Computer Networks 52 (2008) 2558–2567

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Self-orienting wireless multimedia sensor networks for occlusion-free viewpoints

Nurcan Tezcan *, Wenye Wang

Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27606, USA

ARTICLE INFO

Available online 14 June 2008

Keywords:
Multimedia coverage
Multimedia sensor network
Visible field of view
Occlusion
Self-orientation
Distributed algorithm

ABSTRACT

Wireless multimedia sensor networks (WMSN) are formations of a large number of compact form-factor computing devices that can capture multimedia content, such as video and audio, and communicate them over wireless channels. The efficiency of a WMSN heavily depends on the correct orientation (i.e., view) of its individual sensory units in the field. In this paper, we study the problem of self-orientation in WMSN, that is finding the most beneficial orientation for all multimedia sensors to maximize multimedia coverage. We propose a new algorithm to determine a node's multimedia coverage and find the sensor orientation that minimizes the negative effect of occlusions and overlapping regions in the sensing field. Our approach enables multimedia sensor nodes to compute their directional coverage leading to an efficient and self-configurable sensor orientation calculation. By using simulations, we show that the occlusion-free viewpoint approach increases the multimedia coverage significantly. The self-orientation methodology is designed in the form of a distributed algorithm, making it a suitable candidate for deployment in practical systems.

1. Introduction

As we manufacture more sophisticated sensing electronics cheaper every day, the nature of the information to be hauled by wireless sensor networks (WSNs) change. We are now able to capture audio-visual information from the environment using low-cost, low-resolution cameras embedded in sensor nodes. The need for using such multimedia sensors is usually driven by the necessity of providing comprehensive information pertaining to a specific region of interest. To be able to support the demand for monitoring, we focus on wireless multimedia sensor nodes with directional sensing views. Performance of directional sensing is very much dependent on the obstacles present in the environment. Therefore, finding the most favorable orientation for the multimedia sensors is an important and challenging problem. For example, deploying a large number of low-resolution image sensors is recently shown to be a good alternative to having a single high-resolution camera [1]. Distributed methods for camera sensor networks also show gains from using a large number of low-power image sensors [1,2]. In such WMSNs, inherent disadvantages due to physical obstacles in an environment (e.g., trees, buildings, etc.) can be turned into a multimodality advantage, with the flexibility to adjust orientations of the multimedia sensors attached to the wireless nodes.

There have been several works on vision planning which take the object geometry information as an input from a database, as well as modifications of the camera and the lens to determine camera poses and settings [3]. Therefore, orientation of multimedia sensors can be performed on site once the multimedia sensors have been deployed. However, such methods need an accurate field information database before deployment and are mostly applied to a small number of multimedia devices. Due to external effects or application-specific queries in WMSNs, multimedia nodes may need to change/re-orient their pose (direction of sensory unit) over time. In WMSNs, nodes

* Corresponding author. Tel.: +1 862 368 5591.
E-mail addresses: ntezcan@ncsu.edu (N. Tezcan), wwang@ncsu.edu (W. Wang).

1389-1286/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2008.05.014

may fail due to battery outage or external effects which should be handled by a dynamic update of the poses which can be performed via local information exchange among sensors.

In this paper, we present a new distributed method to find the most beneficial orientations for the sensors used in a WMSN. We specifically consider (i) minimizing the effects of occlusion in the environment and (ii) improving the cumulative quality of the information sensed from the region of interest. Let us consider a WMSN with a large number of scattered nodes, each having neighbors with which it can communicate directly. Using a *distributed* method outlined in this paper, each node can discover its neighbors and examine possible overlapping sensing regions as well as the obstacles in the environment. In our scheme, each sensor node determines the most beneficial orientation for its multimedia sensor so that the entire image of a field can be constructed using low-resolution snapshots from multiple sensors. Our approach enables multimedia sensor nodes to monitor their coverage performance, provisioning self-configurable sensor orientations in an efficient way.

The proposed algorithm also decreases the obstacles' detrimental effect on the quality of the sensed information while maximizing total covered area. As discussed in [4,5], WMSNs have stringent constraints of limited communication bandwidth, processing capability, and power supply to deliver multimedia context. It is crucial to capture the most recent occlusion-free multimedia context from the environment. This helps newly designed WMSN protocols [5] delivering efficient multimedia context with the limited bandwidth resource.

In this context, we summarized the contributions of this paper as follows: (i) the proposed algorithm is fully distributed using local information: thus communication overhead is incurred only between neighboring nodes; (ii) with the flexibility to adjust orientations of the multimedia sensors, multimedia sensor nodes update the orientation of multimedia sensors on the fly to increase the multimedia coverage significantly, (iii) overlapped and occluded regions in the sensing field can be decreased by collecting the current pose of neighboring nodes and (iv) coverage is increased even for sparse networks by using self-orientation instead of random orientations, for arbitrary obstacles in the sensor field.

The remainder of the paper is organized as follows. In Section 2, we review the existing work on sensing coverage and multimedia coverage in WMSNs. We summarize the challenges on multimedia converge and define the multimedia coverage problem in Section 3, and propose a new distributed algorithm for multimedia coverage calculation in Section 4. Performance evaluation is discussed in Sections 5 and 6 concludes the paper.

2. Related work

Maintaining and maximizing the coverage of an area have been studied in great depth in the fields of multimedia, robotics and wireless sensor networking. From the perspective of sensor networking, considerable work is present for the omnidirectional coverage problem [6–9] that aims to cover a plane by arranging circles on the plane. However, the proposed solutions for omnidirectional coverage cannot be used for the coverage of bidirectional and field-of-view sensors such as low-resolution video cameras. A common limitation of these existing protocols [6,10,11] is that the collected information on phenomena (e.g., temperature, concentration of a substance, light intensity, pressure, humidity, etc.) are assumed to come from an omnidirectional sensing. However, multimedia sensors, (i.e., low-resolution cameras, microphones, etc.) have the unique feature of capturing direction-sensitive multimedia content. Especially, video sensors can only capture useful images when there is line of sight between the event and the sensor [12]. Hence, coverage models developed for traditional wireless sensor networks are not sufficient for deployment planning of a multimedia sensor network.

In [13], a preliminary investigation of the coverage problem for video sensor networks is addressed. The concept of sensing range is replaced with the camera's *field of view*, which is the maximum volume visible from the camera when sensors are placed on the floor. All camera nodes are assumed to be situated on a plane (at the ceiling of the monitored room), and they shoot the images of the scene from a parallel plane. Such a ceiling placement, however, may only fit specific indoor applications. Then, authors proposed a routing scheme for the video sensors based on cameras' field of view metrics. Video sensors are directed to the floor, and coverage is determined by disk shaped scenes on the floor, without considering effects of any occlusions. On the other hand, a wide range of multimedia applications require outdoor placement of multimedia sensors. Several projects on habitat monitoring use acoustic and video feeds from the multimedia sensors scattered in the environment. Similarly, a large number of video sensors are already used by oceanographers to observe sandbars via image processing techniques.

In addition, triangular view regions are used for computing multimedia coverage of sensor networks in [14]. The major goal of this work is to find the minimum observed distance to any multimedia sensor that any target traveling through the field must have, even if the target optimally tries to avoid the sensors. Sensors are assumed to have a isosceles triangular coverage (field of view) placed on a square field. Using mathematical modeling, worst-case breach coverage is calculated using a polynomial time algorithm. One limitation of this work is the lack of occlusions which is the most common problem of multimedia sensors. Any obstacle in the Field of View (FoV) region result in occlusion which should be considered while calculating the worst-case breach coverage. Second, the proposed algorithm determines the closest observable distance to a sensor that any target must have for a given a deployment. Differing from this study, our goal is to determine and then increase the multimedia coverage of each individual sensor and in total by designing a local algorithm to self-orient the pose of the sensors.

In terms of occlusion effect, [1] has several investigations for wireless camera networks. The paper shows that deploying a large number of low-resolution image sensors

is a better alternative compared to a single high-resolution camera in highly-occluded environments. Therefore, distributed methods for camera sensor networks have gains over using a large number of low-power image sensors [1,2]. They observed that a collection of low resolution (short sensing range) sensors outperforms a single sensor with equivalent coverage as the degree of occlusion in the environment increases [1].

On the other hand, the geometric variations of the classic camera placement problem are also related to our problem. However, none of these variations in the literature addressed cameras as individual multimedia sensor nodes which can communicate to each other. In [15], an in-depth theoretical analysis of the problem is shown to maximize camera coverage of an area, where the camera fields of view do not overlap. In [16], the art gallery framework is further refined by introducing a resolution quality metric. In addition, Isler et al. extended the formulation of the minimum guard coverage art gallery problem to incorporate the minimum-set cover problem. They derived reduced upper bounds for two cases of exterior visibility for two- and three-dimensions [17]. In the field of robotics, a system was developed to perform dynamic sensor planning for a camera mounted on a moving robotic arm in order to compute optimal viewpoints for a preplanned robotic grasping task. In [3], a planning method was presented to determine optimal camera placement given task-specific observational requirements such as field of view, visibility, and depth of field. Our research differs from the existing works since it calculates the optimal orientation of sensor nodes using *local* information only after the initial deployment. In addition, our method considers a large number of sensor nodes with multimedia sensors having much lower resolution than the multimedia devices discussed in [3,17].

3. Multimedia coverage and self-orientation

As audio-visual sensors take their places on wireless nodes, the omnidirectional sensing assumption loses ground significantly since a typical audio or video sensor has a sectoral perception and is effected heavily by surrounding obstacles [12]. Therefore, we envision that sensing coverage planning for the wireless multimedia sensor networks (WMSN) will be different from what first-generation sensor networks used. Multimedia sensors, such as cameras, are powerful multidimensional sensors that can capture a directional view, usually called Field of View (FoV). The most commonly used low-resolution camera module is equipped with a lens providing a 45° FoV [3]. For example, the human eye, without any rolling movement, can only see objects lying inside a cone having a 25° half-angle [3]. To obtain a much wider view of the surroundings, fisheye lenses with a FoV of 150° FoV have been developed [2]. In this work, we assume sensors nodes have *fixed* lenses providing a field of view with angle Θ, and they can only pan to adjust their FoV as shown in Fig. 1. We use the term "*camera sensors*" for simplicity to represent wireless multimedia sensors including video and audio sensors having a directional view.

Multimedia nodes are densely deployed, providing detailed visual and audio information from multiple disparate viewpoints. Low resolution sensors can be used for many WMSN applications such as environmental monitoring, and health care. A sensor resolution of 128×128 is usually enough for typical applications, whereby 320×240 might be required for some applications requiring object recognition [18]. Each camera node is responsible for extracting necessary data out of the captured video stream and sending it to a base station. A multimedia sensor network with N sensors is represented by $S = \{s_1, s_2, \ldots, s_N\}$, which can be deployed in a polygonal sensing field, denoted by $\mathscr{A}$. We also assume that each node is equipped to learn its location information via any lightweight localization technique for wireless sensor networks [19].

Let us denote the distance between s_i and s_j by $d(i,j)$, where $d(i,j) = |\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}|$. If $d(i,j) < 2R_s$, where R_s denotes multimedia sensing range, sensors s_i and s_j are *FoV neighbors* that share sensing area in common. These sensors use FoV neighbor information to compute non-overlapping viewpoints. A communication link exists between sensor s_i and sensor s_j if a single-hop transmission from s_i to s_j and s_j to s_i can be performed successfully. Sensors s_i and s_j are *transmission neighbors*; if there is a symmetric link between them. Each sensor s_i is associated with quad $(x_i, y_i, \mathscr{T}_i, \mathscr{C}_i)$, where $\mathscr{T}_i$ represents *transmission neighbors*, and $\mathscr{C}_i$ represents the set of *FoV neighbors*.

A sensor is called *self-orienting*, if it is capable of adjusting its pose at the point of deployment (low-cost multimedia sensors [20] that are capable of panning). In this context, some definitions that will be used in the rest of the paper are as follows:

Definition 1. Field of View (FoV): The term *field of view* refers to the directional view of a multimedia sensor and is assumed to be an isosceles triangle (two-dimensional approximation) as shown in Fig. 2. A field of view of a sensor s_i is denoted by $\Lambda_i^{\Theta_i}$, where the parameter Θ_i is the vertex angle of the isosceles triangle.

Definition 1.1. Visible FoV (vFoV): Visible FoV, denoted by $v\Lambda_i^{\Theta_i}$, is a FoV of a sensor node s_i which is visible to the sensor itself, i.e., has not been blocked by any obstruction within FoV, $\forall$ obs$_j$ in $\mathscr{A}$, if obs$_j \cap \Lambda_i^{\Theta_i} = \emptyset$, then $\Lambda_i^{\Theta_i} \Rightarrow v\Lambda_i^{\Theta_i}$, where obs_j is an obstacle in the sensing field.

Definition 1.2. Occluded FoV (oFoV): The contrary of vFoV is the *occluded FoV* such that, $\forall$obs$_j$ in $\mathscr{A}$, if obs$_j \cap \Lambda_i^{\Theta_i} \neq \emptyset$, then $\Lambda_i^{\Theta_i} \Rightarrow o\Lambda_i^{\Theta_i}$.

Definition 1.3. Overlapping FoV (xFoV): The visible FoV is referred to as *overlapping FoV* if it intersects with any of the neighboring sensor's visible FoV, $\forall s_j \in \mathscr{C}_i$, if there exists $v\Lambda_i^{\Theta_i} \cap v\Lambda_j^{\Theta_j} \neq \emptyset$, then $\Lambda_i^{\Theta_i} \Rightarrow x\Lambda_i^{\Theta_i}$.

Definition 2. FoV disk: The FoV disk associated with a sensor defines the set of all possible FoVs. For simplicity, we assume that the orientation of all sensors can be anywhere in between $[0^0, 360^0]$; thus, FoV disk is a circular disk having a radius of R_s, i.e., the maximum distance to capture with a given resolution.

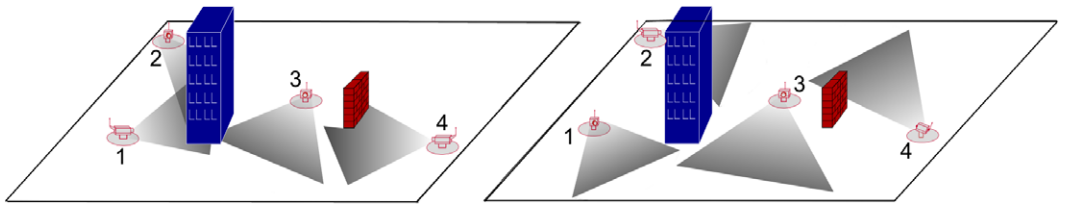


Fig. 1. Two dimensional representation of a wireless multimedia sensor network having four low-resolution sensors. In the left figure, the nodes' orientation is randomly determined. The right one illustrates the same area with slight changes in orientation. Gray areas are the regions visible by the sensors.

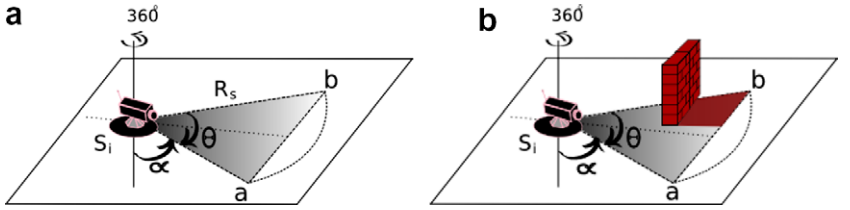


Fig. 2. Illustration of two dimensional field of view (FoV) of a multimedia sensor node, where α is the vertical angle to the boundary edge of FoV showing the current orientation, Θ is the FoV vertex angle, and R_s is the maximum multimedia sensing range.

4. A distributed solution to multimedia sensors self-orientation problem

Our solution is designed in the form of a *distributed* algorithm. The main reason behind the distributed approach is making our algorithm a suitable candidate for deployment in practical systems. Centralized orientation of WMSN systems may not clearly scale since WMSNs are usually composed of a large number multimedia nodes. In addition, we may need to update the orientations which would be more costly in a centralized approach compared to that of a distributed approach.

In the rest of this section, we will explain the details of the self-orientation algorithm that has three major phases: (i) initial messaging; (ii) distributed FoV detection; and (iii) self-orientation algorithm as shown in Fig. 3. Each phase has specific tasks and uses a set of messages. Next, we walk through each phase in detail.

4.1. Initial messaging phase

Sensor nodes exchange messages between neighbors to collect the neighborhood information. All sensors broadcast a `HELLO_MSG` indicating their unique sensor IDs and their location coordinates. We assume that stationary sensors having identical FoV ranges are located in the sensing field. Finally, each sensor processes the `HELLO_MSG` and lists overlapping FoV neighbors. The initial messaging phase ensures that every sensor is aware of its neighbors and their locations.

4.2. Distributed FoV detection

Distributed FoV detection uses *three* consecutive tests to detect sensor's maximum visible FoVs. The first test, namely *perimeter test*, checks the existence of a visible FoV within $[0^0, 360^0]$. If a sensor fails to find a visible FoV during the perimeter-test, it moves to the second test called *neighbor-distance test* which examines the distance to FoV neighbors. Finally, the third test, called *obstacle-distance test*, is performed if the sensor fails the neighbor-distance test. It compares the occluded FoVs to find the largest visible FoV. Here, we explain these three tests as follows:

4.2.1. Perimeter test

In perimeter-test, each sensor scans its *FoV disk perimeter* to determine whether a visible FoV (which cannot be captured by any other FoV neighbor) exists in its FoV disk. The reason is that FoV disk perimeter can effectively show occlusions and possible overlapping regions. The intersection points of any tangent touching an existing obstacle on the perimeter can be used to determine the size of the

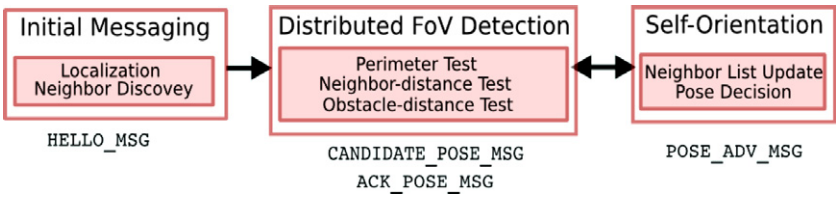


Fig. 3. Three major steps in self-orientation of multimedia sensors.

occlusion. For example in Fig. 4, the FoV disk of sensor s_1 is illustrated. There are two obstacles inside its FoV disk which are close enough to s_1 so that an occlusion may result. The intersections of the tangents on the perimeter are shown with points *F* and *G* for the first obstacle (obs1); *H* and *A* for the second obstacle (obs2). Therefore, a sensor s_i can determine that if there exists a $\Lambda_i^{\Theta_i}$ where $\Lambda_i^{\Theta_i} \cap \Lambda_i^{\angle FOG} = 0$ or $\Lambda_i^{\Theta_i} \cap \Lambda_i^{\angle HOA} = 0$ then $\Lambda_i^{\Theta_i}$ is a *visible* FoV and we refer to arcs $\widehat{FG}$ (counter clock-wise) and $\widehat{HA}$ as *occluded arcs* on the FoV disk of s_1 (see Figs. 5 and 6).

The perimeter-test not only finds the visible FoV but also helps to determine non-overlapping FoVs in a FoV disk. In this step, sensors do not know the orientations of their FoV neighbors. However, they can determine possible overlapping FoVs inside their FoV disks. Similar to occluded arcs, each sensor finds possible *overlapping arcs* on its perimeter using the location information received from its neighbors. To do this, the intersection points of the arcs are determined and the perimeter is scanned as illustrated in Fig. 4. For example, sensor s_1 has an overlapping arc $\widehat{BD}$ and $\widehat{CE}$.

By examining each FoV neighbor and obstacles, a sensor decides whether occluded and overlapped arcs enclose its perimeter from 0° to 360° [21]. If there is a $v\Lambda_i^{\Theta_i}$ with $\Theta_i \geqslant \Theta$ such that $x\Lambda_i^{\Theta_i}$ does not exist, we say that the "perimeter-test" is passed. This means that the sensor has a visible FoV which has not been captured by any other sensor in any orientation. Since our goal is to maximize the visible FoV in the total sensing region, sensors which pass the perimeter-test will adjust their pose. On the other hand, sensors that do not pass the perimeter-test continue the FoV detection with the neighbor-distance test, which will be explained in the following subsection.

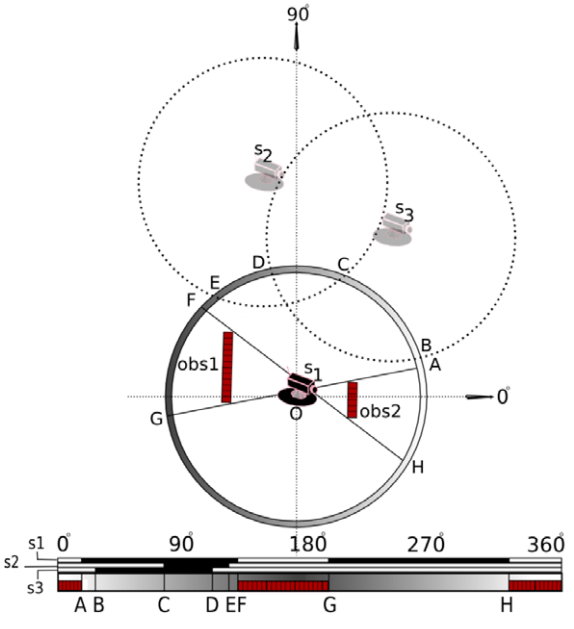


Fig. 4. An example showing the perimeter test for sensor s_1.

perimeter_test () {
  for each $\{obs_k$ in FoV disk$\}$
    compute the occluded arc of obs_k
  for each $\{s_j \in \mathcal{C}_i\}$
    compute the overlapped arc of s_j
  scan perimeter $(0^0, 360^0)$ for any arc $\widehat{P_mP_n}$
    if $\{(\widehat{P_mP_n}) > \Theta\}$ *and* $\{$visible($\widehat{P_mP_n}$) == TRUE $\}$ *and*
      $\{$overlapped ($\widehat{P_mP_n}$) == FALSE $\}$
        return PASS;
    else $\{$**return** FAIL;$\}$

Fig. 5. Pseudo code of perimeter test.

neighbor_distance_test () {
  scan perimeter $(0^0, 360^0)$ for any arc $\widehat{P_mP_n}$
    if $\{$occluded($\widehat{P_mP_n}$) == FALSE $\}$ *and* $\{$ $(\widehat{P_mP_n}) > \Theta\}$
      if $\{x\Lambda_i^{\widehat{P_mP_n}}$ exists with FoV neighbor $s_j\}$
        find the distance $d(i,j)$
      return PASS;
      store s_j with max $d(i,j)$;
    else $\{$ **return** FAIL; $\}\}$

Fig. 6. Pseudo code of neighbor-distance test.

4.2.2. Neighbor-distance test

Passing the parameter test implies that a sensor has a visible FoV, which cannot be covered by its neighbors in any orientation (non-overlapped in any case). In the neighbor-distance test, however, we examine whether a sensor has visible FoV which might be overlapped. If a sensor has a $v\Lambda_i^{\Theta_i}$ with an angle $\Theta_i \geqslant \Theta$ in its perimeter, it is assumed to pass the neighbor-distance test, otherwise it moves to obstacle-distance test. Sensors that pass the neighbor-distance test then find the largest visible FoV based on neighbor's distances.

Even though the final orientations of the neighbors are not known, FoV neighbors might have overlapping FoVs. In this case, sensors need to find the smallest overlapping FoV by scanning visible arcs and calculating the distances between each neighbor. A closer neighbor implies a larger overlapping FoV. In Fig. 7, the FoV disk of sensor s_1 and its neighbors are shown. Since perimeter of s_1 is enclosed by an occluded arc $\widehat{FH}$ and overlapping arcs $\widehat{FA}$, $\widehat{BC}$, $\widehat{DE}$, and $\widehat{GA}$, sensor s_1 fails the perimeter-test. However, it passes the neighbor-distance test, since arc $\widehat{HF}$ is visible which is greater than Θ, the FoV angle of the camera sensors which is assumed to be fixed. Among the neighbors s_2, s_3, s_4 and s_5, sensor s_2 has the largest distance to s_1, denoted by $d(1,2)$, indicating smallest possible overlapping FoV, shown as dark shaded areas inside the FoV disk.

4.2.3. Obstacle-distance test

Finally in the obstacle-distance test, sensors with no vFoV are examined. Fig. 8 shows an example sensor s_1 surrounded by four obstacles. Since there is no *visible* arc in

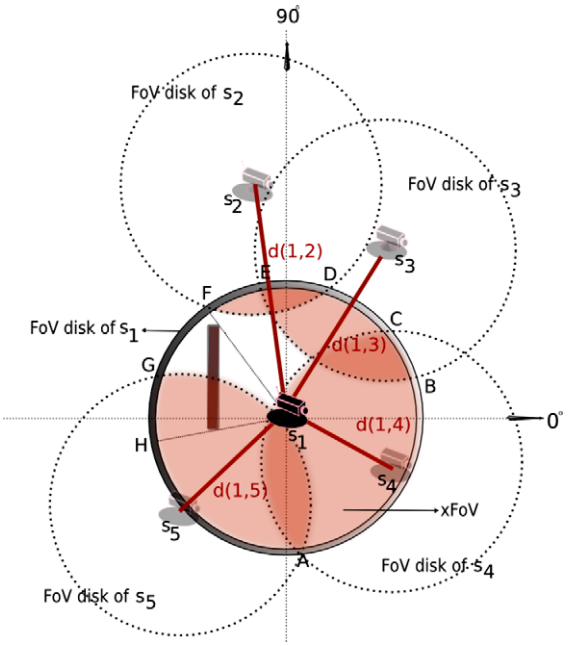


Fig. 7. An example showing the neighbor-distance test for sensor s_1.

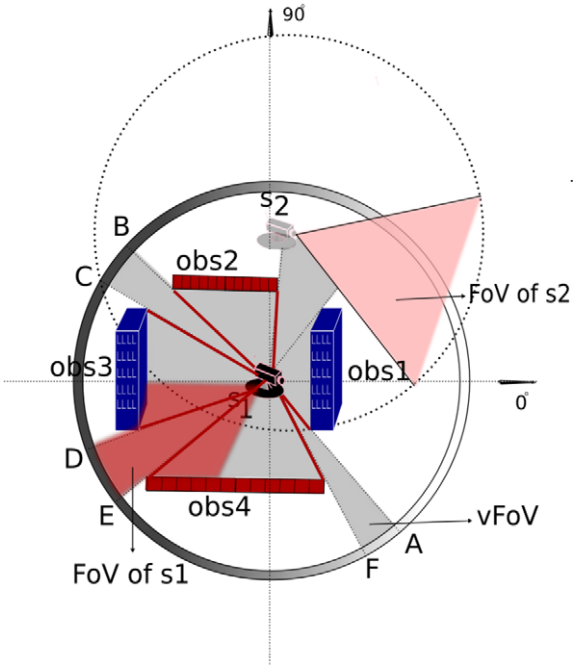


Fig. 8. An example showing the obstacle-distance test condition for sensor s_1.

the perimeter greater than Θ, the final orientation of sensor s_1 will not have a visible FoV. However, by finding the distances between the obstacles and the sensor node, the occluded FoV can be minimized by keeping the visible FoV maximized. Similar to the neighbor-test, a closer obstacle means a larger occluded FoV. In such conditions,

obstacle_test () {
  for each $\{obs_k$ in FoV $\}$
   find the distance $d(i,k)$
  return obs_k with max $d(i,k)$ $\}$

Fig. 9. Pseudo code of obstacle-distance test.

a sensor scans the perimeter in order to find the most beneficial arc Θ, to maximize the visible FoV (see Figs. 9, 10).

Note that the perimeter of the FoV disk may not be fully-occluded or fully-overlapped. For example, In Fig. 8, arc $\widehat{FA}$ and $\widehat{DE}$ are visible and non-overlapped arcs, but smaller than Θ. In such cases, these small segments can be included in the FoV. In Fig. 8, the FoV of sensor s_1 is shown as a shaded region which includes the arc $\widehat{CD}$ and occluded regions with larger distances from obstacles.

Note that, in our algorithm, multimedia sensors can update their neighbor lists and orientations periodically by taking the advantage of local information exchange. Thus, all tests are performed using up-to-date FoV neighbors and their orientation decisions.

4.3. Distributed FoV detection-based heuristic algorithm for self-orientation

In this section, we present a heuristic algorithm to obtain the most beneficial pose of the sensors for the multimedia sensors self-orientation problem. Under the 360° pan-capability assumption, multimedia sensors will determine their pose for self-orienting by using their local one-hop neighborhood information. The dimensions and the locations of the obstacles are assumed to be known by sensors before self-orientation. We do not consider the multimedia sensors as obstacles with respect to the other multimedia sensors due to their small size.

Based on the test for distributed FoV detection presented in Section 4.2, we propose a heuristic algorithm described as follows:

Step 1: Sensors send the `HELLO_MSG` that indicates the location of the sensors. For self-orientation, sensors must build a list of FoV neighbors that are close enough to have an overlapping FoV. A received `HELLO_MSG` is then used to update the neighbor lists. Note that, we assume that the maximum sensing range, R_s, is equal to or smaller than the transmission range of the multimedia sensors.

Step 2 and Step 3: After exchanging the `HELLO_MSG`, each sensor has an up-to-date FoV neighbor list with their locations and apriori-known obstacle locations. The next step is performing the perimeter test. As we explained in Section 4.2.1, perimeter test checks if a sensor s_i has a visible FoV, $v\Lambda_i^{\Theta_i}$, which cannot be captured by any other FoV neighbor in a FoV disk. Thus, when the perimeter test is passed, the sensor s_i can self-orient to $v\Lambda_i^{\Theta_i}$ and finalize the self-orienting algorithm. On the other hand, sensors failing the perimeter test will continue the algorithm with the neighbor-distance test.

In particular, the perimeter test shows the existence of *at least one vFoV* that cannot be observed by others in any orientation. However, there may be more than than one

```
0. for each sensor s_i do {
1.   broadcast HELLO_MSG for exchanging location information and finding FoV neighbors.
2.   perform perimeter_test()
       if (perimeter_test == PASS)
         self-orient to pose Θ_i^* then broadcast POSE_ADV_MSG.
         return();
       else {
3.       update FoV neighbors based on POSE_ADV_MSG received.
4.       perform neighbor_distance_test()
         if (neighbor_distance_test == PASS)
           find candidate pose and broadcast CANDIDATE_ADV_MSG
5.         update FoV neighbor based on CANDIDATE_POSE_MSG received.
           send ACK_POSE_MSG for each CANDIDATE_POSE_MSG
           self-orient to pose Θ_i^* then broadcast POSE_ADV_MSG
           return();
         else {
6.         perform obstacle_distance_test()
           self-orient to pose Θ_i^* and broadcast POSE_ADV_MSG
           return(); }
       }
 }
```

Fig. 10. The general approach of the self-orientation algorithm.

visible FoVs that result in passing the perimeter test. Then sensors change their pose to the most beneficial *vFoV*. Here, the term *beneficial* corresponds to having *smallest panning angle* to a self-orienting multimedia sensor. Therefore, a sensor selects a $vA_i^{\Theta_i}$ with a vertical angle of α_i to the boundary (as shown in Fig. 2) such that $|\alpha_i - \alpha_0|$ is the smallest among all possible *vFoVs*, where α_0 is the current vertical angle. After changing the pose, a sensor should advertise its decision to all its neighbors with a POSE_ADV_MSG and finalize the self-orienting procedure.

In *Step 3*, sensors that have failed the perimeter test update their neighbor list based on the POSE_ADV_MSGs they received. If a sensor receives a POSE_ADV_MSG from a FoV neighbor, it updates its neighbor list by adding the pose of its neighbor for the next steps. *Step 4 and Step 5:* In *Step 4*, sensors invoke the neighbor-distance test to find a occlusion-free FoV. By passing the neighbor distance test, a sensor determines the existence of a visible FoV in the FoV disk. From the visible FoVs, it selects the pose toward the FoV neighbor s_d with maximum distance, using the candidate pose selection procedure in *Step 5* and sends its candidate pose by a CANDIDATE_ACK_MSG to the neighbor s_d. This message indicates the candidate pose of the sensor to its neighbors.

Since sensor nodes perform the self-orienting simultaneously, sensors then receive a CANDIDATE_ACK_MSG from their neighbors who have passed the neighbor-distance test, thus replying with a ACK_POSE_MSG if no *xFoV* occurs. Whenever a sensor receives ACK_POSE_MSG, it indicates that the sensor can select this pose safely and finalize the self-orienting procedure. Otherwise, a sensor should repeat the *Step 5* with the second minimum distance neighbor. *Step 6:* Finally, sensors that failed the perimeter and neighbor distance test perform the last test, the obstacle-distance test. Since they have failed the previous tests, no visible FoV exists in their FoV disk. Thus, in *Step 6*, sensors will select an occluded FoV with maximum coverage which is the pose toward the obstacle with maximum distance similar to the neighbor-distance test. If there is a visible FoV with an angle smaller than Θ, the final pose will be selected from the small vFoV including the occluded FoV to maximize the visible region that the sensor will capture.

The self-orienting algorithm uses $O(N)$ messages which takes $O(N)$ time. Every node sends a HELLO_MSG and POSE_ADV_MSG once. And sensors which execute the neighbor-distance test exchange one CANDIDATE_POSE_MSG and one ACK_POSE_MSG. Thus, the total number of these messages is $O(N)$.

At the end of this algorithm, each sensor selects its pose and self-orients to maximize total visible FoVs over the sensing field $\mathscr{A}$. One may argue that, the system may require overlap in the coverage of cameras in some applications, e.g., mobile tracking using camera sensor networks. Overlapping coverage can help to localize objects [22], especially moving objects. In such applications, sensors should self-orient for occlusion-free but overlapping FoVs which can be achieved by simple modifications in our algorithm. Each sensor calculates the intersection points of $\mathscr{A}'$ and its FoV disk such that in the perimeter test, any visible arc overlapped with $\mathscr{A}'$ is selected without considering the

neighboring overlapped regions. In this way, our algorithm cannot only be used to maximize the visible coverage but also serve for any application where overlapping FoV is necessary. Next, we show our extensive simulation results for different scenarios.

5. Performance evaluation

5.1. Simulation settings

We have used Ns-2 simulator [23] for the performance evaluation of our algorithms. Simulations have been performed for randomly placed sensor nodes in a rectangular two-dimensional terrain. All sensor nodes have been configured with an *FoV* vertex angle $\Theta = 60^\circ$, and an R_s of 30 m. Communication between two sensors are assumed to be possible (i.e., an edge exists between two sensors nodes in the connectivity graph), if the distance between the transmitter and receiver is no more than 60 m. A sensing field spanning an area of 250×250 m^2 has been used in which the number of sensors were varied to study the system performance from sparse to dense deployments. In the basic scenario, 50 static multimedia sensor nodes are deployed with self-orientation capabilities. Initial orientation of these sensors is randomly determined. The sensing field $\mathscr{A}$ has several predefined rectangular obstacles which adversely affect the visible viewpoints of multimedia sensors.

5.2. Simulation results

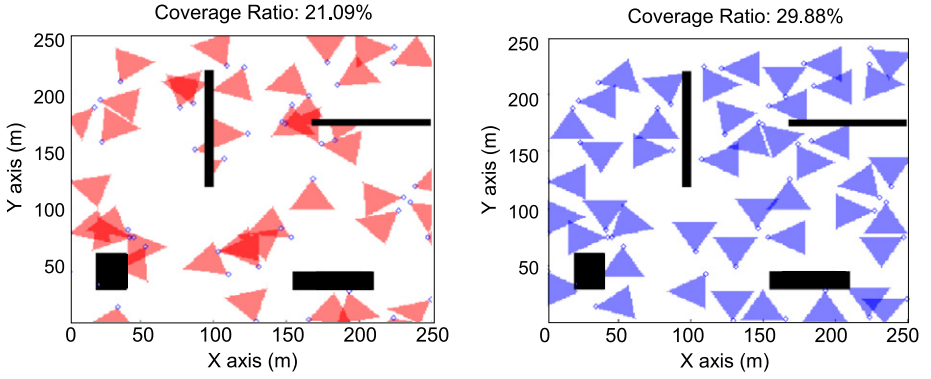
In our simulations, we consider multimedia coverage and messaging overhead as the two key metrics to evaluate the performance of our *self-orienting algorithms*. We assume that sensors are configured with their deployment locations (or capable of determining the same using a localization technique [19]). Note that, the overhead of localization scheme is not considered in our performance evaluations since it is not the main focus of this paper. Global access to obstacle locations on a calibrated coordinate system is available for the sensors. Recall from Section 4.3 that sensors perform the tests in sequence and those who send the `POSE_ADV_MSG` self-orient their viewpoints to an appropriate final FoV. When the initial messaging phase starts, sensor nodes set a timer and broadcast a `HELLO_MSG`. The value of this timer is determined using the average degree of connectivity of the sensor. When expired, each sensor node updates its coverage and transmission neighbor list. This $O(N)$ message complexity operation is repeated periodically (i.e., for each phase) throughout the lifetime of the network.

In the simulation, each sensor scans the perimeter of the *FoV* disk to determine possible *oFoVs* and xFoVs. This is a low-computational-cost operation even for low-capacity multimedia sensors. Once the $\Lambda_i^{\alpha_i}$ of s_i is determined, the size of the visible $v\Lambda_i^{\alpha_i} \subseteq \Lambda_i^{\alpha_i}$ is calculated to find the total visible *FoV*. However, *vFoV* might be an arbitrary polygon due to several obstacles, and calculating exact shapes require a complex geometric library support. Therefore, total visible *FoV* is calculated in a bitmap fashion using 62,500 bins (i.e., $1\text{ m} \times 1\text{ m}$) bins for each point) on the $250\text{ m} \times 250\text{ m}$ field $\mathscr{A}$. Bins that fall into a sensor's triangular *FoV* are tested for line of sight (LOS) view (i.e., the line segment from the bin corresponding to the *FoV* point to the camera sensor should not intersect with any obstacle on the field). Using this test for all *FoV* points of all sensors, we determine the total vFoVs in all scenarios.

The effect of self-orientation on coverage: In Fig. 11a, an experimental outcome with random orientation is illustrated, resulting 21.09% overall coverage of the field. Although sensors had the capability to exchange information regarding their neighbors and obstacles, due to the lack of proper coordination, several sensors went overlap-

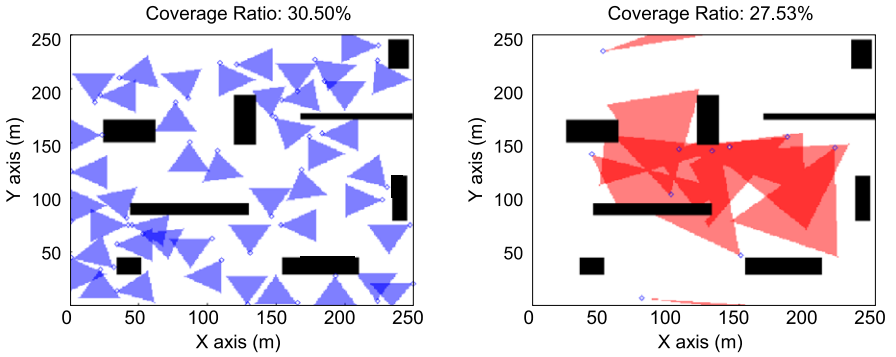
Nodes	Obstacles	Multimedia coverage gain %
50	4	8.93%
50	8	6.06%
100	4	16.39%
100	8	12.08%
150	4	9.3%
150	8	7.8%

Fig. 12. Multimedia coverage ratios.



(a) Random orientation. (b) Self-orientation algorithm.

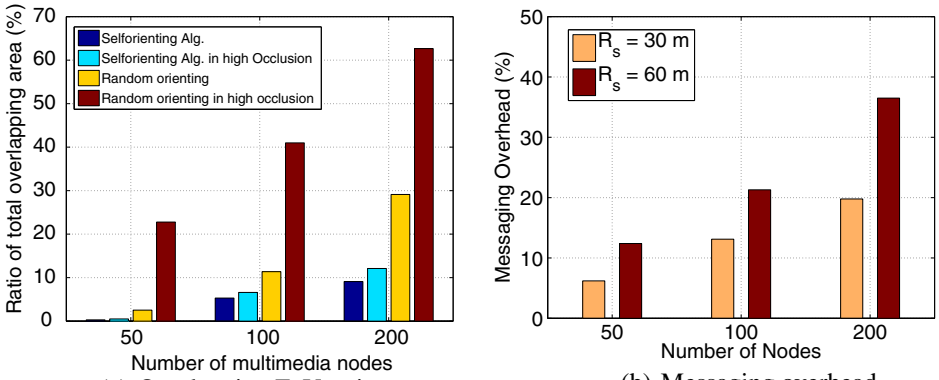
Fig. 11. Multimedia coverage.



(a) Self-orientation algorithm.

(b) Random orientation for high resolution cameras.

Fig. 13. Highly-occluded sensing field.



(a) Overlapping FoV ratio.

(b) Messaging overhead.

Fig. 14. Messaging overhead.

ping. Mostly, occluded FoVs are a serious waste of resources. However, in Fig. 11b, sensors were programmed to determine their FoV disk, scan their coverage neighbors, obstacles and communicate with their neighbors to decide on the optimal pose. We observed that by using our approach in a 50 node network, a coverage ratio of 29.88% could be achieved, which is very close to the maximum possible coverage with 50 sensors of 30 m range on this field. A set of resultant coverage gains (%) of self-orienting algorithms is also given in Fig. 12 for different scenarios. Here, coverage gain is defined as the increase (in %) when self-orientation algorithm is used compared to random orientation in the same deployment. The results are the average of five iterations of each test.

The effect of resolution vs. node density: In Fig. 13, a sensing field with several obstacles (represented by black rectangular areas) and 50 multimedia sensors is shown. Each multimedia sensor is illustrated with a "small diamond" and its vFoV is shown with a dark shaded area. We observe that in highly-occluded environments, a small number of high-resolution camera sensors[1] perform much worse than a large number of low-resolution camera sensors. In Fig. 13a, a sensing field with randomly placed eight obstacles of different sizes is given. This field has 50 low-resolution camera nodes deployed with 30 m range. On the other hand, in Fig. 13b, the same sensing field is installed with 10 high-resolution camera nodes of 100 m range. Triple-ranged camera sensors in Fig. 13 result in about 10% degraded total coverage compared to the low-resolution sensors. Note that the FoVs for the network in Fig. 13 are set using the self-orienting algorithm explained earlier.

The effect of self-orientation on overlapping area: The self-orienting algorithm not only determines occlusion-free viewpoints for sensors but also avoids overlapping FoVs using neighbor-distance test, as explained in Section 4.2.2. For example, in Fig. 11, coverage ratio gains up to 41% were obtained by using self-orientation. Preventing overlapping FoVs contributed 12% of the the total increase in coverage. In Fig. 14a, we show the ratio of overlapping FoV when self-orientation algorithm is used. We observe that an increase in the number of nodes causes a dramatic increase in the total overlapping area. Self-orientation results in at most 9% overlapping area, whereas random orientation results in overlapping areas up to 29%.

The overhead of the self-orientation algorithm: For the first test we present, multimedia sensors with a 30 m range on a field of 250 m $\times$ 250 m are used. In Fig. 14b,

[1] Note that in this context, a high-resolution camera refers to the one that can capture information from a larger sensing area.

we show the ratio of total number of messages used by the self-orienting algorithm to the total number of control messages, including routing. As we explained in Section 4.3, our algorithm uses $O(n)$ messages which is 6% of all control messages on average when $N = 50$. The ratio increases only up to 35% of total control traffic when $N = 200$ and $R_s = 60$ m, indicating a very dense network with high-degree of connectivity.

5.3. Observations

From the presented experiments, our observations on multimedia coverage and self-orientation can be summarized as follows:

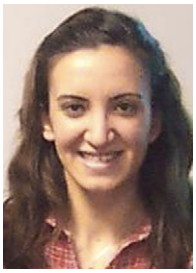
- Increasing the node density does not increase the coverage ratio proportionally, on the contrary, it results in large overlapping areas.
- In highly-occluded fields, many low-resolution cameras constitute a much better alternative to few high-resolution camera nodes.
- Self-orientation is a very effective way to increase coverage ratio while avoiding occlusions and overlapping FoVs.

6. Conclusion

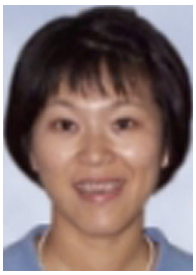
In this paper, we proposed a self-orienting algorithm for multimedia wireless sensor networks in order to attain occlusion-free coverage. We find that (i) the proposed algorithm uses local information; that is, communication overhead is incurred only between neighboring nodes with a complexity of $O(N)$, (ii) the proposed algorithm is fully distributed, which can operate after initial deployment and update the orientation of multimedia sensors on the fly, (iii) the proposed algorithm can support prioritized or accurate observation that require more than multiple inputs from more than one sensor node, and (iv) coverage can be increased even for sparse networks by using self-orientation instead of random orientations, for arbitrary obstacles in the sensor field.

References

[1] M. Rahimi, S. Ahmadian, D. Zats, R. Laufer, D. Estrin. Magic of numbers in networks of wireless image sensors, in: Proc. of SenSys Workshop on Distributed Cameras, 2006.
[2] P. Sambhoos, A.B. Hasan, R. Han, T. Lookabaugh, J. Mulligan. Weeblevideo – wide angle field-of-view video sensor networks, in: Proc. SenSys Workshop on Distributed Cameras, 2006.
[3] K.A. Tarabanis, R.Y. Tsai, A. Kaul, Computing occlusion-free viewpoints, IEEE Transactions on Pattern Analysis and Machine Intelligence 2 (1996) 273–292.
[4] O.B. Akan, Performance of transport protocols for multimedia communications in wireless sensor networks, IEEE Communications Letters 11 (10) (2007) 826–828.
[5] Y. Gu, Y. Tian, E. Ekici, Real-time multimedia processing in video sensor networks, Signal Processing: Image Communication Journal 22 (3) (2007) 237–251.
[6] M. Cardei, M. Thai, W. Wu, Energy-efficient target coverage in wireless sensor networks, in: Proc. IEEE Infocom, Miami, Forida, USA, 2005.
[7] H. Gupta, S.R. Das, Q. Gu, Connected sensor cover: self-organization of sensor networks for efficient query execution, in: Proc. ACM Mobihoc, Annapolis, Maryland, USA, 2003.
[8] S. Slijepcevic, M. Potkonjak, Power efficient organization of wireless sensor networks, in: Proc. IEEE International Conference on Communications, Helsinki, 2003, pp. 472–476.
[9] D. Tian, N.D. Georganas, A coverage-preserving node scheduling scheme for large wireless sensor networks, in: Proc. First ACM Int. Workshop on Wireless Sensor Networks and Applications, Georgia, USA, 2002.
[10] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: Proc. SenSys, Los Angeles, California, USA, 2003, pp. 28–40.
[11] H. Zhang, J. Hou, Maintaining sensor coverage and connectivity in large sensor networks, in: Proc. NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Adhoc Wireless, and Peer-to-Peer Networks, 2004.
[12] I.F. Akyildiz, T. Melodia, K.R. Chowdhury, A Survey on Wireless Multimedia Sensor Networks, Computer Networks.
[13] S. Soro, W.B. Heinzelman, On the coverage problem in video-based wireless sensor networks, in: Proc. of the Second Workshop on Broadband Advanced Sensor Networks (BaseNets '05), 2005.
[14] J. Adriaens, S. Megerian, M. Potkonjak, Optimal worst-case coverage of directional field-of-view sensor networks, in: Proc. of IEEE SECON, 2006.
[15] J. O'Rourke, Art Gallery Theorems and Algorithms, Oxford University Press, New York, 1987.
[16] S. Fleishman, D. Cohen-Or, D. Lischinski, Automatic camera placement for image-based modeling, in: Proc. Pacific Graphics, vol. 99, 1999, pp. 12–20.
[17] V. Isler, S. Kannan, K. Daniilidis, Vc-dimension of exterior visibility, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 667–671.
[18] A. Williams, D. Xie, S. Ou, Distributed smart cameras for aging in place, in: Proc. SenSys Workshop on Distributed Cameras, 2006.
[19] T. He, C. Huang, B.M. Blum, J.A. Abdelzaher, Range-free localization schemes for large scale sensor networks, in: In Proc. ACM Mobicom, San Diego, California, USA, 2003, pp. 81–95.
[20] M. Nicolescu, G. Medioni, Electronic pan-tilt-zoom: a solution for intelligent room systems.
[21] M.-F. Huang, Y.-C. Tseng, The coverage problem in a wireless sensor network, in: Proc. ACM WSNA, San Diego, CA, USA, 2003.
[22] P. Kulkarni, D. Ganesan, P. Shenoy, Q. Lu, SensEye: a multi-tier camera sensor network, in: Proc. ACM Multimedia, 2005.
[23] Ns-2, http://www.isi.edu/nsnam/ns.



Nurcan Tezcan received the B.S. and M.S. degrees in computer engineering from Yeditepe University, Turkey in 2001 and Bogazici University, Turkey, in 2004, respectively. She is currently working toward the Ph.D. degree in computer engineering at the North Carolina State University, Raleigh. Her current research interests are in transport layer protocols, energy-efficient algorithms and performance analysis of wireless networks.



Wenye Wang (M'98/ACM'99) received the B.S. and M.S. degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 1986 and 1991, respectively. She also received the M.S.E.E. and Ph.D. degree from Georgia Institute of Technology, Atlanta, Georgia in 1999 and 2002, respectively. She is now an Assistant Professor with the Department of Electrical and Computer Engineering, North Carolina State University. Her research interests are in mobile and secure computing, quality-of-service (QoS) sensitive networking protocols in single- and multihop networks. She has served on program committees for IEEE INFOCOM, ICC, ICCCN in 2004. Dr. Wang is a recipient of NSF CAREER Award in 2006. She has been a member of the Association for Computing Machinery since 2002.