Multi-Cue Integration for Multi-Camera Tracking

Kuan-Wen Chen[1] and Yi-Ping Hung[1,2]
[1]*Dept. of Computer Science and Information Engineering, National Taiwan University, Taiwan*
[2]*Graduate Institute of Networking and Multimedia, National Taiwan University, Taiwan*
{r93014, hung}@csie.ntu.edu.tw

Abstract

For target tracking across multiple cameras with disjoint views, previous works usually employed multiple cues and focused on learning a better matching model of each cue, separately. However, none of them had discussed how to integrate these cues to improve performance, to our best knowledge. In this paper, we look into the multi-cue integration problem and propose an unsupervised learning method since a complicated training phase is not always viable. In the experiments, we evaluate several types of score fusion methods and show that our approach learns well and can be applied to large camera networks more easily.

1. Introduction

Camera networks have been extensively used in visual surveillance system. Recently, more and more works investigated the multi-camera tracking problem with non-overlapping field-of-views (FOVs). Two visual cues were usually employed: spatio-temporal cue and appearance cue. Previous works [1],[4],[6] focused on learning better relationships of both cues among cameras and demonstrated that a better estimated model would improve tracking. All of them combined the matching probabilities of each cue with equal weights and never discussed whether selecting a better fusion weight will lead to better results.

Instead of equal weights, we hypothesize that one cue would get better results in some conditions, and a higher fusion weight should be given to it. For example, the spatio-temporal cue may outperform appearance cue when tracking vehicles, because the vehicles usually keep the distance but their colors are limited. Thus, we combine the matching probabilities with score level fusion and tried to learn better fusion weights for tracking.

Score fusion have been widely studied in multi-modal biometric authentication [5],[7],[9] or automatic speech recognition (ASR) system [8],[10]. However, to multi-camera tracking problem, which method is better remains unclear. In addition, to authentication or recognition issues, a supervised training phase is usually available. It is not always acceptable to the tracking problem though, especially when there are a lot of cameras. Furthermore, our experiments showed that the fusion weights should be varied in different environments or even different illumination conditions. Hence, a supervised learning method is not a viable option to a changing environment.

2. Background

Multi-Camera tracking with disjoint views seeks to establish correspondence between observations of objects across cameras. This is often termed as object "handover," where one camera transfers a tracked object or person to another camera. The handover list is a set of observations having left from one camera view within the maximum allowable reappearance period. Suppose that a person P enters the view of one camera. Then, the best corresponding person h in the handover list H_s is selected. Denote the probability of the target P belonging to h in the handover list as $p(P = h)$. The most likely correspondence could be obtained as follows:

$$h^* = \arg\max_{h \in H_s}(p(P = h)) \cdot \quad (1)$$

From [1], $p(P = h)$ can be estimated from $p_{st}(P = h)$ and $p_{app}(P = h)$, where $p_{st}(P = h)$ and $p_{app}(P = h)$ are the matching probabilities of p and h with spatio-temporal cue and appearance cue, respectively.

To combine these probabilities, we apply the most common score fusion method, the sum rule, which had been shown being not significantly affected by the probability estimation errors [7]. Then, we have

$$h^* = \arg\max_{h \in H_s} (w_{st} \times p_{st}(P = h) + w_{app} \times p_{app}(P = h)) \quad , s.t.\ w_{st} + w_{app} = 1. \tag{2}$$

where w_{st} and w_{app} are the fusion weights of spatio-temporal cue and appearance cue, respectively.

3. Multi-Cue Integration

In this section, three basic methods and four supervised learning methods will be introduced first. Then, we propose an unsupervised learning method based on the discriminability of genuine and impostor score distributions.

3.1. Basic method

First, we introduce three basic and widely applied methods without learning. The quantity p^m represents the score of two targets are the same one for matcher *m*. *M* is the number of matcher. The fused score is denoted as *f*. Notice that each p^m is a probability in the range [0, 1] in our system, and so no score normalization process is needed in advance.

- **Simple-Sum (Sum):** $f = \frac{1}{M}\sum_{m=1}^{M} p^m$. It is also called equal-weighted approach.
- **Max-Score (Max):** $f = \max(p^1, p^2, \ldots, p^M)$.
- **Min-Score (Min):** $f = \min(p^1, p^2, \ldots, p^M)$.

3.2. Supervised learning method

Two types of supervised learning methods are presented: Global-Weighting (GW) and Camera-Pair-Weighting (CPW). GW methods learn a set of global fusion weights for matchers [7],[8],[9],[10]. On the contrary, CPW approach assigns different fusion weights to different pairs of cameras. Jain and Ross [5] proposed user-weighting approach and declared that the fusion weights should be different for authenticating different users. Based on the same idea, we consider that the fusion weights should also be varied to different camera pairs in different environments, conditions, or even illumination. We term it as Camera-Pair-Weighting.

- **Matcher-Weighting (MW):** It is a Global-Weighting approach. Weights are assigned to each matcher based on the error rates of the matcher. Potamianos et al. [8] had shown that the optimal fusion weight is inversely proportional to the single-cue classification errors, denoted as e^m, in most practical cases. More accurate matchers are with higher weights. Then, the weight w^m associated with matcher *m* is calculated as

$$w^m = \frac{\left(1 \Big/ \sum_{m=1}^{M} \frac{1}{e^m}\right)}{e^m}. \tag{3}$$

  Then, the fused score is

$$f = \sum_{m=1}^{M} w^m p^m. \tag{4}$$

- **CPW-MW:** Weights are assigned with the same rule as MW method, except the weights are estimated separately for each pair of cameras.
- **CPW-Lambness (CPW-L):** The calculation of the weights is based on the *wolf-lamb* concept [3]. Snelick et al. [9] developed a metric of *lambness* for each matcher, which is estimated according to the distance between genuine and impostor distributions. Denote the mean and standard deviation of the genuine and impostor distributions of matcher *m* are μ_{gen}^m, σ_{gen}^m, μ_{imp}^m, and σ_{imp}^m, respectively. The *lambness* metric is formulated by the distance between two distributions, and d-prime metric is used as the measure. Then, we have

$$d^m = \frac{\mu_{gen}^m - \mu_{imp}^m}{\sqrt{\left(\sigma_{gen}^m\right)^2 + \left(\sigma_{imp}^m\right)^2}}, \tag{5}$$

  where d^m is the *un-lambness* of matcher *m*. If d^m is larger, the matcher *m* can distinguish the impostor from genuine more easily. For each camera pair, the fusion weights are calculated as follows:

$$w^m = \frac{d^m}{\sum_{m=1}^{M} d^m}. \tag{6}$$

- **CPW-Exhaustively-Search (CPW-ES):** Weights for each pair of cameras are calculated by exhaustively searching a coarse sampling of the weight space [5]. It finds the optimal solution of the training data.

3.3. Unsupervised learning method

Our unsupervised learning method is Camera-Pair-Weighting and based on the *wolf-lamb* concept [3]. For simplicity, it is abbreviated as CPW-UL method.

With the *wolf-lamb* concept, we need to estimate the mean and standard deviation of genuine and impostor distributions for calculating the *lambness* metric. For each matcher *m*, suppose that there are *N* match scores after data collection with $N = N_{gen} + N_{imp}$, where N_{gen} and N_{imp} are the number of genuine match and the number of impostor match, respectively.

Assume that the distributions of genuine and impostor scores are both Gaussian distributions. The distribution of N match scores can be considered as a mixture of two Gaussian distributions. Then, we use EM algorithm [2] to learn two separate Gaussian distributions. The distribution with larger mean value is considered as the genuine distribution. According to Equation (5) and (6), the fusion weights are calculated. Notice that there will be only one distribution estimated sometimes, which means the matcher cannot distinguish the genuine from the impostor well, and hence the weight will be set as zero.

4. Experimental Results

In this section, two match probabilities $p_{st}(P = h)$ and $p_{app}(P = h)$ are obtained by using the method proposed by Chen et al. [1]. We experiment in two environments. The first environment is shown in Figure 1. To evaluate with different illumination conditions, we turned on and off the lights in the view of E1_Cam 1, as shown in Figure 1(b). Therefore, there are three pairs of connection between cameras: E1_Cam 1 L1 and E1_Cam 2, E1_Cam 1 L2 and E1_Cam 2, and E1_Cam 2 and E1_Cam 3, abbreviated as "*E1_C12_L1*," "*E1_C12_L2*," and "*E1_C23*," respectively. We record a 6-hour period in the daytime (3 hours for E1_Cam 1 L1 and the other 3 hours for E1_Cam 1 L2). All of these 6-hour data are with ground-truthed correspondences.

The second environment is shown in Figure 2. There are three pairs of connection between cameras: E2_Cam 1 and E2_Cam 2, E2_Cam 2 and E2_Cam 3, and E2_Cam 3 and E2_Cam 4, abbreviated as "*E2_C12*," *E2_C23*," and "*E2_C34*," respectively. We record a 4-hour period in the daytime. All data are with ground-truthed correspondences.

In the following, tracking accuracy is defined as a ratio of the number of objects tracked correctly to the total number of objects passing through the scene.

First, three basic methods: Sum, Max, and Min, are evaluated. As shown in Figure 3, Max and Min methods are instable. Sum method performs best with its average tracking accuracy, 68.99%, higher than those of Max (66.52%) and Min (68.88%) methods.

Because the absolute value of tracking accuracy of different camera pairs are quite distinct and thus is not significant for comparison. We evaluate the following results by using a relative value, and Sum method is chosen as the baseline method for comparison. The improvement ratio, IR_{FM}, is represented the relative value of tracking accuracy by using the fusion method *FM*. It is defined as follows:

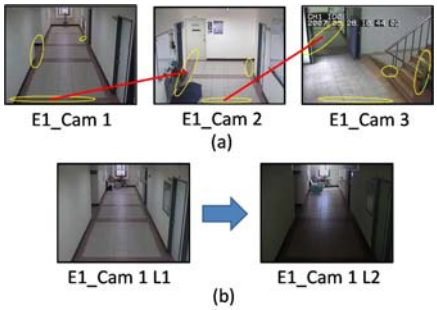

Figure 1. An indoor environment with three cameras. (a) The results estimated by Chen's method. (b) Two different illumination conditions of E1_Cam 1, named L1 for the lights turned on, and L2 for the lights turned off.

Figure 2. An outdoor environment with four cameras and the results estimated by Chen's method.

$$IR_{FM} = \frac{ACC_{FM} - ACC_{Sum}}{ACC_{Opt.} - ACC_{Worst}}, \quad (7)$$

where ACC_{FM}, ACC_{Sum}, $ACC_{Opt.}$, and ACC_{Worst} are the tracking accuracy by using *FM*, Sum method, the optimal fusion weights, and the worst fusion weights, respectively. The optimal and worst fusion weights are obtained by exhaustively searching a coarse sampling of the weight space for the ground-truthed test data.

Second, we evaluate four supervised learning method with 1-hour period of data for training and the others for testing. From Figure 4, we observe that the performance of MW method, which learns a global fusion weight, is almost the same as Sum method. The estimated weight w_{st} is 0.51, almost identical to equal-weighted approach, because the tracking accuracy of one cue in different environments is quite different. That is why using a global fusion weight is not suitable to multi-camera tracking.

CPW-MW method has better performance than MW method in *E2_C23* and *E2_C34*, because of the camera-pair-dependent weights. However, it got worse result in *E1_C23*. The estimated weight w_{st} of *E1_C23* is 1, because the tracking accuracy by using spatio-temporal cue is 100% in the training phase. It demonstrates that learning with a fixed period of data may suffer the problem of insufficient training data, which is a typical problem of supervised learning.

CPW-L method achieves the best performance in our experiments. The average improvement ratio to Sum method is 18.52% over 0% (MW), -4.83% (CPW-MW), and 0.46% (CPW-ES). CPW-ES method

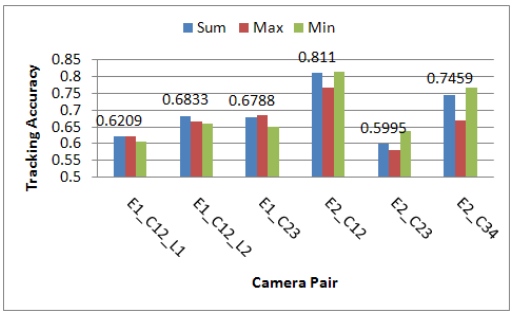

Figure 3. The tracking accuracy by using three basic methods. The values above the bars are the tracking accuracy by using Sum method.

would sometimes overfit the training data, and hence performs worse in the testing phase.

Finally, we evaluate the CPW-UL method. As shown in Figure 4, CPW-UL method outperforms Sum method in all camera pairs, and the average improvement ratio is 11.26%. Although the overall performance of CPW-UL method is not better than CPW-L method, it overcomes the problem of insufficient training data suffered by CPW-L method in *E1_C12_L2*. Most important of all, it learns without any hand-labeled training data.

Table 1 shows the estimated weights. The worst weights are always close to 0 or 1, i.e. only one cue is used. It suggests that to integrate multiple cues improves the tracking performance. The optimal weights of different camera pairs are much distinct. It shows that the fusion weights should be varied in different conditions. As we can see, the directions of bias of the weights, i.e. larger than 0.5 or smaller than 0.5, estimated by CPW-UL method are all identical to those of optimal weights. It demonstrates that the proposed CPW-UL method learns the fusion weights well.

5. Conclusion

For target tracking across multiple cameras with disjoint views, the multi-cue integration problem had never been discussed. In this paper, we introduced and evaluated several types of score fusion methods, and further proposed an unsupervised learning method. In the experiments, we have demonstrated that the fusion weights should be varied in different conditions. Hence, a supervised learning method is no longer a viable option, especially when the environments are changing. Furthermore, our method learns well without any hand-labeled training data and thus can be applied to a large camera network more easily.

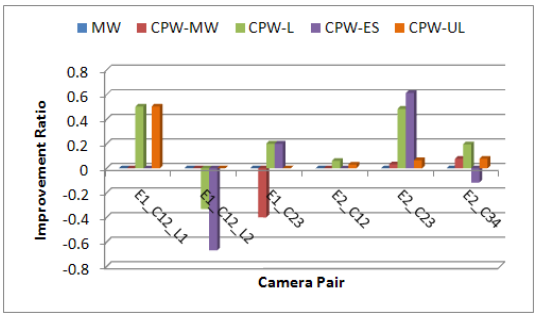

Figure 4. The improvement ratio to Sum method.

	Worst w_{st}	Optimal w_{st}	CPW-UL w_{st}
E1_C12_L1	0	0.52	0.55
E1_C12_L2	1	0.23	0.32
E1_C23	1	0.29	0.46
E2_C12	0	0.65	0.71
E2_C23	0	0.96	0.62
E2_C34	0.02	0.79	0.6

Table 1. The worst, optimal, and CPW-UL weights w_{st}.

References

[1] K. W. Chen, C. C. Lai, Y. P. Hung, and C. S. Chen, "An Adaptive Learning Method for Target Tracking across Multiple Cameras," In *CVPR*, 2008.

[2] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, B-39(1):1-38, 1977.

[3] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds, "Sheeps, Goats, Lambs and Wolves: A Statistical Analysis of Speaker Performance in the NIST 1998 Speaker Recognition Evaluation," In *ICSLD*, 1998.

[4] A. Gilbert and R. Bowden, "Incremental, Scalable Tracking of Objects Inter Camera," *CVIU*, 111(1), 2008, pp. 43-58.

[5] A. Jain and A. Ross, "Learning User-Specific Parameters in a Multibiometric System," In *ICIP*, 2002.

[6] O. Javed, K. Shafique, Z. Rasheed, and M. Shah, "Modeling Inter-Camera Space-Time and Appearance Relationships for Tracking across Non-overlapping Views," *CVIU*, 109(2), 2008, pp. 146-162.

[7] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On Combining Classifiers," *TPAMI*, 20(3): 226-239, Mar. 1998.

[8] A. Potamianos, E.S. Soto, and K. Daoudi, "Stream Weight Computation for Multi-Stream Classifiers," In *ICASSP*, 2006.

[9] R. Snelick, U. Uludag, A. Mink, M. Indovina, and A. Jain, "Large-Scale Evaluation of Multimodal Biometric Authentication Using State-of-the-Art Systems," *TPAMI*, 27(3): 450-455, Mar. 2005.

[10] E.S. Soto, A. Potamianos, and K. Daoudi, "Unsupervised Stream Weight Estimation Using Anti-Models," In *ICASSP*, 2007.