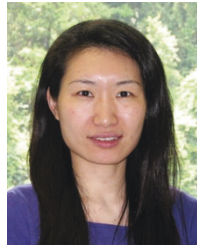
Edmund Y. Lam received the B.S. degree (with distinction) in 1995, the M.S. degree in 1996, and the Ph.D. degree in 2000, all in electrical engineering from Stanford University, Stanford, Calif. At Stanford, he developed image processing algorithms for the Programmable Digital Camera project. He also consulted for industry in the areas of digital camera systems design and algorithms development. Before returning to academia, he was affiliated with the Reticle and Photomask Inspection Division (RAPID) of KLA-Tencor Corporation in San Jose, Calif, as a senior engineer, working in the design of defect detection tools for the core die-to-die and die-to-database inspections. He is currently an Assistant Professor of electrical and electronic engineering at The University of Hong Kong, as well as the Director of its Imaging Systems Laboratory. His research interests include electronic and computational imaging, image processing applications in semiconductor manufacturing, biomedical engineering, and sensor networks.