Experimental Results from using a Rank and Fuse Approach for Multi-target Tracking in CCTV Surveillance

D.M. Lyons, D. F. Hsu, C. Usandivaras, F. Montero
Dept. of Computer and Information Sciences
Fordham University, New York, NY 10023
dlyons@cis.fordham.edu

Abstract

In this paper we study a novel approach to the problem of fusion of sensory information in tracking multiple targets in CCTV surveillance video. The approach, called "Rank and Fuse" (RAF) is based on multiple feature ranking and merging as opposed to a more typical combination of all scores (similarity or probability) in a single ranking. This has the advantages of low computational complexity, easy scalability to multiple features, and low-latency. Experimental results are presented to illustrate two aspects of the RAF approach for a "difficult" example from CCTV surveillance: the advantage of rank versus score combination, and the use of the rank versus score curve to decide which features to fuse.

1 Introduction

The standard approaches developed for point target tracking, e.g., MHT [1] and JPDAF[2], have been applied to video target tracking with some success, e.g., [3, 4]. However, the information from a video sequence, even from a single camera, is much richer than from the point tracking applications with which multi-target methods originated. For this reason a crucial problem becomes the integration of multiple sensory cues [5, 6].

In this paper, we investigate an approach to the problem of multi-target tracking in video sequences that is based on using data fusion in a flexible and efficient manner. The approach is called "Rank and Fuse" [7] or RAF for short, and is based on a *combinatorial* approach to enumerating and merging sensory information from multiple sources. The approach is inspired by work in information retrieval [13] and we argue it can handle difficult sensory fusion issues on which a more traditional Bayesian approach would fail.

2 Fusion for Target Tracking

Approaches to tracking in the literature include *nearest neighbor* [8], *Joint Probabilistic Data Association Filters* or JPDAFs [2, 4] and *Multiple Hypothesis Tracking* or MHT[1, 9]. MHT is a provably optimal approach in which all combinations of matches between previous tracks and current measurements are calculated. This allows successful tracking of a large number of simultaneous targets even if the correct association is not immediately made, as may happen when targets *occlude* each other or otherwise become temporarily ambiguous (e.g., as in a crowded airport terminal scene).

The collection of sensory information from multiple sources can improve target tracking. In tracking surveillance targets in a CCTV video sequence, the video, even if it comes from a single camera, can yield a number of data sources from each image which can be treated as separate cues or features. These include foreground region properties such as centroid position, color and shape descriptions, etc [6].

MHT typically tracks based on a single feature, usually position. We ask: how can we extrapolate the MHT concept to handle multiple features in a manner appropriate for video tracking? Bayesian combination for multiple cues is proposed for this in [10] (and for JPDAF in [4]). Evidence combination and data fusion have also been considered in information retrieval [11, 12]). More recently Hsu et al. [13] initiated the study of ranking and data fusion in a new direction by considering the problem of when the *rank combination* performs better than the *score combination*. A Bayesian combination is a form of score combination, and while this does work well in many cases, there are many other, common cases in which it doesn't. In these cases, a non-linear combination is required for better performance [11, 12].

3 The *RAF* Framework

The core of the RAF framework consists of two stages (*Rank* stage and *Fusion* stage). In the Rank stage, we enumerate all possible matches of measurements to tracks per feature in a fashion similar to MHT. For each feature we calculate the score of each trajectory using a similarity measure. The RAF method can apply to different levels in a multilevel, multi-target tracking system. For example, at the object level, the goal is to score and rank the trajectories according to different features (such as positions or colors) and then combine those ranks into a simple coherent result. While at the multi-sensor level, ranks from similarity measures for each of the many sensors are integrated into a rank which has better information about the target.

3.1 Rank and Fusion Architecture and Approach

The method that we propose consists of a two-stage framework (Figure 1). The first stage (*Rank stage*) is a process of ranking the collection of possible trajectories according to each of the selected features. In this regard, each of the trajectories in the collection for a target is

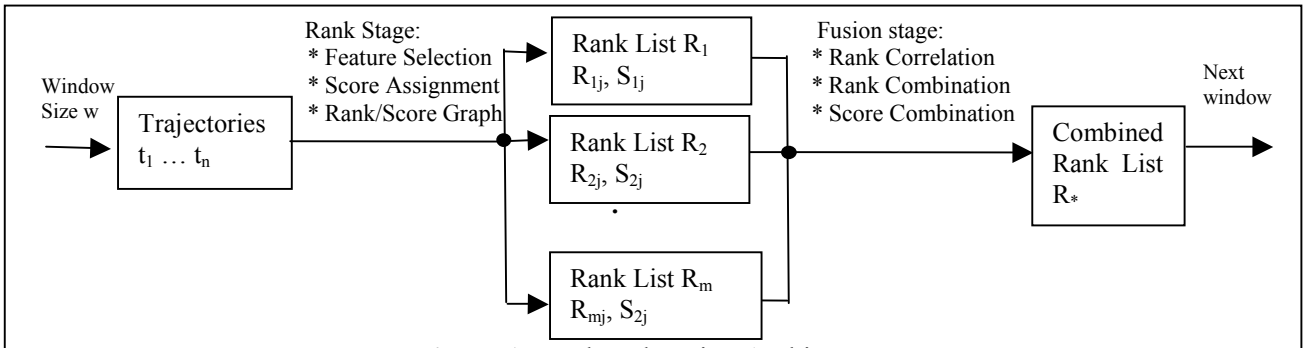

Figure 1: Rank and Fusion Architecture

assigned a score (which can be a measurement of similarity or probability) depending on a specific sensory feature. Sorting the collection on the score leads to a ranking of the trajectories in the collection. The second stage (*Fusion stage*) is a process of combining the rankings obtained from the first stage. Here the rankings are combined using different methods.

Suppose that *m* rankings have resulted from the *m* features in the rank stage. The fusion stage aims to find an optimal way to combine (or merge) the *m* rankings so that the resultant combination (and ranking) would tell us which trajectories are best supported by all measurements so far. These are the trajectories that will be ranked higher in the combination.

	1	2	3	4	5	6	7	8	9	10
R_1	t_1	t_3	t_5	t_7	t_9	t_2	t_4	t_6	t_8	t_{10}
R_2	t_2	t_5	t_8	t_1	t_4	t_{10}	t_9	t_3	t_9	t_6
R_3	t_3	t_7	t_1	t_5	t_2	t_4	t_6	t_8	t_9	t_{10}
R_*	t_1	t_5	t_3	t_2	t_7	t_4	t_8	t_9	t_6	t_{10}
V_1	(t_1	t_2	t_3)	(t_5	t_7)	t_8	(t_4	t_9)	t_{10}	t_6
V_2	(t_1	t_5)	t_2	(t_4	t_7)	t_3	(t_8	t_9)	(t_6	t_{10})

Figure 2: 3 Rankings and 3 Combinations with R_*=*LC(average)*, V_1=*VS1=Min*, V_2=*VS2=Max*.

In Hsu et al. [13] a ranking of *n* different elements is treated as a permutation of these *n* elements. Therefore, the rank space (the set of all possible rankings on *n* elements) is the set of all permutations of *n* elements, which is a group (i.e., the so-called *symmetric group* S_n). Moreover, by suitably choosing a generating set *S* of the group, and using its elements to define adjacency between nodes, the group S_n is turned into a graph $G(S_n, S)$, called the Cayley graph of the group S_n with generating set *S* (see e.g.,[13, 14]). By harnessing the group and graph structure and studying the rank correlation, the fusion stage of our RAF method can be modeled in a dynamic fashion (see e.g., [13, 15]).

There are several ways of combining the *m* rankings that are generated. Since each ranking consists of rank and score, there are both rank and score combinations. For the rank combination in general, two directions are followed. The first one involves *consensus building*. This combines the results of a list of multiple systems by using a weighted sum from each of the component systems. The second direction is the *voting model* [16] which selectively chooses a rank from some of the *m* rankings. As an example, we show three combinations: LC (averge), VS1 (minimum), VS2 (maximum). Let R_1, R_2 and R_3 be three rankings attained from the rank stage (Fig. 1) on $n=10$ trajectories labeled as t_i $i=1,..,10$, shown in Figure 2. Figure 3 shows the procedure to obtain R_*=LC (average). Each of the combinations V_1 and V_2 simply considers the rank of the trajectory t_j to be the minimum (maximum) among $\{R^{-1}_i(t_j) \mid i=1,2,3\}$. Accordingly, trajectories with the same rank are enclosed by parentheses (Fig. 2). We note that tied ranking can occur for R_* combinations also. (In this paper, we will resolve tied combinations by simply using the order of the trajectory labels.)

(a): $f(t_i)=\frac{1}{3}\sum_{j=1}^{3} R_j^{-1}(t_i)$

t_i	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
$f(t_i)$	2.66	4	3.66	6	3	8.33	4.33	6.66	7.66	8.6

(b): Sort $f(t_i)$ in ascending order

n	1	2	3	4	5	6	7	8	9	10
$S_f(n)$	2.66	3	3.66	4	4.33	6	6.66	7.66	8.33	8.6

(c): $R_*(n) = f^{-1}(S_f(n))$

n	1	2	3	4	5	6	7	8	9	10
$R_*(n)$	t_1	t_5	t_3	t_2	t_7	t_4	t_8	t_9	t_6	t_{10}

Figure 3: Procedure to obtain R_*

The RAF method is computationally efficient. For each of the *m* rankings obtained, it requires *n log n* complexity to sort the list of *n* trajectories. Moreover, the Fusion stage only takes *(m-1)n* additions and an additional *n log n* steps to obtain the combined rank. The combined rank is the result of a sensory fusion, since its input was

separate sensory rankings. Any of the standard trajectory management techniques used with MHT can now also be applied to the combined rank.

To make clear that this is a concrete application and to demonstrate that our preliminary results are promising, we begin with the problem to fuse *two sensory cue streams from a single video camera*: a position-tracking stream and a color-tracking stream.

3.2 Video Sequence Notation

We will adopt the following standard notation:

- A *Video Sequence* $\mathcal{F}$ is a set of n frames denoted $\mathcal{F} = \{F_0, F_1, F_2, \ldots\ldots, F_{n-1}\}$ where F_i is frame number $0 \le i \le n-1$. Each *frame* consists of an image I_i with a time stamp t_i, for $0 \le t_i \le t_{max}$.
- *Segmentation* is the process of identifying regions in an image that correspond to objects of interest. A segmented image $S(I_i)$ is a set of regions I_{ij} for $j \in \{0..n_i\}$.
- Each region can be characterized by a number of *feature measurements* obtained by applying a *measurement function* to the region.

3.3 Feature Selection and Score Assignment

The *trajectory* T_k of a tracked object is a sequence of regions, one element of the sequence per frame in the segmented video sequence, $T_k=(I_{ij})_i$ for object trajectory k. If there are nt_i object trajectories at image i and there are n_i regions in image i then the *association matrix* is a $nt_i \times n_i$ matrix where entry a_{kj} represents the cost of or score for associating a measured region j of the current image with an existing object trajectory hypothesis k.

Each a_{kj} is arrived at by calculating the similarity between the measured features of region j in the current frame, and of the predicted values of the last region in trajectory k, given the time difference between the measurement of j and of the measurement of the last region in k. Let ti_k be the last region in trajectory k and let tt_k be the time of the frame in which the last region of k occurs. Similarity is typically calculated as:

$$a_{kj} = p\,(f_i(ti_k), t_i\text{-}tt_k) - f_i(I_{ij})$$

where $p(f,t)$ is the *prediction function* that maps the last measured features of a region to the expected values of those features after a time t has elapsed and f_i is the measurement function for frame i. Our approach is to calculate similarity separately for each feature. A measurement function f_{iq} is defined for *each* feature q in frame i:

$$f_{iq} : S(I_{ij}) \rightarrow D_q \text{ and } f_{iq}(I_{ij}) = d_q \;,\; q \in \{1 \ldots m\},\; d_q \in D_q$$

And we define a prediction function for each feature,

$$p_q : D_q \times \{0 \ldots t_{max}\} \rightarrow D_q \text{ where } p_q(d_q,t) = \hat{d}_q$$

and $\hat{d}_q$ is the predicted value for feature q which had value d_q in the frame t time units before frame i. An association matrix is produced for each feature:

$$a_{qkj} = p_q\,(f_q(ti_k), t_i\text{-}tt_k) - f_q(I_{ij})$$

This of course requires that the association matrix construction and trajectory generation be done q times, instead of just once. However, note that these operations can be accomplished in polynomial time per feature [9].

3.4 Scoring Trajectories

After the trajectory generation phase for frame i [1], a score is assigned to each trajectory based on the association matrix value a_{qkj} and on the score of trajectory k. Reid [1] introduced a probabilistic scoring scheme that was an extension of Kalman filtering. We do not follow this approach for several reasons. Firstly, Kalman filters are not an appropriate prediction mechanism for watching human targets [17, 18]. Secondly, to keep the computational efficiency of the approach as high as possible, and to allow for a low-latency response, we will reduce the scoring mechanism to its minimal form.

The score assigned to a trajectory is simply the sum of the association matrix value and the existing trajectory, normalized to the new length of the trajectory. If the trajectory is a new trajectory, then the score is the maximum similarity value for the feature. Let l_{qk} be the length and s_{qk} be the score of trajectory k based on feature q, then for each generated trajectory $\bar{k}$:

$$s_{q\bar{k}} = \begin{cases} \dfrac{s_{qk} + a_{qkj}}{l_k + 1} & \text{if existing trajectory} \\ \max_q & \text{if new trajectory} \end{cases}$$

3.5 Sensory Fusion and Trajectory Management

We accomplish trajectory filtering and sensory fusion using the same technique: ranking and merging lists of trajectories and eliminating trajectories below a critical rank. The scoring function in the previous section allows us to establish a ranking per feature. We adopt an *N scan-back*

Figure 4: Example Frames from Video Sequence (B).

filter approach: that is, ambiguity is allowed to accumulate for a certain number of frames (the "window") and then reduced. At the end of the window in frame *i*, each trajectory feature ranking is merged into a consolidated list. This list is then reduced to its *n* top ranked members.

There are a number of different methods to merge ranked lists. A constraint that we wish to impose is that the merged permutation be *as good or better* than each individual permutation. Hsu et al. [13] have established this as long as certain conditions hold on the input rankings.

4 Experiments

Three short video sequences (A, B, and C) were used in which a human target entered a room, removed his coat and placed it on a chair, and then walked to the other side of the room (Figure 4 shows frames from A). Normally the use of both color and position cues *should* give a much more robust tracker than simply a color or position tracker on its own. However, in this difficult but rather common case, the majority of the color information comes from the coat, and hence will introduce incorrect trajectories at the time that the coat is removed.

Each video was processed by passing it through a background subtraction module that used a *non-parametric* statistical approach to building a color model of what constitutes background for each pixel in the image [19, 20] and then doing a connected-components analysis for each pixel whose value did not fall into the background. The resultant foreground regions for each frame *i* are the inputs I_{ij} for our approach.

(a) **(b)**

Figure 5: Successful (a) and unsuccessful (b) Trajectories on the sequence in Fig. 5 (video B).

4.1 Measurement and Prediction Functions

Two measurement and prediction functions were used: a position feature measure and a color feature measure: The position feature was the position of the centroid of the region. The color feature was the average color of the region. The shape feature was simply the bounding box ratio. Prediction of the position feature was accomplished by calculating the frame to frame velocity of the centroid [21] *v* and using it as follows:

$$p_{pos}(x,t) = x+vt$$

The prediction function for color was the identity function.

4.2 Ranking , Merging and Evaluation

For the purpose of this experiment, *the window size w* was set to be a function of the number of trajectories. Whenever the number of trajectories exceeded 250% of the critical value *N=100*, then the number of trajectories was sorted and reduced to *N*. Each trajectory ranked in the top *q* (*q=20 here)* in the resultant ranking (Fig. 6) is checked for correctness (Figure 5). We often obtain more than one correct trajectory in the resultant combination.

A Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	..	32
r_p	**0**	64	**37**	**1**	**3**	65	**26**	**38**	67	**2**	**40**	**4**	**57**	66	**27**	**39**	68	**29**	**41**	**16**		
r_c	64	67	65	68	239	231	241	233	235	237	99	244	240	232	242	234	236	238	257	260		**63**
r_{p+c}	64	67	65	**0**	68	66	**3**	**1**	**4**	**37**	**2**	34	**40**	99	**63**	36	**38**	**26**	35	**73**		
r_{VS1}	64	**0**	67	65	**37**	68	**1**	239	**3**	231	**26**	241	**38**	223	235	237	**40**	**4**	99	244		
r_+	**0**	64	67	**37**	65	**1**	68	**3**	239	231	**26**	241	**38**	223	235	**40**	237	**4**	99	**57**		

B Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
r_p	**0**	45	**21**	58	**1**	**8**	**13**	46	49	63	**22**	**72**	**90**	**77**	**86**	51	**194**	**25**	**82**	**21**
r_c	45	51	100	174	176	175	191	177	193	179	192	178	171	173	**168**	172	**0**	181	**170**	**169**
r_{p+c}	45	**0**	**21**	51	**13**	49	58	**8**	54	**90**	**16**	**196**	**86**	**82**	172	48	**167**	**27**	**4**	63
r_{VS1}	**0**	45	51	54	**21**	58	48	**8**	174	176	**13**	49	191	63	177	193	**72**	206	**90**	179
r_+	**0**	45	51	**21**	54	58	48	**8**	174	**13**	176	49	191	63	177	**72**	193	**90**	206	**77**

C Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
r_p	**18**	92	**0**	**3**	**2**	74	77	**1**	**26**	**4**	76	**5**	**12**	75	**13**	78	**11**	**21**	**55**	79
r_c	**7**	**8**	**103**	**9**	**107**	**112**	**6**	**149**	**102**	**151**	**161**	**142**	**162**	**152**	**143**	**155**	**163**	**156**	**354**	**136**
r_{p+c}	92	**18**	74	77	76	307	**0**	**22**	79	75	**2**	86	**4**	85	**5**	**1**	80	84	95	87
r_{VS1}	81	**18**	92	82	**0**	83	80	**3**	78	**2**	79	74	84	77	86	**1**	85	**26**	**4**	76
r_+	**18**	81	92	82	**0**	83	**3**	80	**2**	78	74	79	77	84	**1**	86	**26**	85	**4**	92

Figure 6: Results for Video Sequences A, B & C.

4.3 Rank and Score Combination Results

Five tracking runs were completed for each video sequence: one that used only the position feature, one that used only the color feature, one that used a measurement function that was a *sum* of the color and position scores, and two rank combinations for position and color – a VS1 combination and a Merge combination. Figure 6 shows the top ranked 20 trajectories for each of the position (r_p), color (r_c), sum of position and color scores (r_{p+c}), combination by minimum rank (r_{V1}), and combination by merged rank (r_+). Successful trajectories are shown in bold font. Trajectories were assigned unique identifiers for each video sequence, based on a run-code encoding. This code is valid for every run in each sequence; it is *not* valid across sequences, of course.

The position-only run produced the correct trajectory of the target as the top ranked trajectory in all cases. The color run did not have the correct trajectory in A in the top 20 ranking, it was the 32nd. This is because the color feature was misdirected when the target removed his coat and placed it on the chair. The sum of the scores run (r_{p+c}) had the correct trajectory in rank 4, since the local combination was weighted down by the incorrect color information. However, a straight merging of the ranked results from the color and position runs (r_+) produces the correct trajectory at rank 1. The combination by minimum rank selection also performed well. These combinations perform in a similar fashion on sequence B.

In sequence C, the color information is correct. Combination by score performs well in this case of course, but the rank combinations perform well also. Hence, rank combinations handle the difficult cases in A and B, but have not lost functionality on the "easy" case.

4.4 Rank/Score Graphs

Hsu et al. [13] remarks that rank combination will outperform score combination under certain conditions. In particular, they note that in the literature it has been observed empirically that when the ranking behavior of the two input ranking processes are sufficiently different, then combination by rank outperforms combination by score. In their own results, they derive a specific criterion for rank outperforming score.

Figure 7 shows the rank versus score graph for three feature cues: position, color, and a new feature, shape. Note that while position and color cues have a somewhat similar overall form, the shape cue appears to have a different form. Based on the empirical observations cited before, we would expect that a rank combination of shape and color would outperform a score combination. Figure 8 shows the results of this experiment, where r_s is the shape cue ranking, r_{s+c} is the score combination of shape and color, and r_{V1}, r_* are the rank combinations by minimum and by average discussed previously. Shape and Color are not particularly good cues for this tracking application. Their score combination is not particularly effective. However, the rank combination by sum performs noticeably better, as predicted.

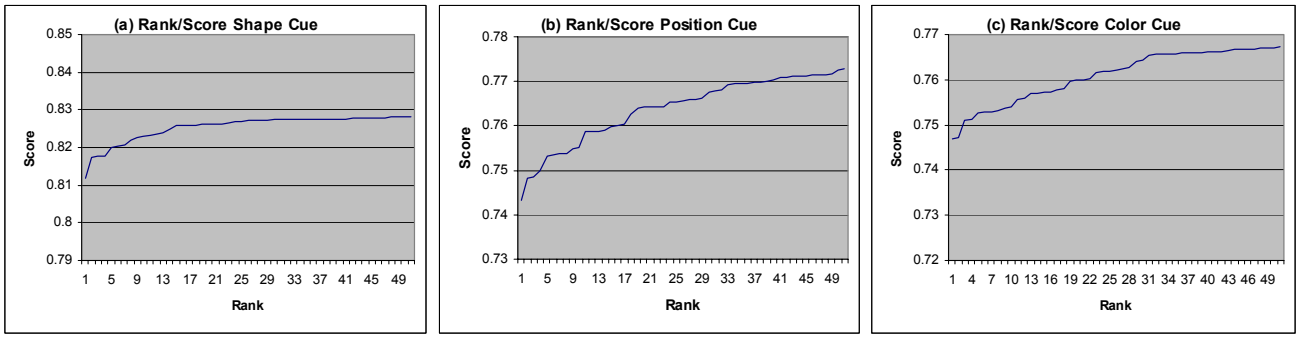

Figure 7: Rank/Score Graphs for (a) shape (b) position (c) color from video sequence B

B Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
r_s	34	37	42	30	56	35	45	33	27	38	**49**	**2**	**5**	**75**	**78**	29	**54**	39	**47**	40
r_c	100	102	101	180	182	181	197	183	199	185	198	184	177	179	**174**	178	**106**	187	**176**	**175**
r_{s+c}	100	202	30	201	200	34	27	42	33	206	37	32	31	201	36	29	205	38	35	203
r_{VS1}	100	34	37	180	42	197	183	30	36	177	**174**	35	**106**	45	187	33	**170**	27	38	103
r_*	100	202	207	200	**250**	203	**258**	206	**243**	204	205	215	**237**	30	201	**271**	**233**	**230**	**273**	283

Figure 8: Results for Video Sequence B fusing color and shape

5 Discussion

We have applied the RAF method to multi-target tracking in CCTV surveillance and have obtained results which indicate that the RAF approach handles difficult

feature fusion cases better than just a score combination method (eg Bayesian) could. In particular, our experiments on this difficult example demonstrated that each of the rank combinations r_+ and r_{V1} performs better than that of r_{p+c} (combination resulting from the sum of feature scores r_p and r_c). However, our combination r_{p+c} in all cases exhibits correct trajectories in its top 20 ranked list. If the typical technique used with MHT of looking at the best ranked trajectory in r_{p+c} were employed, no correct trajectory would have been found however. We have also shown some empirical validation of the remarks in [13] and elsewhere that the more different the component rank/score graphs are, the better rank combination should perform over score combination.

We discuss the significance and advantages of RAF operation under two key headings:

(1) *Computational Efficiency.* Sorting a list of nt_i items as in our examples has complexity $nt_i*log\ nt_i$. All the features can thus be sorted in $m*nt_i*log\ nt_i$ worst case. Note that the combination operation does not have to proceed beyond the critical rank.

(2) *Scalability.* The mechanism scales *linearly* by the number of features. This encourages us to believe that this mechanism can be used as broadly as possible. That is, not only for multi-cue fusion from a single camera, but also multi-camera fusion, and ultimately extension to other sensory modalities than vision.

Note that while the RAF appproach handles multiple targets and occlusion issues, it does *not* handle the merging or splitting of trajectories when dealing with groups. We argue that this should be handled separately, e.g., [21], based on the tracking data supplied by RAF.

The current study indicates several directions for future work:

(1) For one video sequence, two trackers are employed, each using (a) different similarity measures, and (b) different pruning measures, tracking one, two and then many objects.

(2) For two (or more) video sequences of the same events, two (or more) identical or different trackers are used for one, two and then more objects

(3) The rank r_{p+c} is obtained from combining the constituent rankings r_p and r_c. On the other hand, r_{V1}, r_{V2}, and r_+ result from combining ranks in r_p and r_c. We will explore other score and rank combinations with various rank/score graphs in future work.

6 References

1. Reid, D.B., *An Algorithm for Tracking Multiple Targets.* IEEE Trans. on Aut. Control, 1979. AC-24(6): p. 843-854.

2. Bar-Shalom, Y., Fortmann, T.,, *Tracking and Data Association*. 1988: Academic Press.

3. Cox, I.J., Hingorani, S.L. *An Efficient Implementation and Evaluation of Reid's Multiple Hypothesis Tracking Algorithm for Visual Tracking*. *Int. Conf. on Pattern Recognition*. 1994.

4. Rasmussen, C., Hager, G.,. *Joint Probabalistic Techniques for tracking Multi-Part Objects*. *Proc. Computer Vision & Pat. Recog.*. 1998. Santa Barbara CA.

5. Varshney, P.K., *Special Issue on Data Fusion.* Proceedings of the IEEE, 1997. 85(1).

6. Loy, G., Fletcher, L., Apostoloff, N., Zelinsky, A. *An Adaptive Fusion Architecture for Target Tracking*. *Proc. of 5th Int. Conf. on Face and Gesture Recog.* 2002. Washington DC.

7. Hsu, D.F., Lyons, D.M., Usandivaras, C., Montero, F. *RAF: A Dynamic and Efficient Approach to Fusion for Multi-target Tracking in CCTV Surveillance*. *IEEE Int. Conf. on Multisensor Fusion*. 2003. Tokyo Japan.

8. Rao, B., *Data Association Methods for Tracking Systems*, in *Active Vision*, A. Blake, Yuille, A.,, Editor. 1992, MIT Press. p. 91-106.

9. Cox, I.J., Miller, M.L., *On Finding Ranked Assignments with Application to Multi-Target Tracking and Motion Correspondence.* IEEE Trans. AES, 1995. 32(1): p. 486-489.

10. Liggins II, M.E., Kadar, I., Vannicola, V., *Distributed Fusion Architectures and Algorithms for Target Tracking.* Proc. of the IEEE, 1997. 85(1): p. 95-107.

11. Vogt, C.C., Cotrell, G.W., *Fusion via a linear combination of scores.* Info. Ret. 1999. 1(3): p. 151-172.

12. Ng, K.B., Kantor, P.B., *Predicting the effectiveness of Niave Data Fusion on the basis of system characteristics.* Jour. American Soc. Info. Sc., 2000. 51(13): p. 1177-1189.

13. Hsu, D.F., Shapiro, J., Taksa, I., *Methods of Data Fusion in Information Retreival: Rank vs. Score Combination*. 2002, DIMACS TR 2002-58.

14. Heydermann, M.C., *Cayley graphs and interconnection networks*, in *Graph Symmetry*, G. Hahn, Sabiclussi, G., Editor. 1997, Kluwer Academic. p. 161-224.

15. Marden, J.I., *Analysing and Modeling Rank Data*. 1995: Chapman & Hall.

16. Parker, J.R., *Multiple Sensors, Voting Methods and Target Value Analysis*. 1998, Univ. of Calgary: Alberta CA.

17. Isard, M., Blake, A. *Contour tracking by stochastic propagation of conditional density*. in *European Conf. on Computer Vision*. 1996. Cambridge UK.

18. Collins, R.T., Lipton, A.J., Kanade, T. *A System for Video Surveillance and Monitoring*. *Proc. American Nuclear Society 8th Int. Topical Meeting on Robotics and Remote Systems*. 1999.

19. Brodsky, T., Cohen, R., Cohen-Solal, E., Gutta, S., Lyons, D., Philomin, V., Trajkovic, M. *Visual Surveillance in Retail Stores and in the Home*. in *Proc. European Workshop on Advanced Video Surveillance*. 2001. Kingsten on Thames UK.

20. Elgammal, A., Harwood, D., Davis, L.S.,. *Nonparametric Model for Background Subtraction*. *Proc. 6th European Conference on Computer Vision*. 2000.

21. Courtney, J.D., *Automatic Video Indexing via Object Motion Analysis.* Pattern Recog, 1997. 30(4): p. 607-625.