

# MULTI-CAMERA SURVEILLANCE WITH VISUAL TAGGING AND GENERIC CAMERA PLACEMENT

Jian Zhao and Sen-ching S. Cheung

University of Kentucky  
Center for Visualization and Virtual Environment  
1 Quality Street, Suite 800, Lexington, Kentucky 40507

## ABSTRACT

A common goal in many vision applications is to identify and track human objects with distinctive visual features or “tags”. Examples range from identifying distinct soccer player by his jersey number to locating the face of an individual that produces a match in a face recognition system. In this paper, we made two contributions to this “visual tagging” problem. First, we propose a general framework for camera placement. This framework can measure the performance of any particular camera placement using simulation method. The optimal placement strategy can then be obtained by iterative grid-based binary integer programming. Second, we focus on tracking specific colored tags used in a privacy-protecting visual surveillance network. By building a color classifier for tag detection and using epipolar geometry between multiple cameras for occlusion handling, our proposed system can identify, track and visually obfuscate individuals whose privacy in the surveillance video needs to be protected.

**Index Terms**— multi-camera tracking, camera placement, epipolar geometry, visual tags, privacy protection

## 1. INTRODUCTION

One of the most important tasks in distributed camera network is to identify and track common objects across disparate camera views. It is a difficult problem because image features like corners or scale-invariant feature transform (SIFT) may vary significantly between different camera views due to disparity, occlusions and variation in illumination. One possible solution is to utilize semantically rich visual features based either on intrinsic characteristics such as faces or gaits, or artificial marks like numbers on sports jersey or special clothing like hats or name tags. We call the problem of identifying distinctive visual features from an object in two or more camera views the “Visual Tagging” problem.

Even though visual tagging may require more sophisticated classifiers for tracking or even cooperation from surveillance subjects, it has a wide range of important applications. For example, using distinctive biometric features, visual tagging allows tracking of terrorist suspects across a large area like an airport that already has a network of video cameras in place. Automatic tracking of jersey numbers of players in a football field can be used to assist coaches in the study of different tactics and strategies. A recent application of visual tagging is to use special tags to identify individuals whose privacy need to be protected in a video surveillance network [1]. Once a person is identified to possess a certain tag, his/her images in all cameras will be obfuscated to protect the identity. The ideal tag should be small, light and easy to carry. This application of visual tagging on privacy

protection is particularly challenging as the goal is to obfuscate the images of an individual in ALL CAMERA VIEWS, regardless of whether the small tag is visible to a particular camera. If a tag is visible in only two camera views, its location can be transferred to a third camera view by projecting the corresponding epipolar lines to the new view as shown in Figure 1. This requires a careful design of distributed algorithms in different cameras so that they can optimally share information about the knowledge of the tags. This is the subject of this paper.



Epipolar  
Lines from  
two other  
cameras

**Fig. 1.** Transfer of tag information via epipolar geometry

In this paper, we study various aspects of the visual tagging problem including tagging performance under different camera placement, optimal camera placement strategy, communication strategy for tag localization and its application in a privacy protection system. The main contributions of our paper include the followings:

1. We present a novel comprehensive visibility metric to measure the performance of observing a visual tag modeled as a small object with orientation in any arbitrary camera placement.
2. We develop an iterative optimization method to simultaneously determine the optimal number and positions of cameras in achieving the desired level of visibility.
3. With colored tags and epipolar geometry between multiple cameras for occlusion handling, we are the first to demonstrate how visual tagging between multiple cameras can be used in a privacy protected video surveillance system.

The rest of the paper is organized as follows. In Section 2, we briefly review the state-of-the-art in camera placement problem and privacy protection schemes. In Section 3, we present a generic model

for measuring the performance of a particular camera placement. Section 4 specializes the generic model to define a metric for the “visual tagging” problem based on the probability of observing a tag from at least two cameras. Using this metric, we formulate in Section 5 the search of the optimal camera placement as a Binary Integer Programming problem. With the optimal camera configuration in place, we describe how we use visual tagging to enhance privacy protection in Section 6. Preliminary experimental results on both simulations and real videos are presented in Section 7. We conclude the paper by discussing future work in Section 8.

## 2. RELATED WORK

The problem of finding the optimal camera placement has been studied for a long time. The earliest investigation can be traced back to the “art gallery problem” in computational geometry. This problem is the theoretical study on how to place cameras in an arbitrary-shaped polygon so as to cover the entire area [2]. While the theoretical difficulties of the camera placement problem are thoroughly studied, few solutions can be directly applied to realistic computer vision problems. Camera placement has also been studied in the field of photogrammetry for building the most accurate 3D model. Various metrics such as visual hull [3] and viewpoint entropy [4] have been developed and optimization are realized by various types of ad-hoc searching and heuristics. These techniques assume very dense placement of cameras and are not applicable to wide-area wide-baseline camera networks.

Recently, Ramakrishnan et al. propose a framework to study the performance of sensor coverage in wide-area sensor networks [5]. Unlike previous techniques, their approach takes into account the orientation of the object. They develop a metric to compute the probability of observing an object of random orientation from one sensor, and use that to recursively compute the performance for multiple sensors. While their approach can be used to study the performance of a fixed number of cameras, it is not obvious on how to extend their scheme to find the optimal number of cameras as well as how to incorporate other constraints such as the visibility from more than one camera.

On the other hand, Horster and Lienhart develop a flexible camera placement model by discretizing the space into grid and denoting the possible placement of camera as a binary variable over each grid point [6]. The optimal camera configuration is formulated as an integer linear programming problem which can incorporate different constraints pertinent to a particular application. While our approach follows a similar optimization strategy, we adopt a more sophisticated probabilistic approach to capture the uncertainty of object orientation. We also improve upon the fixed grid point strategy to provide more efficient traversal of the search space. None of the techniques above is suitable for the “visual tagging” problem because they do not consider the requirement of viewing an object at arbitrary location and orientation from two or more cameras.

The application of visual tagging in privacy protected video surveillance is first proposed by Schiff et al. [1]. They use an Adaboost classifier to identify hard hats and apply particular filtering to track them through time. The privacy of an individual wearing such a hat is protected by having his/her face covered by a black box. The choice of hard hats is to provide a significant target for tracking and recognition and to minimize occlusion. On the other hand, its prominent presence may be singled out in certain environments. In addition, their scheme does not incorporate any cues from multiple cameras. While Schiff et al.’s may be the only scheme that addresses the identification problem in a privacy protected video surveillance, many

others have proposed schemes to obfuscate identity information in video. Datong chen et al.[7] present a system obscuring the human body while preserving the structure and motion information. Newton et al. develop a face modification algorithm to counter face recognition [8]. Wickramasuri et al. use RFID to track individuals and visually replace them with a static background [9]. Our previous work demonstrate an efficient video in-painting algorithm to erase individuals for privacy protection [10] and present a data hiding scheme to preserve privacy information in compressed video [11].

## 3. GENERAL VISIBILITY MODEL

In this section, we outline a general model to compute the visibility of a single tag  $P$  in a confined environment. We assume a two-dimensional model but the analysis can be easily extended to three-dimensional. The 2D model, however, is usually adequate for visibility calculations, assuming that the cameras are mounted on the low ceiling typically seen in many indoor office environment.

Given the number of cameras and their placement in the environment, we can model the visibility of tag  $P$  as a positive function  $V_m(x, y, \theta, r)$ , based on the coordinates  $(x, y)$  of  $P$ , its pose  $\theta$  with respect to a reference direction, and its half-length  $r$ . An example is shown in Figure 2.

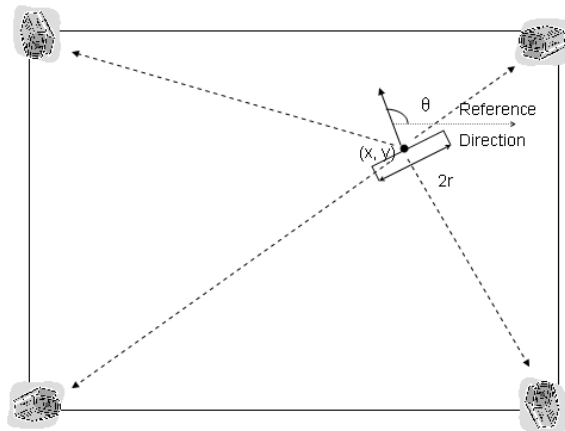


Fig. 2. General visibility model of a tag  $P$

The visibility function  $V_m$  provides an aggregate measure of the projected sizes of tag  $P$  on the image planes of different cameras. Based on the particular application,  $V_m$  can use different aggregation method and incorporate a variety of camera and environmental constraints. The specific visibility function suitable for visual tagging will be introduced in Section 4. Note that the dependency of  $V_m$  on  $\theta$  allows us to model self-occlusion. It does not, however, model occlusion from other objects and thus we assume that there is only one tag in the environment. While the half-length  $r$  of the tag is relatively constant, the coordinates and the pose are random variables governed by a prior distribution  $f(x, y, \theta)$ . This prior distribution can be used to incorporate prior knowledge about the environment. For example, if an application is interested in locating faces, the likelihood of the head positions and poses are affected by furnishings and attractions such as television sets and paintings.

To correctly identify and track any visual tag, a classification algorithm would require the tag size on the image to be bigger than a certain minimum size, though a larger projected size usually does

not make much difference. Assuming that this minimum size is  $T$  pixels, this requirement can be modeled by binarizing the visibility function as follows:

$$V_b(x, y, \theta, r|T) = \begin{cases} 1 & V_m(x, y, \theta, r) > T \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Finally, we define  $\eta$ , the *mean visibility*, to be the single metric for measuring the visibility of  $P$  over the entire parameter space:

$$\eta = \int V_b(x, y, \theta, r|T) \cdot f(x, y, \theta) dx dy d\theta \quad (2)$$

Except for the most straightforward environment such as the single camera case discussed in Section 4.1, Equation (2) does not admit a closed-form solution. Nevertheless, it can be easily estimated by using standard Monte-Carlo sampling and its many variants.

#### 4. VISIBILITY MODEL FOR VISUAL TAGGING

In this section, we present a visibility model for the visual tagging problem. This model is a specialization of the general model in Section 3. The goal is to design a visibility function  $V_m$  that can measure the performance of a camera placement for viewing a tag in two or more cameras. In addition to the assumptions listed in Section 3, we further assume that the environment of interest is convex. This implies that the visibility of the tag with respect to a camera depends only on the coordinates and poses of the tag and the camera. We also assume the basic pinhole camera model. Let us start with the simple case for one camera.

##### 4.1. Visibility for single camera

Recall the visibility function  $V_m(x, y, \theta)$  measures the projected size of the tag on the image plane. Instead of arbitrarily choosing a coordinate system and a reference direction, we directly compute the projected size  $l$  based on the following geometrical quantities of the tag  $P$  and the camera  $C$  as well as the optical properties of  $C$ :

$d$	the length of line segment $\overline{PC}$ that joins the camera pinhole to the center of the tag
$\alpha$	the angle between the tag orientation and $\overline{PC}$
$\beta$	the angle between the camera projection direction and $\overline{PC}$
$f$	the camera's focal length
$FOV$	the camera's field of view
$p$	the width of a pixel on the image plane

These quantities are illustrated in Figure 3. For convenience, we assume that the image plane is distance  $f$  in front of the camera pinhole.

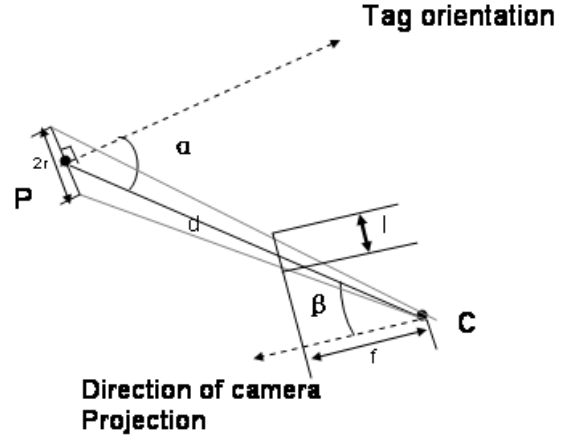
First, it is straightforward to see that  $P$  is visible by the camera if and only if the following two conditions hold:

1.  $P$  is within the camera's field of view or

$$|\beta| < FOV/2 \quad (3)$$

2.  $P$  is not self-occluded when seen from the camera or

$$|\alpha| < \pi/2 \quad (4)$$



**Fig. 3.** The size of object of interest in the image,  $l$ , is determined by the object size  $r$ , distance  $d$ , and angles  $\alpha, \beta$

We thus define the visibility  $V(P, C)$  between the tag  $P$  and a single camera  $C$  as follows:

$$\begin{aligned} V(P, C) &= \begin{cases} l & \alpha, \beta \text{ satisfy conditions (3) and (4)} \\ 0 & \text{otherwise} \end{cases} \\ &= U\left(\frac{\pi}{2} - |\alpha|\right) \cdot U\left(\frac{FOV}{2} - |\beta|\right) \cdot l \end{aligned} \quad (5)$$

where  $U(\cdot)$  is the unit step function. A threshold version is sometimes more convenient:

$$V_b(P, C|T) = \begin{cases} 1 & \text{if } V(P, C) > T \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Second, we express the projected tag size  $l$  in terms of the basic geometric properties defined above. Define  $\angle 1$  and  $\angle 2$  as the angles between  $\overline{PC}$  and the lines joining either ends of the tag to the pinhole. We have four different cases:

**Case 1:**  $\beta$  and  $\angle 2$  are on the same side with  $\beta > \angle 2$ . This is illustrated in Figure 4(a). Using the two right-angled triangles formed among the pinhole, the optical center and the two respective endpoints of the projection, we can compute  $l$  in terms of the number of pixels as follows:

$$l = [\tan(\beta + \angle 1) - \tan(\beta - \angle 2)] \cdot \frac{f}{p} \quad (7)$$

**Case 2:**  $\beta$  and  $\angle 1$  are on the same side with  $\beta > \angle 1$ . This is illustrated in Figure 4(b). Based on a similar argument as case 1, we have

$$l = [\tan(\beta + \angle 2) - \tan(\beta - \angle 1)] \cdot \frac{f}{p} \quad (8)$$

**Case 3:**  $\beta$  and  $\angle 1$  are on the same side with  $\beta < \angle 1$ . This is illustrated in Figure 4(c). Based on a similar argument as case 1, we have

$$l = [\tan(\beta + \angle 2) + \tan(\angle 1 - \beta)] \cdot \frac{f}{p} \quad (9)$$

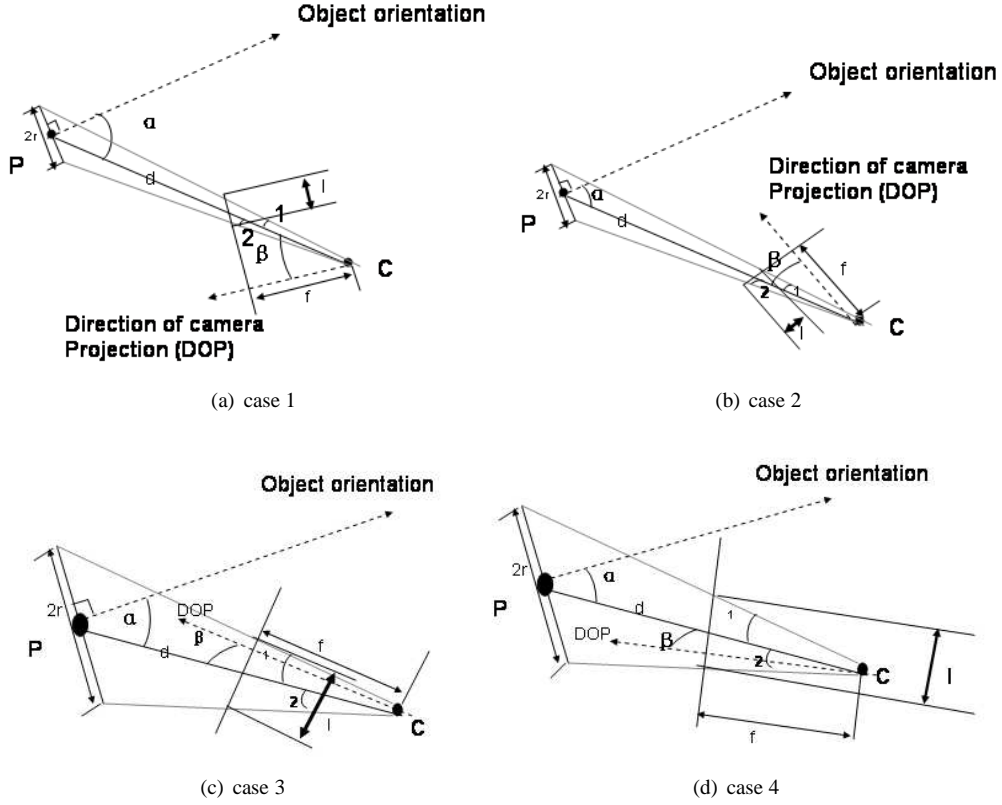


Fig. 4. Four cases to compute the projected tag size  $l$ .

**Case 4:**  $\beta$  and  $\angle 2$  are on the same side with  $\beta < \angle 2$ . This is illustrated in Figure 4(d). Based on a similar argument as case 1, we have

$$l = [\tan(\beta + \angle 1) + \tan(\angle 2 - \beta)] \cdot \frac{f}{p} \quad (10)$$

Note that Equation (7) is equivalent to Equation (10), and Equation (8) is equivalent to Equation (9). The two sets of equations can be differentiated based on whether  $\alpha$  and  $\beta$  are on the same side of  $\overline{PC}$ . In addition, we can express the tangent of  $\angle 1$  and  $\angle 2$  in terms of the tag parameters  $\alpha$ ,  $\beta$ ,  $r$  and  $d$  as follows:

$$\tan \angle 1 = \frac{r \cos \alpha}{d + r \sin \alpha} \quad (11)$$

and

$$\tan \angle 2 = \frac{r \cos \alpha}{d - r \sin \alpha} \quad (12)$$

Combining the above observations with Equation (11) and (12), we can compute  $l$  as a function of  $\alpha$ ,  $\beta$ ,  $r$  and  $d$ :

$$l(\alpha, \beta, d, r) = \begin{cases} \frac{2dr \cos \alpha (\tan^2 \beta + 1) \cdot (f/p)}{d^2 - r^2 \sin^2 \alpha + r^2 \tan \beta (\tan \beta \cos^2 \alpha - \sin 2\alpha)} & \alpha, \beta \text{ on the same side of } \overline{PC} \\ \frac{2dr \cos \alpha (\tan^2 \beta + 1) \cdot (f/p)}{d^2 - r^2 \sin^2 \alpha + r^2 \tan \beta (\tan \beta \cos^2 \alpha + \sin 2\alpha)} & \text{otherwise} \end{cases} \quad (13)$$

## 4.2. Visibility for Visual Tagging

To extend the single-camera case in Section 4.1 to multiple cameras, we note that the visibility of the tag from one camera does not affect the other and so each camera can be treated independently. The visual tagging problem requires that the tag must be visible by at least two cameras. Given  $N$  cameras  $C_1, C_2, \dots, C_N$ , we define the visibility function  $V_m(x, y, \theta, r)$  for visual tagging to be the *second largest projected tag size among all the cameras*:

$$V_m(x, y, \theta, r) = \max_{i \in \{1, 2, \dots, N\}, i \neq k} V(P, C_i) \quad (14)$$

where  $V(P, C_i)$  is the visibility of tag  $P$  with respect to camera  $C_i$  as defined in Equation (5) and  $C_k$  is the camera that captures the largest tag image or  $k = \arg \max_{j \in \{1, 2, \dots, N\}} V(P, C_j)$ .

Even if the environment is densely covered with cameras, there is no guarantee that a tag at an arbitrary position will be visible to at least two cameras – a tag next to and facing the wall is only visible if there are two cameras right in front of the tag. In the actual design of camera networks, we would like to avoid such pathological cases and to adopt the design if most of the environment is perfectly visible. We call the area in the environment a *Perfect Zone* in which a tag of half-length  $r$ , regardless of its pose, is visible to two cameras. In

other words, the Perfect Zone can be defined as

$$\begin{aligned} \text{Perfect Zone} &= \{(x, y) : V_m(x, y, \theta, r) > 0 \text{ for all } \theta\} \\ &= \{(x, y) : V_b(x, y, \theta, r|T) = 1 \text{ for all } \theta\} \end{aligned} \quad (15)$$

## 5. OPTIMAL CAMERA PLACEMENT

In the previous sections, we show how to compute visibility of arbitrary camera placement. In this section, we demonstrate how to compute the optimal camera placement – the minimum number of cameras used, their poses, and their positions in the environment in order to achieve a target  $\eta_t$ .

Due to the difficulty in obtaining a continuous solution over an arbitrary-shape environment, we follow a similar approach as in [6] by finding an approximate solution over a discrete grid. We first discretize the environment into a lattice *gridP* of  $N_p$  grid points  $\{P_i : i = 1, 2, \dots, N_p\}$  where we are interested in finding the tag visibility. We also discretize the camera space, that includes both the 2D locations and the orientation, into a lattice *gridC* of  $N_c$  grid points  $\{C_i : i = 1, 2, \dots, N_c\}$ . To formulate the optimization problem, we associate each camera grid point  $C_i$  with a binary variable  $b_i$  such that

$$b_i = \begin{cases} 1 & \text{a camera is present at } C_i \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

The optimization problem can be described as the minimization of the number of cameras:

$$\min \sum_{i=1}^{N_c} b_i \quad (17)$$

subjected to the visual tagging constraint:

$$\sum_{i=1}^{N_c} b_i \cdot V_b(P_j, C_i|T) \geq 2 \quad (18)$$

for each possible tag configuration  $P_j$  and the ‘‘single camera’’ constraint:

$$\sum_{\text{all } C_i \text{ at } (x, y)} b_i \leq 1 \quad (19)$$

The first constraint (18) represents the requirement of visual tagging that all tags must be visible by at least two cameras. As defined in Equation (6),  $V_b(P_j, C_i|T)$  measures the visibility of tag  $P_j$  with respect to camera at  $C_i$ . In other words,  $P_j$  satisfying the constraint (18) must be in the perfect zone. The second constraint in (19) is a set of inequalities to guarantee that only one camera is placed at any 2D location. The optimization problem in (17) with constraints (18) and (19) forms a standard Binary Integer Programming (BIP) problem. While the general BIP problem is NP-hard, fast approximate solutions exist and can be obtained using software libraries such as `lp_solve` [12].

The choice of the grid points in *gridP* and *gridC* affect the outcome of the optimization. As discussed in Section 4.2, there is no guarantee that a tag at a random location can be visible by two cameras even if there is a camera at every camera grid point. Thus, the optimization problem may not have a solution if the tag grid points are randomly placed. Instead, we start our search process at a configuration that guarantees a solution and then traverse the search space by gradually increasing the density of both *gridP* and *gridC*. The

initial configuration of *gridP* is to place a single tag at the center of the largest circle inscribed within the environment. This tag is guaranteed to be in the perfect zone if we put five center-facing cameras equally spaced on the circle – no matter what the tag orientation is, the tag will always be visible to two or more cameras. In order to produce a better estimate of  $\eta$ , *gridP* then grows uniformly in density within the interior of the environment but remains at least one interval away from the boundary. *gridC* maintains its initial density until the BIP solver fails to return an answer, at which point the density of *gridC* is increased. The iteration terminates when the target  $\eta_t$  is achieved or the density of *gridC* exceeds a predefined limit. The above process is described in Algorithm 1. The algorithm is guaranteed to terminate by setting a very high *maxDensity* so that the entire environment can be covered with cameras. In practice, as we will show in Section 7, it only requires a few iteration to arrive at a very high level of  $\eta$ .

**Input:** initial grid points for cameras *gridC* and tag *gridP*,  $\eta_t$ , maximum grid density *maxDensity*

**Output:** Camera placement *camPlace*

Set  $\eta = 0$ ;

**while**  $\eta \leq \eta_t$  **AND**  $\text{density}(\text{gridC}) \leq \text{maxDensity}$  **do**

**foreach**  $C_i$  in *gridC* **do**

**foreach**  $P_j$  in *gridP* **do**

            | Calculate  $V_b(P_j, C_i|0)$ ;

**end**

**end**

    Solve *newCamPlace* = `lp_solve(gridC, gridP,  $V_b$ )`;

**if** *newCamPlace* == `EMPTY` **then**

        | Increase density of *gridC*;

        | Decrease density of *gridP*;

**else**

        | *camPlace* = *newCamPlace*;

**end**

    Calculate  $\eta$  for *camPlace* by Monte Carlo Sampling;

    Increase density of *gridP*;

**end**

**Algorithm 1:** Optimal camera placement algorithm

## 6. VISUAL TAGGING FOR PRIVACY PROTECTION

In this section, we describe a system that uses visual tagging to protect privacy of selected individuals in a multi-camera video surveillance network. The cameras are positioned based on the optimal placement strategy described in Section 5. All cameras are calibrated such that camera  $C_k$  knows the set of fundamental matrices  $F_{ik}$  from camera  $C_i$  to  $C_k$ . In other words, if  $\mathbf{x}_i$  (in homogeneous coordinate) is a point on the image plane of  $C_i$ , then  $\mathbf{l}_k = F_{ik}\mathbf{x}_i$  is the epipolar line to  $\mathbf{x}_i$  on the image plane of  $C_k$ .

Individuals whose privacy need to be protected are wearing small rectangular colored tags. Each tag has a unique color for which we have prepared a color classifier. Our current implementation uses a pre-trained Gaussian Mixture Model classifier on the hue and saturation of each color. Using these classifiers, each camera identifies all pixels that match these colors, performs component grouping on pixels with the same color, and computes the centroid of each group. The image coordinates of the centroids and their corresponding colors along with the camera’s own ID are broadcast to all other cameras.

After receiving all messages from its peers, each camera  $C_k$  decodes the messages and builds a contingency table between cameras and color tags. If more than one camera provide information about a tag with a specific color, camera  $C_k$  computes the corresponding epipolar lines and estimates their point of intersection on its own image plane. Since these epipolar lines must come from the same tag, they will intersect at a point. By cross-correlating epipolar lines from other cameras, it is possible for camera  $C_k$  to identify the image location of these “virtual tags” even though they are not directly observed by  $C_k$ . In practice, these epipolar lines intersect in a small region instead of a single point due to the uncertainty in identifying the projection of the centroid from each camera and in computing the fundamental matrices between wide baseline cameras. While this error can be reduced by taking advantage of the specific shape of the tags, it is usually tolerable as it will be further correlated with the identified moving objects as discussed in the sequel.

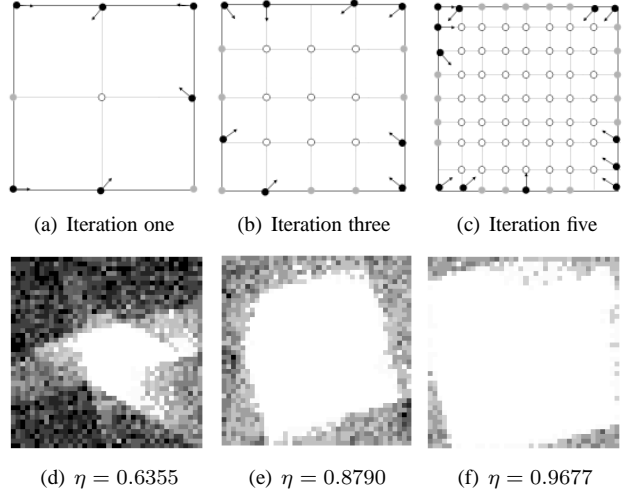
Based on our earlier work in [10], each camera combines a background subtraction algorithm with a probabilistic tracer to motion-segment individual objects in the video. Objects overlapped with any directly-observed tags or virtual tags are erased from the video by using an efficient object-template based in-painting scheme [10]. Preliminary results of this system are shown in Section 7.

## 7. EXPERIMENTAL RESULTS

In this section, we report three sets of experiments. In the first set of experiments, we apply the algorithm introduced in Section 5 to derive the optimal camera placement for a target mean visibility  $\eta_t$ . To facilitate testing of the camera placement strategy, our second set of experimental results are based on simulating a virtual environment in which cameras can be placed in arbitrary positions. Finally, we illustrate the use of visual tagging for privacy protection in an actual three-camera surveillance network. All the simulations assume a room of dimension  $10\text{m} \times 10\text{m}$  and a tag of half-length  $r = 10\text{cm}$  long. For the camera and lens models, we assume a pixel width of  $5.6 \mu\text{m}$ , focal length of  $8 \text{ cm}$  and the field of view of  $120$  degrees. The threshold  $T$  for visibility is set to 5 pixels.

By setting the target mean visibility  $\eta_t = 0.95$ , the optimal placement algorithm terminates after five iterations. The snapshots after the first, third and fifth iteration are shown in Figure 5(a) to 5(f). Figure 5(a) to 5(c) show the tag grid points (hollowed circles), the camera grid points (solid circles) and the resulting optimal camera positions and poses (solid black circles with camera orientations). Both the camera and tag grids are refined over the course of these iterations. Figure 5(d) to 5(f) show the corresponding visibility function  $V_m(x, y, r, \theta)$  at each of the random sample location  $(x, y)$ , averaged over all possible  $\theta$  under an uniform distribution. A brighter pixel indicates a higher average visibility and a white pixel belongs to the perfect zone – that it is visible to two or more cameras regardless of the orientation. The mean visibility  $\eta$  is computed based on these random samples. Notice that while both the room and the grids are symmetric under  $90^\circ$  rotations, the optimal solution at each stage does not possess such symmetry. The reason is that our software randomly picks one of the many optimal configurations that satisfy the optimality criteria.

To validate the obtained optimal camera placement, we simulate a 3-D environment of the same dimension in OpenGL and put twelve cameras according the optimal placement computed in Figure 5(c). A humanoid wearing a visual tag with random position and pose is created [13]. A sample of camera views are shown in Figure 6. Out of 140 random humanoid scenes, our visual inspection identifies 14 scenes in which the tag is not visible to at least two cameras. This re-



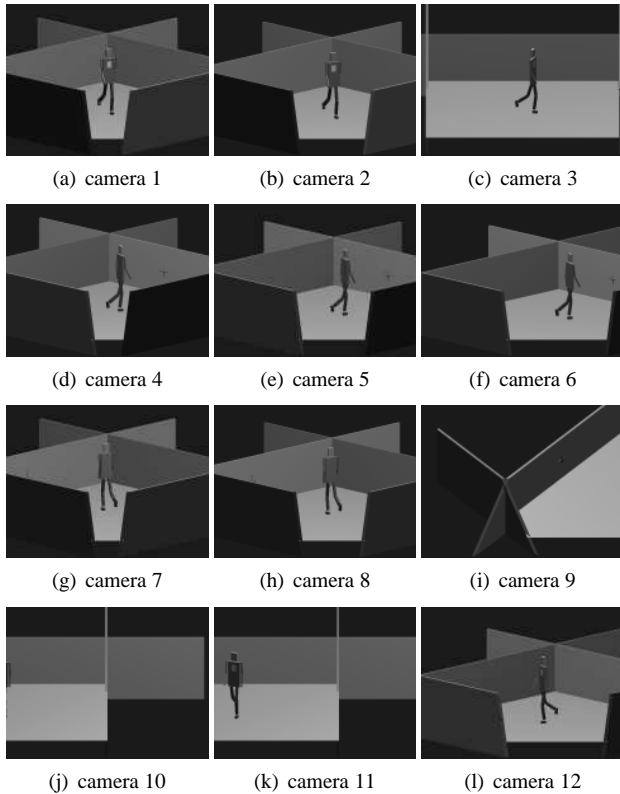
**Fig. 5.** First few iterations of the camera placement algorithm. Figures 5(a) to 5(c) show the grid points as well as the optimal camera poses and locations. The tag grid consists of only the hollowed circles and the camera grid consists of all the solid grid points. The black dots show the optimum camera position after the iteration and the arrows show the camera pose. The corresponding average visibility performances are shown in Figure 5(d) to 5(f).

sults in an estimate of  $\eta \approx 0.9$ , lower than the expected 0.9677. Part of the reasons is that our model does not take into account the elevation of the camera, which makes the tag invisible when the humanoid is too closed to the camera. However, if we restrict the positions of the humanoid to be within the middle  $7.5\text{m} \times 7.5\text{m}$  area of the room, we obtain only 1 miss out 150 random scenes. This confirms the presence of the perfect zone in the middle of the environment.

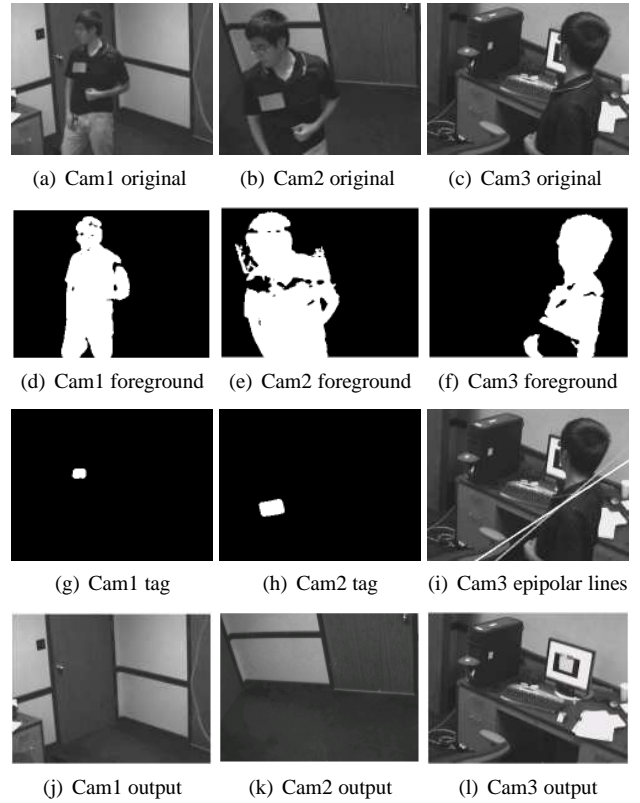
The final experiment involves using three real cameras mounted on the ceiling of our laboratory. Figure 7(a) to 7(c) show the three views of the same person. The foreground moving blobs are shown in Figure 7(d) to 7(f). Note that the tag is only visible in Cam1 and Cam2 but not in Cam3. Figure 7(g) and 7(h) show that our classifier can clearly identify the tag in Cam1 and Cam2. No such pixels are detected in Cam3, which resorts to calculating the two epipolar lines in Figure 7(i) based on information sent by Cam1 and Cam2. For Cam1 and Cam2, the tag pixels overlap with those of the foreground blobs. As a result, the foreground blobs are erased and the output frames are shown in Figure 7(j) and 7(k). As for Cam3, since the intersection of the two epipolar lines lie within the foreground blob, the blob is also erased as shown in Figure 7(l).

## 8. FUTURE WORK

In this paper, we have proposed a multi-camera surveillance system with visual tagging and camera placement model. By building a camera placement metric using planar geometry, we have derived an optimal camera placement strategy using iterative grid based binary integer programming. Equipped with the optimal camera placement, we have presented a multi-camera surveillance system capable of robustly identifying the visual tag and protecting the privacy of selected individuals by obfuscating their images in all camera views. We are currently extending the camera model to handle mutual occlusion among multiple tags so that robust tracking in crowded en-



**Fig. 6.** Different views of the virtual environment. The cameras are positioned as in Figure 5(f) with camera 1 being the one at the lower left corner and the camera number increases counter-clockwise around the perimeter of the room.



**Fig. 7.** Outputs of the privacy protection system. 7(a) to 7(c) are the originals and 7(d) to 7(f) are the foreground blobs. 7(g) and 7(h) are the detected tags and 7(i) are the two epipolar lines computed based on the positions of the tag from other cameras. The output in-painted frames are shown in 7(j) to 7(l).

vironment can be realized. We are also improving the registration techniques between different camera views by combining all available cues into a probabilistic framework.

## 9. REFERENCES

- [1] J. Schiff, M. Meingast, D. Mulligan, S. Sastry, and K. Goldberg, "Respectful cameras: Detecting visual markers in real-time to address privacy concerns," in *International Conference on Intelligent Robots and Systems (IROS)*, 2007.
- [2] Joseph O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987.
- [3] D. Yang, J. Shin, A. Ercan, and L. Guibas, "Sensor tasking for occupancy reasoning in a camera network," in *1st Workshop on Broadband Advanced Sensor Networks (BASENETS)*. IEEE/ICST, 2004.
- [4] P.-P. Vazquez, M. Feixas, M. Sbert, and W. Heidrich, "Viewpoint selection using viewpoint entropy," in *Proceedings of the Vision Modeling and Visualization Conference (VMV01)*, 2001.
- [5] S. Ram, K. R. Ramakrishnan, P. K. Atrey, V. K. Singh, and M. S. Kankanhalli, "A design methodology for selection and placement of sensors in multimedia surveillance systems," in *VSSN '06: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, New York, NY, USA, 2006, pp. 121–130, ACM Press.
- [6] E. Horster and R. Lienhart, "On the optimal placement of multiple visual sensors," in *VSSN '06: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, New York, NY, USA, 2006, pp. 111–120, ACM Press.
- [7] D. Chen, Y. Chang, R. Yan, and J. Yang, "Tools for protecting the privacy of specific individuals in video," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. Article ID 75427, 9 pages, 2007, doi:10.1155/2007/75427.
- [8] E. N. Newton, L. Sweeney, and B. Main, "Preserving privacy by de-identifying face images," *IEEE transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 232–243, February 2005.
- [9] J. Wickramasuri et al., "Privacy protecting data collectino in media spaces," *ACM Multimedia*, pp. 48–55, October 2004.
- [10] S.-C. Cheung, J. Zhao, and V. Venkatesh M., "Efficient object-based video inpainting," in *Proceedings of IEEE International Conference on Image Processing, ICIP 2006*, 2006, pp. 705–708.
- [11] W. Zhang, S.-C. Cheung, and M. Chen, "Hiding privacy information in video surveillance system," in *Image Processing, IEEE International Conference on*. IEEE, 2005.
- [12] "Introduction to lp\_solve 5.5.0.10," <http://lpsolve.sourceforge.net/5.5/>.
- [13] K. Agarway and P. Winston, "Walker," <http://www.sgi.com/products/software/opengl/examples/glut/demos/zip/walker.zip>.