Vehicle Identification between Non-Overlapping Cameras without Direct Feature Matching

Ying Shan Harpreet S. Sawhney Rakesh Kumar
Vision Technologies Laboratory
Sarnoff Corporation
201 Washington Road, Princeton, NJ 08540
{yshan, hsawhney, rkumar}@sarnoff.com

Abstract

We propose a novel method for identifying road vehicles between two non-overlapping cameras. The problem is formulated as a same-different classification problem: probability of two vehicle images from two distinct cameras being from the same vehicle or from different vehicles. The key idea is to compute the probability without matching the two vehicle images directly, which is a process vulnerable to drastic appearance and aspect changes. We represent each vehicle image as an embedding amongst representative exemplars of vehicles within the **same** *camera. The embedding is computed as a vector each of whose components is a non-metric distance for a vehicle to an exemplar. The non-metric distances are computed using robust matching of oriented edge images. A set of truthed training examples of same-different vehicle pairings across the two cameras is used to learn a classifier that encodes the probability distributions. A pair of the embeddings representing two vehicles across two cameras are then used to compute the same-different probability. In order for the vehicle exemplars to be representative for both cameras, we also propose a method for jointly selection of corresponding exemplars using the training data. Experiments on observations of over 400 vehicles under drastically illumination and camera conditions demonstrate promising results.*

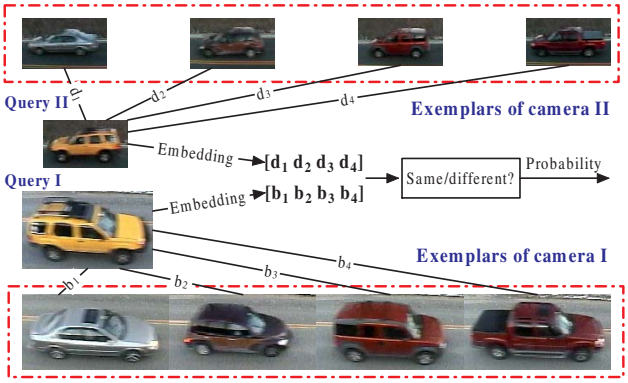

Figure 1. Directly matching vehicle objects such as Query I and Query II between a pair of non-overlapping cameras can be very difficult due to drastic illumination/appearance and aspect changes. The key idea of our work is to use non-metric distance embeddings $[d_1, d_2, d_3, d_4]$ of vehicle observations within a camera and their corresponding embeddings $[b_1, b_2, b_3, b_4]$ in another camera as a means of characterizing similarities and differences between vehicles across cameras. The embeddings are with respect to exemplars precomputed for each camera, and the output is the probability of whether the two vehicles are the same or different

1. Introduction

We address the problem of matching vehicles across non-overlapping camera pairs. An example scenario is tracking vehicles on a complex road network with fixed cameras where the number of cameras is minimized to provide large area coverage, and hence the cameras do not overlap. Complexities of the matching problem include large appearance/illumination and aspect changes across the cameras. The vehicle chips as shown in Fig. 1 are extracted through a real-time in-camera tracker dedicated to each camera. In previous work [14], we addressed the problem of computing the probability that two vehicle observations across two cameras are derived from the same vehicle or different vehicles. The proposed solution used alignment and matching of oriented edge images of pairs of vehicles across two cameras on the basis of which the same-different probabilities are learned. However, given the variations in appearances and aspect of the same vehicle across disparate observations, direct matching may not always provide a reliable means of computing same-different probabilities.

In this work, we explore the possibility of computing the same-different probabilities for pairs of disparate observations *without* the need for directly matching these observations. Figure 1 illustrates the basic idea and the key components of the proposed approach. The intuition is that for the class of objects like vehicles, it should be possible to

represent an observation of a vehicle in a given camera using distances to a set of representative vehicle observations within the *same* camera. To compute these distances, comparison of observations is required within the same camera only. Since the camera is fixed, all observations will be similarly affected by illumination and other environmental effects as well as the camera aspect. Therefore, robust alignment and matching within the same camera can accurately capture the similarities and differences amongst the vehicles. With any vehicle in any camera represented as a vector of distances to the respective exemplars within that camera, two such vector representations can be compared to produce the probabilities of two disparate observations being of the same or different vehicles.

We capture the above intuition within a formal framework, in which we first address the problem of computing robust distance measures between a pair of vehicle objects. Since the distances are computed on a pairwise basis using robust alignment and matching, they are typically non-metric. We address the problem of selecting exemplars from two sets of corresponding vehicle objects based on the non-metric distance measures. We also address the problem of embedding the vehicle object with respect to the exemplars in each individual camera, and use the embedded distance vectors of both query vehicles to compute the same-different probabilities using probabilistic SVM, which is trained based on a set of truthed training examples of same-different vehicle pairings between the two cameras in question. Experimental results using a number of different camera pairs and over 400 different vehicle examples shows very encouraging results of the proposed approach.

2. Related Work

Vetter and Poggio first propose the idea of learning shape changes from the 2D prototype shapes of two distinctive views [16]. Under some linearity assumptions, they decompose the 2D shape of an object $\mathbf{x}$ as a linear combination of a set of basis shapes $\{\mathbf{x}_i\}$ such that $\mathbf{x} = \sum \alpha_i \mathbf{x}_i$. They then conclude that if $y = \sum \beta_i \mathbf{y}_i$, where y and y_i are the corresponding shape and the basis of objects in another view, it holds that $\alpha_i = \beta_i$. To facilitate some of their underlying assumptions in the algorithm, they decomposed the object image into shape and texture, and treated them separately. Instead of computing a linear projection within the space spanned by the basis shapes/textures, our approach uses an embedding process that requires only the distances of the query image with respect to each basis image. The distance measure can be non-metric. Moreover, the mapping between the embedding coordinate system of two views is automatically learned and is not assumed to be linear.

Another seminal work related to our approach is the 3D model-based vehicle matching method [10], where a deformable model for 5 classes of vehicle objects is estimated from the video sequence. The estimated parameters can then be used for vehicle recognition and classification. Instead of using a hand crafted 3D model, our approach uses 2D exemplars automatically selected for each camera.

One major technical component of the proposed work is related to the previous work on feature embedding for object recognition and classification. Athitsos et. al. [2] use Lipschitz embedding to approximate the Chamfer distance [3] for hand pose recognition with large database. Since the direct Chamfer distance is non-metric, some additional errors will be introduced into the approximation process when triples of images violate triangle inequality. Also the exemplars for embedding are randomly selected. To address the non-metric problem, Athitsos et. al. [1] propose a nice approach using AdaBoost to learn the embedding with the triangle inequality enforced. Grauman et. al. [4] use Locality Sensitive Hashing-based embedding (LSH-embedding) [6] to approximate the expensive Earth Mover's distance. By design, the LSH-embedding works only with the Earth Mover's distance.

Another technical component of our work is related to previous work on edge-based object matching. In [3, 2, 15], edge features were used to detect traffic signs, pedestrians, and for recognizing hand gestures. Examples of traditional edge-based match measures include Chamfer distance [3], Hausdorff distance [5], and Earth Mover's distance [4]. In [12, 15], both edge locations and orientations are used to define a combined edge measure, which is reported to improve the matching and classification performance significantly. Many previous works use clean edge maps for at least one of the edge maps. Truncated Chamfer distance [5] or robust Hausdorff distance [12] may work for these cases, but not for the cases when both edge maps are not clean and there is significant clutter in the scene. Our approach uses the robust edge-based distance proposed in [14] that has been proved to work with the cases when both query and model are not clean. Since the distance measure we use is non-metric, we extend the non-metric embedding algorithm proposed in [7] to compute more meaningful distances.

On the application side, [8] also deals with object matching between non-overlapping cameras and on-line learning of camera topology and path probabilities. The work in [9] and [13] propose a nice framework for object matching and feature learning. All these methods rely on directly matching vehicle objects across multiple cameras.

The remainder of the paper is organized as follows. We mathematically state our problem in Section 3 and present the overall algorithm in Section 4. We define a robust distance measure between edge maps in Section 5. We describe the algorithm to select exemplar pairs in Section 6 and present the embedding, learning, and classification al-

gorithms in Section 7. Finally in Section 8, we present detailed experimental results.

3. Problem Statement and Notations

For a given pair of cameras C_i and C_j, we want to estimate the probability density functions:

$$\begin{aligned} P(\mathbf{y} \mid \text{same}, C_i, C_j) &\equiv P(\mathbf{y} \mid \mathcal{S}_{i,j}) \\ P(\mathbf{y} \mid \text{different}, C_i, C_j) &\equiv P(\mathbf{y} \mid \mathcal{D}_{i,j}) \ , \end{aligned} \tag{1}$$

where $P(\mathbf{y} \mid \mathcal{D}_{i,j})$ and $P(\mathbf{y} \mid \mathcal{S}_{i,j})$, are the probability density functions of the measurement vector $\mathbf{y}$ given that the two observations are of same/different vehicles, and

$$\mathbf{y} = f_{i,j}(E_k^i, E_l^j) \ , \tag{2}$$

where $f_{i,j}$ is a function of two observed edge maps, E_k^i and E_l^j, corresponding to the kth and lth observations in cameras C_i and C_j, respectively. Both edge maps could be contaminated by noise, scene clutter, and obscuration.

For training, given a set of edge maps $\mathcal{E}_i = \{E_k^i, k = 1, \cdots, N\}$ for the ith camera, and a set of corresponding edge maps $\mathcal{E}_j = \{E_k^j, k = 1, \cdots, N\}$ for the jth camera, the problem is to compute the probability density functions in (1), without directly matching the edge maps E_k^i and E_l^j. Note here the correspondences between two edge sets are manually labeled, and the number of edge maps N is the same for both sets.

For testing, given a pair of edge maps E_x^i and E_y^j, we want to compute the measurement vector $\mathbf{y}$ between them without direct matching, and compute the same-different probabilities according to the learned probability density functions in (1).

4. Algorithm Overview

The algorithm consists of a training stage and a classification stage. The first step of the training stage is to select representative exemplars from the corresponding edge sets of two cameras. To select exemplars that are representative for edge sets of both cameras, we propose an algorithm that selects a set $\mathcal{C} = \{(E_k^i, E_k^j) \,|\, k = 1, \cdots, O\}$ of exemplar pairs jointly from both sets of edge maps $\mathcal{E}_i$ and $\mathcal{E}_j$. Note that O is the number of the exemplars, and (E_k^i, E_k^j) are the pairs of corresponding edge maps selected from both sets of the edge maps. Details are described in Section 5 and Section 6.

The second step of the training stage is to construct a training set $\mathcal{T} = \{(E_k^i, E_k^j, l_k) \,|\, k = 1, \cdots, T\}$, where $l_k = \{0, 1\}$ is the truth label for the pair of edge maps E_k^i and E_k^j, and T is the number of training samples. An embedding vector is then computed for each training sample as described in Section 7. Embedding uses the exemplars selected during the first step of the training process. A probabilistic SVM [17] classifier is trained based on the embedding vectors. The embedding process is essentially computing the function $f_{i,j}$ in (2), and the resulting classifier encodes the same-different probability distributions in (1).

During the classification stage, given a pair of query edge maps (E_x^i, E_y^j), we use (2) to compute its embedding vector $\mathbf{y}$, and use the trained classifier to compute its same-different probabilities.

5. Robust Distance Measure between a Pair of Edge Maps

Figure 2 shows a typical alignment result of two edge maps and their corresponding images. Yellow pixels in the figure are perfect matches. Edge pixels such as those shown in the circle are approximate matches. Other pixels such as those in the square are outliers that do not have any close matches. We exploit the content in the edge maps by computing a robust distance measure that takes into account information in both the inlier and outlier pixels.

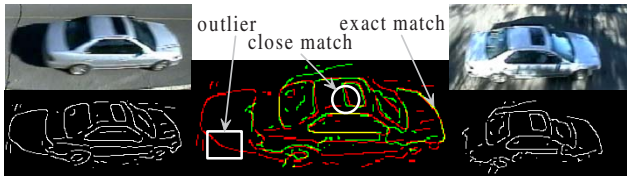

Figure 2. An example of edge map alignment (middle image) of two vehicle images (top left and top right), and their corresponding edge maps (bottom left and bottom right). This figure is best viewed with color. The two vehicles in this figure are selected from two different cameras to make the point. However in this work, inter-camera matching is not used

5.1. Edge-based Measures

Figure 3 shows three edge-based raw measures that are extracted from a pair of edge maps. Since we are only interested in shape-based matching, the magnitude different measure is not used in this paper. Suppose that E_A and E_B are two aligned edge maps, p is a point in E_A, and q is the closest point to p in E_B, we define

$$d_{A \mapsto B}^p = \|p - q\|_1 \ , \tag{3}$$

$$a_{A \mapsto B}^p = \theta_p - \theta_q \ , \tag{4}$$

where d and a denote distance and angular difference, respectively; θ_p and θ_q are the edge directions defined at the

edge points p and q, respectively. The subscript $A \mapsto B$ denotes that the closest point is defined from E_A to E_B.

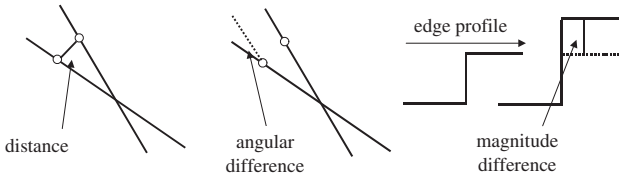

Figure 3. Edge-based measures: pointwise distance, angular difference, and gradient magnitude difference

5.2. Robust Distance Measure

Based on the first two edge-based measures, a robust match score between a pair of edge maps is derived as:

$$\gamma = \frac{\sum_{A \mapsto BI} h(d^p, \delta) h(a^p, \alpha) + \sum_{B \mapsto A} h(d^p, \delta) h(a^p, \alpha)}{N(A) + N(B)} , \tag{5}$$

where $N(A)$ and $N(B)$ are the numbers of edge pixels of the edge maps E_A and E_B, $\gamma \equiv \gamma_{A,B}$, $h(x, c) = (1 - |x|/c)$ for $|x| < c$, $h(x, c) = \rho$ for $|x| \geq c$, ρ is a small positive number, and d^p and a^p are defined as in (3) and (4). The constants δ and α can either be predefined and kept the same for all pairs of cameras, or statistically computed from the data of each camera as by estimating the inlier and outlier processes as in [14]. Since the score is in the range of $[0, 1]$, we simply define the distance measure as

$$d_{A,B} = 1 - \gamma_{A,B} . \tag{6}$$

It can be seen from (5) and (6) that the score converts the pointwise distance and angular difference into a single robust match measure. A similar score has been used in [14], and proved to be superior than the truncated Chamfer distance [5], and the robust Hausdorff distance [12]. Also note that the distance measure in (5) is symmetric. However, like many other robust distances, (5) is not a metric because the triangle inequality is not guaranteed.

For all pairs of edge maps in both edge sets $\mathcal{E}_i$ and $\mathcal{E}_j$, we then compute dissimilarity matrices $\mathbf{D}_i$ and $\mathbf{D}_j$, according to the distance measure defined in (6). An entry of matrix $\mathbf{D}_i$ represents the distance for a pair of edge maps in camera i and similar for $\mathbf{D}_j$.

6. Exemplar Pairs for Non-Metric Embedding

From the edge sets $\mathcal{E}_i$ and $\mathcal{E}_j$, we want to find the set of exemplar pairs $\mathcal{C} = \{(E_k^i, E_k^j) \,|\, k = 1, \cdots, O\}$ that are representative for both cameras. A standard method of computing representative objects from a given metric dissimilarity matrix is to use clustering such as Kolmogorov clustering algorithm. For non-metric distances, [7] computes *redundancy* as a more meaningful measure of whether one object can be replaced by another. As a result, we first convert matrices $\mathbf{D}_i$ and $\mathbf{D}_j$ into redundancy matrices $\mathbf{Q}_i$ and $\mathbf{Q}_j$ as in Section 6.1. In order to enforce the constraint that the exemplars are representative for both cameras, we then define a joint dissimilarity matrix $\mathbf{J}$ based on $\mathbf{Q}_i$ and $\mathbf{Q}_j$ (Section 6.2), and use Kolmogorov clustering algorithm to compute the exemplars.

6.1. Redundancy Matrix from Distance-based Dissimilarity Matrix

Given a robust distance-based dissimilarity matrix $\mathbf{D} = \{d_{A,B}\}$, where E_A and E_B are two aligned edge maps, we define a matrix $\mathbf{Q} = \{q_{A,B}\}$ such that

$$q_{A,B} = \mathrm{corr}(\mathbf{v}_A, \mathbf{v}_B) , \tag{7}$$

where "corr" denotes correlation coefficients, and $\mathbf{v}_A$ and $\mathbf{v}_B$ are the distance vectors defined as:

$$\mathbf{v}_X = \{d_{X,K} \,|\, \forall K \in \mathcal{E}, K \neq A, B\} , \tag{8}$$

where X is either A or B. $\mathbf{v}_X$ is a vector of distances between the edge map X to all other edge maps except A and B in the same edge set $\mathcal{E}_i$ or $\mathcal{E}_j$. The quantity $q_{A,B}$ defined in (7) is essentially a correlation-based estimation of *redundancy* [7], which is the probability of $|d_{A,K} - d_{B,K}|$ being small for an arbitrary edge map K. In the case when the distance measure d is Euclidean, $|d_{A,K} - d_{B,K}|$ is guaranteed to be small if $|d_{A,B}|$ is small, because of the triangle inequality. However, when the distance is non-metric, (7) provides a more meaningful estimation of the redundancy.

6.2. Exemplars for a Pair of Edge Sets

Given a pair of edge maps (E_A^i, E_B^i) for the ith camera, and a corresponding pair of edge maps (E_A^j, E_B^j) for the jth camera, the event $|d_{A,K}^i - d_{B,K}^i|$ being small is independent of the event $|d_{A,K}^j - d_{B,K}^j|$ being small. Therefore the joint probability/redundancy $u_{A,B}$ can be computed as

$$u_{A,B} = q_{A,B}^i * q_{A,B}^j , \tag{9}$$

where $q_{A,B}^i$ and $q_{A,B}^j$ are the redundancies of A and B for the ith camera and the jth camera, respectively. We define a dissimilarity matrix $J = \{v_{A,B}\}$ based on the joint redundancy, where $v_{A,B} = 1 - u_{A,B}$, and compute the exemplars using Kolmogorov clustering. By construction, the edge maps of the exemplars thus computed are representative in both edge sets.

7. Embedding and Classification

From the set of exemplar pairs $\mathcal{C} = \{(E_k^i, E_k^j) \,|\, k = 1, \cdots, O\}$, each edge map of a pair of query edge maps (X, Y) can be embedded into a vector space as following:

$$\begin{aligned} \mathbf{v}_X &= [d_{X,E_1^i}, d_{X,E_2^i}, \cdots, d_{X,E_O^i}] \,, \\ \mathbf{v}_Y &= [d_{Y,E_1^j}, d_{Y,E_2^j}, \cdots, d_{Y,E_O^j}] \,. \end{aligned} \tag{10}$$

This is Lipschitz embedding of the query edge map with respect to the exemplar edge maps of each camera. The basic assumption of the Lipschitz embedding is that two nearby points have similar distances to any third point. In general, this property does not hold for non-metric distance measures such as the one that we are using. However, it has been observed that in practice the cases that the triangle inequality is violated are rare and have limited impact on the accuracy of the embedding. Instead of using non-metric embedding method such as in [1], we choose to use directly the embedding vectors in (10) to form the final representation of the pair of query edge maps:

$$\mathbf{y} = [\mathbf{v}_X, \mathbf{v}_Y] \,, \tag{11}$$

where $\mathbf{y}$ is simply the concatenation of two embedding vectors $\mathbf{v}_X$ and $\mathbf{v}_Y$. It is important to note that the computation of $\mathbf{y}$ does not involve any direct matching between the two query edge images.

Given a training set $\mathcal{T} = \{(E_k^i, E_k^j, t_k) \,|\, k = 1, \cdots, T\}$, where $t_k = \{0, 1\}$ is the truth label for the pair of edge maps E_k^i and E_k^j, and T is the number of training samples. We compute the representation $\mathbf{y}_k$ for each sample in the set. We then use a probabilistic version of SVM [17] to train a classifier using the truthed representation set $\{(\mathbf{y}_k, t_k) \,|\, k = 1, \cdots, T\}$. Given a pair of query edge maps, the same-different probability is computed from the trained classifier.

8. Experiments

We collected three data sets of vehicle chips from 3 pairs of cameras for the experiments. These three cameras are selected from a multiple camera system running in real time under all weather conditions (except night) in the field. One of the key components of the system is our previous edge-based matching algorithm [14], which has proved to be very effective after extensive testing and evaluation in the real environment. In [14] we have also compared our previous approach with many existing edge-based methods and demonstrated that it outperforms these approaches under a variety of environmental conditions. We see our previous method as a representative approach of directly matching vehicles across cameras, and we will compare the proposed approach against it throughout the experiment section. We will refer to the previous algorithm as the *inter-camera method*, and the method proposed in this paper as the *intra-camera method*.

Each database contains about 100-300 pairs of vehicle chips manually labeled as the same objects. The first two databases cover typical situations for which our previous inter-camera matching algorithm has high error rates. The third database represents a situation for which our previous algorithm has good performance.

Two major results are given in this section. First, we demonstrate that the intra-camera method significantly outperforms the inter-camera method when the geometric and illumination changes between two cameras are large. Second, we show that the performance of the intra-camera method is only slightly worse than the inter-camera method when the geometric and illumination changes between camera are minimal. Intermediate results on exemplar selection, and its impact on the overall performance are also presented with detailed discussions.

8.1. Experiment on Vehicles with Large Illumination Changes

Figure 4. Sample vehicle images with large illumination changes between two cameras. Top row is from camera #3, and bottom row is from camera #1

The first experiment tests the performance of the proposed method under large illumination changes between a pair of cameras. Vehicles shown in Fig. 4 are selected from the observations of camera #3 and camera #1 recorded from 3:30 pm to 4:00 pm on a sunny afternoon. Because of glare, self shadows, and the shadows cast by trees, the appearances of the corresponding vehicles look very different, and a lot of details important for vehicle matching cannot be used. There are 132 pairs of vehicles labeled as the same objects. For the intra-camera method, we randomly select 82 pairs of vehicles, from which we select 15 exemplar pairs. We then used the rest of the 82 vehicle pairs to construct a training data set with 67 true matches and 134 false matches, and a test data set with 50 true matches and 100 false matches. For both training and test data sets, the false matches are randomly generated. The same sets of training data and test data are used for the experiments with the inter-camera method. The results are shown in Table 3. It can be seen that the error rates of the intra-camera method are 10-11% lower than the inter-camera method.

Error rate	Intra-camera	Inter-camera
Training	0.18	0.29
Test	0.22	0.32

Table 1. Training and test error rates for matching vehicle objects under large illumination changes. The proposed intra-camera approach outperforms the inter-camera approach by 10-11%

8.2. Experiment on Vehicles with Large Perspective Changes

The second experiment tests the performance of the proposed method under large perspective changes between a pair of cameras. Vehicles shown in Fig. 5 are selected from the observations of camera #43 and camera #42 recorded around 4:30 pm on a cloudy afternoon. Camera #43 is on the near side of the road, while camera #42 is on the far side of the road. Therefore the perspective changes of the vehicle objects between the two cameras are large. From 4:30 pm to 5:00 pm we collected 105 pairs of vehicles and labeled them as the same objects. For the intra-camera method, we randomly select 70 pairs of vehicles, from which we select 15 exemplar pairs. We use the rest of the 70 vehicle pairs to construct a training data set with 55 true matches and 110 false matches, and a test data set with 50 true matches and 100 false matches. For both training and test data sets, the false matches are randomly generated. The same sets of training data and test data are used for the experiments with the inter-camera method. The results are shown in Table 2. It can be seen from Table 2 that the error rates of the intra-camera method are 16-19% lower than the inter-camera method.

Figure 5. Sample vehicle images with large perspective changes between two cameras. Top row is from camera #43, bottom row is from camera #42

Error rate	Intra-camera	Inter-camera
Training	0.096	0.260
Test	0.118	0.310

Table 2. Training and test error rates for matching vehicle objects under large perspective changes. The proposed intra-camera approach outperforms the inter-camera approach by 16-19%

Figure 6. Sample vehicle images with small changes between two cameras. Top row is from camera #1, bottom row is from camera #3

8.3. Experiment on Vehicles with Small Changes

In the previous two experiments we see that the performance of the proposed method is better when the shape and illumination changes are large between two cameras. It is natural to ask the question whether it can outperform the inter-camera matching method even when the changes are small.

In this experiment, we choose a set of data recorded from camera #3 and camera #1 on a cloudy morning from 10:30am to 11:00 am. This is one of the best conditions for the inter-camera matching algorithm, because both the perspective and illumination changes between the two cameras are minor. Figure 6 shows some samples vehicles from this data set, where 192 pairs of vehicles are labeled as the same objects. For the intra-camera method, we randomly select 112 pairs of vehicles, from which we select 15 exemplar pairs. We use the rest of the 112 vehicle pairs to construct a training data set with 97 true matches and 194 false matches, and a test data set with 80 true matches and 160 false matches. For both training and test data sets, the false matches are randomly generated. The same sets of training data and test data are used for the experiments with the inter-camera method. The results are shown in Table 3. It can be seen from Table 3 that the error rates of the intra-camera method are 3.6% higher than the inter-camera method.

The precise reason for this behavior needs further investigation. It seems that if inter-camera matching is possible, then direct matching exploits fine scale similarity and dif-

Error rate	Intra-camera	Inter-camera
Training	0.068	0.032
Test	0.086	0.050

Table 3. Training and test error rates for vehicle objects under small changes between cameras. The error rates of the proposed approach are 3.6% higher than the inter-camera approach

ference well. However when direct matching is not possible due to large differences, intra-camera embedding and classification may be the best that can be achieved.

8.4. Impact of Exemplar Selection Methods

Approaches	Redundancy	Random	Distance
Training	0.068	0.121	0.084
Test	0.147	0.226	0.235

Table 4. Training and test error rates using different approaches for exemplar selection. The error rates for the random selection approach are the average of 10 independent runs. All experiments in the table use the same number of 15 exemplars

In this experiment we study the impact of the exemplar selection methods. For the same data set as in Section 8.2, we applied three methods for exemplar selection and list the error rates of each method in Table 4. Note here that the embedding and classification approach is still the same as described in Section 7. The *redundancy* approach uses the joint redundancy measure described in (9). From the table, this approach has the lowest training and test error rates. The *random* approach selects exemplars randomly, and the results shown in Table 4 are the average of 10 independent runs. The *distance* approach combines the distance measure from two corresponding dissimilarity matrices in a way similar to the joint redundancy measure, but uses the robust distance measure instead. Surprisingly, the test error for this approach is even higher than the random approach.

Figure 7 shows the 15 pairs of exemplars generated as the output of the redundancy approach. One interesting observation is that there is only one sedan in the exemplar list, despite the fact that about 40-50% of the vehicles in the data set are sedans. Figure 8 shows some intermediate results of the clustering algorithm. The left dendrogram shows the clustering based on the joint distance measure as described above, while the right dendrogram shows the clustering based on the joint redundancy measure. It can

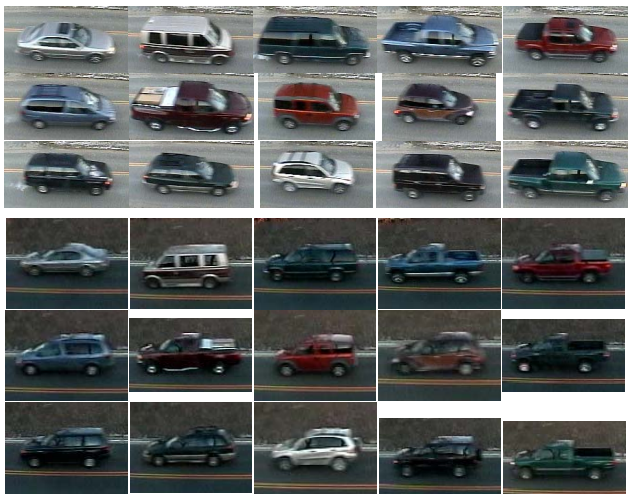

Figure 7. Pairs of exemplars generated using the joint redundancy measure. The top 15 vehicles (the first three rows) are selected from camera #43, and the bottom 15 vehicles are selected from camera #42. The top left vehicle corresponds to the left most vehicle on the fourth row, and other vehicles correspond each other accordingly

be seen from the dendrograms that the redundancy measure generates more meaniful clusters. This partially explains the reason why the error rates in Table 4 for the redundancy method is lower than the distance approach.

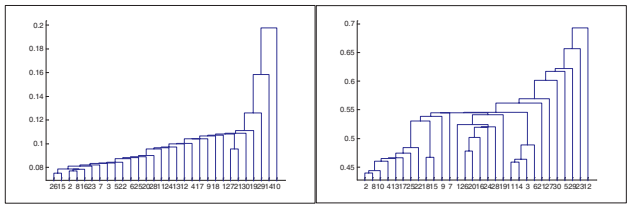

Figure 8. Dendrograms computed from both distance-based (left) and redundancy-based (right) dissimilarity matrices

8.5. Impact of the Number of Exemplars

In this experiment we study the impact of the number of exemplars on the performance of the proposed method. For the same data set as in Section 8.2, we vary the number of exemplars from 5 to 20, and report the error rates in Table 5. As expected, the error rates, especially the test error rate increases as the number of exemplars decreases. It is inter-

esting to see that when the number of exemplar increases to 20 the test error rate increases significantly. This may have something to do with the negative effect of increasing the dimensionally of the embedding features.

Number	5	10	15	20
Training	0.103	0.15	0.068	0.09
Test	0.31	0.167	0.147	0.262

Table 5. Training and test error rates for different number of exemplars. Using 15 exemplars generates the lowest error rates

9. Conclusion and Future Work

The major contribution of this paper is the idea of computing the same-different probabilities without directly matching vehicle objects across non-overlapping cameras. We have exploited this idea with the proposed approach that computes intra-camera non-metric embedding, and inter-camera learning and classification. We have demonstrated the strength of this idea with substantial amount of data collected from a real system under various environmental conditions.

The proposed approach has laid out a framework for many future extensions and improvements. Since our approach requires only intra-camera matching, other robust image features such as SIFT [11] that are more discriminating but may be sensitive to illumination changes can also be applied. Given the embedding vectors in (11), it is also possible to try other classifiers instead of SVM. It will be worthwhile to replace the embedding part of the current approach with other methods such as [1] that explicitly deal with non-metric embeddings. The current approach separates the exemplar selection, embedding, and learning into three individual processes. It will be interesting to see if there is an approach that integrates these three steps for a more optimal solution of the problem. Also, the current approach requires manual labeling of certain amount of data. An interesting but hard problem is whether we can remove this restriction and use unsupervised learning. Finally, it will be informative to combine the embedding vectors in (11) with the inter-camera features such as in [14] to see if the overall performance can be improved.

References

[1] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR04)*, pages 268–275, 2004.

[2] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR03)*, page 432, 2003.

[3] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *International Conference on Computer Vision (ICCV99)*, pages 87–93, 1999.

[4] K. Grauman and T. Darrell. Fast contour matching using approximate Earth Mover's Distance. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR04)*, volume I, pages 220–227, 2004.

[5] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 9(15):850–863, 1993.

[6] P. Indyk and N. Thaper. Fast image retrieval via embeddings. In *International Conference on Computer Vision (ICCV03)*, 2003.

[7] D. Jacobs, D. Weinshall, and Y. Gdalyahu. Class representation and image retrieval with non-metric distances. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 22(6):583–600, 2000.

[8] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. In *International Conference on Computer Vision (ICCV03)*, volume 2, pages 952–957, 2003.

[9] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR99)*, 1999.

[10] D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision (IJCV)*, 10(3):257–281, 1993.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[12] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Trans. Image Processing*, 6(1):103–113, 1997.

[13] H. Pasula, S. J. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *International Joint Conferences on Artificial Intelligence (IJCAI99)*, pages 1160–1171, 1999.

[14] Y. Shan, H. S. Sawhney, and R. Kumar. Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR05)*, 2005.

[15] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Hand pose estimation using hierarchical detection. In *International Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, vol. 3058, Springer (2004)*, 2004.

[16] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):733–742, 1997.

[17] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 2004.