

2017-03-01

PRISM: Person Reidentification via Structured Matching

Z. Zhang, V. Saligrama. 2017. "PRISM: Person Reidentification via Structured Matching." IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, Volume 27, Issue 3, pp. 499 - 512 (14). <https://doi.org/10.1109/TCSVT.2016.2596159>

<https://hdl.handle.net/2144/44090>

"Downloaded from OpenBU. Boston University's institutional repository."

PRISM: Person Re-Identification via Structured Matching

Ziming Zhang, and Venkatesh Saligrama, *Member, IEEE*

Abstract—Person re-identification (re-id), an emerging problem in visual surveillance, deals with maintaining entities of individuals whilst they traverse various locations surveilled by a camera network. From a visual perspective re-id is challenging due to significant changes in visual appearance of individuals in cameras with different pose, illumination and calibration. Globally the challenge arises from the need to maintain structurally consistent matches among all the individual entities across different camera views. We propose PRISM, a structured matching method to jointly account for these challenges. We view the global problem as a weighted graph matching problem and estimate edge weights by learning to predict them based on the co-occurrences of visual patterns in the training examples. These co-occurrence based scores in turn account for appearance changes by inferring likely and unlikely visual co-occurrences appearing in training instances. We implement PRISM on single shot and multi-shot scenarios. PRISM uniformly outperforms state-of-the-art in terms of matching rate while being computationally efficient.

Index Terms—Person Re-identification, Structured Matching, Visual Co-occurrences, Single/Multi-Shot

1 INTRODUCTION

Many surveillance systems require autonomous long-term behavior monitoring of pedestrians within a large camera network. One of the key issues in this task is *person re-identification (re-id)*, which deals with as to how to maintain entities of individuals as they traverse through diverse locations that are surveilled by different cameras with non-overlapping camera views. As in the literature, in this paper we focus on finding entity matches between two cameras.

Re-id presents several challenges. From a vision perspective, camera views are non-overlapping and so conventional tracking methods are not helpful. Variation in appearance between the two camera views is so significant — due to the arbitrary change in view angles, poses, illumination and calibration — that features seen in one camera are often missing in the other. Low resolution of images for re-id makes biometrics based approaches often unreliable [1]. Globally, the issue is that only a subset of individuals identified in one camera (location) may appear in the other.

We propose, PRISM, a structured matching method for re-id. PRISM is a *weighted bipartite matching* method that simultaneously identifies potential matches between individuals viewed in two different cameras. Fig. 1(a) illustrates re-id with two camera views, where 4 images labeled by green form the so-called probe set, and 4 entities labeled by red form the so-called gallery set. Graph matching requires edge weights, which correspond to similarity between entities viewed from two different cameras.

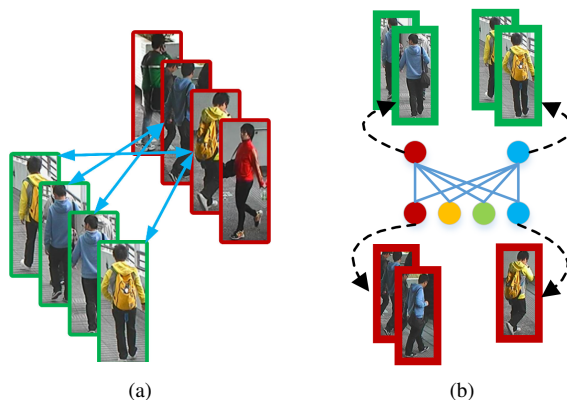


Fig. 1. (a) Illustration of re-id, where color red and green label the images from two different camera views, and arrows indicate entity matches. (b) Illustration of the weighted bipartite graph matching problem for (a), where each row denotes a camera view, each node denotes a person entity, different colors denote different entities, and the edges are weighted by 0 or 1, indicating missing matches or same entities. Each entity per view can be associated with single or multiple images.

We learn to estimate edge weights from training instances of manually labeled image pairs. We formulate the problem as an instance of structured learning [2] problem. While structured learning has been employed for matching text documents, re-id poses new challenges. Edge weights are obtained as a weighted linear combination of basis functions. For texts these basis functions encode shared or related words or patterns (which are assumed to be known a priori) between text documents. The weights for the basis functions are learned from training data. In this way during testing edge weights are scored based on a weighted combination of related words. In contrast, visual words (*i.e.* vector representations of appearance information, similar to

• Dr. Z. Zhang and Prof. V. Saligrama are currently with the Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215, US.

E-mail: zzhang14@bu.edu, srv@bu.edu



Fig. 2. Illustration of visual word co-occurrence in positive image pairs (*i.e.* two images from different camera views per column belong to a *same* person) and negative image pairs (*i.e.* two images from different camera views per column belong to *different* persons). For positive (or negative) pairs, in each row the enclosed regions are assigned the same visual word.

the words in texts) suffer from well known visual ambiguity and spatial distortion. This issue is further compounded in the re-id problem where visual words exhibit significant variations in appearance due to changes in pose, illumination, *etc.*

To handle the visual ambiguity and spatial distortion, we propose new basis functions based on co-occurrence of different visual words. We then estimate weights for different co-occurrences from their statistics in training data. While co-occurrence based statistics has been used in some other works [3], [4], [5], ours has a different purpose. We are largely motivated by the observation that the co-occurrence patterns of visual codewords behave similarly for images from different views. In other words, the transformation of target appearances can be statistically inferred through these co-occurrence patterns. As seen in Fig. 2, we observe that some regions are distributed similarly in images from different views and robustly in the presence of large cross-view variations. These regions provide important discriminant co-occurrence patterns for matching image pairs. For instance, statistically speaking, the first column of positive image pairs shows that “white” color in Camera 1 can change to “light blue” in Camera 2. However, “light blue” in Camera 1 can hardly change to “black” in Camera 2, as shown in the first column of negative image pairs.

In our previous work [6], we proposed a novel visual word co-occurrence model to capture such important patterns between images. We first encode images with a sufficiently large codebook to account for different visual patterns. Pixels are then matched into codewords or visual words, and the resulting spatial distribution for each codeword is embedded to a kernel space through *kernel mean embedding* [7] with latent-variable conditional densities [8] as kernels. The fact that we incorporate the spatial distribution of codewords into appearance models provides us with locality sensitive co-occurrence measures. Our

approach can be also interpreted as a means to *transfer* the information (*e.g.* pose, illumination, and appearance) in image pairs to a common latent space for meaningful comparison.

In this perspective appearance change corresponds to transformation of a visual word viewed in one camera into another visual word in another camera. Particularly, our method does not assume any smooth appearance transformation across different cameras. Instead, our method learns the visual word co-occurrence patterns statistically in different camera views to predict the identities of persons. The structured learning problem in our method is to determine important co-occurrences while being robust to noisy co-occurrences.

In summary, our main contributions of this paper are:

- We propose a new structured matching method to simultaneously identify matches between two cameras that can deal with both single-shot and multi-shot scenarios in a unified framework;
- We account for significant change in appearance design of new basis functions, which are based on visual word co-occurrences [6];
- We outperform the state-of-the-art significantly on several benchmark datasets, with good computational efficiency in testing.

1.1 Related Work

While re-id has received significant interest [1], [9], [10], much of this effort can be viewed as methods that seek to *classify* each probe image into one of gallery images. Broadly re-id literature can be categorized into two themes with one focusing on cleverly designing local features [6], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25] and the other focusing on metric learning [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38]. Typically local feature design aims to find a re-id specific representation based on the some

properties among the data in re-id, *e.g.* symmetry and centralization of pedestrians in images [13], color correspondences in images from different cameras [24], [23], spatial-temporal information in re-id videos/sequences [12], [14], discriminative image representation [6], [11], [17], viewpoint invariance prior [25]. Unlike these approaches that attempt to match local features our method attempts to learn changes in appearance or features to account for visual ambiguity and spatial distortion. On the other hand, metric learning aims to learn a better similarity measure using, for instance, transfer learning [29], dictionary learning [30], distance learning/comparison [31], [33], [34], similarity learning [35], dimension reduction [36], template matching [37], active learning [38]. In contrast to metric learning approaches that attempt to find a metric such that features from positively associated pairs are close in distance our learning algorithm learns similarity functions for imputing similarity between features that naturally undergo appearance changes.

Re-ID can also be organized based on so called single-shot or multi-shot scenarios. For *single-shot learning*, each entity is associated with only one single image, and re-id is performed based on every single image pair. In the literature, most of the methods are proposed under this scenario. For instance, Zhao *et al.* [39] proposed learning good discriminative mid-level filters for describing images. Yang *et al.* [23] proposed a saliency color based image descriptor and employed metric learning with these descriptors for re-id. For *multi-shot learning*, each entity is associated with at least one image, and re-id is performed based on multiple image pairs. How to utilize the redundant information in multiple images is the key difference from single-shot learning. Wu *et al.* [40] proposed a locality-constrained collaboratively regularized nearest point model to select images for generating decision boundaries between different entities, which are represented as sets of points in the feature space. Bazzani *et al.* [41] propose a new image representation by focusing on the overall chromatic content and the presence of recurrent local patches.

Our work in contrast deals with these different scenarios within one framework. In addition we allow for no matches for some entities and can handle cases where the numbers of entities in both probe and gallery sets are different. Meanwhile, our basis function can handle both single-shot and multi-shot learning directly while accounting for appearance changes.

While special cases of our method bears similarity to Locally-adaptive Decision Functions (LADF) described in [42], they are fundamentally different. LADF proposes a second-order (*i.e.* quadratic) decision function based on metric learning. In contrast we compute similarities between entities and do not need to impose positive semidefinite conditions during training. Our method is also related to [43] where an integer optimization method was proposed to enforce network consistency in re-id during testing, *i.e.* maintaining consistency in re-id results across the network. For instance, a person A from camera view 1 matches a person B from view 2, and A also matches a person C

from view 3, then based on consistency B should match C as well. This network consistency helps improve the camera pairwise re-id performance between all the individual camera pairs. In contrast, graph-structure is integral to both training and testing in our proposed approach. We *learn-to-estimate* bipartite graph structures during testing by pruning the feasible solution space based on our a priori knowledge on correct matching structures. Recently, Paisitkriangkrai *et al.* [44] and Liu *et al.* [45] proposed utilizing structured learning to integrate the metric/color model ensembles, where structured learning is taken as a means to enhance the re-id performance of each individual model. In contrast, we consider structured learning as a way to learn the classifier, working with our own features for re-id.

To summarize our contributions, our method learns to assign weights to pairs of instances using globally known feasible assignments in training data. Unlike text data or other conventional approaches our weights incorporate appearance changes and spatial distortion. We express the weights as a linear combination of basis functions, which are the set of all feasible appearance changes (co-occurrences). Our decision function is a weighting function that weights different co-occurrences. During training, our structural constraints induce higher scores on ground-truth assignments over other feasible assignments. During testing, we enforce a globally feasible assignment based on our learned co-occurrence weights.

Very recently, open-world re-id [46], [47], [48] has been introduced, where persons in each camera may be only partially overlapping and the number of cameras, spatial size of the environment, and number of people may be unknown and at a significantly larger scale. Recall that the goal of this paper is to identify the persons given aligned images, which are the cases in most person re-identification benchmark datasets, while open-world re-id this is more a system level concept that must deal with issues such as person detection, tracking, re-id, data association, *etc.* Therefore, open-world re-id is out of scope of our current work.

Structured learning has been also used in the object tracking literature (*e.g.* [49]) for data association. The biggest difference, however, between our method and these tracking methods is that in our re-id cases, we do not have any temporal or location information with data, in general, which leads to totally different goals: our method aims to find the correct matches among the entities using structured matching in testing based on only the appearance information, while in tracking the algorithms aim to associate the same object with small appearance variations in two adjacent frames locally.

The rest of this paper is organized as follows: Section 2 explains our structured prediction method in detail. Section 3 lists some of our implementation details. Section 4 reports our experimental results on the benchmark datasets. We conclude the paper in Section 5.

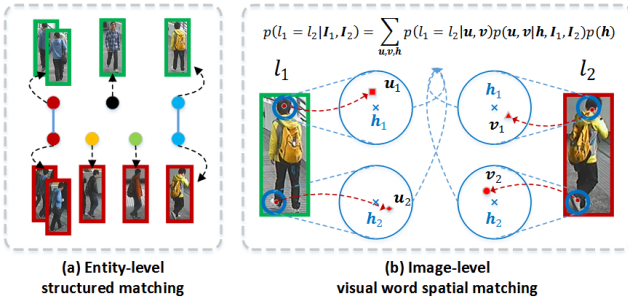


Fig. 3. Overview of our method, PRISM, consisting of two levels where (a) entity-level structured matching is imposed on top of (b) image-level visual word deformable matching. In (a), each color represents an entity, and this example illustrates the general situation for re-id, including single-shot, multi-shot, and no match scenarios. In (b), the idea of visual word co-occurrence for measuring image similarities is illustrated in a probabilistic way, where l_1, l_2 denote the person entities, u_1, u_2, v_1, v_2 denote different visual words, and h_1, h_2 denote two locations.

2 PRISM

In this paper we focus on two camera re-id problems, as is common in the literature. In the sequel we present an overview of our proposed method.

2.1 Overview

Let us first describe the problem we often encounter during testing. We are given N_1 probe entities (Camera 1) that are to be matched to N_2 gallery entities (Camera 2). Fig. 3 depicts a scenario where entities may be associated with a single image (single-shot), multiple images (multi-shot) and be unmatched to any other entity in the probe/gallery (e.g. “black”, “orange”, and “green” entities in (a)). Existing methods could fail here for the reason that entities are matched independently based on pairwise similarities between the probes and galleries leading to the possibility of matching multiple probes to the same entity in the gallery. Structured matching is a framework that can address some of these issues.

To build intuition, consider \bar{y}_{ij} as a binary variable denoting whether or not there is a match between i^{th} probe entity and j^{th} gallery entity, and s_{ij} as their similarity score. Our goal is to predict the structure, $\bar{\mathbf{y}}$, by seeking a maximum bipartite matching:

$$\max_{\forall i, \forall j, \bar{y}_{ij} \in \{0,1\}} \sum_{i,j} \bar{y}_{ij} s_{ij}, \text{ s.t. } \bar{\mathbf{y}} = [\bar{y}_{ij}]_{\forall i, \forall j} \in \mathcal{Y} \quad (1)$$

where \mathcal{Y} could be the sub-collection of bipartite graphs accounting for different types of constraints. For instance, $\mathcal{Y} = \{\bar{\mathbf{y}} \mid \forall i, \sum_j \bar{y}_{ij} \leq r_i, \forall j, \sum_i \bar{y}_{ij} \leq g_j\}$ would account for the relaxed constraint to identify at most r_i potential matches from the gallery set for probe i , and at most g_j potential matches from the probe set for gallery j . Hopefully the correct matches are among them.

Learning Similarity Functions: Eq. 1 needs similarity score s_{ij} for every pair of probe i and gallery j , which is a priori unknown and could be arbitrary. Therefore, we

seek similarity models that can be learned from training data based on minimizing some *loss function*.

Structured learning [2] formalizes loss functions for learning similarity models that are consistent with testing goals as in Eq. 1. To build intuition, consider the example of text documents, where each document is a collection of words chosen from a dictionary \mathcal{V} . Let $\mathcal{D}_i, \mathcal{D}_j$ be documents associated with probe i and gallery j . Let \mathcal{D} denote the tuple of all training probe and gallery documents. A natural similarity model is one based on *shared-words* in the two documents, namely, $s_{ij} = \sum_{v \in \mathcal{V}} w_v \mathbf{1}_{\{v \in \mathcal{D}_i \cap v \in \mathcal{D}_j\}} \cdot w_v$ denotes the importance of word v in matching any two arbitrary documents. The learning problem reduces to learning the weights w_v for each word from training instances that minimizes some loss function. A natural loss function is one that reflects our objectives in testing. In particular, substituting this similarity model in Eq. 1, we obtain $\sum_{i,j} \bar{y}_{ij} s_{ij} = \sum_{v \in \mathcal{V}} w_v \sum_{i,j} \bar{y}_{ij} \mathbf{1}_{\{v \in \mathcal{D}_i \cap v \in \mathcal{D}_j\}}$. We denote as $f_v(\mathcal{D}, \bar{\mathbf{y}}) = \sum_{i,j} \bar{y}_{ij} \mathbf{1}_{\{v \in \mathcal{D}_i \cap v \in \mathcal{D}_j\}}$ the *basis function* associated with word v . It measures the frequency with which word v appears in matched training instances. A loss function must try to ensure that,

$$\sum_{v \in \mathcal{V}} w_v f_v(\mathcal{D}, \mathbf{y}) \geq \sum_{v \in \mathcal{V}} w_v f_v(\mathcal{D}, \bar{\mathbf{y}}), \forall \bar{\mathbf{y}} \in \mathcal{Y} \quad (2)$$

where $\bar{\mathbf{y}}$ is any bipartite matching and \mathbf{y} is the ground-truth bipartite matching. Hinge losses can be used to penalize violations of Eq. 2. Note that such loss functions only constrain the weights so that they perform better only on alternative bipartite matchings, rather than any arbitrary $\bar{\mathbf{y}}$.

Similarity Models for Re-ID are more complex relative to the example above. First, we typically have images and need a way to encode images into visual words. Second, visual words are not typically shared even among matched entities. Indeed a key challenge here is to account for significant visual ambiguity and spatial distortion, due to the large variation in appearance of people from different camera views.

We propose similarity models based on cross-view visual word co-occurrence patterns. **Our key insight is that aspects of appearance that are transformed in predictable ways, due to the static camera view angles, can be statistically inferred through pairwise co-occurrence of visual words.** In this way, we allow the same visual concepts to be mapped into different visual words, and account for visual ambiguity.

We present a probabilistic approach to motivate our similarity model in Fig. 3(b). We let the similarity s_{ij} be equal to the probability that two entities are identical, i.e. ,

$$\begin{aligned} s_{ij} &\triangleq p(\bar{y}_{ij} = 1 | \mathbf{I}_i^{(1)}, \mathbf{I}_j^{(2)}) \\ &= \sum_{\mathbf{u} \in \mathcal{U}, \mathbf{v} \in \mathcal{V}, \mathbf{h} \in \Pi} p(\bar{y}_{ij} = 1 | \mathbf{u}, \mathbf{v}) p(\mathbf{u}, \mathbf{v} | \mathbf{h}, \mathbf{I}_i^{(1)}, \mathbf{I}_j^{(2)}) p(\mathbf{h}) \\ &= \sum_{\mathbf{u} \in \mathcal{U}, \mathbf{v} \in \mathcal{V}} p(\bar{y}_{ij} = 1 | \mathbf{u}, \mathbf{v}) \left[\sum_{\mathbf{h} \in \Pi} p(\mathbf{u}, \mathbf{v} | \mathbf{h}, \mathbf{I}_i^{(1)}, \mathbf{I}_j^{(2)}) p(\mathbf{h}) \right], \end{aligned} \quad (3)$$

where $\mathbf{I}_i^{(1)}, \mathbf{I}_j^{(2)}$ denote two images from camera view 1 (left) and 2 (right), respectively, $\mathbf{u} \in \mathcal{U}, \mathbf{v} \in \mathcal{V}$ denote the visual words for view 1 and view 2, and $\mathbf{h} \in \Pi$ denotes the shared spatial locations.

Following along the lines of the text-document setting we can analogously let $w_{uv} = p(\bar{y}_{ij} = 1 | \mathbf{u}, \mathbf{v})$ denote the likelihood (or importance) of co-occurrence of the two visual words among matched documents. This term is data-independent and must be learned from training instances as before. The basis function, $f_{uv}(\cdot)$ is given by $\sum_{\mathbf{h} \in \Pi} p(\mathbf{u}, \mathbf{v} | \mathbf{h}, \mathbf{I}_i^{(1)}, \mathbf{I}_j^{(2)}) p(\mathbf{h})$ and must be *empirically* estimated. The basis function $f_{uv}(\cdot)$ measures the frequency with which two visual words co-occur after accounting for spatial proximity. The term $p(\mathbf{u}, \mathbf{v} | \mathbf{h}, \mathbf{I}_i^{(1)}, \mathbf{I}_j^{(2)})$ here denotes the joint contribution of the visual words at location \mathbf{h} . To handle spatial distortion of visual words, we allow the visual words to be deformable, similar to deformable part model [50], when calculating their joint contribution. $p(\mathbf{h})$ denotes the importance of location \mathbf{h} for prediction.

In summary, our similarity model handles both visual ambiguity (through co-occurring visual words) and spatial distortion simultaneously. We learn parameters, w_{uv} , of our similarity model along the lines of Eq. 2 with analogous structured loss functions that penalize deviations of predicted graph structures from ground-truth annotated graph structures. In the following sections we present more details of the different components of our proposed approach.

2.2 Structured Matching of Entities in Testing

Now let us consider the re-id problem as a bipartite graph matching problem, where all the entities are represented as the nodes in the graph, forming two sets of nodes for the probe set and the gallery set, respectively, and the matching relations are represented as the edges with weights from $\{0, 1\}$, as illustrated in Fig. 3(a).

The **key insight** of our structured matching in testing is to narrow down the feasible solution space for structured prediction in weighted bipartite graph matching based on the prior knowledge on correct matching structures.

During training, since the bipartite graph can be defined based on the training data, the degree of each node can be easily calculated. But during testing we have to predict the degree of each node. Usually the node degrees in the probe can be given beforehand. For instance, we would like to find *at most* r_i entity matches in the gallery set for entity i in the probe set so that hopefully the correct match is among them, then the degree of node i in the graph is r_i . However, this is not the case for the nodes in the gallery.

Therefore, without any prior on the graph structure during testing, we enforce it to have the following structural properties, which are very reasonable and practical:

- (1) All the entities in either gallery or probe set are *different* from each other, and every test entity i in the probe can be *equally* matched with any entity in the gallery. It turns out that in this way we actually maximize the matching likelihood for the test entity i .

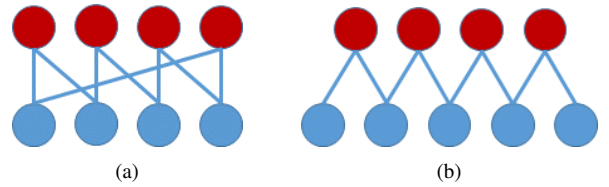


Fig. 4. Illustration of our predicted bipartite matching graphs, where red and blue nodes represent the probe and gallery sets, respectively. Both graphs in (a) and (b) are examples which satisfy the conditions (1) and (2) during testing.

- (2) We constrain the nodes in the gallery set to have *similar* degrees. This helps avoid the mismatched cases such as multiple entities in the probe being matched with the same entity in the gallery.

We illustrate two examples for our predicted graphs during testing that satisfy the both conditions in Fig. 4, and we would like to find at most $r_i = 2$ matches in the gallery for each probe. Then the red node degrees can be no large than 2. Accordingly, the total degree in the gallery set, where each node degree needs to be similar to others, should be the same as that in the probe set. By minimizing the entropy of the node degrees in the gallery, we can easily calculate the upper bound of the gallery node degrees in (a) as $\lceil \frac{4 \times 2}{4} \rceil = 2$ and in (b) as $\lceil \frac{4 \times 2}{5} \rceil = 2$, respectively.

By incorporating these node degree upper bounds, we can narrow down the feasible solution space for correct matching structures from $\{0, 1\}^{N_1 \times N_2}$, where N_1 and N_2 denote the numbers of nodes in the bipartite graph from view 1 (probe) and view 2 (gallery), to \mathcal{Y} such that

$$\mathcal{Y} = \left\{ \mathbf{y} \mid \forall i, \forall j, y_{ij} \in \{0, 1\}, \sum_j y_{ij} \leq r_i, \sum_i y_{ij} \leq \left\lceil \frac{\sum_i r_i}{N_2} \right\rceil \right\}, \quad (4)$$

where $\forall i, r_i$ denotes the predefined degree for node i in the probe, and $\lceil \cdot \rceil$ denotes the ceiling function. As we see, r_i and $\left\lceil \frac{\sum_i r_i}{N_2} \right\rceil$ are used to control the node degrees in the probe and gallery, respectively, and $\left\lceil \frac{\sum_i r_i}{N_2} \right\rceil$ enforces the gallery node degrees to be similar to each other.

Then we can formulate our structured matching, *i.e.* weighted bipartite graph matching, for re-id *during testing* as follows:

$$\mathbf{y}^* = \arg \max_{\bar{\mathbf{y}} \in \mathcal{Y}} \mathbf{w}^T f(\mathcal{X}, \bar{\mathbf{y}}) = \arg \max_{\bar{\mathbf{y}} \in \mathcal{Y}} \left\{ \sum_{i,j} \bar{y}_{ij} \mathbf{w}^T \phi(\mathbf{x}_{ij}) \right\}, \quad (5)$$

where $\mathbf{x}_{ij} \in \mathcal{X}$ denotes an entity pair between entity i in the probe and entity j in the gallery, $\phi(\cdot)$ denotes the *similarity measure function*, \mathbf{w} denotes the weight vector for measuring entity similarities, $(\cdot)^T$ denotes the matrix transpose operator, $\bar{\mathbf{y}} \in \mathcal{Y}$ denotes a matching structure from the structure set \mathcal{Y} , and \mathbf{y}^* denotes the predicted matching

structure for re-id. Note that $f(\mathcal{X}, \bar{\mathbf{y}}) = \sum_{i,j} \bar{y}_{ij} \phi(\mathbf{x}_{ij})$ is our *basis function* for re-id.

Functionally, \mathbf{w} and $\phi(\mathbf{x}_{ij})$ in Eq. 5 stand for $p(\bar{y}_{ij} = 1|\mathbf{u}, \mathbf{v})$ and $\sum_{\mathbf{h} \in \Pi} p(\mathbf{u}, \mathbf{v}|\mathbf{h}, \mathbf{I}_i^{(1)}, \mathbf{I}_j^{(2)})p(\mathbf{h})$ in Eq. 3, respectively. $\forall i, \forall j, \mathbf{w}^T \phi(\mathbf{x}_{ij})$ defines the edge weight between node i and node j in the bipartite graph. Our method learns the 0/1 assignments for the edges under the structural conditions, so that the total weight over the bipartite graph is maximized. Given these edge weights, we can utilize linear programming to solve Eq. 5, and then threshold the solution to return the 0/1 assignments. Notice that our structured matching can handle the general entity matching problem as illustrated in Fig. 3(a), which is different from conventional re-id methods.

2.3 Similarity Models

Now we come to the question of as to how we define our similarity measure function $\phi(\cdot)$ in Eq. 5. Recall that our method has to deal with (1) single-shot learning, (2) multi-shot learning, (3) visual ambiguity, and (4) spatial distortion. Following Fig. 3(b), we define $\phi(\cdot)$ based on the cross-view visual word co-occurrence patterns.

2.3.1 Locally Sensitive Co-occurrence [6]

We need co-occurrence models that not only account for the locality of appearance changes but also the random spatial and visual ambiguity inherent in vision problems. Recall that we have two codebooks $\mathcal{U} = \{\mathbf{u}\}$ and $\mathcal{V} = \{\mathbf{v}\}$ for view 1 and view 2, respectively. Our codebook construction is global and thus only carries information about distinctive visual patterns. Nevertheless, for a sufficiently large codebook distinctive visual patterns are mapped to different elements of the codebook, which has the effect of preserving local visual patterns. Specifically, we map each pixel at 2D location $\boldsymbol{\pi} \in \Pi$ of image \mathbf{I} in a view into one codeword to cluster these pixels.

To emphasize local appearance changes, we look at the spatial distribution of each codeword. Concretely, we let $\Pi_u = \mathcal{C}(\mathbf{I}, \mathbf{u}) \subseteq \Pi$ (*resp.* $\Pi_v = \mathcal{C}(\mathbf{I}, \mathbf{v}) \subseteq \Pi$) denote the set of pixel locations associated with codeword \mathbf{u} (*resp.* \mathbf{v}) in image \mathbf{I} and associate a spatial probability distribution, $p(\boldsymbol{\pi}|\mathbf{u}, \mathbf{I})$ (*resp.* $p(\boldsymbol{\pi}|\mathbf{v}, \mathbf{I})$), over this observed collection. In this way visual words are embedded into a family of spatial distributions. Intuitively it should now be clear that we can use the similarity (or distance) of two corresponding spatial distributions to quantify the pairwise relationship between two visual words. This makes sense because our visual words are spatially locally distributed and small distance between spatial distributions implies spatial locality. Together this leads to a model that accounts for local appearance changes.

While we can quantify the similarity between two distributions in a number of ways, the kernel mean embedding [7] method is particularly convenient for our task. The basic idea to map the distribution, p , into a reproducing kernel Hilbert space (RKHS), \mathcal{H} , namely, $p \rightarrow \mu_p(\cdot) = \sum K(\cdot, \boldsymbol{\pi})p(\boldsymbol{\pi}) \triangleq E_p(K(\cdot, \boldsymbol{\pi}))$. For universal kernels,

such as RBF kernels, this mapping is injective, *i.e.*, the mapping preserves the information about the distribution [7]. In addition we can exploit the reproducing property to express inner products in terms of expected values, namely, $\langle \mu_p, \Phi \rangle = E_p(\Phi)$, $\forall \Phi \in \mathcal{H}$ and obtain simple expressions for similarity between two distributions (and hence two visual words) because $\mu_p(\cdot) \in \mathcal{H}$.

To this end, consider the codeword $\mathbf{u} \in \mathcal{U}$ in image $\mathbf{I}_i^{(1)}$ and the codeword $\mathbf{v} \in \mathcal{V}$ in image $\mathbf{I}_j^{(2)}$. The co-occurrence matrix (and hence the appearance model) is the inner product of visual words in the RKHS space, namely,

$$\begin{aligned} [\phi(\mathbf{x}_{ij})]_{uv} &= \langle \mu_{p(\cdot|\mathbf{u}, \mathbf{I}_i^{(1)})}, \mu_{p(\cdot|\mathbf{v}, \mathbf{I}_j^{(2)})} \rangle \\ &= \sum_{\boldsymbol{\pi}_u \in \Pi} \sum_{\boldsymbol{\pi}_v \in \Pi} K(\boldsymbol{\pi}_u, \boldsymbol{\pi}_v) p(\boldsymbol{\pi}_u|\mathbf{u}, \mathbf{I}_i^{(1)}) p(\boldsymbol{\pi}_v|\mathbf{v}, \mathbf{I}_j^{(2)}), \end{aligned} \quad (6)$$

where we use the reproducing property in the last equality and $[\cdot]_{uv}$ denotes the entry in $\phi(\mathbf{x}_{ij})$ for the codeword pair (\mathbf{u}, \mathbf{v}) .

Particularly, in [6] we proposed a *latent spatial kernel*. This is a type of probability product kernel that has been previously proposed [8] to encode generative structures into discriminative learning methods. In our context we can view the presence of a codeword \mathbf{u} at location $\boldsymbol{\pi}_u$ as a noisy displacement of a true latent location $\mathbf{h} \in \Pi$. The key insight here is that the spatial activation of the two codewords \mathbf{u} and \mathbf{v} in the two image views $\mathbf{I}_i^{(1)}$ and $\mathbf{I}_j^{(2)}$ are conditionally independent when conditioned on the true latent location \mathbf{h} , namely, the joint probability factorizes into $p\{\boldsymbol{\pi}_u, \boldsymbol{\pi}_v|\mathbf{h}, \mathbf{I}_i^{(1)}, \mathbf{I}_j^{(2)}\} = p\{\boldsymbol{\pi}_u|\mathbf{h}, \mathbf{I}_i^{(1)}\} p\{\boldsymbol{\pi}_v|\mathbf{h}, \mathbf{I}_j^{(2)}\}$. We denote the noisy displacement likelihoods, $p\{\boldsymbol{\pi}_u|\mathbf{h}, \mathbf{I}_i^{(1)}\} = \kappa(\boldsymbol{\pi}_u, \mathbf{h})$ and $p\{\boldsymbol{\pi}_v|\mathbf{h}, \mathbf{I}_j^{(2)}\} = \kappa(\boldsymbol{\pi}_v, \mathbf{h})$ for simplicity. This leads us to $K(\boldsymbol{\pi}_u, \boldsymbol{\pi}_v) = \sum_{\mathbf{h}} \kappa(\boldsymbol{\pi}_u, \mathbf{h}) \kappa(\boldsymbol{\pi}_v, \mathbf{h}) p(\mathbf{h})$, where $p(\mathbf{h})$ denotes the spatial probability at \mathbf{h} . By plugging this new K into Eq. 6, we have

$$\begin{aligned} [\phi(\mathbf{x}_{ij})]_{uv} &= \sum_{\boldsymbol{\pi}_u \in \Pi} \sum_{\boldsymbol{\pi}_v \in \Pi} \sum_{\mathbf{h} \in \Pi} \kappa(\boldsymbol{\pi}_u, \mathbf{h}) \kappa(\boldsymbol{\pi}_v, \mathbf{h}) p(\mathbf{h}) \\ &\quad \cdot p(\boldsymbol{\pi}_u|\mathbf{u}, \mathbf{I}_i^{(1)}) p(\boldsymbol{\pi}_v|\mathbf{v}, \mathbf{I}_j^{(2)}) \\ &\leq \sum_{\mathbf{h}} \max_{\boldsymbol{\pi}_u} \left\{ \kappa(\boldsymbol{\pi}_u, \mathbf{h}) p(\boldsymbol{\pi}_u|\mathbf{u}, \mathbf{I}_i^{(1)}) \right\} \\ &\quad \cdot \max_{\boldsymbol{\pi}_v} \left\{ \kappa(\boldsymbol{\pi}_v, \mathbf{h}) p(\boldsymbol{\pi}_v|\mathbf{v}, \mathbf{I}_j^{(2)}) \right\} p(\mathbf{h}), \end{aligned} \quad (7)$$

where the inequality follows by rearranging the summations and standard upper bounding techniques. Here we use an upper bound for computational efficiency, and assume that $p(\mathbf{h})$ is a uniform distribution for simplicity without further learning. The main idea here is that by introducing the latent displacement variables, we have a handle on view-specific distortions observed in the two cameras. Using different kernel functions κ , the upper bound in Eq. 7 results in different latent spatial kernel functions.

Fig. 5 illustrates the whole process of generating the latent spatial kernel based appearance model given the codeword images, each of which is represented as a collection of codeword slices. For each codeword slice, the max operation is performed at every pixel location to search for

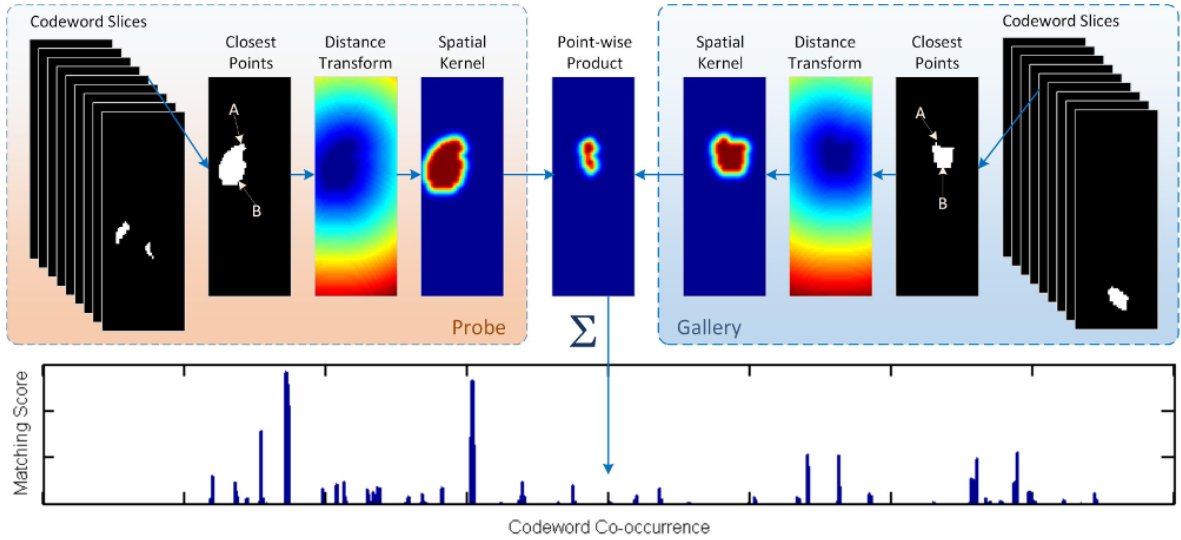


Fig. 5. Illustration of visual word co-occurrence model generation process in [6]. Here, the white regions in the codeword slices indicate the pixel locations with the same codeword. “A” and “B” denote two arbitrary pixel locations in the image domain. And “ Σ ” denotes a sum operation which sums up all the values in the point-wise product matrix into a single value $[\phi(\mathbf{x}_{ij})]_{uv}$ in the model.

the spatially closest codeword in the slice. This procedure forms a distance transform image, which is further mapped to a spatial kernel image. It allows each peak at the presence of a codeword to be propagated smoothly and uniformly. To calculate the matching score for a codeword co-occurrence, the spatial kernel from a probe image and another from a gallery image are multiplied element-wise and then summed over all latent locations. This step guarantees that our descriptor is insensitive to the noise data in the codeword images. This value is a single entry in the bin indexing the codeword co-occurrence in our descriptor for matching the probe and gallery images. As a result, we have generated a high dimensional sparse appearance descriptor. Note that we simplify the computation of this model by utilizing the indicator function for $p(\pi_u|\mathbf{u}, \mathbf{I}_i^{(1)})$ and $p(\pi_v|\mathbf{v}, \mathbf{I}_j^{(2)})$, respectively. Namely, $p(\pi_u|\mathbf{u}, \mathbf{I}_i^{(1)}) = 1$ (resp. $p(\pi_v|\mathbf{v}, \mathbf{I}_j^{(2)}) = 1$) if the pixel at location π_u (resp. π_v) in image $\mathbf{I}_i^{(1)}$ (resp. $\mathbf{I}_j^{(2)}$) is encoded by codeword \mathbf{u} (resp. \mathbf{v}); otherwise, 0.

2.3.2 Multi-Shot Visual Word Co-occurrence Models

By comparing the simplified model in Eq. 7 with Eq. 3, we can set

$$p(\mathbf{u}, \mathbf{v}|\mathbf{h}, \mathbf{I}_i^{(1)}, \mathbf{I}_j^{(2)}) \triangleq \max_{\pi_u \in \Pi_u} \kappa(\pi_u, \mathbf{h}) \cdot \max_{\pi_v \in \Pi_v} \kappa(\pi_v, \mathbf{h}), \quad (8)$$

where $\max_{\pi_u \in \Pi_u} \kappa(\pi_u, \mathbf{h})$ and $\max_{\pi_v \in \Pi_v} \kappa(\pi_v, \mathbf{h})$ can be computed independently once for comparing similarities, making the calculation much more efficient. This model cannot, however, handle the multi-shot scenario directly.

In this paper we extend the visual word co-occurrence model in [6] to the multi-shot scenario, and propose three more efficient spatial kernel functions for κ .

Let $\forall p, q, \mathcal{I}_q^{(p)} = \{\mathbf{I}_{m_q}^{(p)}\}_{m_q=1, \dots, N_q^{(p)}}$ be the image set with the same image resolution for entity q from view p ,

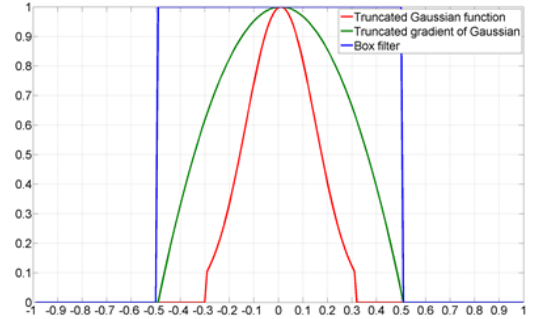


Fig. 6. Illustration of the three efficient spatial kernels for κ .

where $\forall m_q, \mathbf{I}_{m_q}^{(p)}$ denotes the $(m_q)^{th}$ image, and σ be the scale of patches centered at these locations. Then we can define our $\phi(\mathbf{x}_{ij})$ as follows:

$$[\phi(\mathbf{x}_{ij})]_{uv} = \sum_{\mathbf{h} \in \Pi} [\psi(\mathcal{I}_i^{(1)}, \mathbf{h}, \sigma)]_u \cdot [\psi(\mathcal{I}_j^{(2)}, \mathbf{h}, \sigma)]_v, \quad (9)$$

where ψ denotes the multi-shot visual word descriptor at location \mathbf{h} for each entity, and $[\cdot]_u$ (resp. $[\cdot]_v$) denotes the entry in the vector for \mathbf{u} (resp. \mathbf{v}) at location \mathbf{h} .

Next, we will take $[\psi(\mathcal{I}_i^{(1)}, \mathbf{h}, \sigma)]_u$ as an example to explain its definition, and accordingly $[\psi(\mathcal{I}_j^{(2)}, \mathbf{h}, \sigma)]_v$ can be defined similarly. Letting $\Pi_u(\mathbf{I}_{m_i}^{(1)})$ be the set of locations where pixels are encoded by visual word \mathbf{u} in image $\mathbf{I}_{m_i}^{(1)}$, based on Eq. 8 we define $[\psi(\mathcal{I}_i^{(1)}, \mathbf{h}, \sigma)]_u$ as follows:

$$[\psi(\mathcal{I}_i^{(1)}, \mathbf{h}, \sigma)]_u = \frac{1}{|\mathcal{I}_i^{(1)}|} \sum_{\mathbf{I}_{m_i}^{(1)} \in \mathcal{I}_i^{(1)}} \max_{\pi_u \in \Pi_u(\mathbf{I}_{m_i}^{(1)})} \kappa(\pi_u, \mathbf{h}, \sigma), \quad (10)$$

where $|\cdot|$ denotes the cardinality of a set, i.e. the number of images for person entity i from view 1, and σ is the

spatial kernel parameter controlling the locality. For multi-shot learning, we take each sequence as a collection of independent images, and utilize the average to represent the entity. Even though we use such simple representation (*e.g.* totally ignoring the temporal relations between images), it turns out that our method can outperform the current state-of-the-art significantly for the multi-shot scenario, as we will demonstrate in Section 4.4.

The choices for the spatial kernel κ in Eq. 10 are quite flexible. To account for computational efficiency, here we list three choices, *i.e.* (1) truncated Gaussian filters (κ_1), (2) truncated gradient of Gaussian filters (κ_2), and (3) box filters [51] (κ_3). Their definitions are shown below:

$$\kappa_1 = \begin{cases} \exp\left\{-\frac{\text{dist}(\boldsymbol{\pi}_s, \mathbf{h})}{\sigma_1}\right\}, & \text{if } \text{dist}(\boldsymbol{\pi}_s, \mathbf{h}) \leq \alpha \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

$$\kappa_2 = \max\left\{0, 1 - \frac{1}{\sigma_2} \cdot \text{dist}(\boldsymbol{\pi}_s, \mathbf{h})\right\}, \quad (12)$$

$$\kappa_3 = \begin{cases} 1, & \text{if } \text{dist}(\boldsymbol{\pi}_s, \mathbf{h}) \leq \sigma_3, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

where $\text{dist}(\cdot, \cdot)$ denotes a distance function, $\sigma_1, \sigma_2, \sigma_3$ denote the corresponding scale parameters in the functions, and $\alpha \geq 0$ in Eq. 11 is a predefined thresholding parameter. These three functions are illustrated in Fig. 6, where the distance function is the Euclidean distance. Compared with the Gaussian function that is used in [6], these three functions produce much sparser features, making the computation more efficient.

2.4 Structured Learning of Similarity Models

Now we come to the other question of as to how we learn the weight vector \mathbf{w} in Eq. 5.

We denote the training entity set as $\mathcal{X} = \{\mathbf{x}_q^p\}_{q=1, \dots, N_p}^{p=1, 2}$, where $\forall p, \forall q, \mathbf{x}_q^p$ denotes the q^{th} person entity from camera view p . We refer to view 1 as the *probe* set, and view 2 as the *gallery* set. Also, we denote $\mathbf{y} = \{y_{ij}\}_{i,j \geq 1}$ as the ground-truth bipartite graph structure, and $y_{ij} = 1$ if $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_j^{(2)}$ are the same; otherwise, $y_{ij} = 0$. Then our method in training can be formulated as the following structured learning problem:

$$\begin{aligned} \min_{\mathbf{w}, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + C\xi \\ \text{s.t.} \quad & \forall \bar{\mathbf{y}} \in \mathcal{Y}, \mathbf{w}^T f(\mathcal{X}, \mathbf{y}) \geq \mathbf{w}^T f(\mathcal{X}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}, \bar{\mathbf{y}}) - \xi, \\ & \xi \geq 0, \end{aligned} \quad (14)$$

where \mathbf{w} is the weight vector, $\bar{\mathbf{y}} \in \mathcal{Y}$ denotes a predicted bipartite graph structure, $f(\mathcal{X}, \mathbf{y}) = \sum_{i,j} y_{ij} \phi(\mathbf{x}_{ij})$ (*resp.* $f(\mathcal{X}, \bar{\mathbf{y}}) = \sum_{i,j} \bar{y}_{ij} \phi(\mathbf{x}_{ij})$) denotes the basis function under the ground-truth (*resp.* predicted) graph structure, $\Delta(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{i,j} |y_{ij} - \bar{y}_{ij}|$ denotes the loss between the two structures, $C \geq 0$ is a predefined regularization constant, and $\|\cdot\|_2$ denotes the ℓ_2 norm of a vector. Here the constraint is enforcing the structured matching score of the ground-truth structure to be the highest among all possible matching structures in \mathcal{Y} . In order to adapt the definition

Algorithm 1 Structured learning of PRISM

Input : training entity set \mathcal{X} , ground-truth matching structure \mathbf{y} , predefined regularization parameter $C \geq 0$

Output: \mathbf{w}

Construct the feasible solution space \mathcal{Y} ;

Randomly sample a subset of matching structures $\bar{\mathcal{Y}} \subset \mathcal{Y}$;

repeat

$\mathbf{w} \leftarrow \text{RankSVM_Solver}(\mathcal{X}, \mathbf{y}, \bar{\mathcal{Y}}, C)$;

$\mathbf{y}^* \leftarrow \arg \max_{\bar{\mathbf{y}} \in \bar{\mathcal{Y}}} \mathbf{w}^T f(\mathcal{X}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}, \bar{\mathbf{y}})$;

$\bar{\mathcal{Y}} \leftarrow \bar{\mathcal{Y}} \cup \mathbf{y}^*$;

until *Converge*;

return \mathbf{w}

of \mathcal{Y} in Eq. 4 to the ground-truth matching structure \mathbf{y} , we can simply set $r_i = \max_i \sum_j y_{ij}$, and substitute this value into Eq. 4 to construct the feasible solution space \mathcal{Y} . Same as the structured matching in testing, in training we also utilize a priori knowledge on the correct matching structures to reduce the chance of mismatching.

In principle we can solve Eq. 14 using 1-slack structural SVMs [52]. We list the cutting-plane algorithm for training PRISM in Alg. 1. The basic idea here is to select most violated matching structure \mathbf{y}^* from the feasible set \mathcal{Y} in each iteration, and add it into the current feasible set $\bar{\mathcal{Y}}$, and resolve Eq. 14 using $\bar{\mathcal{Y}}$. In this way, the solution searching space is dependent on $\bar{\mathcal{Y}}$ rather than \mathcal{Y} . In each iteration, we can simply adopt RankSVM solver [53] to find a suitable \mathbf{w} . For inference, since we have $\Delta(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{i,j} |y_{ij} - \bar{y}_{ij}| = \sum_{i,j} (y_{ij} - \bar{y}_{ij})^2$ (because of $\forall y_{ij} \in \{0, 1\}, \forall \bar{y}_{ij} \in \{0, 1\}$), we indeed solve a binary quadratic problem, which can be efficiently solved using the similar thresholding trick for inference in testing.

Note that in order to speed up the learning we can alternatively adopt large-scale linear RankSVMs [53] (or even linear SVMs [54] as we did in [6]) with a large amount of randomly sampled matching structures from \mathcal{Y} to approximately solve Eq. 14. This trick has been widely used in many large-scale training methods (*e.g.* [55]) and demonstrated its effectiveness and efficiency without notable performance loss. Similarly, in our re-id cases we implement both learning strategies and have found that the performance loss is marginal.

3 IMPLEMENTATION

We illustrate the schematics of our method in Fig. 7. At training stage, we extract low-level feature vectors from randomly sampled patches in training images, and then cluster them into codewords to form a codebook, which is used to encode every image into a codeword image. Each pixel in a codeword image represents the centroid of a patch that has been mapped to a codeword. Further, a visual word co-occurrence model (descriptor) is calculated for every pair of gallery and probe images, and the descriptors from training data are utilized to train our classifier using Eq. 14. We perform re-id on the test data using Eq. 5.

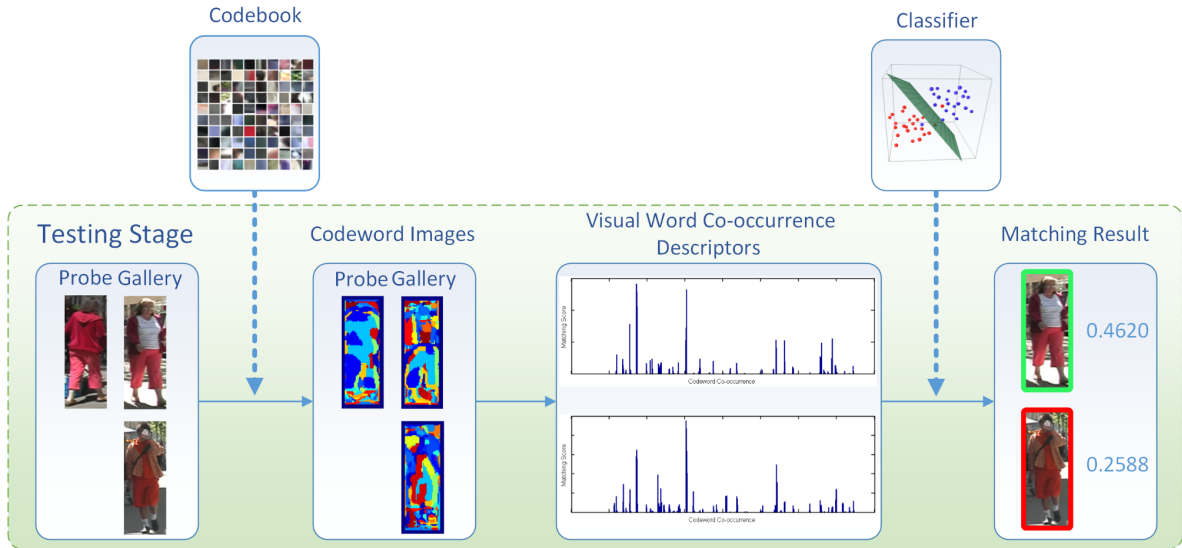


Fig. 7. The pipeline of our method, where “codebook” and “classifier” are learned using training data, and each color in the codeword images denotes a codeword.

Specifically, we extract a 672-dim Color+SIFT¹ [21] feature vector from each 5×5 pixel patch in images, and utilize K-Means to generate the visual codebooks based on about 3×10^4 randomly selected Color+SIFT features per camera view. Then every Color+SIFT feature is quantized into one of these visual words based on minimum Euclidean distance. The number of visual words per view is set by cross-validation.

We employ the chessboard distance for Eq. 11, 12 and 13, consider every pixel location as \mathbf{h} , and set the scale parameter σ by cross-validation for the spatial kernel κ . Similarly the regularization parameter C in Eq. 14 is set by cross-validation.

During testing, for performance measure we utilize a standard metric for re-id, namely, Cumulative Match Characteristic (CMC) curve, which displays an algorithm’s recognition rate as a function of rank. For instance, a recognition rate at rank- r on the CMC curve denotes what proportion of queries are correctly matched to a corresponding gallery entity at rank- r or better. Therefore, we set $\forall i, r_i = r$ in Eq. 5, and solve the optimization problem.

Note that we can further save on computational time for prediction during testing. This follows from the fact that we do not need the exact solution for Eq. 5, as long as we can use the current solution to find the ranks of entities in the gallery for each probe, to determine the top matches. Therefore, we ask the linear programming solver to run for 10 iterations at most in our experiments.

4 EXPERIMENTS

We test our method on three benchmark datasets, *i.e.* VIPeR [56], CUHK Campus [21], and iLIDS-VID [57],

1. We downloaded the code from https://github.com/Robert0812/saliency_match.

for both single-shot and multi-shot scenarios. We do not re-implement comparative methods. Instead, we try to cite numbers/figures of comparative methods either from released codes or from the original papers as accurately as possible (*i.e.* for methods LAFT [58] and LDM [59] in Table 1 and Table 2, respectively), if necessary. Also, we compare our method against currently known state-of-the-art on these datasets. Our experimental results are reported as the average over 3 trials.

We denote the three derivatives of our method based on different spatial kernels as (1) PRISM-I for using κ_1 in Eq. 11, (2) PRISM-II for using κ_2 in Eq. 12, and (3) PRISM-III for using κ_3 in Eq. 13, respectively.

4.1 Datasets and Experimental Settings

VIPeR [56] consists of 632 entities captured in two different camera views, denoted by CAM-A and CAM-B, respectively. Each image is normalized to 128×48 pixels. We follow the experimental set up described in [21]. The dataset is split in half randomly, one partition for training and the other for testing. Samples from CAM-A and CAM-B form the probe and gallery sets, respectively.

CUHK Campus [58], [21] consists of 1816 people captured from five different camera pairs, labeled from P1 to P5 and denoted as CAM-1 and CAM-2 per camera pair which form the probe and gallery sets, respectively. Each camera view has 2 images per entity, and each image contains 160×60 pixels. We follow the experimental settings in [58], [21], and use only images captured from P1. We randomly select 485 individuals from the dataset for training, and use the rest 486 individuals for testing.

iLIDS-VID [57] is a new re-id dataset created based on two non-overlapping camera views from the i-LIDS Multiple-Camera Tracking Scenario (MCTS) [60]. For single-shot learning, there are 300 image pairs for 300

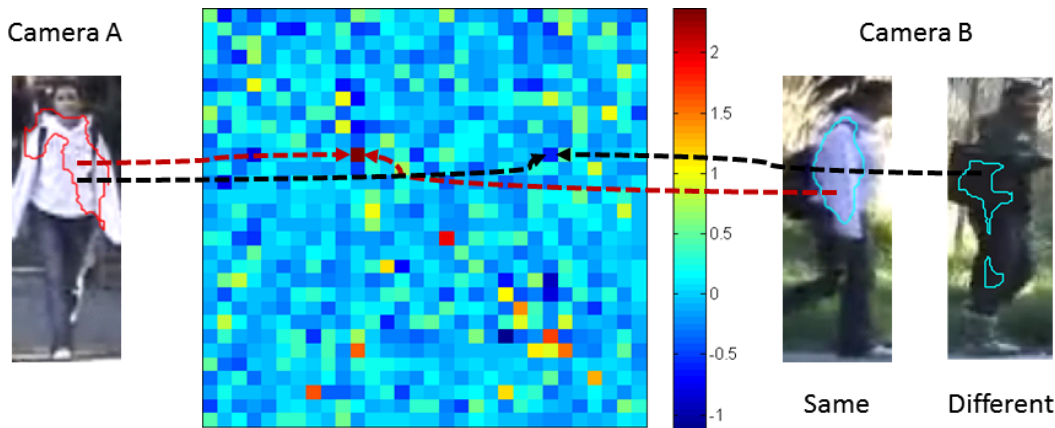


Fig. 8. The interpretation of our learned model parameter w in Eq. 14. The enclosed regions denote the pixels encoded by the same visual words, as used in Fig. 2. The learned weight for the visual word pair “white” and “light-blue” from the two camera views has a positive value, contributing to identifying the same person. On the other hand, the learned weight for the visual word pair “white” and “black” is negative, which contributes to identifying different persons.

randomly selected people with image size equal to 128×64 pixels. For multi-shot learning, there are 300 pairs of image sequences for the 300 people. The length of each image sequence varies from 23 to 192 frames with average of 73. Following [57], we randomly select 150 people as training data, and use the rest 150 people as testing data. The data from the first and second camera views forms the probe and gallery sets, respectively.

4.2 Model Interpretation

We start by interpreting our learned model parameters w in Eq. 14. We show a typical learned w matrix in Fig. 8 with 30 visual words per camera view. Recall that $w \triangleq p(\bar{y}_{ij} = 1 | \mathbf{u}, \mathbf{v})$ denotes how likely two images come from a same person according to the visual word pairs, and our spatial kernel κ always returns non-negatives indicating the spatial distances between visual word pairs in two images from two camera views. As we see in Fig. 8, by comparing the associated learned weights, “white” color in camera A is likely to be transferred into “light-blue” color (with higher positive weight), but very unlikely to be transferred into “black” color (with lower negative weight) in camera B . Therefore, when comparing two images from camera A and B , respectively, if within the same local regions the “white” and “light-blue” visual word pair from the two images occurs, it will contribute to identifying the same person; on the other hand, if “white” and “black” co-occur within the same local regions in the images, it will contribute to identifying different persons.

4.3 Single-Shot Learning

Now we discuss our results for single-shot learning (see the definition in Section 1.1). Table 1 lists our comparison results on the three datasets, where the numbers are the matching rates over different ranks on the CMC curves.

Here we divide comparative methods into 2 subcategories: non-fusion based and fusion based methods. Fusion

TABLE 1

Matching rate comparison (%) for single-shot learning, where “-” denotes no result reported for the method.

Rank $r =$	1	5	10	15	20	25
	VIPeR					
SCNCD [23]	20.7	47.2	60.6	68.8	75.1	79.1
LADF [42]	29.3	61.0	76.0	83.4	88.1	90.9
Mid-level filters [39]	29.1	52.3	65.9	73.9	79.9	84.3
Mid-level filters+LADF [39]	43.4	73.0	84.9	90.9	93.7	95.5
VW-CooC [6]	30.7	63.0	76.0	81.0	-	-
RQDA [36]	34.7	65.4	78.6	-	89.6	-
Semantic (super. single) [24]	31.1	68.6	82.8	-	94.9	-
Polynomial kernel [35]	36.8	70.4	83.7	-	91.7	-
QALF [36]	30.2	51.6	62.4	-	73.8	-
Semantic (super. fusion) [24]	41.6	71.9	86.2	-	95.1	-
SCNCD _{final} (ImgF) [23] ²	37.8	68.5	81.2	87.0	90.4	92.7
Ensemble Color Model [45]	38.9	67.8	78.4	-	88.9	-
Metric ensembles [44]	45.9	77.5	88.9	-	95.8	-
Kernel ensembles-I [27]	35.1	68.2	81.3	-	91.1	-
Kernel ensembles-II [27]	36.1	68.7	80.1	-	85.6	-
Ours: PRISM-I	35.8	69.9	80.4	86.7	89.6	90.5
Ours: PRISM-II	36.7	66.1	79.1	85.1	90.2	92.4
Ours: PRISM-III	35.4	66.1	77.9	85.1	87.7	90.5
	CUHK01					
LAFT [58]	25.8	55.0	66.7	73.8	79.0	83.0
Mid-level filters [39]	34.3	55.1	65.0	71.0	74.9	78.0
VW-CooC [6]	44.0	70.5	79.1	84.8	-	-
Semantic (super. single) [24]	32.7	51.2	64.4	-	76.3	-
Semantic (super. fusion) [24]	31.5	52.5	65.8	-	77.6	-
Metric ensembles [44]	53.4	76.4	84.4	-	90.5	-
Ours: PRISM-I	51.8	72.0	79.5	83.7	86.9	88.2
Ours: PRISM-II	50.1	70.1	79.4	82.9	85.4	87.8
Ours: PRISM-III	52.0	71.8	79.9	84.0	85.9	87.8
	iLIDS-VID					
Colour&LBP [61]+RSVM	9.1	22.6	33.2	45.5	-	-
SS-SDALF [13]	5.1	14.9	20.7	31.3	-	-
Saliency [22]	10.2	24.8	35.5	52.9	-	-
Ours: PRISM-I	22.0	43.3	52.0	62.7	73.3	77.3
Ours: PRISM-II	20.0	39.3	52.7	60.0	70.0	76.7
Ours: PRISM-III	16.7	36.7	52.0	56.7	67.3	74.7

based methods aim to combine multiple features/metrics to improve the matching performance, while non-fusion methods perform recognition using single type of features or a metric. Overall, fusion based methods achieve better performance than non-fusion based methods (including

2. ImgF: image-foreground feature representations

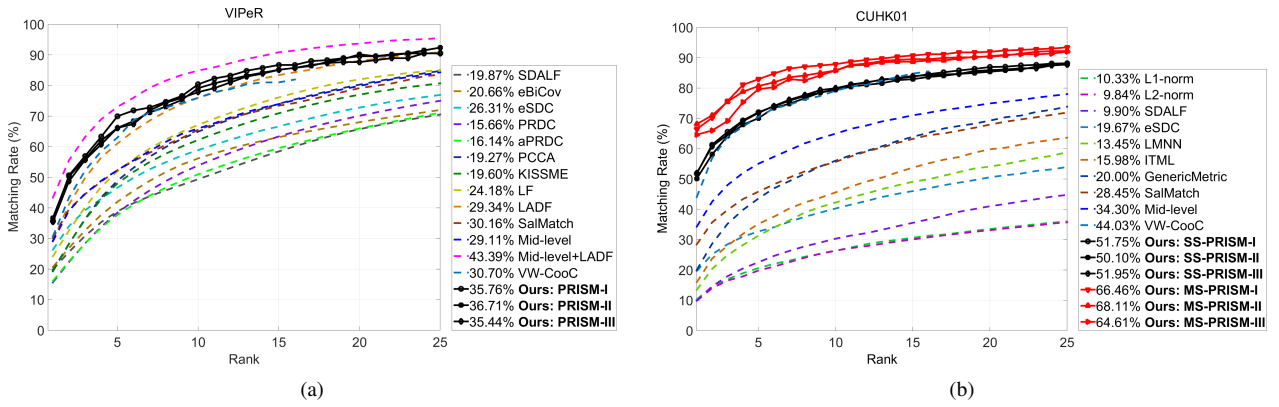


Fig. 9. CMC curve comparison on (a) VIPeR and (b) CUHK01, where “SS” and “MS” denote the single-shot and multi-shot, respectively. Notice that except our results, the rest are copied from [39].

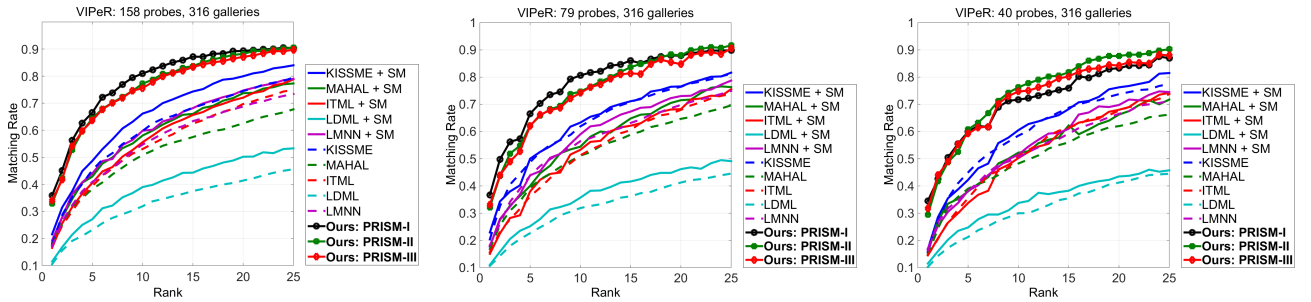


Fig. 10. CMC curve comparison on VIPeR using different numbers of entities in the probe set for robust re-id.

ours, which is always comparable). These methods, however, lack of clear interpretability of why the performance is better. Among non-fusion based methods, on VIPeR “Mid-level filters+LADF” from [39] is the current best method, which utilized more discriminative mid-level filters as features with a powerful classifier, and “SCNCD_{final}(ImgF)” from [23] is the second, which utilized only foreground features. Our results are comparable to both of them. However, PRISM always outperforms their original methods significantly when either the powerful classifier or the foreground information is not used. On CUHK01 and iLIDS-VID, PRISM performs the best. At rank-1, it outperforms [6] and [22] by **8.0%** and **11.8%**, respectively. Some CMC curves of different methods on VIPeR and CUHK01 are compared in Fig. 9. Our current method only utilizes the visual word co-occurrence model. Integration of multiple features will be explored in our future work.

Compared with our previous work in [6], our improvement here mainly comes from the structured matching in testing by precluding the matches that are probably wrong (*i.e.* reducing the feasible solution space). Clearly our method outperforms [6] by **6.0%** on VIPeR and **8.0%** on CUHK01 at rank-1 rank in terms of matching rate.

4.4 Multi-Shot Learning

For multi-shot learning (see the definition in Section 1.1), since VIPeR does not have multiple images per person,

3. The code is downloaded from <http://lrs.icg.tugraz.at/research/kissme/>.

TABLE 2

Matching rate comparison (%) for multi-shot learning, where “-” denotes no result reported for the method.

Rank $r =$	1	5	10	15	20	25
CUHK01						
LAFT [58]	31.4	58.0	68.3	74.0	79.0	83.0
LDM [59]	12.1	31.7	41.7	48.3	54.0	58.0
Ours: PRISM-I	66.5	82.9	87.9	90.7	92.0	93.4
Ours: PRISM-II	68.1	80.7	85.8	88.7	90.3	92.2
Ours: PRISM-III	64.6	79.6	85.8	89.5	90.5	92.0
iLIDS-VID						
MS-SDALF [13]	6.3	18.8	27.1	-	37.3	-
MS-Colour&LBP+RSVM	23.2	44.2	54.1	-	68.8	-
DVR [57]	23.3	42.4	55.3	-	68.4	-
MS-SDALF+DVR	26.7	49.3	61.0	-	71.6	-
MS-Colour&LBP+DVR	34.5	56.7	67.5	-	77.5	-
Saliency [22]+DVR	30.9	54.4	65.1	-	77.1	-
Ours: PRISM-I	60.7	86.7	89.3	94.7	96.0	96.7
Ours: PRISM-II	62.0	86.0	90.0	94.7	96.0	96.7
Ours: PRISM-III	62.0	86.0	90.0	94.7	96.0	96.7

we compare our method with others on CUHK01 and iLIDS-VID only, and list the comparison results in Table 2. Clearly, PRISM beats the state-of-the-art significantly by **36.7%** on CUHK01, and **27.5%** on iLIDS-VID, respectively, at rank-1. Note that even compared with the best fusion method [44] on CUHK01, our method outperforms it by **14.7%** at rank-1. Our multi-shot CMC curves on CUHK01 are also shown in Fig. 9(b) for comparison.

The improvement of our method for multi-shot learning mainly comes from the multi-instance setting of our latent

TABLE 3
Matching accuracy comparison (%) for robust re-id.

Rank $r =$	158 probes						79 probes						40 probes					
	1	5	10	15	20	25	1	5	10	15	20	25	1	5	10	15	20	25
KISSME ³	18.0	45.3	59.7	68.3	74.5	79.5	20.4	48.7	61.9	70.9	76.5	81.3	15.8	45.0	58.5	68.8	73.8	77.5
KISSME + SM	21.5	48.7	66.1	74.3	80.0	84.0	22.9	50.0	63.3	71.1	76.6	81.6	16.8	42.3	60.5	68.3	76.3	81.5
Ours: PRISM-I	35.9	66.5	81.0	87.2	89.3	90.4	36.7	66.6	80.6	86.1	87.9	89.9	34.5	60.0	71.8	76.0	83.3	87.0
Ours: PRISM-II	32.9	64.3	77.2	83.7	88.2	90.5	32.2	62.2	74.7	83.7	88.0	91.7	29.5	60.8	76.3	81.8	87.8	90.3
Ours: PRISM-III	34.0	63.6	75.6	83.4	87.1	89.7	33.2	62.0	74.2	81.4	84.8	90.5	31.8	59.3	74.8	80.3	84.3	88.0

spatial kernel in Eq. 10. It has been clearly demonstrated as we compare our performances using single-shot learning and multi-shot learning on CUHK01. By averaging over all the gallery images for one entity in multi-shot learning, the visual word co-occurrence model constructed is more robust and discriminative than that for single-shot learning, leading to significant performance improvement.

4.5 Robust Person Re-identification

In this experiment, we would like to demonstrate the robustness of our method by including the missing match scenarios for re-id. Here we compare different methods only on VIPeR for the demonstration purpose.

We utilize KISSME [62] to do the comparison, which includes 5 different metric learning methods, namely, KISSME [62], MAHAL (*i.e.* Mahalanobis distance learning), ITML [63], LDML [64], and LMNN [65]. These metric learning methods learn the similarities between image pairs, which are equivalent to $\mathbf{w}^T \phi(\mathbf{x}_{ij})$ in Eq. 5. Then we apply our structured matching (SM for short) in Eq. 5 on top of each method above by utilizing these image pair similarities for comparison.

We first simulate the re-id scenario where every probe has its match in the gallery set, but not all the galleries have matches in the probe set. Fig. 10 shows our comparison results using (a) 158 probes, (b) 79 probes, and (c) 40 probes, respectively, with 316 entities in the gallery set. As we see, for all the 5 metric learning methods, structured matching helps improve their performances, in general, under different settings. PRISM always performs best among all the methods.

Table 3 summarizes the numbers at different ranks for KISSME, KISSME+SM, and our PRISM in Fig. 10, since KISSME and KISSME+SM are the most comparable methods in Fig. 10. At rank-1, PRISM outperforms them significantly by at least **14.4%**. As the number of probes decreases, in general, at every rank the matching rates of all the methods degrades. However, as we see, for PRISM the matching rates are much more stable. By comparing these results with those in Table 1, we can see that these results are similar, again demonstrating the robustness of our structured matching method.

We display representative matching results at rank-1 in Fig. 11 using PRISM with/without structured matching for robust re-id. As we see, without structured matching all the probes are matched with the same entity in the gallery, inducing incorrect matches. However, structured matching



Fig. 11. Examples of matching result comparison on VIPeR at rank-1 using PRISM with/without structured matching for robust re-id. The sizes of the probe and gallery sets are 40 and 316, respectively.

TABLE 4
Average matching accuracy (%) for robust re-id on VIPeR.

# probe	KISSME	KISSME+SM	PRISM-I	PRISM-II	PRISM-III
158	14.9	20.6	20.9	20.4	21.0
79	15.0	16.5	20.8	20.2	20.8
40	14.4	15.1	22.5	22.2	22.6

can correct this type of errors, find their true matches, and thus improve the matching rates.

Next we simulate another re-id scenario where not all the probes/galleries have matches in the gallery/probe sets. This is a common situation in re-id where missing matches occur all the time. During testing, we randomly select 158/79/40 probes to be matched with randomly selected 158 galleries, and list in Table 4 the results in terms of average *matching accuracy*, *i.e.*, in the probe set the total number of true positives (true matches) and true negatives (true non-matches) divided by the total number of entities. Still structured matching helps improve the performance, and PRISM achieves the best.

4.6 Storage & Computational Time

Storage (S_t for short) and computational time during testing are two critical issues in real-world applications. In our method, we only need to store the image descriptors for calculating similarities between different entities. The computational time can be divided into three parts: (1) image descriptors T_1 , (2) entity-matching similarities T_2 , and (3) entity-level structured matching T_3 . We do not consider the time for generating Color+SIFT features, since we directly use the existing code without any control.

We record the storage and computational time using 500 visual words for both probe and gallery sets on VIPeR.

TABLE 5

Average storage and computational time for our PRISM.

	S_t (Kb)	T_1 (ms)	T_2 (ms)	T_3 (s)
PRISM-I	158.9	77.4	1.3	1.3
PRISM-II	113.7	73.3	1.4	1.5
PRISM-III	16.2	56.4	1.3	1.3

The rest of the parameters are the same as described in Section 3. Roughly speaking, the storage per data sample and computational time are linearly proportional to the size of images and number of visual words. Our implementation is based on unoptimized MATLAB code. Numbers are listed in Table 5 for identifying the matches between 316 probes and 316 galleries, including the time for saving and loading features. Our experiments were all run on a multi-thread CPU (Xeon E5-2696 v2) with a GPU (GTX TITAN). The method ran efficiently with very low demand for storage.

5 CONCLUSION

In this paper, we propose a structured matching based method for re-id in the contexts of (1) single-shot learning, and (2) multi-shot learning. We formulate the core of the re-id problem, *i.e.* entity matching, as a weighted bipartite graph matching problem, and try to predict such graph structures. To handle the huge appearance variation (*e.g.* visual ambiguity and spatial distortion) as well as achieving computational efficiency, we propose a new basis function to capture the visual word co-occurrence statistics. Our experiments on several benchmark datasets strongly demonstrate the power of our PRISM for re-id in both scenarios. Low demand of storage and good computational efficiency indicate that our method can be potentially applied to real-world applications.

Several questions will be considered as our future work. It would be useful to further reduce the computational complexity of calculating our pair-wise latent spatial kernels. One possibility is to modify the learning algorithm by decomposing the weight matrix w into two separable matrices, because our appearance model can be decomposed into two parts, one from the probe image and the other from the gallery image. Such decomposition will accelerate the computation. Second, it would be interesting to learn the optimal spatial kernels and see how they affect the behavior of our visual word co-occurrence model. Third, it would be also interesting to extend our current structured matching framework to multi-camera settings by adding more constraints on the matched/dismatched entity pairs to enforce the structural information (*e.g.* temporal) in the network.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Grant Award 2013-ST-061-ED0001. The views and conclusions

contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

REFERENCES

- [1] R. Vezzani, D. Baltieri, and R. Cucchiara, "People reidentification in surveillance and forensics: A survey," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 29:1–29:37, Dec. 2013.
- [2] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin, "Learning structured prediction models: A large margin approach," in *ICML*, 2005, pp. 896–903.
- [3] P. Banerjee and R. Nevatia, "Learning neighborhood cooccurrence statistics of sparse features for human activity recognition," in *AVSS*, 2011, pp. 212–217.
- [4] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *CVPR*, June 2008.
- [5] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Graph cut based inference with co-occurrence statistics," in *ECCV*, 2010, pp. 239–253.
- [6] Z. Zhang, Y. Chen, and V. Saligrama, "A novel visual word co-occurrence model for person re-identification," in *ECCV 2014 Workshops*, vol. 8927, 2015, pp. 122–133.
- [7] A. J. Smola, A. Gretton, L. Song, and B. Schölkopf, "A hilbert space embedding for distributions," in *ALT*, 2007, pp. 13–31.
- [8] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels," *JMLR*, vol. 5, pp. 819–844, Dec. 2004.
- [9] G. Doretto, T. Sebastian, P. Tu, and J. Rittscher, "Appearance-based person reidentification in camera networks: problem overview and current approaches," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 2, pp. 127–151, 2011.
- [10] X. Wang and R. Zhao, "Person re-identification: System design and evaluation overview," in *Person Re-Identification*, 2014, pp. 351–370.
- [11] S. Bak, E. Corvee, F. Bremond, and M. Thonnat, "Multiple-shot human re-identification by mean riemannian covariance grid," in *AVSS*, 2011, pp. 179–184.
- [12] M. Bauml and R. Stiefelhagen, "Evaluation of local features for person re-identification in image sequences," in *AVSS*, 2011, pp. 291–296.
- [13] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," in *CVPR*, 2010, pp. 2360–2367.
- [14] N. Gheissari, T. B. Sebastian, and R. Hartley, "Person reidentification using spatiotemporal appearance," in *CVPR*, vol. 2, 2006, pp. 1528–1535.
- [15] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *ECCV*, 2008, pp. 262–275.
- [16] C. Liu, S. Gong, C. C. Loy, and X. Lin, "Person re-identification: What features are important?" in *ECCV Workshops*, vol. 7583, 2012, pp. 391–401.
- [17] B. Ma, Y. Su, and F. Jurie, "Bicov: a novel image representation for person re-identification and face verification," in *BMVC*, 2012.
- [18] V.-H. Nguyen, K. Nguyen, D.-D. Le, D. A. Duong, and S. Satoh, "Person re-identification using deformable part models," in *ICONIP*, 2013, pp. 616–623.
- [19] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian, "Local fisher discriminant analysis for pedestrian re-identification," in *CVPR*, 2013, pp. 3318–3325.
- [20] B. Prosser, W.-S. Zheng, S. Gong, T. Xiang, and Q. Mary, "Person re-identification by support vector ranking," in *BMVC*, vol. 1, no. 3, 2010, p. 5.
- [21] R. Zhao, W. Ouyang, and X. Wang, "Person re-identification by salience matching," in *ICCV*, 2013.
- [22] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," in *CVPR*, 2013, pp. 3586–3593.
- [23] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, and S. Z. Li, "Salient color names for person re-identification," in *ECCV*, 2014.
- [24] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *CVPR*, 2015.
- [25] Z. Wu, Y. Li, and R. Radke, "Viewpoint invariant human re-identification in camera networks using pose priors and subject-discriminative features," *TPAMI*, 2014.

- [26] M. Dikmen, E. Akbas, T. S. Huang, and N. Ahuja, "Pedestrian recognition with a learned metric," in *ACCV*, 2011, pp. 501–512.
- [27] F. Xiong, M. Gou, O. Camps, and M. Szaier, "Person re-identification using kernel-based metric learning methods," in *ECCV*, 2014, pp. 1–16.
- [28] O. Javed, K. Shafique, Z. Rasheed, and M. Shah, "Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views," *CVIU*, vol. 109, no. 2, pp. 146–162, Feb. 2008.
- [29] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *ACCV*, 2012, pp. 31–44.
- [30] X. Liu, M. Song, D. Tao, X. Zhou, C. Chen, and J. Bu, "Semi-supervised coupled dictionary learning for person re-identification," in *CVPR*, 2014.
- [31] A. Mignon and F. Jurie, "PCCA: a new approach for distance learning from sparse pairwise constraints," in *CVPR*, 2012, pp. 2666–2672.
- [32] F. Porikli, "Inter-camera color calibration by correlation model function," in *ICIP*, vol. 2, 2003, pp. II–133.
- [33] W.-S. Zheng, S. Gong, and T. Xiang, "Person re-identification by probabilistic relative distance comparison," in *CVPR*, 2011, pp. 649–656.
- [34] W. Zheng, S. Gong, and T. Xiang, "Re-identification by relative distance comparison," *TPAMI*, vol. 35, no. 3, pp. 653–668, 2013.
- [35] D. Chen, Z. Yuan, G. Hua, N. Zheng, and J. Wang, "Similarity learning on an explicit polynomial kernel feature map for person re-identification." *CVPR*, 2015.
- [36] S. Liao, Y. Hu, and S. Z. Li, "Joint dimension reduction and metric learning for person re-identification," *arXiv preprint arXiv:1406.4216*, 2014.
- [37] Y. Xu, L. Lin, W.-S. Zheng, and X. Liu, "Human re-identification by matching compositional template with cluster sampling," in *ICCV*, 2013.
- [38] C. Liu, C. C. Loy, S. Gong, and G. Wang, "Pop: Person re-identification post-rank optimisation," in *ICCV*, 2013.
- [39] R. Zhao, W. Ouyang, and X. Wang, "Learning mid-level filters for person re-identification," in *CVPR*, 2014, pp. 144–151.
- [40] Y. Wu, M. Mukunoki, and M. Minoh, "Locality-constrained collaboratively regularized nearest points for multiple-shot person re-identification," in *Proc. of The 20th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, 2014.
- [41] L. Bazzani, M. Cristani, A. Perina, and V. Murino, "Multiple-shot person re-identification by chromatic and epitomic analyses," *Pattern Recogn. Lett.*, vol. 33, no. 7, pp. 898–903, May 2012.
- [42] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith, "Learning locally-adaptive decision functions for person verification," in *CVPR*, 2013, pp. 3610–3617.
- [43] A. Das, A. Chakraborty, and A. K. Roy-Chowdhury, "Consistent re-identification in a camera network," in *ECCV*, 2014.
- [44] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Learning to rank in person re-identification with metric ensembles," *ArXiv e-prints*, Mar. 2015.
- [45] X. Liu, H. Wang, Y. Wu, J. Yang, and M.-H. Yang, "An ensemble color model for human re-identification," in *WACV*, 2015.
- [46] S. Gong, M. Cristani, C. C. Loy, and T. M. Hospedales, "The re-identification challenge," in *Person Re-Identification*, 2014, pp. 1–20.
- [47] W.-S. Zheng, S. Gong, and T. Xiang, "Transfer re-identification: From person to set-based verification," in *CVPR*, 2012, pp. 2650–2657.
- [48] B. Cancela, T. Hospedales, and S. Gong, "Open-world person re-identification by multi-label assignment inference," in *BMVC*, 2014.
- [49] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *CVPR*, 2011, pp. 263–270.
- [50] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [51] R. Rau and J. H. McClellan, "Efficient approximation of gaussian filters," *TSP*, vol. 45, no. 2, pp. 468–471, 1997.
- [52] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural svms," *Mach. Learn.*, vol. 77, no. 1, pp. 27–59, oct 2009.
- [53] C.-P. Lee and C.-J. Lin, "Large-scale linear ranksvm," *Neural Comput.*, vol. 26, no. 4, pp. 781–817, Apr. 2014.
- [54] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *JMLR*, vol. 9, pp. 1871–1874, 2008.
- [55] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *JMLR*, vol. 2, pp. 45–66, Mar. 2002.
- [56] D. Gray, S. Brennan, and H. Tao, "Evaluating appearance models for recognition, reacquisition, and tracking," in *PETS*, 2007.
- [57] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by video ranking," in *ECCV*, 2014.
- [58] W. Li and X. Wang, "Locally aligned feature transforms across views," in *CVPR*, 2013, pp. 3594–3601.
- [59] L. Yang, R. Jin, R. Sukthankar, and Y. Liu, "An Efficient Algorithm for Local Distance Metric Learning," in *AAAI*, 2006.
- [60] U. H. Office, "i-LIDS multiple camera tracking scenario definition," 2008.
- [61] M. Hirzer, P. M. Roth, M. Köstinger, and H. Bischof, "Relaxed pairwise learned metric for person re-identification," in *ECCV*, 2012, pp. 780–793.
- [62] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *CVPR*, 2012.
- [63] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *ICML*, Corvallis, Oregon, USA, 2007, pp. 209–216.
- [64] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? metric learning approaches for face identification," in *ICCV*, sep 2009, pp. 498–505.
- [65] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *JMLR*, vol. 10, pp. 207–244, Jun. 2009.