18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Multiway Histogram Intersection for Multi-target Tracking

Yu Pang[♮] **Xinchu Shi**[♮] **Bin Jia**[‡] **Erik Blasch**[♯] **Carolyn Sheaff**[♯] **Khanh Pham**[♯]
Genshe Chen[‡] **Haibin Ling**[♮]

[♮]Computer & Information Science Department, Temple University, Philadelphia, PA, USA
[‡]Intelligent Fusion Technology, Inc, Germantown, MD, USA
[♯]Air Force Research Lab, USA

{*yu.pang,xinchu.shi,hbling*}*@temple.edu,* {*bin.jia,gchen*}*@intfusiontech.com,*
{*erik.blasch,carolyn.sheaff,khanh.pham.1*}*@us.af.mil*

***Abstract*—In recent studies of multi-target tracking, high-order association and its corresponding high-order affinity (or similarity) is often preferred over pairwise comparisons to capture high-order discriminative information. A naturally raised challenge is to calculate affinity (or similarity) among more than two target candidates. When target appearance is represented by histograms, such as the popular SIFT and HOG descriptors, pairwise matching measurements, such as Histogram Intersection (HI) *etc.*are often combined to fit the high-order request in an ad hoc way. However, such combinations may be ineffective and inefficient.**

In this paper, we address the pairwise matching issue by proposing a novel multi-histogram similarity named Multiway Histogram Intersection (MHI). MHI naturally extends HI by summing over the "min" value of all histograms in each bin. MHI applies to any number of histograms, fits the request of multi-target tracking better and requires less time than previously used affinities. To demonstrate its superiority, we integrate MHI into a recently proposed rank-1-tensor-approximation multi-tracking framework and apply it to vehicle tracking in wide aerial video surveillance. The advantage of using MHI is clearly supported by the experimental results against six common approaches on two public benchmark datasets.

Keywords: multiway histogram intersection, feature comparison, tensor, multi-target tracking, WAMI, etc.

I. Introduction

To tackle *multi-target tracking* (MTT) problems, a good affinity representation is always of great importance. Roughly speaking, affinity can be viewed as the similarity between a set of targets (or target candidates) and the observed image. Similar appearances or motion patterns should yield higher affinity scores and vice versa. While pairwise affinity has been widely used in many vision tasks, MTT often prefers affinities over more than two targets such as the likelihood over a set of target candidates which form a trajectory or tracklet. Normally, the trajectory or tracklet consists of a set of targets[1] extracted from consecutive frames, one target per frame. When there are more than two targets involved, their affinity is often referred to as a high-order affinity.

[1]For conciseness, in this paper we do not distinguish between target and target candidates, since our focus is on the multiway affinity.

Attributing to the increasing popularity of encoding high-order information in MTT (*e.g.*, through multi-target association), high-order affinity becomes an important factor in modern MTT algorithms. Since histogram-based representation like Scale-invariant feature transform (SIFT) and Histogram of oriented gradients (HOG) are widely used for target representation, measuring the similarity over a set of histograms plays a key role in defining high-order affinity for MTT. Previous studies (details in Sec. II) often construct such affinity on top of pairwise histogram similarities, *e.g.*, by averaging histogram intersections over all pairs of targets. However, such affinities do not capture the high-order information in a natural way and may also suffer from low efficiency as illustrated in Sec. IV.

In this paper, we design a new high-order affinity, named *Multiway Histogram Intersection* (MHI), to measure the similarity of a set of histograms. Defined as the sum of multiway min operation over each histogram bin, MHI naturally generalizes the popular *Histogram Intersection* (HI) [32] from a pairwise affinity to a groupwise one. Compared with previously used multiway versions of HI, MHI has several advantages in both affinity accuracy and computational efficiency, which make it suitable for MTT tasks.

For our application, we integrate MHI into a recently proposed tensor-based MTT algorithm [30] by replacing the original affinity with the one generated by MHI. The algorithm is then tested on two recent challenging benchmarks for tracking vehicles in wide area video surveillance. In both experiments, our method not only consistently outperforms the baseline algorithms with traditional affinities, but also achieves the state-of-the-art results in comparison with recently tested MTT algorithms.

The rest of the paper is organized as follows: Sec. II introduces the related work. Sec. III and IV show the definition of Multiway Histogram Intersection and its properties with experiments. Sec. V explains how it can be used in multi-target tracking. Sec. VI illustrates its performance on two real datasets. Finally, Sec. VII concludes our work.

II. Related Work

Multi-target tracking has gained increasing attention due to its wide application in many areas, such as surveillance, security and biomedical applications. Most of the current studies focus on designing models, frameworks and affinity models. For an in-depth survey, readers are encouraged to read [16]. Nevertheless, there are few efforts paid on the similarity measurement. Yet, it is an important factor to improve the performance.

In general, similarity measurements are components in affinity models which contain appearance, motion, interaction, and exclusion models [16]. Given a specific model, a similarity measurement is used to compare a set of targets and use their likelihood to determine if they should be in the same group or different groups. For example, given a set of color histograms, we can measure their appearance similarity; given point positions, we can measure whether the speed is stable or not *etc.*. More specifically, for histogram-based appearance models, straightforward selections include L_1, L_2, χ^2 distances [13], Histogram Intersection [30], Bhattacharyya distance [35, 39], correlation coefficients [12] or the earth mover's distance (EMD) [10, 26]. However, all the above mentioned distance comparison methods perform pairwise comparisons.

When there are more than two targets to compare, *e.g.*, targets in a trajectory (or tracklet), various strategies have been designed for effective tracking. One way is to represent the target set as *single feature vector*. For example, in [31] key points are used to represent the tracklets, such as the beginning or the end point of the tracklet. In [24] the average histogram is used to represent the tracklet.

Another way is to first compute a set of pairwise similarities using the above listed distance-based methods, then combine them to get the final *multiway similarity*. A direct solution would be using a greedy algorithm [35] or Hungarian algorithm [20] to find correspondences between consecutive frames. In [10, 12, 21, 30], linear combinations or multiplications over pairwise similarities are used. To solve the association problem, a linear programming model is proposed in [11], cost-flow network models are used in [1, 4, 21, 39], or a low rank tensor decomposition is used in [29, 30]. When measuring trajectory affinities, most of these works use pairwise distances computed from consecutive frames, *i.e.*, $(t_1, t_2), \ldots, (t_k, t_{k+1}), \ldots, (t_{n-1}, t_n)$ for the n frames. There are also some studies that use all pairwise distances over all pairs in a tracklet. For example, a complete graph over all the candidates in a segment of frames is first constructed in [38]; then the generalized minimum clique problem is solved over the graph. Intrinsically, the resulting clique uses information from every pair. [27] stacks a set of histograms into a matrix, then finds a row mapping between the two matrices that minimize the residuals.

A third way is to learn an affinity representation from the *training data*. For example, the HybridBoost algorithm is used in [13] to learn ranking classifiers given any two pairs. In [36, 37], a CRF model is created between tracklets and the parameters are learned. Still, all of the aforementioned literatures utilize pairwise comparison within and between tracklets.

There are a few studies in the literature consider high-order affinity for *motion models*. For a k-th order affinity, the method in [19] uses the first $(k-1)$ items to fit a motion model, then test how well the last one fits in the model. In [9], many 3-order affinities are provided based on geometric invariance, such as similarity, affine or projective invariance. But few considers high-order appearance affinity.

Histogram intersection (HI) [32] is popularized for its simplicity and effectiveness. Besides, HI is also robust to occlusion, change of view, image resolution and certain degree of background distraction. Later on, it is proved in [1] that HI is a Mercer's kernel, which enlarges its application across the fields. In terms of SVM, [17, 33] show that by building a HI kernel, SVM can achieve runtime complexity of logarithmic in the number of support vectors. Furthermore, by some approximating algorithms HI can reduce to even constant runtime and space requirements. In tracking, HI is often used in comparing image histograms. Other commonly used measurements in tracking include χ^2 distance, Bhattacharyya distance and EMD. Bhattacharyya distance is popularized by the MeanShift traker [7] as a successful tracker at the early 2000, while EMD [26] is designed for content-based image retrieval (CBIR). None of these measurements can be easily extended to high-order measurements, whereas HI can be directly generalized as we explain in Sec. III.

III. Multiway Histogram Intersection

A. Definition

Let $\mathcal{H} = \{H_1, H_2, \ldots, H_N\}$ be a set of N histograms, each histogram $H_n = (h_n(1), h_n(2), \ldots, h_n(K))^\top$ has K bins, $n = 1, 2, \ldots, N$. In the MTT scenario, these N histograms can be extracted from N targets located at N consecutive frames. We define the *Multiway Histogram Intersection* (MHI) of $\mathcal{H}$ as

$$\mathrm{MHI}(\mathcal{H}) = \sum_{k=1}^{K} \min\left(h_1(k), h_2(k), ...h_n(k)\right) . \qquad (1)$$

In other words, MHI is defined as the sum of the minimums along each histogram bin over all input histograms in $\mathcal{H}$. Note that, though by definition we do not restrict histograms to be normalized, in practice it is usually desirable to ensure all the histograms are of the same scale.

MHI naturally generalizes the popular histogram interaction (HI) that is defined over a pair of histograms. Or, HI can be viewed as a special case of MHI for $N = 2$. In this way, MHI inherits properties of HI for histogram similarity.

The calculation of MHI can also be done incrementally by

$$\begin{aligned}\mathrm{MHI}(\mathcal{H}) &= \sum_{k=1}^{K} \min\Big(h_1(k), h_2(k), ...h_n(k)\Big) \\ &= \sum_{k=1}^{K} \min\Big(h_n^{(N-1)}(k), h_n(k)\Big) \\ &= \mathrm{HI}(\mathcal{H}^{(N-1)}, H_N)\ . \end{aligned} \tag{2}$$

where $\mathrm{HI}(\cdot,\cdot)$ is the histogram intersection, and $\mathcal{H}^{(m)} = \min(H_1, \ldots, H_m)$ is the element-wise minimum of the first m histograms in $\mathcal{H}$.

B. Other HI-based Affinities

Previously, there are two main ways to use HI to build high-order affinity in MTT. The first way computes the average HI over all neighbouring pairs of histograms in $\mathcal{H}$, corresponding to neighboring targets in a trajectory. This results in the *Neighboring HI* (NHI) [30] as

$$\mathrm{NHI}(\mathcal{H}) = \frac{1}{N-1}\sum_{i=1}^{N-1} \mathrm{HI}(H_i, H_{i+1}). \tag{3}$$

Similarly, the second method is to average HI over the complete set of histogram pairs, yielding the so called *Complete HI* (CHI) [38] as

$$\mathrm{CHI}(\mathcal{H}) = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N} \mathrm{HI}(H_i, H_j). \tag{4}$$

Comparing these two methods, CHI is better than NHI in terms of capturing more reliable appearance similarity. Given a short tracklet, we always assume the target appearance does not change much. Thus, comparing all pairs reduces the chance that a single appearance is similar to the previous one, but differs significantly from even earlier ones, which happens when using NHI. However, CHI is much more computationally expensive as summarized in Table I.

As for MHI, it captures both advantages of NHI and CHI. MHI naturally considers all the histograms in a trajectory (or tracklet), meanwhile it is of low computational cost. Furthermore, NHI and CHI need to be normalized by their number of HIs when the number of hypothesises in a trajectory varies. MHI is inherently between 0 and 1 no matter how many inputs are involved. In other words, no explicit normalization is needed to handle different input sizes, which is often desirable in MTT settings.

The computational costs for the three affinities are summarized in Table I. MHI needs the least number of operations. NHI needs a set of additional "+" operations to calculate the HI value for each pair, while MHI only needs to do it once. CHI has a higher complexity because all combinations of pairs are in the set.

TABLE I: Computational cost of HI-related high-order affinities. N is the number of histograms and K is the number of bins.

	"min" operation	"+" operation
MHI	$(N-1)K$	$K-1$
NHI	$(N-1)K$	$(N-1)K$
CHI	$N(N-1)K/2$	$N(N-1)K/2$

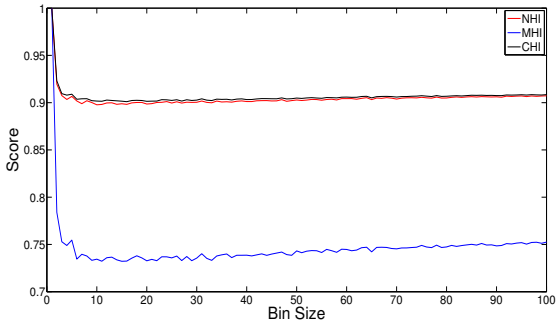
IV. Property Analysis

To study the performance of HI-related high-order affinities including the proposed MHI, we run a set of synthetic experiments as follows: Given a random histogram of bin size K as the *base histogram*, we generate N histograms in which we add some Gaussian noise $\mathcal{N}(0, \sigma^2)$ to each bin of the base histogram (we enforce bin values to be non-negative). Then we record and plot the scores (affinity values) of each measurement. The three main factors that are studied here include:

- K: the bin size,
- N: the number of histograms involved,
- σ: the noise level of the histograms.

A. Bin size

Fig. 1: The affinity scores of different bin size.



We fixed $N = 10$ and $\sigma = 0.2K$ (because a histogram needs to be normalized to 1, so the σ is smaller for a larger K). We can see from Fig. 1, the bin size does not affect the scores much. They quickly become stable when $K \geq 3$ as the bin size increases. However, MHI has a much lower stable score than both NHI and CHI.

B. Number of histograms

We fixed $K = 20$ and $\sigma = 0.01$. We can see from Fig. 2, NHI and CHI become stable very quickly after $N >= 6$ while MHI keeps decreasing. Also, MHI is much lower than the other two at the same N. MHI keeps the minimum value of each bin from all the histograms, so the bin with largest negative noise is kept. On the other hand, HI and CHI are based on pairs, so they average out the noise effect when more histograms are involved.

Fig. 2: The scores of different number of histograms

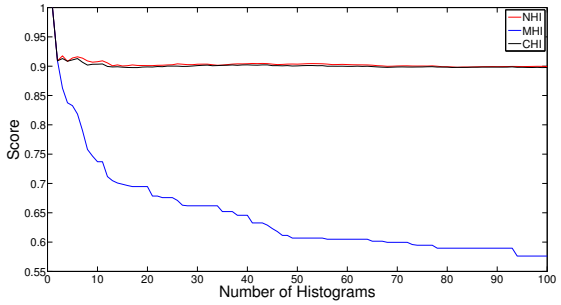


Fig. 3: The affinity scores at different noise level.

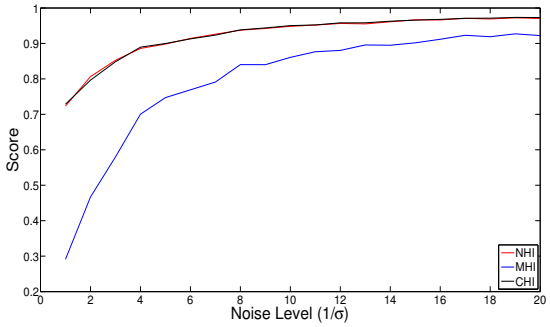


Fig. 4: The affinity scores of randomly generated histograms.

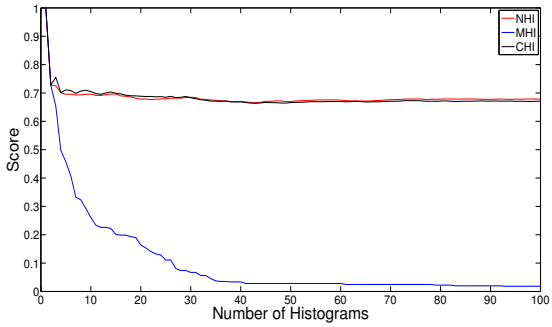
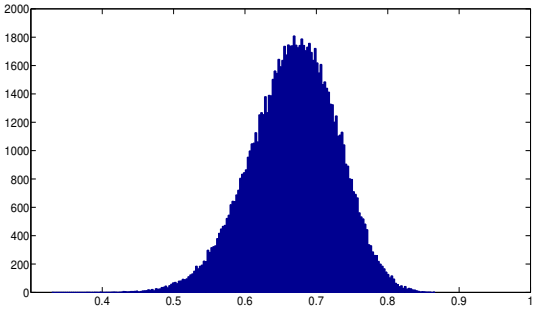


Fig. 5: The distribution of affinity scores of HI estimated from 100K pairs of randomly generated histograms.



C. Noise level

We fixed $N = 10$ and $K = 20$ and the noise level is $L = 1/\sigma$.

We can see from Fig. 3, the scores increased as the noise reduced. Also, the steeper curve of MHI indicates it has a wider noise response range. To test the extreme case, we conducted another experiment in which we computed scores of a set of randomly generated histograms. We fixed $K = 20$, as the number of histograms (N) increases, the score should be always close to 0 ideally. The result is shown in Fig. 4, MHI drops to close to 0 very quickly, while NHI and CHI become stable at around 0.67.

To further understand the behavior of MHI, NHI and CHI, we generate a pair of random histograms and compute their HI score. Running $100K$ times, we draw its distribution in Fig. 5. The distribution is similar to a Gaussian shape with mean value at around 0.67. Although HI has the range from 0 to 1, the random histogram scores are not near the bottom (around 0) of the range. This suggests that in most scenarios, HI has a much shorter range of its representation ability. However, in multiway scenario, MHI compensates for this drawback of HI and gives a wider representation range.

V. Multiway Histogram Intersection for Multi-target Tracking

The MHI measurement is suitable for any high-order histogram-based matching. A natural scenario is multi-target tracking (MTT). In most MTT settings as mentioned in Sec. II, the task is to aggregate the detected points or tracklets into trajectories. No matter whichever algorithm one would use, measuring similarity score of a hypothesis trajectory is always needed. Previously, such higher order measurement is achieved through combining pairwise measurements. MHI, on the other hand, provides a straightforward comparison between any arbitrary number of inputs.

There are many algorithms that could adopt MHI and one of the recently proposed methods is the rank-1 tensor approximation [30]. Rank-1 tensor explicitly requires a similarity between a set of points and their appearance feature histograms. In the rank-1 model, a tensor is a high dimensional form of a matrix. Each dimension represents edge relationships between two consecutive frames. In this sense, the index of the elements in the tensor would represent a set of connected edges which forms a potential trajectory. So the value in that element, which is called affinity, can be regarded as the similarity between those points that form such a trajectory.

The basic assumption is that the tensor is constructed from a series of outer products of binary vectors, thus a rank-1 tensor. So given a tensor $\mathcal{A}$ estimated from the similarity measurement, we want to minimize Eq. 5

$$\min_{\lambda,\mathbb{V}} ||\mathcal{A} - \lambda V^{(1)} * \cdots * V^{(K)}||_F^2, \tag{5}$$

where $V^{(k)}$ is a binary vector indicating whether an edge exists or not, λ is a scaling factor and $*$ means the outer product of a tensor and vector. The approximation of Eq. 5 minimizes the Frobenius norm of the difference between the tensor $\mathcal{A}$ and its reconstructed rank-1 tensor. More detailed formulations and solutions on the rank-1 tensor decomposition are discussed in [30].

We use the source code provided by the authors of the [30] with the same histogram features that are computed from HOG [8]. Each candidate patch is normalized to a fixed size and a 96-bin histogram is constructed to capture its gradient distribution.

The main difference in our work compared to [30] is how to construct the tensor $\mathcal{A}$. Their original paper used NHI to compute affinities in $\mathcal{A}$. That is, given an element position in $\mathcal{A}$, the index represents a potential trajectory. They then compute NHI from the appearances of the points in that trajectory and use it as the affinity value. So, in the experiments in Sec. VI, we simply replace NHI with MHI and fix some scaling issues, whereas everything else is kept the same.

VI. Experiments

A. WAMI data

Wide aerial motion imagery (WAMI) is a type of image source that is shot from the air, where the targets are extremely small compared to other indoor or outdoor scenarios. Typically a target is around 20×20 in pixels and the image size is normally greater than 2000×2000. One image may contain hundreds of targets.

Compared with other types of sequences, WAMI datasets are harder to deal with. Firstly, due to the small region of targets and the gray scale image quality, the targets are extremely noisy in terms of their appearance models. This is a good test for different measurements whether or not they can handle the noise well. Secondly, because the number of candidates is very large and the capturing rate is very low in such datasets, the potential number of candidate trajectories can be huge.

The key issue is by assigning higher affinity values to the correct trajectories and lower the values of the wrong ones, the algorithms are more likely to pick up the true ones. So, WAMI datasets are challenging scenarios for MTT affinity testing. A summary of WAMI tracking methods can be found in [2, 3].

B. Other baseline measurements

Besides NHI in their original paper [30], we also propose a few other baseline measurements to compare with. For all the measurements mentioned below, we only replace the tensor construction part as mentioned in Sec. V and keep all the rest unchanged. The first one is the Complete Histogram Intersection (CHI) as mentioned in Sec. III-B. Bhattacharrya is used similar to NHI, except we replace the HI measurement with Bhattacharrya distance. Jensen-Shannon (JS) divergence is proposed in [14] and its generalized version can deal with multiple distributions all at once. JS is defined as:

$$JS_{\pi}(p_1, p_2, ...p_n) = H(\sum_{i=1}^{n} \pi_i p_i) - \sum_{i=1}^{n} \pi_i H(p_i),$$

where p_i are a set of distributions, π_i are a set of weight (or prior distribution), $H(\cdot)$ represents the entropy function. Without any prior assumption, π_i can be set to $\pi_i = \frac{1}{n}$. If we consider histograms as distributions, the purpose for Jensen-Shannon divergence is very similar to the MHI measurement.

C. CLIF Dataset experiment

This experiment is conducted on Columbus Large Image Format (CLIF) dataset [5, 15, 18]. The image size is 4016×2672. We use the same three sub-sequences as in [28, 30], each having 100 frames with detailed annotation. Sequence 1 is a heavy traffic scene with more than 200 vehicles on average. The other two are sparser with 80 vehicles on average.

The correct matching percentage P_c and wrong matching percentage P_w are defined as:

$$P_c = 100 \times \frac{\Sigma_t cm(t)}{\Sigma_t g(t)}, P_w = 100 \times \frac{\Sigma_t wm(t)}{\Sigma_t g(t)} \qquad (6)$$

respectively, where $cm(t)$ is the number of correct matching and $wm(t)$ is the number of wrong matching, while $g(t)$ is the ground truth. Note here, wrong matching may consist of *true detection* matches *true detection* but wrong pair, or *true detection* matches *false alarm* or even *false alarm* matches *false alarm*. So the sum of P_c and P_w may not necessarily add up to 1.

The results of the first 3 methods in Tab. II are copied from [30]. It can be shown that by merely altering the affinity measurement to MHI, we can obtain a consistently better result than all the other baseline measurements and other methods. Because both the correct percentage is increasing and wrong percentage is decreasing, we can see MHI has a better discriminative ability to show the similarity of a bundle of objects.

Taking NHI and MHI for example, in Fig. 7, red is good and green is bad. Red boxes 1 and 2 are good in both cases. Thus, red boxes 1 to 5 show the correct trajectory and the green boxes 3 to 5 show an incorrect trajectory. From the corresponding histograms, the three green boxes have quite similar histograms, thus have a high HI comparison value. However, the three red boxes 3 to 5 are less similar thus yield low HI values. NHI gives 0.51 to the red boxes trajectory and 0.68 to the red-green mixed trajectory. On the other hand, MHI is determined by the lowest scores from all the histogram bins as shown at the end of each histograms column. The correct MHI has a higher score 0.36 than the wrong one 0.23. So, MHI corrects such potential errors, thus has a higher accuracy and lowers track switching.

D. WPAFB Dataset experiment

Using another WAMI dataset called WPAFB [34], we again tested the MHI. The scene is similar to CLIF, but is much more

Fig. 6: An example from the CLIF dataset and our tracking results. Top: a cropped region from a frame in CLIF. Bottom: tracking results from our algorithm; cyan and red show individual trajectories for five frames and different colors show the two directions.

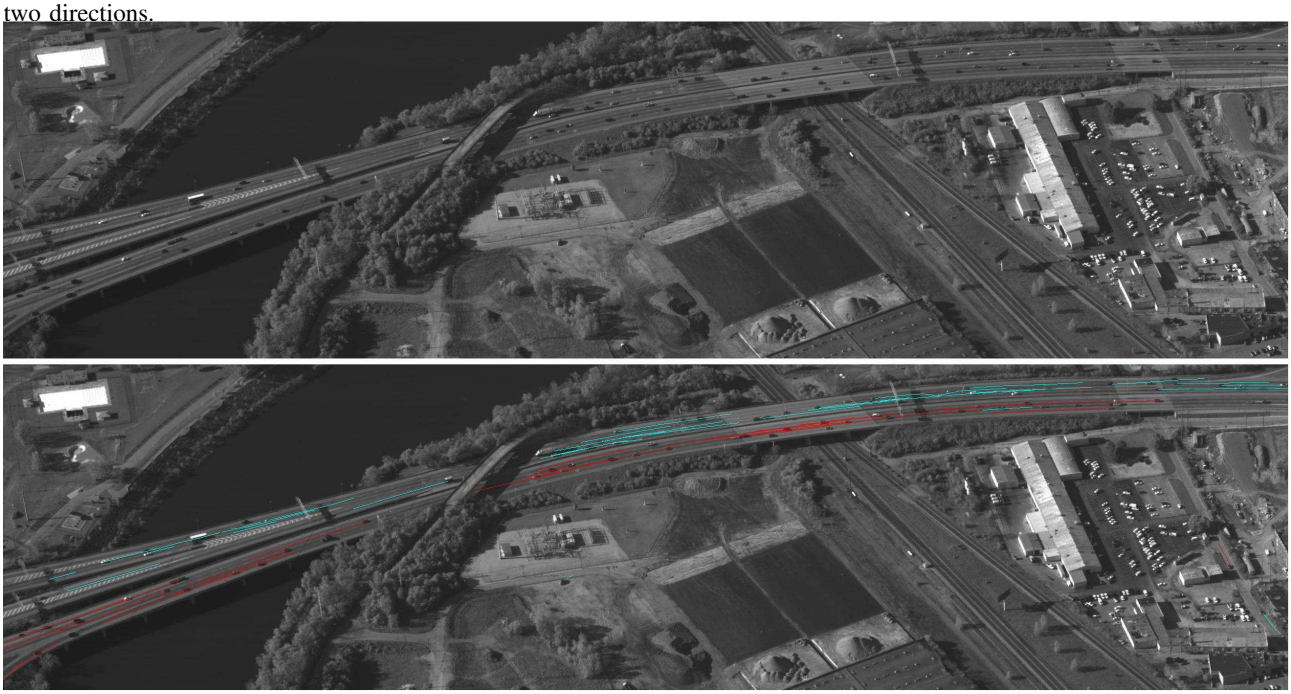


TABLE II: Tracking results on the CLIF dataset. Note: results of HUN, ICM, TENSOR_NHI are obtained from [30]. The **best** performance is shown in **red**, the second is shown in blue.

	Correct matching percentage			Wrong matching percentage		
	Seq1	Seq2	Seq3	Seq1	Seq2	Seq3
HUN[30]	75.8	86.8	83.3	25.2	11.3	16.5
ICM[6]	83.1	89.6	87.3	16.5	10.3	12.9
TENSOR_NHI[30]	91.1	92.1	91.4	11.9	9.4	9.4
CHI	90.5	92.7	93.5	6.8	**3.3**	5.6
Bhattacharrya	87.7	91.9	92.1	9.4	4.3	9.6
Jensen-Shannon	91.7	93.0	93.7	5.1	4.0	7.4
TENSOR_MHI	**93.2**	**93.3**	**94.4**	**3.5**	3.4	**5.2**

complicated in the direction that a vehicle may move. We used the pre-processed dataset by [23] as described in their paper and is shown in Fig. 8. The image size is 1408×1408. There are 1125 frames with average target number of 24 per frame.

We obtained the initial detection result and ground truth from the authors and used it as our input. However, the detection contains a large amount of false alarms and only a fraction of the ground truth. The ground truth contains 24,594 targets across all frames, while the detection contains about 13,000 targets with about 125,000 false alarms.

We used the same configuration as in Sec. VI-C. It focuses on association only and assume all input are valid, thus it does not tell if a candidate trajectory is false alarm or not. On the other hand, [23] handles such false alarms naturally in their algorithm, so the result may not be fair if we compare them directly. We applied a simple method to eliminate many false alarms. We labeled a small fraction of 10% of the detections and threw them into an SVM with HOG features. The SVM classifier was used to classify all the detections. We still fed all the detections into the baseline methods and the MHI method. The only post-processing we performed is if all the points in a trajectory are classified as negative samples, the trajectory was discarded. In this way, the number of false alarms was greatly reduced.

The results are shown in Tab. III. It is similar as in Sec. VI-C, where the MHI results show a better performance than using NHI or other baseline measurements. Meanwhile, the tensor method with MHI can achieve similar or better performance than other state-of-the-art methods. Note here, we did not perform any post processing except for discarding all the negative trajectories, so the swaps and breaks are slightly higher than the baseline methods.

Fig. 8: Left: full view of a frame in the WPAFB dataset. Middle: the cropped part(the dash rectangle in the left image) used in our experiment. Right: the trajectories generated using our algorithm.

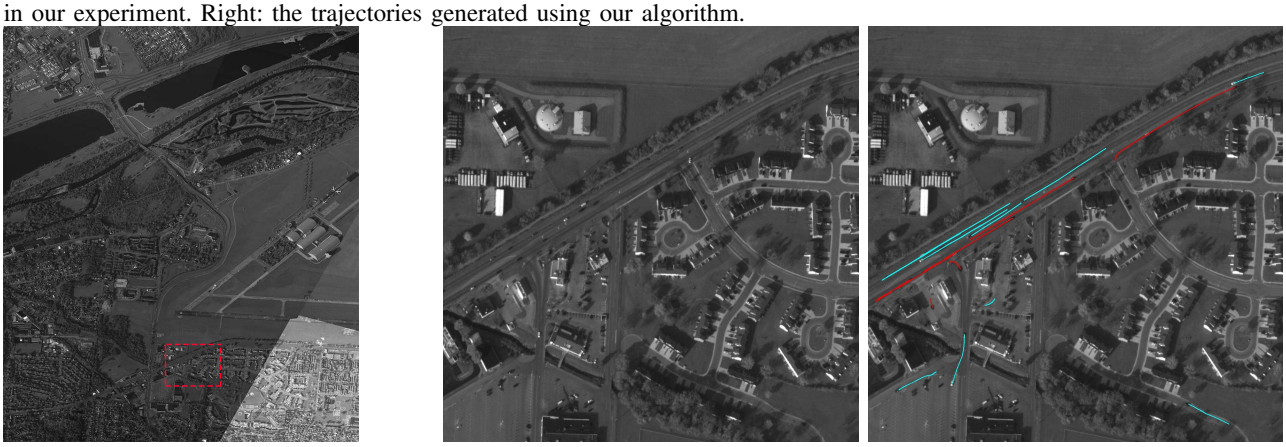


TABLE III: Tracking results on the WPAFB dataset. Results of [21–23, 25] are obtained from [23].The **best** performance is shown in **red**, the second is shown in blue.

	MHI	CHI	Bhattacharrya	Jensen-Shannon	NHI[30]	[23]	[22]	[21]	[25]
Detect. Rate (Recall)	0.47	0.46	0.47	0.46	0.47	**0.48**	0.41	0.36	0.44
Precision	0.96	0.94	0.85	0.91	0.89	0.92	**0.97**	0.14	0.14
False Pos. per Frame	0.49	0.66	1.93	1.10	1.32	1.03	**0.35**	53.5	65.1
False Pos. per GT	0.02	0.03	0.08	0.05	0.06	0.04	**0.01**	2.23	2.72
MODA	**0.45**	0.43	0.39	0.42	0.41	0.44	0.39	-1.87	-2.27
Track Swaps	0.48	0.50	1.11	0.45	0.55	**0.20**	0.36	1.23	1.31
Track Breaks	2.06	2.09	2.45	2.00	2.13	**0.99**	1.77	2.80	3.10
MOTA	**0.44**	0.42	0.37	0.41	0.40	0.43	0.39	-1.90	-2.30

VII. Conclusion

In this paper, we present a new method for multi-target tracking which is able to compare multiple histograms all at once. The Multiway Histogram Intersection(MHI) technique is generalized from the Histogram Intersection method. MHI both reduces the computational cost and increases the representation power and accuracy. We conducted several synthetic data analysis to understand the behavior of MHI and showed its superiority over other HI-related methods in different scenarios. Finally, we presented experiments on two WAMI datasets and showed that using MHI is a better choice over many other MTT approaches and it can achieve the state-of-the-art performance. Besides a multi-target tracking scenario, MHI can potentially be used in various fields as long as multiple features need to be compared at the same time.

Acknowledgement

This research was partly supported by the United States Air Force under contract number FA8750-15-C-0025 and FA8750-13-C-0110, and by the NSF CAREER Award IIS-1350521. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Air Force.

References

[1] A. Barla, F. Odone, and A. Verri. Histogram intersection kernel for image classification. In *ICIP, 2003.*

[2] E. Blasch, G. Seetharaman, S. Suddarth, K. Palaniappan, G. Chen, H. Ling, A. Basharat. Summary of Methods in Wide-Area Motion Imagery (WAMI). In *SPIE, Vol. 9089, 2014.*

[3] E. Blasch, G. Seetharaman, K. Palaniappan, H. Ling, G. Chen. Wide-Area Motion Imagery (WAMI) Exploitation Tools for Enhanced Situation Awareness. In *Proc. IEEE Applied Imagery Pattern Recognition (AIPR) Workshop: Computer Vision: Time for Change, 2012.*

[4] A. A. Butt and R. T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *CVPR, 2013.*

[5] Clif 2006 dataset. www.sdms.afrl.af.mil/index.php?collection=clif2006.

[6] R. T. Collins. Multitarget data association with higher-order motion models. In *CVPR, 2012.*

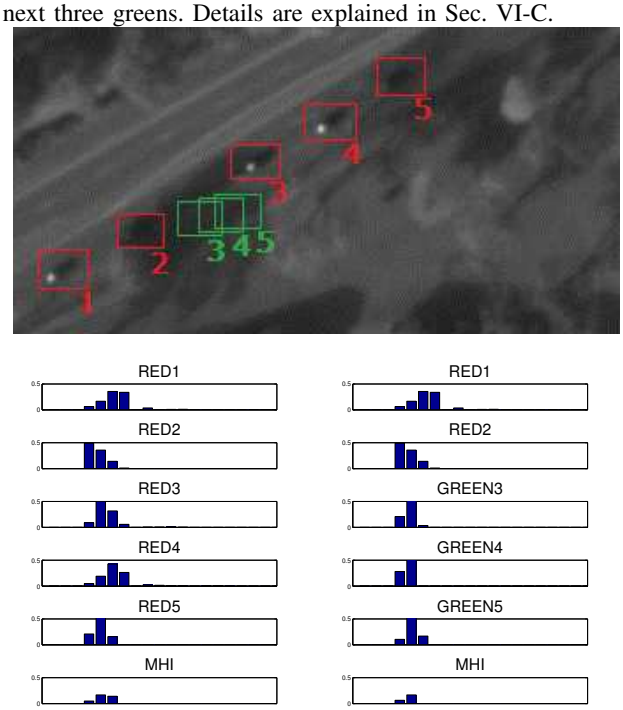
[7] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR, 2000.*

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR, 2005.*

[9] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *TPAMI, 33(12):2383–2395, 2011.*

[10] W. Ge and R. T. Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *BMVC, 2008.*

Fig. 7: An example of NHI vs. MHI. The red boxes are the correct trajectory while the green ones are wrong. The bottom left is the corresponding histograms of red boxes path, the bottom right is the histograms of the first two reds and the next three greens. Details are explained in Sec. VI-C.



[11] H. Jiang, S. Fels, and J. J. Little. A linear programming approach for multiple object tracking. In *CVPR, 2007.*

[12] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR, 2011.*

[13] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR, 2009.*

[14] J. Lin. Divergence measures based on the Shannon entropy. *Transaction on Information Theory, 37(1):145–151, 1991.*

[15] H. Ling, Y. Wu, E. Blasch, G. Chen, and L. Bai. Evaluation of Visual Tracking in Extremely Low Frame Rate Wide Area Motion Imagery. *Info Fusion, 2011.*

[16] W. Luo, X. Zhao, and T.-K. Kim. Multiple object tracking: A review. *arXiv preprint arXiv:1409.7618, 2014.*

[17] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR, 2008.*

[18] O. Mendoza-Schrock, J. A. Patrick, and E. Blasch. Video Image Registration Evaluation for a Layered Sensing Environment. In *NAECON, 2009.*

[19] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *CVPR, 2012.*

[20] A. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR, 2006.*

[21] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR, 2011.*

[22] J. Prokaj, M. Duchaineau, and G. Medioni. Inferring tracklets for multi-object tracking. In *CVPRW, 2011.*

[23] J. Prokaj and G. Medioni. Persistent tracking for wide area aerial surveillance. In *CVPR, 2014.*

[24] Z. Qin and C. R. Shelton. Improving multi-target tracking via social grouping. In *CVPR, 2012.*

[25] V. Reilly, H. Idrees, and M. Shah. Detection and tracking of large number of targets in wide area surveillance. In *ECCV, 2010.*

[26] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *IJCV, 40(2):99–121, 2000.*

[27] D. Shapira, S. Avidan, and Y. Hel-Or. Multiple histogram matching. In *ICIP, 2013.*

[28] X. Shi, P. Li, W. Hu, E. Blasch, and H. Ling. Using Maximum Consistency Context for Multiple Target Association in Wide Area Traffic Scenes. In *ICASSP, 2013.*

[29] X. Shi, H. Ling, W. Hu, C. Yuan, and J. Xing. Multi-target tracking with motion context in tensor power iteration. In *CVPR, 2014.*

[30] X. Shi, H. Ling, J. Xing, and W. Hu. Multi-target tracking by rank-1 tensor approximation. In *CVPR, 2013.*

[31] B. Song, T.-Y. Jeng, E. Staudt, and A. K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *ECCV, 2010.*

[32] M. J. Swain and D. H. Ballard. Color indexing. *IJCV, 7(1):11–32, 1991.*

[33] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *TPAMI, 34(3):480–492, 2012.*

[34] Afrl wpafb 2009 dataset. https://www.sdms.afrl.af.mil/index.php?collection=wpafb2009.

[35] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *CVPR, 2006.*

[36] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a crf model. In *CVPR, 2011.*

[37] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *CVPR, 2012.*

[38] A. R. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV, 2012.*

[39] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR, 2008.*