People Movement Linkage Based on Path Revision Across Multiple Cameras

I-Cheng Chang[1], Chieh-Yu Liu[2], Chung-Lin Huang[3], Kunal Kabi[1]

[1] Dept. of Computer Science and Information Engineering, National Dong Hwa University, Taiwan
[2] Dept. of Electrical Engineering, National Tsing Hua University, Taiwan
[3] Dept. of Applied Informatics and Multimedia, Asia University, Taiwan
icchang@gms.ndhu.edu.tw, m9661619@oz.nthu.edu.tw, clhuang@asia.edu.tw, kunal018kabi@gmail.com

Abstract

This work presents an object-based people motion linkage system, which tracks and records the behavior of each person across multiple cameras. The proposed system includes two phases: path construction phase and path revision phase. The spatiotemporal relationships among cameras are trained by batch-learning procedures, and the appearance model is improved by color calibration for different cameras. When a person moves across cameras, the spatiotemporal relationship and appearance model are used to obtain the correspondence. However, object tracking may be lost due to some unexpected events. We revise the tracking paths by using the backward tracking technique to connect the missed links. Moreover, the trained Hidden Markov Models are further used to reconstruct the normal paths of these objects. In the experimental results, we demonstrate the efficiency of our approach by showing the reconnection of missing paths under different conditions.

Keywords: Human tracking, Dynamic programming, Hidden Markov Models (HMMs), Multiple cameras with non-overlapping views, Backward tracking

1 Introduction

Recently, the visual surveillance system has become essential for many open spaces as well as private residential areas. The video recording mechanism of multiple cameras surveillance systems usually records the video scene from each camera and stored in various video files. It is a so-called camera-based video recording system. To investigate the video for a specific object, one must watch all the video files and perform a manual search for connections between different videos. Here, we aim to develop a recording system that can link and store the video of a specific object moving across different cameras, called the object-based video recording.

The task of object tracking across multi-camera is to establish the correspondence between observations of objects appearing in different cameras. To identify objects in the video, Black et al. [1] used HIS color space to improve illumination invariance, while keeping color details. Chang and Gong [2] and Dockstader and Tekalp [3] used a Bayesian network for tracking in cameras with overlapping views. Lee et al. [4] proposed an approach for tracking in cameras with overlapping field of views (FOVs) that does not require calibration. Khan et al. [5] used FOVs line constraints to track objects in overlapping cameras. Yilmaz et al. [6] provided an extensive survey of object tracking approaches and introduced some related topics. Other researchers ([7-9]) also discussed multi-camera tracking with overlapping FOVs.

Another research topic concerns disjoint camera views. Kettnaker and Zabih [10-11] proposed a Bayesian approach to track objects across cameras with disjoint views. They reconstruct the paths and solve a linear programming problem to establish the correspondence. Porikli and Divakaran [12] tracked objects using spatiotemporal and appearance cues and proposed a new solution to the inter-camera color calibration problem. Javed et al. [13] presented a system using Parzen windows to learn the camera network topology and path probabilities of objects. Individual tracks are corresponded by seeking the maximal posterior probability of the spatiotemporal and color appearances. Oh et al. [14] applied Markov Chain Monte Carlo principle for data association in multi-camera multiple hypothesis tracking. Yoon et al. [15] proposed an improved tracking algorithm to solve the multi-camera tracking problem in a disjoint camera view environment. Kuo et al. [16] applied a combination of appearance and space-time information obtained from each target in a disjoint multi-camera view environment. Simonjan et al. [17] proposed a distributed self-aware multi-camera tracking approach where the tracking of every object is performed in a distributed manner, and each camera is connected in a self-organizing way without central control.

*Corresponding Author: I-Cheng Chang; E-mail: icchang@gms.ndhu.edu.tw
DOI: 10.3966/160792642020072104030

Dick and Brooks [18] described the observed pattern of people motion in terms of a stochastic transition matrix, which can be used within and between FOVs. In the training phase, a person with an easily identified marker generates the observations. The camera network topology and transition model are assumed to be known. Ellis et al. [19] proposed a method without the need of hand-labeled correspondence or a training phase. They used a threshold technique to search for peaks in the time distribution of travel times between the entrance-exit pairs. Stauffer [20] extended this approach, and Tieu et al. [21] defined a transition motion based on statistical significance. They constructed a correspondence model for the entire set of cameras where cameras with overlapping and non-overlapping FOVs are adopted. Gilbert and Bowden [22] proposed a modified approach that can automatically and probabilistically find the main entry and exit areas within a camera using the incremental learning method. The proposed approach models both the color variations and posterior probability distributions of the spatiotemporal links between cameras. Nam et al. [23] used a Merge-Split (MS) method to solve object occlusion in a single camera and a grid-based approach to extract accurate object features. The method estimates transition times between various entry and exit zones and graphically represented camera topology as an undirected weighted graph using the transition probabilities. Tang et al. [24] proposed a wireless sensor network based on a pre-defined division structure for multi-target tracking. The novel idea of this approach is to control the number of camera sensors to minimize energy consumption and maximize operational utilization. Junejo et al. [25] proposed a framework which can self-calibrate dynamically moving and zooming cameras and determine the orientations. Raty [26] reviewed the development of video surveillance systems and presented the contemporary state of modern surveillance systems. Cao et al. [27] proposed an algorithm to detect water areas in both satellite images and camera views, and use water and coastlines as the features for geo-registration. Srivastava et al. [28] proposed a color correction technique for achieving better color consistency for each detection across multiple cameras. Andriyenko et al. [29] proposed a linear programming method with discrete-continuous optimization for data association and trajectory estimation. The discrete optimization handles data association for each target, and the continuous optimization handles the trajectory optimization.

Some researches proposed different re-identification approaches to multi-camera tracking problems. Mazzon and Cavallaro [30] proposed a novel Special Force Model (SFM) to model each target based on their motion to facilitate person reidentification in a non-overlapping camera environment. Shen et al. [31] proposed a two-step framework for vehicle re-identification. It used visual-spatial-temporal querying at starting and ending states and then applied a Siamese-CNN with Path-LSTM for spatial-temporal regularization. Zapletal and Herout [32] used 3D rectangular alignment with a HOG feature model to solve the re-identification problem of multi-view vehicles. Zhang et al. [33] proposed a hierarchical clustering technique for person reidentification on DukeMTMC dataset. Ristani and Tomasi [34] proposed an improved version of the tracking technique based on triplet loss and identity mining for person re-identification. He et al. [35] proposed a two-stage tracking approach. The first stage uses ResNet50 with clustering and trajectory consistency loss for vehicle re-identification, and the second stage uses a trajectory-based weighted ranking method to improve the performance. Chen et al. [36] proposed a fusion of CNN and RNN to jointly learn spatial and temporal information of each detection for video-based person reidentification.

This paper proposes a surveillance system that can track multiple persons across multiple cameras and link all video shots to a complete tracking sequence for each person. The proposed approach adopts spatiotemporal and appearance cues to track individual objects across cameras, but the tracking path may be lost due to the changing environment. Not many works addressed the issue of lost tracking. The proposed approach handles the missing problem with a path revision approach. We use a dynamic programming algorithm to retrace two or three levels to search for a correspondence between the disconnected paths. Furthermore, we generate the connected path utilizing the concept of the regular path and use Hidden Markov Models to verify whether a connected path is a true tracking path.

Figure 1 shows the structure of the proposed system, which includes two processing phases: path construction and path revision.

In the first phase, the moving objects within each camera view are first tracked to form a set in which the related object information is recorded. Then the spatiotemporal and appearance clues are used to compute the correspondence between observations and build the tracking paths. In the path revision phase, we trace the handover lists and use dynamic programming to search the missing object, and further apply the merge-spilt method to detect the occluded objects. Finally, we explore the regular paths and adopt Hidden Markov Models to confirm if they are plausible tracking paths.

2 Tracking Within Single Camera View

To track the moving object, we apply background subtraction to extract the foreground object from the video sequence. After identifying the object trajectories,

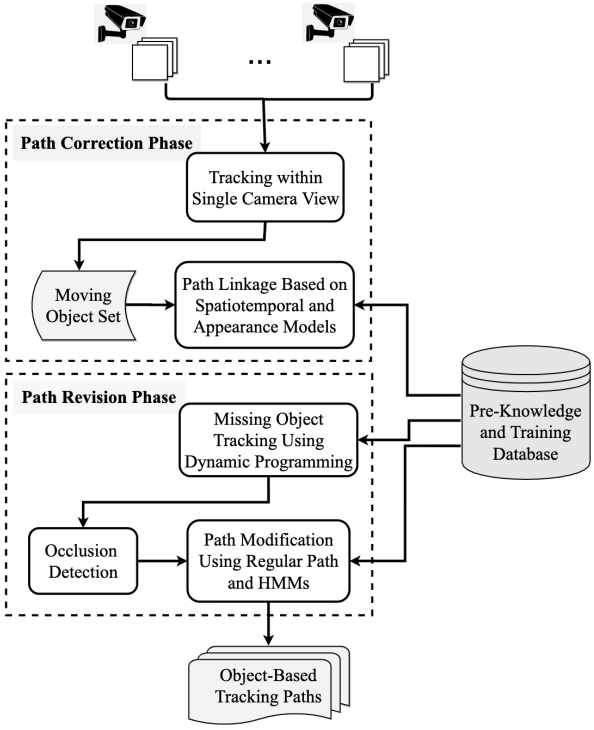


Figure 1. System overview

the I/O entries of each view are determined using the method of GMMs.

2.1 Foreground Object Extraction

We train a background model to extract the foreground objects. After background subtraction, some noise in the foreground image (I_a) may occur. We apply morphological filtering and labeling processes to remove noise and obtain a clean-cut object silhouette.

$$B_{t+1} = \alpha \cdot B_t + (1-\alpha) \cdot (\sim M_t) \cdot D_t \quad \textbf{(1)}$$

where the weighting coefficient α is a real number from 0 to 1, D_t is the input image at time t, and M_t is the binary people mask, which is a binary representation of the people extracted from the video sequence at time t. The background model is defined by the 'brightness' within the image sequence and the estimated stationary background. Because the background may change, we update the background model periodically:

$$B_t = \alpha \cdot B_t + (1-\alpha) \cdot D_t, t = \tau \times n \quad \textbf{(2)}$$

where τ is the update period for the background model and $n \in R$.

After obtaining the foreground object silhouette, we apply morphological filtering using the cascade closing and opening operations to remove noise from the silhouette. We then apply the labeling process that assigns the same label to the connected component and remove the small components, which are treated as noise. Figure 2 shows the foreground images extracted from each camera after removing noise.

(a) Original images

(b) Results of foreground extraction after noise removal

Figure 2. Foreground extraction

2.2 Entry/ Exit Identification

A single-camera view usually contains several entry/exit zones, including door, flow, or hallway. Our goal is to find the areas that people constantly pass to determine walking paths and enter/exit times. We adopt the method from [37]. The distribution of the entry/exit points within each camera view is modeled as a Gaussian Mixture Model (GMM), where the parameters of GMM are estimated using an Expectation-Maximization (EM) algorithm, and the number of clusters is determined by the Bayesian Information Criterion.

A GMM can approximate the probability density distribution using adjustable parameters, which are mixture weights (w_i), mean vector (μ_i), and covariance matrix (Σ_i). The set of entry/exit points can be regarded as a GMM described as follows:

$$p(X_N \mid \lambda) = \sum_{i=1}^{M} w_i g_i(X_N) \quad \textbf{(3)}$$

where w_i is the mixing parameter satisfying $\sum_{i=1}^{M} w_i = 1$.

Our system detects the events when people enter or exit in a single camera view and records the center of the foreground object to serve as the training data. In the training phase, we collect a set of center points from each camera view. We then determine the number of clusters for each camera using a Bayesian Information Criterion (BIC). We apply the *k-means clustering* to partition the total samples into k clusters, i.e., $S = \{S_1, S_2, \ldots, S_k\}$ to minimize the within-cluster sum of squares (WCSS):

$$WCSS = \arg\min_s \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - \mu_i \|^2 \quad \textbf{(4)}$$

where x is a set of samples and μ_i is the mean of S_i.

We can then estimate the parameters (λ) of each component. The likelihood function can be described as follows:

$$P(X \mid \lambda) = \prod_{i=1}^{n} P(x_i \mid \lambda) \quad (5)$$

where $X = \{x_1, x_2, \ldots, x_n\}$ and i = 1 ~ n are independent events. The EM algorithm is based on the Maximum Likelihood Estimate (MLE), which is determined by the marginal likelihood of the observed data $P(X \mid \lambda)$, after applying the EM algorithm. Figure 3 illustrates the results of GMMs for different camera views.

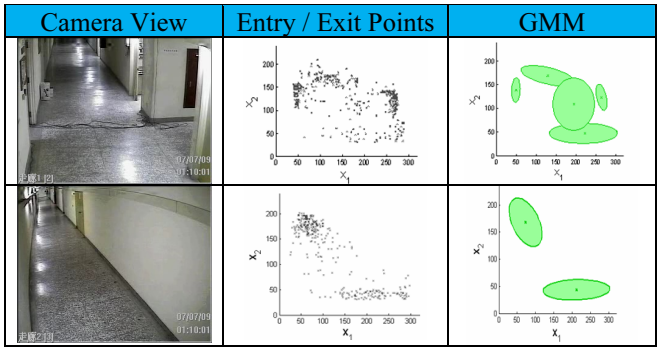


Figure 3. The GMMs of different camera view

3 Tracking across Multiple Cameras

In the tracking process, an observation model of the object is given for correspondence analysis. The color histogram has been proven to be a useful model for the scaling and orientation of the object.

3.1 Color Feature Generation

We first introduce the adopted color feature and present the color calibration between cameras.

3.1.1 Color Histogram

In our system, we adopt an HS (Hue-Saturation) color histogram model. The hue and saturation ranges are uniformly quantized into 10 levels. In this manner, a color region in an image can be described by 10×10 bins on the Hue-Saturation plane. The tracking process detects the moving object that is enclosed by a rectangle represented as $s = \{x, yh, h, r\}$, where (x, y) represents the bottom center location of the rectangle, and (h, r) represents the height and aspect ratios. In each rectangle, the foreground object is described by a color histogram with 101 bins. A similarity measurement between two distributions can be described by the Bhattacharyya coefficient ρ (Comaniciu et al. [38]). The distance between two distributions is described as:

$$d = \sqrt{1 - \rho[p,q]} \quad (6)$$

where the Bhattacharyya coefficient ρ ranges from 0 to 1.

3.1.2 Color Calibration

The images captured by different cameras have different color characteristics. Therefore, color calibration [39] is required in finding the correspondence between camera views. We adopt the method proposed by Ruderman et al. [40], which can minimize the correlation between the three natural scene channels by using a $l\alpha\beta$ color space. Before an image is transformed from RGB space to $l\alpha\beta$ space, the image is first transformed to LMS space (***L***=*log L, **M**=log M, and **S**=log S*). Finally, the maximal de-correlation between the three axes is computed using a principal component analysis (PCA) and moved to the nearby integer coefficients. The transformation is described below:

$$\begin{bmatrix} l \\ a \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{L} \\ \boldsymbol{M} \\ \boldsymbol{S} \end{bmatrix} \quad (7)$$

where the l axis represents an achromatic channel, and the a and β channels are chromatic yellow-blue and red-green opponent channels.

After color transformation, the means (μ) and standard deviations (σ) for each axis separately in $l\alpha\beta$ space are computed. In this manner, we calculate these color measures for both the source and target images. The formulation is described as follows:

$$\begin{cases} l' = l - \mu_s(l) \\ \alpha' = \alpha - \mu_\alpha(\alpha) \\ \beta' = \beta - \mu_\beta(\beta) \end{cases} \rightarrow \begin{cases} l_f = \frac{\sigma_t(l)}{\sigma_s(l)} l' + \mu_t(l) \\ \alpha_f = \frac{\sigma_t(\alpha)}{\sigma_s(\alpha)} \alpha' + \mu_t(\alpha) \\ \beta_f = \frac{\sigma_t(\beta)}{\sigma_s(\beta)} \beta' + \mu_t(\beta) \end{cases} \quad (8)$$

where l_f, α_f, and β_f are the result data points. We then transfer the result back to RGB space for display.

3.2 Training Phase for Spatiotemporal Relationships

The transition time and the transition probability are two crucial factors in describing the spatiotemporal relationship in a camera network. The transition time means the time duration of an object moving from an exit zone to another entry zone, and the transition

probability means the probability between two observations at two different zones. To improve the spatiotemporal estimation, we employ the prior knowledge of camera network topology (Chen et al. [37]):

(1) Information of all adjacent camera pairs.

(2) The blind regions, which are not monitored by any camera, may be closed or open. Usually, the blind region is assumed to be closed; that means there are neither entries nor exits.

The prior knowledge of camera network topology is important for decreasing the computation complexity and removing invalid links.

In our system, the spatiotemporal relationships are based on the link between the entry and exit zones, so we need to locate the entry/exit zones for every single camera. We also learn the transition probability for each possible link using the prior knowledge of camera network topology. Assume that there is a possible link between two zones *a* and *b*, where zone *a* is within the view of camera 1 and zone *b* within that of camera 2. $P_{ab}(T)$ describes the transition probability that people move from zone a to zone *b* at time *T*. Object *i* exits from zone *a* at time T_i, and object *j* enters zone *b* at time T_j. The transition probability is defined as

$$P_{ab}(T) = \Sigma_i \Sigma_j \begin{cases} S(i,j), if\ (-T_i) = T \\ 0 \quad , \quad otherwise \end{cases} \tag{9}$$

$$NP_{ab}(T) = \frac{P_{ab}(T)}{\Sigma P_{ab}(T)} \tag{10}$$

where $NP_{ab}(T)$ is the normalized transition probability of $P_{ab}(T)$, and $S(i, j)$ is the color similarity between objects *i* and *j*, defined as

$$S(i,j) = \sum_{h=1}^{m} \min(B(h,i), B(j,j)) \tag{11}$$

where *m* is the bin number in the HS plane of the appearance color histogram.

In addition to the spatiotemporal relationships between cameras, we also consider the spatiotemporal information within the cameras. When an object moves from an entry zone to an exit zone in a single camera view, the time duration is called the travel time of the object (Figure 4). The travel time and internal transition probability of a single camera are used in the path revision phase to recover the missing linkage.

3.3 Correspondence Analysis

Tracking between multiple cameras with a disjoint view aims to establish a set of correspondences between observations of objects across multiple cameras. We construct the correspondence between the current observation and observations in the handover list. The handover list is defined as a set of observations before the current tracked object within

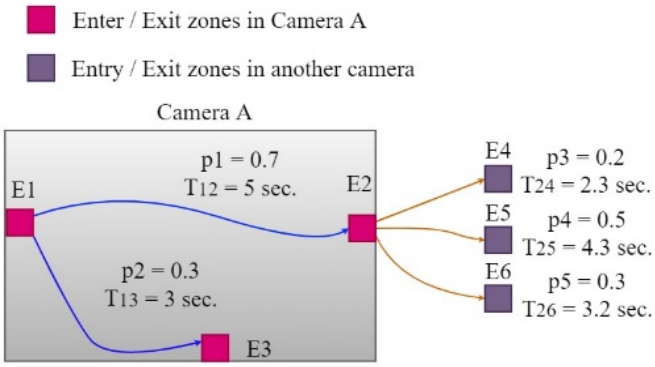


Figure 4. The blue line describes the mean travel time and internal transition probability from zone E1 to another zone in Camera A. The orange line describes the mean transition time and transition probability between a pair of Entry/Exit zones

the maximum allowable period in the valid links. Here we modify the method [37] to find the correspondence.

Assume that a person A enters the view of one camera, whose observation is denoted as O_A. Observation O_A contains two cues: spatiotemporal cue $O_A(st)$ and appearance cue $O_A(app)$. $O_A(st)$ contains the camera ID, entry/exit zone where A entered, the position $O_A(x, y)$, and entry time $O_A(time)$. The best corresponding person with observation O_h is selected from the handover list by using the spatiotemporal relationships between cameras. If the highest probability exceeds the threshold, we label the new arrival observation O_A and observation O_h in the handover list as the same person. Otherwise, object A is a newcomer to our system. The probability of observation O_A belonging to observation O_h is denoted as $p(O_A, O_h)$. The most likely correspondence is found below:

$$\varphi = \arg\max_{h \in H} p(O_A, O_h) \tag{12}$$

where *H* is the handover list.

Assume that the spatiotemporal and appearance cues are independent. The method takes the log-likelihoods and merges them with a weight *w*. According to Bayes Theorem, the most likely correspondence can be described as follows:

$$\begin{aligned} \varphi &= \arg\max_{h \in H} \ln p(O_A, O_h) \\ &= \arg\max_{h \in H} [\ln p(O_A(st), O_h(st))^w \\ &\quad \times \ln p(O_A(app), O_h(app))^{1-w}] \\ &= \arg\max_{h \in H} [w \ln p(O_A(st), O_h(st)) \\ &\quad + (1-w) \ln p(O_A(app), O_h(app))] \end{aligned} \tag{13}$$

where the $p(O_A(app), O_h(app))$ is the probability of color histogram similarity, and $p(O_A(st), O_h(st))$ is estimated as follows:

$$p(O_A(st), O_h(st) = \Sigma_{\forall Z_A}\Sigma_{\forall Z_A}[NP_{Z_A Z_h}(T)p(O_A(x,y)|Z_A)p(O_h(x,y)|Z_h)] \quad (14)$$

where $NP_{Z_A Z_h}(T)$ is the transition probability distribution with travel time between O_h and O_A, and $p(O_*(x,y)|Z_*)$ is the probability of observation O_* entering or exiting zone Z_*.

After constructing the correspondence between observations, we can build the object-based human activity video linkage stored in the video database as shown in Figure 5.

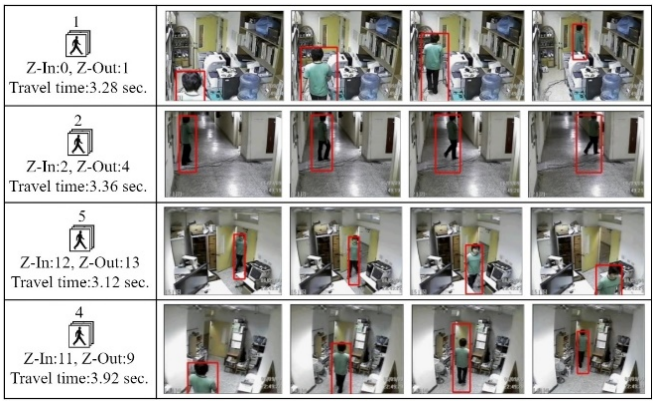


Figure 5. One person activity linkage recording in a video database

4 Tracking Path Revision

Under normal conditions, the spatiotemporal relationships and correspondence can be used for object-based people tracking across cameras, but some unexpected events may result in missed tracking of an object. To cope with the problems, we use the dynamic programming process to reconnect the tracking paths based on the spatiotemporal relationships, and then apply Hidden Markov Models to fix the missing linkage that cannot be found in the dynamic programming process.

4.1 Tracking of Missed Objects

Dynamic programming is appropriate for solving the overlapping sub-problems and optimal substructure. To find the most probable tracking path, we employ the stepwise back-tracking process to find a plausible path of the highest probability. We record and re-use the most probable path and maximal probability at each step.

Assume that object O_1 enter Zone A in the view of camera C_i. The condition probability $P_{O_1}(zone\ B|zone\ A)$ of the same object is computed based on another zone B in the view of camera C_j. We then calculate the most probable path from zone A to zone B using the following equation:

$$P_{O_1}(zone\ B|zone\ A) = P(zone\ B|zone\ S_1)...P(zone\ S_k|zone\ A) \quad (15)$$

where the sequence $\{S_1, S_2 ....., S_k\}$ is the path of the highest probability from zone A to zone B. Figure 6 shows the researched transition path.

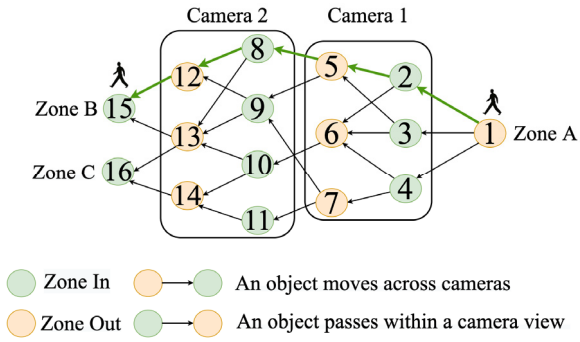


Figure 6. The green arrows depict the highest probability path from zone A to zone B

In the video linking process, the missed linkage problem is mostly due to two issues: (a) color features of the same people change between different cameras, and (b) some people may stay in a blind region for a while. The former results in low correspondence between two observations for the same target, whereas the latter decreases the transition probability. In both cases, the new observation is treated as a new person entering the system. Hence, the linkage video of the same person is broken into two independent video sequences.

To solve this problem, we employ the dynamic programming algorithm to locate the most probable path from zone B to zone A by backward tracing. We then select the person who corresponds best with observation O_B in zone B. There may be many observations in zone B, so we employ the spatiotemporal relationships to find the best one. We usually trace back approximately 2- or 3- levels to find the zone from which someone in the handover list exited. Figure 7 depicts the backward tracking in our system.

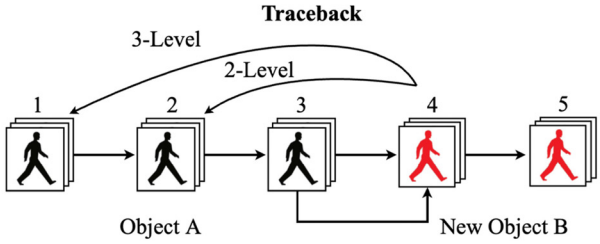


Figure 7. Missed object is tracked using a dynamic programming algorithm. Object A and Object B are recognized as the same people if the correspondence probability of two observations exceeds a threshold between camera 2 and camera 4 (2-Level) or camera 1 and camera 4 (3-Level)

The probability that observation O_A relates to O_B in the handover list is denoted as $p(O_A, O_B)$. The time distribution from zone B to zone A can be estimated stepwise. The most likely correspondence can be obtained by calculating both the similarity of the color histograms and the transition probability from zone B to zone A. The algorithm can trace back and solve the missed tracking conditions. However, some other problems still cannot be solved: (1) color features between the observations of two or three-level and new observation remain dissimilar, *e.g.*, occlusion; (2) no observation exits in the handover list, and (3) there is a low transition probability when we trace back at each level.

4.2 Identification of Occluded Objects

To solve the occlusion problem, we employ the Merge-Split (MS) approach to detect the merged and split events and label the occlusion human blob, defined as a region covering a group of people. In the video clips, human blobs may enter, exit, merge, or split in the scene. Our system detects that two or more blobs may have merged into a group blob, thus decreases the total number of human blobs in the frame. Two or more human blobs in the previous frame overlap with a human blob in the current frame. Conversely, the group blob may split into two blobs. We detect this event when both the total number of blobs in the frame increases and several blobs in the current frame overlap with a group blob in the previous frame.

To assign labels after splitting, we record the color feature of each human blob when it enters the view. We assume that each person appears in one human blob after the splitting process. We label the human blobs by comparing the color feature between the current and previous blobs before occlusion. Figure 8 shows the result of tracking under occlusion. In frame 534, the same camera sees both human blobs *H0* and *H1*. In frame 543, two human blobs merge into a group *G1*. In frame 564, group *G1* splits into *H0* and *H1*.

(a) Frame 534 (b) Frame 543 (c) Frame 564

Figure 8. People movement within a single camera

If a group enters the camera view, the occlusion blob contains a large number of pixels, and the area is larger than an ordinary blob of a single person. The occlusion blob is labeled as a group blob (Figure 9). The information will be used in path modification.

(a) A group enters camera view (b) Detected objects in a group blob

Figure 9. Group detection

4.3 PATH Modification

Usually, people activities follow certain paths in the environment. For example, the moving path of a worker is generally the same in his office on a normal day. We thus define the most probable path as the "regular path" and incorporate Hidden Markov Models (Rabiner [41]) to modify the linked paths.

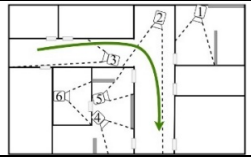
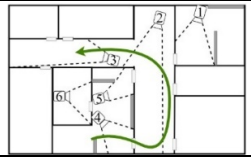
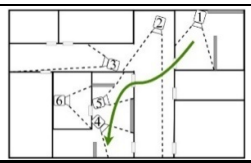
4.3.1 Regular Path Establishment in Training Process

We locate the entry/exit zones in each camera view and model them with GMMs. For each person entering the view of one camera and exiting after a period, we can collect information about his entry and exit. Observing a long video sequence can identify different paths, and if the occurrence times of a path is larger than pre-defined count, then the path is considered as a "regular path."

4.3.2 Regular Path Recognition

A path is a linkage of path segments of which a path segment is defined as the tracking path in a single view. It contains the information of camera ID and entry/exit zone numbers. Table 1 shows regular paths of different lengths.

Table 1. Regular paths across cameras

Path across 2 cameras

Path across 3 cameras

Path across 4 cameras


We apply HMMs to model a regular path for every possible sequence of path segments. An observation codebook is used to convert each path segment to an observation symbol for the HMMs. In our system, we use 2-state, 3-state, and 4-state left-right HMMs. A path may be broken into several broken segments due to environmental influence. We try to reconnect these broken path segments to a connective path and use HMMs to recognize whether it is a regular path. If so, the color feature similarity and transition probability are further to be checked. When one of them is larger than the pre-defined threshold, the connection path is therefore confirmed.

4.3.3 Path Reconstruction

Figure 10 shows an example. If a person goes through cameras 1, 2, 3, and 4, the false tracked results due to the feature dissimilarity or low transition probability. Though the two objects are the same person, the incorrect tracking from camera 2 to camera 3 results in two broken segments: object A goes through camera 1 to camera 2, and object B passes from camera 3 to camera 4.

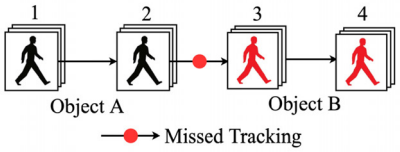


Figure 10. Tracking of the same person is missed between camera 2 and camera 3. The video is divided into two independent broken segments

To correct the result, we first recognize the regular path. Here, both path 1 and path 2 are recognized as irregular paths. And then, we connect these two irregular paths and check if the connection path is a regular path, as in Figure 11. Once the connection path is recognized as a regular path, we further check whether the color feature similarity or the transition probability is larger than pre-defined thresholds. If all requirement is met, the connection path is then confirmed. Finally, object A and object B are recognized as the same person.

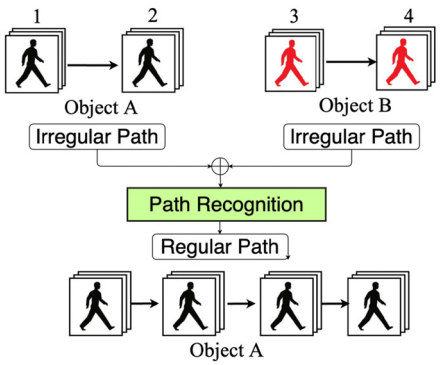


Figure 11. Regular path reconstruction

Another example is that a group splits in the blind region (Figure 12). When these videos are processed, the paths containing segments 2, 3, and 4 are recognized as irregular paths, as is the path containing segments 5 and 6. Because group A contains more than one person, we can connect group A to both objects B and C. After computing the path probability, if paths *1-2-3-4* and *1-5-6* are recognized as regular paths, objects B and C are likely in group A. If the color similarity or the transition probability is larger than the pre-designed threshold, we can connect group A and objects B and C in the same path, as in Figure 13.

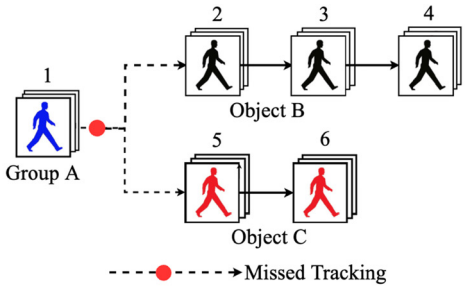


Figure 12. Group A splits into two objects in the blind region, and objects B and C are recognized as new objects

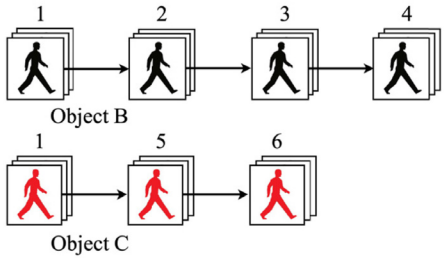
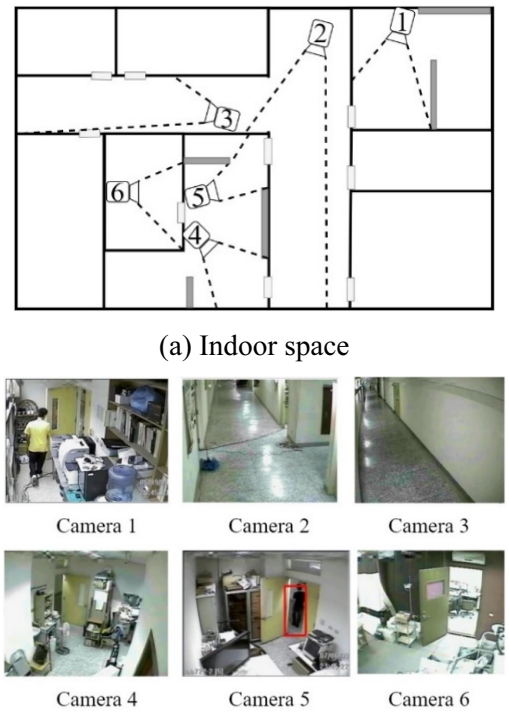


Figure 13. Two objects are represented after path revision using HMM

5 Experimental Results

This section presents the experimental results to evaluate the performance of the proposed system. The image resolution is 320*240, and the frame rate is 25 frames/s. Figure 14(a) shows the layout of the indoor space and the location of cameras, and Figure 14(b) shows the corresponding camera views.

Our experimental environment is an indoor space installed with six cameras. To represent the dynamic path in the layout, we employ the Direct Linear Transformation (DLT) algorithm. In the proposed work, we define the set of object video sequences as {*i*: the number *i* indicates that the person is observed in the view of the *i-th* camera}; for example, {1, 5, 7} means that the person appears in the views of cameras 1, 5, and 7.



(a) Indoor space

(b) Camera views

Figure 14. Environment setup

We took a video of 5 hours for each camera, and 3 hours for training, 2 hours for testing. The training database is selected from these videos, which includes a single person and multi-person activities under different situations; however, the lighting variation is limited. We use 11 hidden models and 124 regular paths to train the model. A hidden state is composed of camera ID and zone IDs. We register the tracking path in the training database and determine the number of states. In the testing database, there are 42 single human activities that contain 89 corresponded pairs. The experimental results are shown in Table 2.

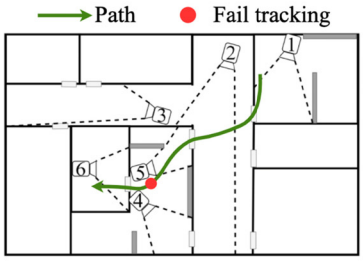
Table 2. Tracking results

Human Activity		42	
Corresponded Pair		89	
Linkage Results	Path Linkage	Missed Object Tracking	Reconstructed Path
	74	5	10

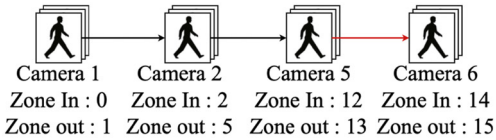
In the results, 74 corresponded pairs are found in the path linkage based on the spatiotemporal and appearance models. Path linkage can work well in most cases, but the testing database contains several unusual events, for example, a person staying in the blind region for a long time. There are still 15 pairs not found. In the path revision phase, 5 pairs are found in missed object tracing, 10 pairs in the path modification process. The above cases could be solved by our proposed method. We show some results in the following sections. Besides these cases, we also tested other conditions, for example, multiple people dressed in similar clothes, but these cases fail.

5.1 Single-Person Tracking

In this section, we show the single-person tracking results. In Figure 15, a person stays in the blind region for a while when passing between cameras 5 and 6. Due to the low transition probability, the initial tracking path is not right and divided into two separate sequences {1, 2, 5} and {6}. Therefore, we employ the dynamic programming algorithm to trace back and attempt to reconnect the two object sequences as the same one. When the dynamic programming tracking does not link the two sequences, the path modification is further applied to determine whether the two objects are the same person. If the probability is larger than a threshold, the two objects in the segments are recognized as the same person, and the sequences are joined.



(a) Walking paths



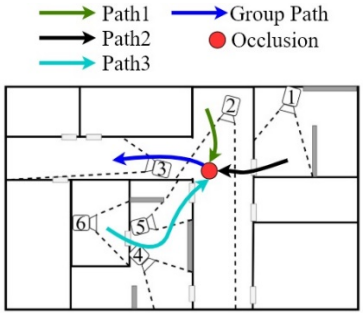
(b) Object-based tracking sequence across cameras

(c) The corresponding tracking videos

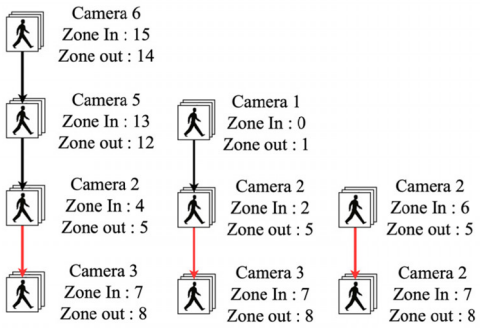
Figure 15. Single-person tracking

5.2 Multiple-Person Tracking

In this section, we show the multiple-person tracking results, which are more complicated than a single-person tracking, as several objects may merge into an object group. In the first case, three persons coming from different places enter the view of camera 2, and they meet and walk together. The three objects merge into an object group, in the view of camera 2. We employ the MS approach to identify the occlusion event in the camera view. Each object can be continuously tracked in the object group after they merge together. However, the initial tracking fails when the object group splits and leaves the view of camera 2, because the color feature of the group is dissimilar to any one of the individual objects. The path revision determines whether these people form an object group or not. If the probability is larger than a threshold, we continuously track the people as an object group in the view of camera 3. Figure 16(a) shows the walking paths for all three persons on the layout, and Figure 16(b) presents detailed path information for each person, including the camera and zone numbers. Figure 16(c) shows the corresponding video in frames.



(a) Walking paths



(b) Object-based tracking sequences across cameras

(c)_1 Tracking video of Person 1

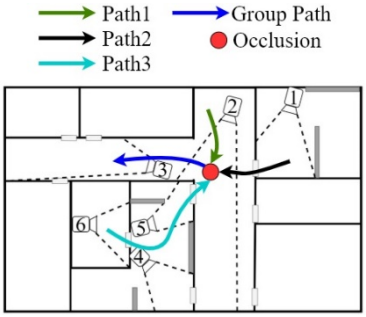
(c)_2 Tracking video of Person 2

(c)_3 Tracking video of Person 3

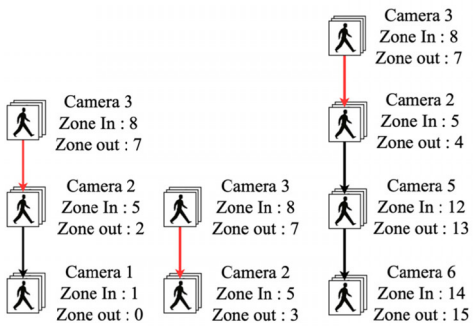
Figure 16. Multiple-person tracking

In the second case, three people initially walk together in the view of camera 3, enter the blind region between cameras 2 and 3, and leave individually. However, the initial tracking fails, as the color feature of the object group is dissimilar to those three objects. It cannot recognize whether those objects are involved in the object group because of the occlusion. Therefore, we employ path modification to determine whether the people are involved

in the group. If the probability is larger than a threshold, we connect the tracking result in the view of camera 3 to those three objects individually. Each object group can be tracked individually after the object group splits into three individual objects. Figure 17(a) shows the walking paths for all three persons in the layout, and Figure 17(b) presents detailed path information for each person, including the camera and zone numbers. Figure 17(c) shows the corresponding video in frames.



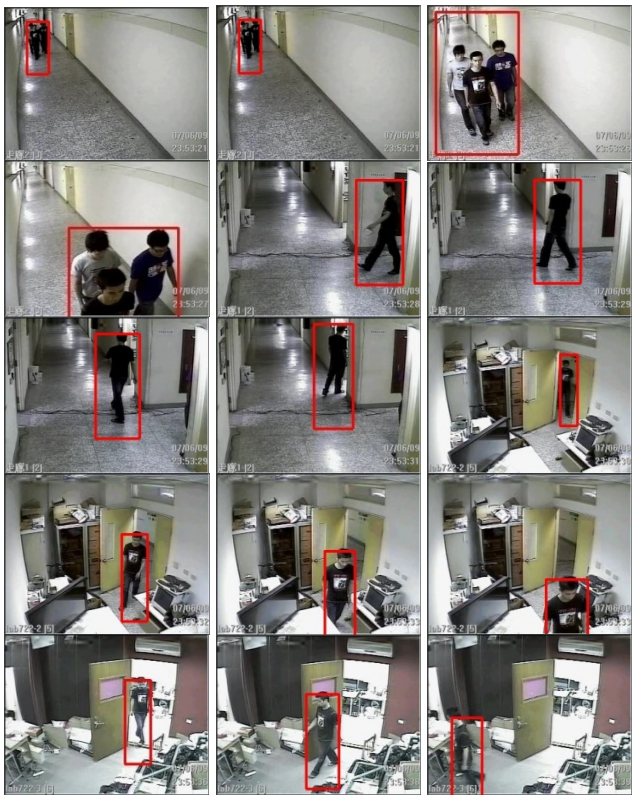
(a) Walking paths



(b) Object-based tracking sequences across cameras

(c)_1 Tracking video of Person 1

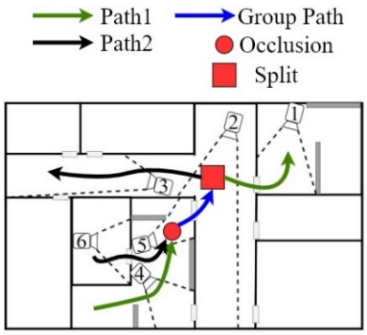
(c)_2 Tracking video of Person 2



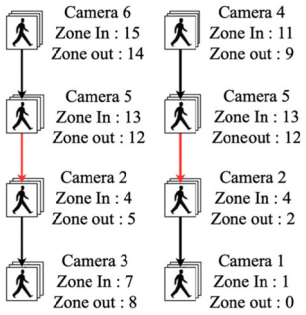
(c)_3 Tracking video of Person 3

Figure 17. Multiple-person tracking

In the third case, two persons from different camera views encounter each other in the view of camera 5 and walk together for a while. Finally, they separate and leave from different camera views. Because the two objects merge into an object group after the encounter, we employ the MS approach to solve the occlusion event in the view of camera 5. Each object can be continuously tracked as an object group after they merge together. They then leave individually. Each object can be tracked individually after the object group splits into two objects. It is difficult to find the correspondence because of insufficient individual information in the views of cameras 2 and 5. Therefore, we employ dynamic programming and obtain more information in the views of cameras 4 and 6. The correspondence is thus found, and the new path is established. Figure 18(a) shows the walking paths for all three persons in the environment, and Figure 18(b) presents detailed path information for each person, including the camera and zone numbers. Figure 18(c) shows the corresponding video in frames.



(a) Walking paths



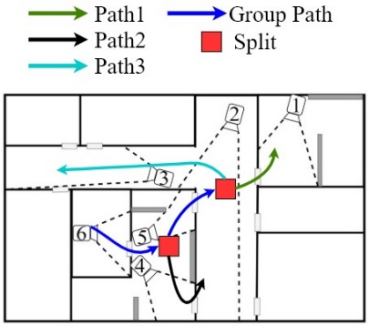
(b) Object-based tracking sequences across cameras

(c)_1 Tracking video of Person 1

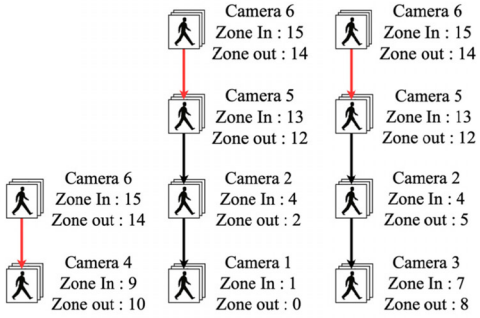
(c)_2 Tracking video of Person 2

Figure 18. Multiple-person tracking

In the fourth case, three persons merge as a three-object group and enter into the view of camera 6; the group then splits into an object and two-object group. The two-object group further splits in the view of camera 2. Figure 19(a) shows the walking paths for all three persons in the environment. All persons can be continuously tracked individually. However, the initial tracking fails because the color feature of the object group is dissimilar to those objects. We employ path modification to determine whether a person is involved in the related object group. When the correspondence is confirmed, we link the tracking video clips for those individual objects. Figure 19(b) presents detailed path information for each person, including the camera and zone numbers. Figure 19(c) shows the corresponding video in frames.



(a) Walking paths remove



(b) Object-based tracking sequences across cameras

(c)_1 Tracking video of Person1

(c)_2 Tracking video of Person 2

(c)_3 Tracking video of Person 3

Figure 19. Multiple-person tracking

6 Conclusion

This paper proposes an automatic object-based tracking system for video surveillance across cameras with non-overlapping views. We exploit spatiotemporal and appearance cues to track human objects across cameras. To learn the relationships among cameras, we use the batch-learning procedure and update all probability matrices for long-term monitoring. Under some situations, human tracking may be lost due to unusual behaviors. In this paper, we adopt the dynamic programming algorithm to solve the problem of missed objects and further apply Hidden Markov Models to re-build the path across cameras. The amendment process could solve the problems for certain conditions.

Acknowledgements

This work was supported by Ministry of Science and Technology, Taiwan, ROC, under Grant 109-2218-E-259-001.

References

[1] J. Black, T. Ellis, D. Makris, Wide Area Surveillance with a Multi-Camera Network, *IDSS-04 Intelligent Distributed Surveillance Systems*, London, UK, 2004, pp. 21-25.

[2] T. Chang, S. Gong, Tracking Multiple People with a Multi-Camera System, *IEEE Workshop on Multi-Object Tracking*, Vancouver, BC, Canada, 2001, pp. 19-26.

[3] S. L. Dockstader, A. M. Tekalp, Multiple Camera Fusion for Multi-object Tracking, *IEEE Workshop on Multi-Object Tracking*, Vancouver, BC, Canada, 2001, pp. 95-102.

[4] L. Lee, R. Romano, G. Stein, Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 758-767, August, 2000.

[5] S. Khan, O. Javed, Z. Rasheed, M. Shah, Human Tracking in Multiple Cameras, *Eighth IEEE International Conference on Computer Vision (ICCV)*, Vancouver, BC, Canada, Canada, 2001, pp. 331-336.

[6] A. Yilmaz, O. Javed, M. Shah, Object Tracking: A Survey, *ACM Computing Surveys*, Vol. 38, No. 4, pp. 1-45, December, 2006.

[7] Q. Cai, J. K. Aggarwal, Tracking Human Motion in Structured Environments Using a Distributed-Camera System, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 11, pp. 1241-1247, November, 1999.

[8] S. Khan, M. Shah, Consistent Labeling of Tracked Objects in Multiple Cameras with Overlapping Fields of View, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 10, pp. 1355-1360, October, 2003.

[9] C. Stauffer, K. Tieu, Automated Multi-camera Planar Tracking Correspondence Modeling, *IEEE International Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, 2003, pp. 1-8.

[10] V. Kettnaker, R. Zabih, Counting People from Multiple Cameras, *IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, 1999, pp. 267-271.

[11] V. Kettnaker, R. Zabih, Bayesian Multi-camera Surveillance, *IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, USA, 1999, pp. 253-259.

[12] F. Porikli, A. Divakaran, Multi-Camera Calibration, Object Tracking and Query Generation, *IEEE International Conference on Multimedia and Expo*, Baltimore, MD, USA, 2003, pp. I-653-I-656.

[13] O. Javed, Z. Rasheed, K. Shafique, M. Shah, Tracking across Multiple Cameras with Disjoint Views, *IEEE Conference on Computer Vision*, Nice, France, 2003, pp. 952-957.

[14] S. Oh, S. Russell, S. Sastry, Markov Chain Monte Carlo Data Association for Multi-Target Tracking, *IEEE Transactions on Automatic Control*, Vol. 54, No. 3, pp. 481-497, March, 2009.

[15] K. Yoon, Y. Song, M. Jeon, Multiple Hypothesis Tracking Algorithm for Multi-Target Multi-Camera Tracking with Disjoint Views, *IET Image Processing*, Vol. 12, No. 7, pp. 1175-1184, July, 2018.

[16] C. H. Kuo, C. Huang, R. Nevatia, Inter-Camera Association of Multi-Target Tracks by On-Line Learned Appearance Affinity Models, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), *European Conference on Computer Vision*, Springer, Berlin, Heidelberg, 2010, pp. 383-396.

[17] J. Simonjan, L. Esterle, B. Rinner, G. Nebehay, G. F. Domínguez, Demo: Demonstrating Autonomous Handover in Heterogeneous Multi-camera Systems, *International Conference on Distributed Smart Cameras (ICDSC '14)*, Venezia Mestre, Italy, 2014, pp.1-3.

[18] A. Dick, M. Brooks, A Stochastic Approach to Tracking Objects across Multiple Cameras, *Australian Conference on Artificial Intelligence*, Cairns, Australia, 2004, pp. 160-170.

[19] T. J. Ellis, D. Makris, J. Black, Learning a Multi-Camera Topology, *Joint IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, Nice, France, 2003, pp. 165-171.

[20] C. Stauffer, Learning to Track Objects Through Unobserved Regions, *IEEE Computer Society Workshop on Motion and Video Computing*, Breckenridge, CO, USA, 2005, pp. 96-102.

[21] K. Tieu, G. Dalley, W. Grimson, Inference of Non-overlapping Camera Network Topology by Measuring Statistical Dependence, *IEEE International Conference on Computer Vision*, Beijing, China, 2005, pp. 1842-1849.

[22] A. Gilbert, R. Bowden, Tracking Objects Across Cameras by Incrementally Learning Inter-camera Colour Calibration and Patterns of Activity, *European Conference on Computer Vision*, Graz, Austria, 2006, pp. 125-136.

[23] Y. Nam, J. Ryu, Y.-J. Choi, W.-D. Cho, Learning Spatio-Temporal Topology of a Multi-Camera Network by Tracking Multiple People, *World Academy of Science, Engineering and Technology*, Vol. 1, No. 6, pp. 1549-1554, 2007.

[24] J. Tang, H. Wu, Z. Zhou, L. Wang, Novel Sensor Selection Technique for Moving Object Tracking Based on Pre-Defined Division Structure in Wireless Sensor Network, *Journal of*

Internet Technology, Vol. 17, No. 7, pp. 1471-1482, December, 2016.

[25] I. N. Junejo, X. Cao, H. Foroosh, Autoconfiguration of a Dynamic Nonoverlapping Camera Network, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 37, No. 4, pp. 803-816, August, 2007.

[26] T. Raty, Survey on Contemporary Remote Surveillance Systems for Public Safety, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 40, No. 5, pp. 493-515, September, 2010.

[27] X. Cao, L. Wu, Z. Rasheed, H. Liu, T. E. Choe, F. Guo, N. Haering, Automatic Geo-Registration for Port Surveillance, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 24, No. 4, pp. 531-555, June, 2010.

[28] S. Srivastava, K. Ka, E. J. Delp, Color Correction for Object Tracking Across Multiple Cameras, *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 1821-1824.

[29] A. Andriyenko, K. Schindler, S. Roth, Discrete-Continuous Optimization for Multi-Target Tracking, *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012, pp. 1926-1933.

[30] R. Mazzon, A. Cavallaro, Multi-Camera Tracking Using a Multi-Goal Social Force Model, *Neurocomputing*, Vol. 100, pp. 41-50, January, 2013.

[31] Y. Shen, T. Xiao, H. Li, S. Yi, X. Wang, Learning Deep Neural Networks for Vehicle Re-Id with Visual-Spatio-Temporal Path Proposals, *IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 1900-1909.

[32] D. Zapletal, A. Herout, Vehicle Re-identification for Automatic Video Traffic Surveillance, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Las Vegas, NV, USA, 2016, pp. 25-31.

[33] Z. Zhang, J. Wu, X. Zhang, C. Zhang, *Multi-Target, Multi-Camera Tracking by Hierarchical Clustering: Recent Progress on DukeMTMC Project*, http://arxiv.org/abs/1712.09531.

[34] E. Ristani, C. Tomasi, Features for Multi-Target Multi-Camera Tracking and Re-Identification, *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 6036-6046.

[35] Z. He, Y. Lei, S. Bai, W. Wu, Multi-Camera Vehicle Tracking with Powerful Visual Features and Spatial-Temporal Cue, *CVPR Workshops*, Long Beach, California, USA, 2019, pp. 203-212.

[36] L. Chen, H. Yang, J. Zhu, Q. Zhou, S. Wu, Z. Gao, Deep Spatial-Temporal Fusion Network for Video-Based Person Re-Identification, *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Hawaii, USA, 2017, pp. 63-70.

[37] K. W. Chen, C. C. Lai, Y. P. Hung, C. S. Chen, An Adaptive Learning Method for Target Tracking Across Multiple Cameras, *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008, pp. 1-8.

[38] D. Comaniciu, V. Ramesh, P. Meer, Real-Time Tracking of Non-Rigid Objects Using Mean Shift, *IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, USA, 2000, pp. 142-149.

[39] E. Reinhard, M. Ashikhmin, B. Gooch, P. Shirley, Color Transfer Between Images, *IEEE Computer Graphics and Applications*, Vol. 21, No. 5, pp. 34-41, September/October, 2001.

[40] D. L. Ruderman, T. W. Cronin, C. C. Chiao, Statistics of Cone Responses to Natural Images: Implications for Visual Coding, *Journal of the Optical Society of America A*, Vol. 15, No. 8, pp. 2036-2045, August, 1998.

[41] L. R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257-286, February, 1989.

Biographies

I-Cheng Chang received a B.S. degree in Nuclear Engineering in 1987, and M.S. and Ph.D. degrees in Electrical Engineering in 1991 and 1999, respectively, all from National Tsing Hua University, Taiwan. In 1999, he joined Opto-electronics and Systems Laboratories, Industrial Technology Research Institute, Taiwan as a project leader. In the autumn of 2003, he joined the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan, and he is currently a professor in National Dong Hwa University. His research interests include image/video processing, computer vision, machine learning and multimedia system design.

Chieh-Yu Liu was born in Taoyuan, Taiwan, in 1985. He received the Master degree from Department of Electrical Enigneering, National Tsing Hua University, Taiwan, in 2009. His research interests include image processing, computer vision, and object tracking.

Chung-Lin Huang received his B.S. degree in Nuclear Engineering from the National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1977, and M.S. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1979 respectively. He obtained his Ph.D. degree in Electrical Engineering from the University of Florida, Gainesville, FL, USA, in 1987. From 1987 to 1988, he worked for the Unisys Co., Orange County, CA, USA, as a project engineer. Since August 1988 he has been with the Electrical Engineering Department, National Tsing Hua University, Hsinchu, Taiwan, R.O.C. Currently, he is a professor in Aisa University, Taiwan. His research inter- ests are in the area of image processing, computer vision, and visual

communication. Dr. Huang is a senior member of IEEE.

Kunal Kabi was born in Odisha, India, in 1992. He received master from School of Computer Engineering, Kalinga Institute of Industrial Technology University, Odisha, India, in 2018. Currently, he is pursuing his doctoral degree in National Dong Hwa University, Taiwan. His current research interests include deep learning, image processing, computer vision, and object tracking.