Deep Deformable Patch Metric Learning for Person Re-identification

Sławomir Bąk Peter Carr
Disney Research
Pittsburgh, PA, USA, 15213
{slawomir.bak,peter.carr}@disneyresearch.com

Abstract—The methodology for finding the same individual in a network of cameras must deal with significant changes in appearance caused by variations in illumination, viewing angle and a person's pose. Re-identification requires solving two fundamental problems: (1) determining a distance measure between features extracted from different cameras that copes with illumination changes (metric learning); and (2) ensuring that matched features refer to the same body part (correspondence). Most metric learning approaches focus on finding a robust distance measure between bounding box images, neglecting the alignment aspects. In this paper, we propose to learn appearance measures for patches that are combined using deformable models. Learning metrics for patches avoids strong dimensionality reduction, thus keeping more information. Additionally, we allow patches to change their locations, directly addressing the correspondence problem. As patches from different locations may share the same metric, our method effectively multiplies the amount of training data and allows patch metrics to be learned on the smaller amounts of labeled images. Different metric learning approaches (KISSME, XQDA, LSSL) together with different deformable models (spring constraints, one-to-one matching constraints) are investigated and compared. For describing patches, we propose to learn a deep feature representation with Convolutional Neural Networks (CNNs), thus obtaining highly effective features for re-identification. We demonstrate that our approach significantly outperforms state-of-the-art methods on multiple datasets.

Index Terms—metric learning, deformable models.

1 Introduction

Person re-identification is the problem of recognizing the same individual across a network of cameras. In a real-world scenario, the transition time between cameras may significantly decrease the search space, but temporal information alone is not usually sufficient to solve the problem. As a result, visual appearance models have received a lot attention in computer vision research [5], [25], [26], [29], [30], [41], [47]. The underlying challenge for visual appearance is that the models must work under significant appearance changes caused by variations in illumination, viewing angle and a person's pose.

Metric learning approaches often achieve the best performance in re-identification. These methods learn a distance function between features from different cameras such that relevant dimensions are emphasized while irrelevant ones are ignored. Many metric learning approaches [10], [19], [24] divide a bounding box pedestrian image into a fixed grid of regions and extract descriptors which are then concatenated into a high-dimensional feature vector. Afterwards, dimensionality reduction is applied, and then metric learning is performed on the reduced subspace of differences between feature vectors. To avoid overfitting, the dimensionality must be significantly reduced. In practice, the subspace dimensionality is about three orders of magnitude smaller than the original. Such strong dimensionality reduction might result in the loss of discriminative information. Additionally, features extracted on a fixed grid (see Fig. 1), may not correspond even though it is the same person (*e.g.* due to a pose change). Metric learning is unable to recover this lost information.

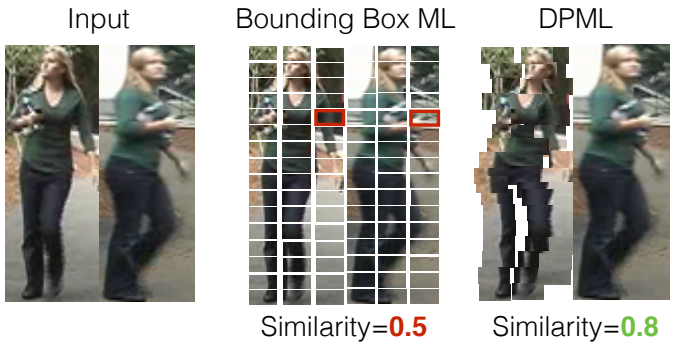


Fig. 1: Full bounding box metric learning *vs.* deformable patch metric learning (DPML). The corresponding patches in the grid (highlighted in red) do not correspond to the same body part because of the pose change. Information from such misaligned features might be lost during the metric learning step. Instead, our DPML deforms to maximize similarity using metrics learned on a patch level.

In this paper, instead of learning a metric for concatenated features extracted from full bounding boxes from different cameras, we propose to learn metrics for 2D patches. Learning metrics for patches is less prone to overfitting (because of lower dimensionality) and it requires less compression. As a result it keeps more information.

Furthermore, we do not assume the patches must be located on

a fixed grid. Our model allows patches to perturb their locations when computing similarity between two images (see Fig. 1). This model is inspired from part-based object detection [12], [43], which decomposes the appearance model into local templates with geometric constraints (conceptualized as springs).

Our main contributions are:

- We propose to learn metrics locally, on feature vectors extracted from patches. These metrics can be combined into a unified distance measure.
- We introduce two deformable patch-based models for accommodating pose changes and occlusions: (1) an unsupervised deformable model that introduces a global one-to-one matching constraint solved by a linear assignment problem, and (2) a supervised deformable model that combines an appearance term with a deformation cost that controls relative placement of patches.
- For describing patches, we propose to learn a deep feature representation with Convolutional Neural Networks (CNNs). The CNN is learned through challenging multi-class identification task. We force the CNN to recognize not only the person identity from which the patch has been extracted but also the patch location. This results in highly effective representation, significantly improving the re-identification accuracy.

Our experiments illustrate the merits of patch-based techniques and achieve new state-of-the-art performance on multiple datasets outperforming existing approaches by large margins.

2 Related work

Person re-identification approaches can be divided into two groups: *feature modeling* [4], [11] designs descriptors (usually handcrafted) which are robust to changes in imaging conditions, and *metric learning* [1], [10], [19], [23], [24], [42], [50] searches for effective distance functions to compare features from different cameras. Robust features can be modeled by adopting perceptual principles of symmetry and asymmetry of the human body [11]. The correspondence problem can be approached by locating body parts [4], [8] and extracting local descriptors (color histograms [8], color invariants [21], covariances [4], CNN [29]). However, to find a proper descriptor, we need to look for a trade-off between its discriminative power and invariance between cameras. This task can be considered a *metric learning* problem that maximizes inter-class variation while minimizing intra-class variation.

Many different machine learning algorithms have been considered for learning a robust similarity function. Gray *et al.* employed Adaboost for feature selection and weighting [14], Prosser *et al.* defined the person re-identification as a ranking problem and used an ensemble of RankSVMs [32]. Recently features learned from deep convolution neural networks have been investigated [1], [7], [23], [35], [37], [40], [48].

However, the most common choice for learning a metric remains the family of Mahalanobis distance functions. These include Large Margin Nearest Neighbor Learning (LMNN) [39], Information Theoretic Metric Learning (ITML) [9] and Logistic Discriminant Metric Learning (LDML) [15]. These methods usually aim at improving k-nn classification by iteratively adapting the metric. In contrast to these iterative methods, Köstinger [19] proposed the KISS metric which uses a statistical inference based on a likelihood-ratio test of two Gaussian distributions modeling positive and negative pairwise differences between features. Owing to its effectiveness and efficiency, the KISS metric is a popular baseline that has been extended to linear [25], [30] and non-linear [29], [41] subspace embeddings. Most of these approaches learn a Mahalanobis distance function for feature vectors extracted from full bounding box images. Integration of feature learning directly with metric learning approach has been proposed in [38]. Mahalanbois-like function together with feature representation is learned with a novel end-to-end framework throughout a triplet embedding.

Recently, a trend of learning similarity measures for patches [2], [3], [33], [34], [51] has emerged. Operating on patches allows to directly address person pose variations and camera viewpoint changes. Shen *et al.* [33] learns the correspondence structure that captures spatial correspondence patterns across camera viewpoints The correspondence structure is represented by patch-wise matching probabilities learned using a boosting-like approach. Patch-wise correspondence is also introduced in [34]. First the body is divided into upper and lower body parts and then clustering trees are independently constructed to find the patch correspondence. Zheng *et al.* [51] shows that introducing the patch-level matching model based on a sparse representation can help in handling inaccurate person detectors as well as the large amount of occlusion.

This paper is based on our previous work [2], where we have proposed to learn dissimilarity functions for patches within bounding boxes, and then combine their scores into a robust distance measure. We have shown that our approach has clear advantages over existing algorithms. In this paper we continue our analysis by evaluating additional parameters (*e.g.* size of patches and their layouts) and by employing novel metric learning approaches. We also investigate additionally an unsupervised deformable model based on one-to-one matching constraint. Finally, we propose to learn patch features directly from data through challenging multi-class patch identification task employing the CNN model [40]. This results in the highly effective representation that brings significant improvement in the performance. Compared with state-of-the art methods, our approach yields significantly higher recognition accuracy.

3 Method

Often the dissimilarity $\Psi(i,j)$ between two bounding box images i and j taken from different cameras is defined as a Mahalanobis metric. The Mahalanobis metric measures the squared distance between feature vectors extracted from these bounding box images, $\mathbf{x}_i$ and $\mathbf{x}_j$

$$\Psi(i,j) = d^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j), \qquad (1)$$

where $\mathbf{M}$ is a matrix encoding the basis for the comparison. $\mathbf{M}$ is usually learned in two stages: dimensionality reduction is first applied on $\mathbf{x}_i$ and $\mathbf{x}_j$ (*e.g.* principle component analysis - PCA), and then metric learning (*e.g.* KISS metric [19]) is performed on the reduced subspace. To avoid overfitting, the dimensionality must be significantly reduced to keep the number of free parameters low [16], [25]. In practice, $\mathbf{x}_i$ and $\mathbf{x}_j$ are high dimensional feature vectors and their reduced dimensionality is usually about three orders of magnitude smaller than the original [19], [25], [29]. Such strong dimensionality reduction might lose discriminative information, especially in case of misaligned features in $\mathbf{x}_i$ and $\mathbf{x}_j$ (*e.g.* highlighted patches in Fig. 1).

We propose to learn a metric for matching patches within the bounding box. We perform dimensionality reduction on features extracted from each patch. The reduced dimensionality is usually only one order of magnitude smaller than the original one, thus keeping more information (see Section 4).

In Section 3.1 we offer a patch-based metric learning. Section 3.2 introduces three state-of-the-art Mahalanobis-like metric learning approaches: KISSME [19], XQDA [25] and LSSL [42]. In Section 3.3, we propose two methods for integrating patch-metrics into a single similarity measure and in Section 3.4 we show how to learn a very effective patch representation with Convolutional Neural Networks.

3.1 Patch-based Metric Learning

We divide bounding box image i into a dense grid with overlapping rectangular patches. From each patch location k, we extract patch feature vector $\mathbf{p}_i^k$. We represent bounding box image i as an ordered set of patch features $\mathcal{X}_i = \{\mathbf{p}_i^1, \mathbf{p}_i^2, \ldots, \mathbf{p}_i^K\}$, where K is the number of patches. Usually in standard metric learning approaches [19], [26], [29], [30], these patch descriptors are further concatenated into a single high dimensional feature vector (*e.g.* $\mathbf{x}_i = [\mathbf{p}_i^1 | \mathbf{p}_i^2 | \ldots | \mathbf{p}_i^K]$) and metric learning together with dimensionality reduction is then performed. Instead, we learn a dissimilarity function Φ for feature vectors extracted from patches. Patch dissimilarities are further combined into a unified dissimilarity by integration function $\mathcal{Z}$ (see Section 3.3). We define the dissimilarity between two images i and j as

$$\Psi(i,j) = \mathcal{Z}_{k,l \in 1 \ldots K} \left(\Phi(\mathbf{p}_i^k, \mathbf{p}_j^l; \theta(k)) \right) \tag{2}$$

where $\mathbf{p}_i^k$ and $\mathbf{p}_j^l$ are the feature vectors extracted from patches at locations k and l, respectively, in bounding box images i and j. Images i and j are assumed to come from different cameras. Set of parameters θ determines function Φ and it is learned using a given metric learning approach (see Section 3.2). Notice that $\Psi(i,j)$ is defined as an asymmetric dissimilarity measure due to k dependency. If symmetry is a concern, one can redefine the final dissimilarity as a function of both $\Psi(i,j)$ and $\Psi(j,i)$, *e.g.* $\Psi'(i,j) = \min(\Psi(i,j), \Psi(j,i))$.

Although, it is possible to learn one metric for each patch location k, this might be too many degrees of freedom. In practice, multiple patch locations might share a common metric, and in the extreme case a single θ could be learned for all patch locations. We investigated re-identification performance with different numbers of patch metrics (see Section 4.2.1) and found that in some cases multiple metrics might perform better than a single one. Regions with statistically different amounts of background noise should have different metrics (*e.g.* patches close to the head contain more background noise than patches close to the torso). However, we also found that the recognition performance is a function of available training data (see Section 4.2.1), which limits the number of patch metrics that can be learned efficiently. In the standard approach, a pair of bounding boxes corresponds to a single training example. Breaking a bounding box into a set of patches increases the amount of training data if a reduced number of metrics is learned (*e.g.* some locations k share the same metric/parameters θ). When a single θ is learned, the amount of training data increases by combining patches for all K locations into a single set ($K\times$ more positive examples for learning a metric compared to the standard approach). In experiments we show that this can significantly boost performance when the training dataset is small (*e.g.* iLIDS dataset).

3.2 Metric learning (Φ)

Given pairs of sample bounding boxes (i,j) we introduce the space of pairwise differences $\mathbf{p}_{ij}^k = \mathbf{p}_i^k - \mathbf{p}_j^k$ and partition the training data into $\mathbf{p}_{ij}^{k+}$ when i and j are bounding boxes containing the same person and $\mathbf{p}_{ij}^{k-}$ otherwise. Note that for learning we use differences on patches from the same location k.

3.2.1 KISS metric learning

Köstinger *et al.* [19] proposed an effective and efficient way of learning a Mahalanobis metric by assuming a Gaussian structure of the difference space (*i.e.* $\mathbf{p}_{ij}^k$). When employing KISSME our patch dissimilarity measure becomes

$$\Phi(\mathbf{p}_i^k, \mathbf{p}_j^k; \theta(k)) = (\mathbf{p}_i^k - \mathbf{p}_j^k)^T \mathbf{M}^{(k)} (\mathbf{p}_i^k - \mathbf{p}_j^k), \tag{3}$$

thus $\theta(k) = \{\mathbf{M}^{(k)}\}$. To learn $\mathbf{M}^{(k)}$ we follow Köstinger [19] and assume a zero mean Gaussian structure on difference space and employ a log likelihood ratio test. This results in

$$\mathbf{M}^{(k)} = \Sigma_{k+}^{-1} - \Sigma_{k-}^{-1}, \tag{4}$$

where Σ_{k+} and Σ_{k-} are the covariance matrices of $\mathbf{p}_{ij}^{k+}$ and $\mathbf{p}_{ij}^{k-}$, respectively

$$\Sigma_{k+} = \sum (\mathbf{p}_{ij}^{k+})(\mathbf{p}_{ij}^{k+})^T, \tag{5}$$

$$\Sigma_{k-} = \sum (\mathbf{p}_{ij}^{k-})(\mathbf{p}_{ij}^{k-})^T. \tag{6}$$

Computing Eq. (4) requires inverting two covariance matrices. In practice, as $\mathbf{p}_i^k$'s are still relatively high dimensional (see Sec. 4.2.3), Σ_{k+} is often singular, thus Σ_{k+}^{-1} cannot be computed. As a result, dimensionality reduction on $\mathbf{p}_i^k$ is usually applied (*e.g.* PCA), which allows to invert Σ_{k+}. Keeping the dimensionality low also avoids overfitting. One can find the optimal number of principal components using cross-validation techniques. Liao *et al.* [25] proposed an alternative solution that simultaneously learns metric $\mathbf{M}$ and low dimensional subspace $\mathbf{W}$, referred to as XQDA.

3.2.2 XQDA metric learning

Using XQDA [25] the patch dissimilarity can be written as

$$\Phi(\mathbf{p}_i^k, \mathbf{p}_j^k; \theta(k)) = (\mathbf{p}_i^k - \mathbf{p}_j^k)^T (\mathbf{W}^{(k)}) \mathbf{M}^{(k)} (\mathbf{W}^{(k)})^T (\mathbf{p}_i^k - \mathbf{p}_j^k), \tag{7}$$

where

$$\mathbf{M}^{(k)} = \left((\mathbf{W}^{(k)})^T \Sigma_{k+} (\mathbf{W}^{(k)}) \right)^{-1} - \left((\mathbf{W}^{(k)})^T \Sigma_{k-} (\mathbf{W}^{(k)}) \right)^{-1}. \tag{8}$$

Original feature dimension d of $\mathbf{p}_i^k$ is reduced by subspace $\mathbf{W}^{(k)} \in \mathbb{R}^{d \times r}$. $\mathbf{W}^{(k)}$ is learned using the Generalized Rayleigh Quotient objective, solved by the generalized eigenvalue decomposition problem similar to LDA [25]. Notice that Σ_{k+} is computed in the original d-dimensional space, thus its singularity remains a problem. Liao *et al.* proposes to add a small regularizer to the diagonal of elements of Σ_{k+}, which is a common trick in LDA-like problems. This makes the estimation of Σ_{k+} more smooth and robust. As a result, learning parameters become $\theta(k) = \{\mathbf{W}^{(k)}, \mathbf{M}^{(k)}\}$.

3.2.3 LSSL metric learning

Yang *et al.* [42] introduced large scale similarity learning (LSSL) that combines feature difference ($\mathbf{p}_{ij}^k = \mathbf{p}_i^k - \mathbf{p}_j^k$) and commonness ($\mathbf{q}_{ij}^k = \mathbf{p}_i^k + \mathbf{p}_j^k$), thus producing more discriminative measure. The main idea comes from insights found in a 2-dimensional Euclidean space. Consider the ℓ_2-normalized 2-dimensional feature space. Notice that for similar vectors (i and j containing the same person) $\mathbf{p}_{ij}^k$ is expected to be small but $\mathbf{q}_{ij}^k$ should be very large, in contrary for dissimilar vectors (i and j containing different people) $\mathbf{p}_{ij}^k$ is expected to be large and $\mathbf{q}_{ij}^k$ should be relatively small. Therefore, by combining difference and commonness we can expect more discriminative metric compared to metric learning methods that only employ differences $\mathbf{p}_{ij}^k$. The patch dissimilarity measure then becomes

$$\Phi(\mathbf{p}_i^k, \mathbf{p}_j^k; \theta(k)) = (\mathbf{p}_{ij}^k)^T \mathbf{M}_p^{(k)} (\mathbf{p}_{ij}^k)^T - \lambda (\mathbf{q}_{ij}^k)^T \mathbf{M}_q^{(k)} (\mathbf{q}_{ij}^k)^T, \tag{9}$$

where both $\mathbf{M}_p^{(k)}$ and $\mathbf{M}_q^{(k)}$ can be inferred analogically to KISS metric learning [19]. Yang *et al.* [42] shows further, that based on a pair-constrained Gaussian assumption, covariance for pairs containing different people ($\mathbf{p}_{ij}^{k-}$ and $\mathbf{q}_{ij}^{k-}$) can be directly deduced from image pairs containing the same person (for details see [42]). Parameter λ is used to balance between difference and commonness of feature vectors. Similarly to [42], we set $\lambda = 1.5$ in all experiments. As a result, learning parameters become $\theta(k) = \{\mathbf{M}_p^{(k)}, \mathbf{M}_q^{(k)}\}$ and PCA is applied on $\mathbf{p}_i^k$ to avoid the covariance singularity problem.

3.3 Integrated dissimilarities for images ($\mathcal{Z}$)

To compute the total dissimilarity between two bounding box images i and j, we propose several strategies for aggregating metrics learned for patches. First, we introduce a rigid model (PML) to illustrate that learning metric for patches keeps more information avoiding strong dimensionality reduction (Section 3.3.1). Additionally, learning metrics on patch level might effectively multiply the amount of training data yielding significant boost in recognition performance for smaller datasets.

Pose changes and different camera viewpoints make re-identification more difficult as features extracted on a fixed grid may not correspond even though it is the same person. Breaking a bounding box image into patches allows us to introduce *deformable* models that can effectively cope with pose changes, enabling patches in one bounding box to perturb their locations (deform) when matching to another bounding box. Independently to metric learning, our task is to find a strategy that can perturb patch locations to simulate pose changes. We investigate two deformable models (1) un unsupervised deformable model based one-to-one matching constraint (HPML) that does not require any additional training apart of metric learning (Section 3.3.2) and (2) a supervised deformable model with geometric constraints (conceptualized as springs) (DPML) that we train by introducing an optimization problem as a relatvie distance comparison of triplets (Section 3.3.3).

3.3.1 Rigid model (PML)

We combine patch dissimilarity scores by summing over all patches

$$\mathcal{Z}_{\text{PML}} = \sum_{k=1}^{K} \Phi(\mathbf{p}_i^k, \mathbf{p}_j^k; \theta(k)). \tag{10}$$

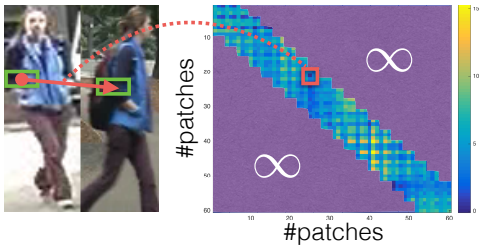


Fig. 2: Deformable models: $K \times K$ cost matrix, which is used as an input to the Hungarian algorithm for finding optimal one-to-one patch correspondence. The dissimilarity between two patches becomes ∞ if the distance between their spatial locations $\eta(\cdot,\cdot)$ is greater than assumed threshold δ.

Compared with the standard approach (*e.g.* in case of KISS metric), this is equivalent to learning a block diagonal matrix

$$\mathcal{Z}_{\text{PML}} = [\mathbf{p}_{ij}^1, \mathbf{p}_{ij}^2, \ldots, \mathbf{p}_{ij}^K] \begin{bmatrix} \mathbf{M}^1 & 0 & \ldots & 0 \\ 0 & \mathbf{M}^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & \mathbf{M}^K \end{bmatrix} \begin{bmatrix} \mathbf{p}_{ij}^1 \\ \mathbf{p}_{ij}^2 \\ \vdots \\ \mathbf{p}_{ij}^K \end{bmatrix} \tag{11}$$

where all $\mathbf{M}^{(k)}$ are learned independently. We refer to this formulation as **PML**.

3.3.2 Unsupervised Deformable Model (HPML)

Patch-based methods [2], [34] often allow patches to adjust their locations when comparing two bounding box images. Sheng *et al.* [34] assumed the correspondence structure to be fixed and learned it using a boosting-like approach. Instead, we define the patch correspondence task as a linear assignment problem. Given K patches from bounding box image i and K patches from bounding box image j we create a $K \times K$ cost matrix that contains patch similarity scores within a fixed neighborhood (see Fig 2). To avoid patches freely changing their location, we introduce a global one-to-one matching constraint and solve a linear assignment problem

$$\begin{aligned} \mathbf{\Omega}_{ij}^* &= \arg\min_{\mathbf{\Omega}_{ij}} \left(\sum_{k=1}^{K} \Phi(\mathbf{p}_i^{\mathbf{\Omega}_{ij}(k)}, \mathbf{p}_j^k; \theta(k)) + \Delta(\mathbf{\Omega}_{ij}(k), k) \right), \\ \text{s.t.} \quad \Delta(\mathbf{\Omega}_{ij}(k), k) &= \begin{cases} \infty, & \eta(\mathbf{\Omega}_{ij}(k), k) > \delta; \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \tag{12}$$

where $\mathbf{\Omega}_{ij}$ is a permutation vector mapping patches $\mathbf{p}_i^{\mathbf{\Omega}_{ij}(k)}$ to patches $\mathbf{p}_j^k$ and $\mathbf{\Omega}_{ij}(k)$ and k determine patch locations, $\Delta(\cdot,\cdot)$ is a spatial regularization term that constrains the search neighborhood, where η corresponds to distance between two patch locations and threshold δ determines the allowed displacement (different δ's are evaluated in Fig 15(a)). We find the optimal assignment $\mathbf{\Omega}_{ij}^*$ (patch correspondence) using the Kuhn-Munkres (Hungarian) algorithm [20]. This yields the total dissimilarity:

$$\mathcal{Z}_{\text{HPML}} = \sum_{k=1}^{K} \Phi\left(\mathbf{p}_i^{\mathbf{\Omega}_{ij}^*(k)}, \mathbf{p}_j^k; \theta(k)\right). \tag{13}$$

We refer to this formulation as **HPML**.

3.3.3 Supervised Deformable model (DPML)

We employ a model which approximates continuous non-affine warps by translating 2D templates [12], [43] (see Fig. 1). We use a spring model to limit the displacement of patches. The deformable dissimilarity score for matching the patch at location k in bounding box i with bounding box j is defined as

$$\psi(\mathbf{p}_i^k, j) = \min_l \left[\Phi(\mathbf{p}_i^k, \mathbf{p}_j^l; \theta(k)) + \alpha_k \Delta(k,l)\right], \qquad (14)$$

where patch feature $\mathbf{p}_j^l$ is extracted from bounding box j at location l; appearance term $\Phi(\mathbf{p}_i^k, \mathbf{p}_j^l; \theta(k))$ computes the feature dissimilarity between patches and deformation cost $\alpha_k \Delta(k,l)$ refers to a spring model that controls the relative placement of patches k and l. $\Delta(k,l)$ is the squared distance between the patch locations. α_k encodes the rigidity of the spring: $\alpha_k = \infty$ corresponds to a rigid model, while $\alpha_k = 0$ allows a patch to change its location freely. Notice the difference to **HPML**, for which the definition of Δ allows us to perform discrete optimization ($\mathbf{\Omega}^*$ stands for optimal global one-to-one assignment). For **DPML** we define Δ a continues function and we first optimize the patch alignment locally ($\psi(\mathbf{p}_i^k, j)$) and then combine these deformable dissimilarity scores into a unified dissimilarity measure

$$\begin{aligned} \mathcal{Z}_{\text{DPML}} &= \sum_{k=1}^{K} w_k \psi(\mathbf{p}_i^k, j) \\ &= \langle \mathbf{w}, \psi_{ij} \rangle, \end{aligned} \qquad (15)$$

where $\mathbf{w}$ is a vector of weights and ψ_{ij} corresponds to a vector of patch dissimilarity scores.

Learning α_k and $\mathbf{w}$: Similarly to [29], we define the optimization problem as a relative distance comparison of triplets $\{i, j, z\}$ such that $\langle \mathbf{w}, \psi_{iz} \rangle > \langle \mathbf{w}, \psi_{ij} \rangle$ for all i, j, z; where i and j correspond to bounding boxes extracted from different cameras containing the same person, and i and z are bounding boxes from different cameras containing different people. Unfortunately, Eq. 14 is non-convex and we can not guarantee avoiding local minima. In practice, we use a limited number of unique spring constants α_k and apply two-step optimization. First, we optimize α_k with $\mathbf{w} = \mathbf{1}$, by performing exhaustive grid search (see Section 4.3) while maximizing Rank-1 recognition rate. Second, we fix α_k and determine the best $\mathbf{w}$ using structural SVMs [18]. This approach is referred to as **DPML**.

3.4 Deep patches

It is common practice in person re-identification to combine handcrafted color and texture descriptors for describing image regions and then let metric learning to discover relevant features and discard irrelevant ones. Often color histograms in different color spaces together with SIFT-like features are concatenated into high-dimensional feature vectors [2]. Xiao *et al.* [40] showed that CNN models also can effectively be applied to person re-identification despite of insufficient data. They proposed to train jointly the CNN with data from multiple datasets and then fine-tune the model to a given camera pair using a domain-guided dropout strategy. In this work we adopt the CNN model from [40], but instead of training it for whole images, we train it for patches to obtain highly robust feature representation. This model learns a set of high-level feature representations through challenging multi-class identification tasks, *i.e.* , classifying a training image into one of C identities. As the generalization capabilities of the learned features increase with the number of classes predicted during training [36], we need C to be relatively large (*e.g.* several thousand). While training the CNN for patches, we modify the training strategy. First, each image is divided into a set of 8 non-overlapping patches of size height$/4$ $\times$ width$/2$ and then each patch (although comes from the same image but from different location) gets assigned a new identity. As a result, the CNN model is forced to determine not only the person identity from which the patch has been extracted but also the patch location. Given a dataset with images of M identities, the task becomes to classify patches into $C = 8M$ identities. When the CNN is trained to classify a large number of identities and configured to keep the dimension of the last hidden layer relatively low (*e.g.* , setting the number of dimensions for fc7 to 256 [40]), it forms compact and highly robust feature representations for re-identification. We found that the learned deep patch feature representation is very effective and combined with metric learning approaches it significantly outperforms state-of-the-art techniques. Figure 3 explains the training of our deep patch features.

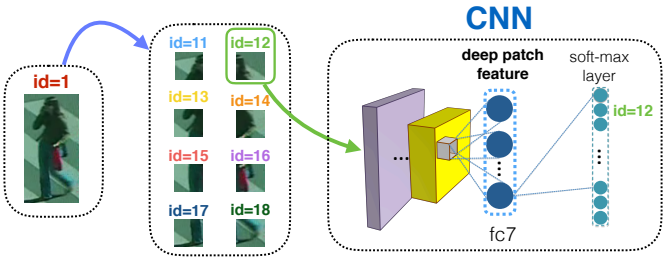


Fig. 3: **Deep patch feature learning** with the CNN: each image is divided into a set of 8 non-overlapping patches. The identity of each patch is extended by its location. As a result, the CNN is forced to recognize not only the person identity but also the patch location.

4 Experiments

We carry out experiments on four challenging datasets: **VIPeR** [13], **i-LIDS** [49], **CUHK01** [22] and **CUHK03** [23]. The results are analyzed in terms of recognition rate, using the *cumulative matching characteristic* (CMC) [13] curve its rank-1 accuracy. The CMC curve represents the expectation of finding the correct match in the top r matches. The curve can be characterized by a scalar value computed by normalizing the area under the curve referred to as $nAUC$ value.

Section 4.1 describes the benchmark datasets used in the experiments. We explore our rigid patch metric model (PML) together with its parameters, including different metric learning approaches (θ) in Section 4.2. The deformable models (HPML and DPML) are discussed in Section 4.3. Finally, in Section 4.4, we compare our performance to other state of the art methods.

4.1 Datasets

VIPeR [13] is one of the most popular person re-identification datasets. It contains 632 image pairs of pedestrians captured by two outdoor cameras. VIPeR images contain large variations in lighting conditions, background, viewpoint, and image quality (see Fig. 4). Each bounding box is cropped and scaled to be 128×48 pixels. We follow the common evaluation protocol for

Fig. 4: Sample images from **VIPeR** dataset. Top and bottom lines correspond to images from different cameras. Columns illustrate the same person.

Fig. 5: Sample images from **i-LIDS** dataset. Top and bottom lines correspond to images from different cameras. Columns illustrate the same person.

this database: randomly dividing 632 image pairs into 316 image pairs for training and 316 image pairs for testing. We repeat this procedure 10 times and compute the average CMC curves for obtaining reliable statistics.

i-LIDS [49] consists of 119 individuals with 476 images. This dataset is very challenging since there are many occlusions. Often only the top part of the person is visible and usually there is a significant scale or viewpoint change as well (see Fig. 5). We follow the evaluation protocol of [29]: the dataset is randomly divided into 60 image pairs used for training and the remaining 59 image pairs are used for testing. This procedure is repeated 10 times for obtaining averaged CMC curves.

CUHK01 [22] contains 971 persons captured with two cameras. For each person, 2 images for each camera are provided. The images in this dataset are better quality and higher resolution than in the two previous datasets. Each bounding box is scaled to be 160×60 pixels. The first camera captures the side view of pedestrians and the second camera captures the frontal view or the back view (see Fig. 6). We follow the common evaluation setting: the persons are split into 485 for training and 486 for testing. We repeat this procedure 10 times for computing averaged CMC curves.

CUHK03 [23] is one of the largest published person re-identification datasets. It contains 1467 persons, where each person has 4.8 images on average. The dataset provides both the manually cropped bounding box images and the automatically detected bounding box images with a pedestrian detector [12]. For evaluation we follow the testing protocol of [23]: the identities are randomly divided into non-overlapping training and test sets. The training set consists of 1367 persons and the test set consists of 100 persons. For testing we only use the automatically detected pedestrians, while training is performed employing both the manually cropped and the automatically detected images. We follow a single-shot setting.

Fig. 6: Sample images from **CUHK01** dataset. Top and bottom lines correspond to images from different cameras. Columns illustrate the same person.

Training deep features: To learn our deep patch representation we used two datasets: **CUHK03** [23] and **PRID2011** [17]. From **CUHK03** we used 1367 identities that were randomly selected for the training [40]. **PRID2011** contains 200 individuals appearing in two cameras and additionally it contains 185 identities that appear in the first camera but do not reappear in the second one, and 549 identities that appear only in the second camera, in total 934 identities. Merging both datasets, we have $M = 2301$ identities and by further dividing images into a set of 8 non-overlapping patches the CNN is forced to perform multi-class identification of $C = 8 \times 2301 = 18408$ identities. The dimensionality of the last hidden layer is kept to be low (256), which stands for our deep feature representation. The architecture and training parameters are kept the same as in [40]. Unlike [40], we do not perform any fine-tuning of the deep patch feature representation on test datasets. Instead, we proposed to perform Mahalanobis metric learning to adjust to the metric to particular camera-pair variations.

4.2 Rigid Patch Metric Learning (PML)

In this section, we first compare our rigid patch model (PML) to the standard full bounding box approach (BBOX). BBOX is equivalent to the method presented in [19].

Each bounding box of size $w \times h$ is divided into a grid of $K = 60$ overlapping patches of size $\frac{w}{4} \times \frac{w}{2}$ with stride $\frac{w}{8} \times \frac{w}{4}$ resulting in a 20×3 layout (different patch layouts are discussed in Section 4.2.2). In this experiment, patches are represented by concatenated histograms in LAB and HSV color space together with color SIFT (see details on different patch representations in Section 4.2.3). For the full bounding box case, we concatenate the extracted patch feature vectors into a high dimensional feature vector. PCA is applied to obtain a 62-dimensional feature space (where the optimal dimensionality is found by cross-validation). Then, the KISS metric [19] is learned in the 62-dimensional PCA subspace. For PML, instead of learning a metric for the concatenated feature vector, we learn metrics for patch features. In this way, we avoid undesirable compression. The dimensionality of the patch feature vector is reduced by PCA to 35 (also found by cross-validation) and metrics are learned independently for each patch location. Fig. 7 illustrates the comparison on three datasets. It is apparent that PML significantly improves the re-identification performance by keeping a higher number of degrees of freedom (35×60) when learning the dissimilarity function.

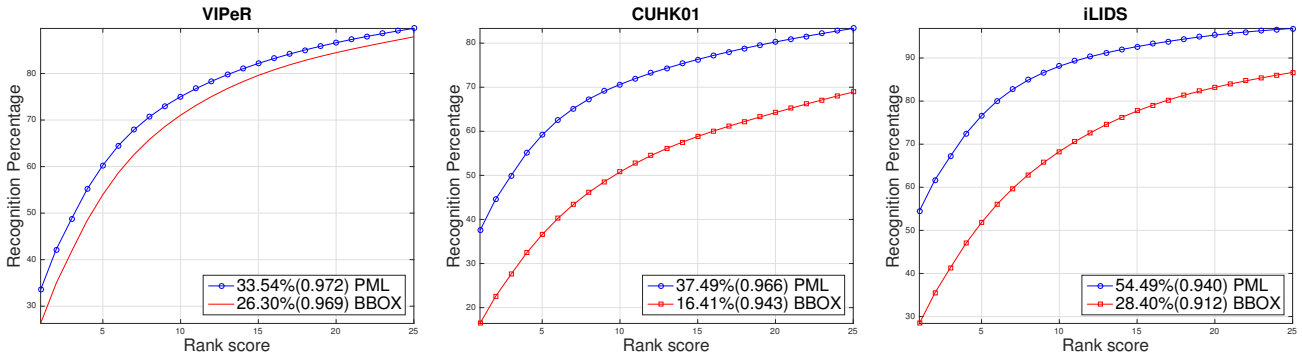


Fig. 7: Performance comparison of Patch based Metric Learning (PML) *vs.* full bounding box metric learning (BBOX). Rank-1 identification rates as well as $nAUC$ values provided in brackets are shown in the legend next to the method name.

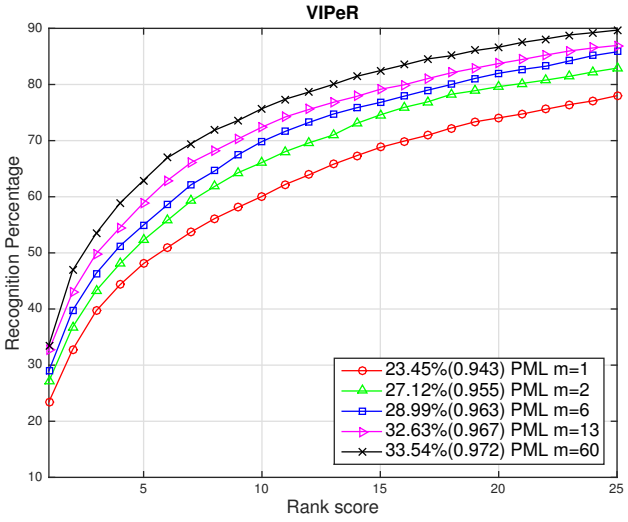
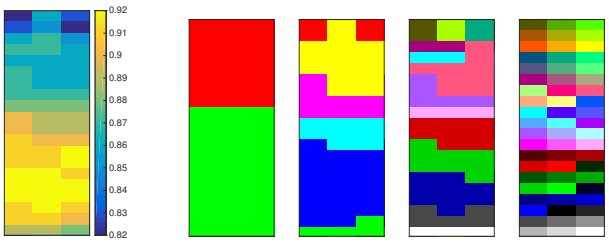


Fig. 8: Performance comparison *w.r.t.* the number of $\mathbf{M}$. Using different metrics for different image regions yields better performance.



(a) $nAUC$ (b) clusters

Fig. 9: Dividing image regions into several metrics. (a) $nAUC$ values *w.r.t.* a location of a learned metric; (b) clustering results for different number of clusters m.

4.2.1 Number of Patch Metrics

As mentioned earlier, our formulation allows θ to be learned per patch location. In practice, there may be insufficient training data for this many degrees of freedom. We evaluate two extremes: learning $m = 60$ independent KISS metrics (one per patch location) and learning a single KISS metric for all 60 patches ($m = 1$), see Fig. 8. The results indicate that multiple metrics lead to significantly better recognition accuracy.

To understand the variability in the learned metrics, we setup the following experiment: learn a metric for a particular location k, and then apply this metric to compute dissimilarity scores for all other patch locations. We plot $nAUC$ values *w.r.t.* to the location of the learned metric in Fig. 9(a). It is apparent that metrics learned at different locations yield different performances. Surprisingly, higher performance is obtained by metrics learned on patches at lower locations within the bounding box (corresponding to leg regions). We believe that it is due to significant number of images in the VIPeR dataset having dark and cluttered backgrounds in the upper regions (see the last 3 top images in Fig. 4). Lower parts of the bounding boxes usually have more coherent background from sidewalks.

Additionally, we cluster patch locations spatially using hierarchical clustering (bottom-up), where similarity between regions is computed using $nAUC$ values. Fig. 9(b) illustrates clustering results *w.r.t.* to the number of clusters. Next, we learn metrics for each cluster of patch locations. These metrics are then used for computing patch similarity in corresponding image regions. Recall from Fig. 8 that the best performance was achieved with $m = 60$. In this circumstance, there appears to be sufficient data to train an independent metric for each patch location. We test this hypothesis by reducing the amount of training data and evaluating the optimal number of patch metrics when fewer training examples are available. Fig. 10 illustrates that the patch-based approach achieves high performance much faster than full bounding box metric learning. Interestingly, for a small number of positive pairs (less than 100), a reduced number of metrics gives better performance. When a common metric is learned for multiple patch locations, the amount of training data is effectively increased because features from multiple patches can be used as examples for learning the same metric (Section 3.1).

4.2.2 Patch layout

Our model consists of a set of rectangular patches extracted on a grid layout. The size of the patch and the grid density (determined by a stride) define the total number of patches K and a level of patch overlap. To investigate the impact of these parameters on the re-identification performance, we evaluate PML with the KISS metric for different patch layouts. Fig. 11(a) shows the results for different K defined by different patch sizes and different

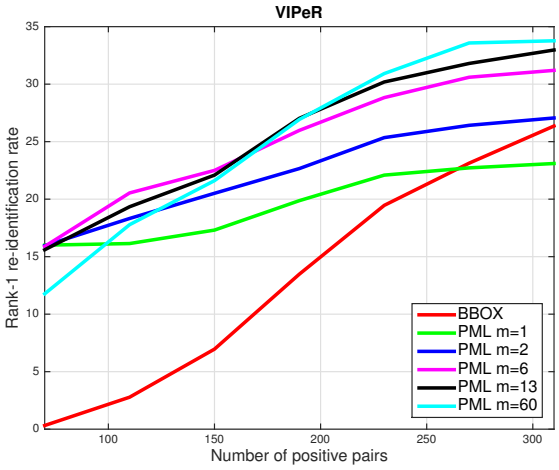


Fig. 10: Rank-1 recognition rate with varying size of training dataset.

strides (*e.g.* $K = 60$ is a result of patch size $\frac{w}{4} \times \frac{w}{2}$ with stride $\frac{w}{8} \times \frac{w}{4}$ and for $K = 8$ stride dimensions are equal to the patch size, which corresponds to a configuration of non-overlapping patches). From Fig. 11(a) we can notice that having small patches (*e.g.* for $K = 140$ with patch size $\frac{w}{4} \times \frac{w}{4}$) might slightly decrease the performance, and in general keeping patches larger (see $K = 39$ and $K = 60$) yields better recognition accuracy. The results also indicate that overlapping patches (when the stride is smaller than the patch size) perform significantly better than non-overlapping patches ($K = 8$ and $K = 20$ in Fig. 11(a) correspond to layouts with non-overlapping patches). Similarly, using overlapping deep patch features yields better performance compared with non-overlapping patches (see Fig. 11(b)). As a result, in further performance evaluations we select layouts that consist of overlapping patches for both handcrafted features as well as deep patch features. For handcrafted features we select $K = 60$ – the best performing configuration in Fig. 11(a). For deep patches we also select a configuration with overlapping patches but with $K = 39$ to match the patch size used during the deep patch training (see Section 3.4). Notice that we train the deep patch feature using non-overlapping patches, which we found to perform slightly better.

4.2.3 Patch representation

It is common practice in person re-identification to combine color and texture descriptors for describing an image. We evaluated the performance of different combinations of representations, including Lab, RGB and HSV histograms, each with 30 bins per channel. Texture information was captured by color SIFT, which is the SIFT descriptor extracted for each Lab channel and then concatenated. In our previous work [2], we selected the combination of Lab, HSV and color SIFT as the best descriptor. The dimensionality of concatenated HSV, Lab and color SIFT is 564 ($30 \times 3 + 30 \times 3 + 128 \times 3 = 564$). In this work, instead of handcrafting the patch representation, we propose to learn patch features directly from data with CNNs through multi-class identification task (Sec. 3.4). As a result, each deep patch is represented by 256-dimensional feature vector (fc7) and we reduce its dimensionality to 60 by PCA before running KISS metric learning. Fig. 12(a) illustrates the averaged CMC curves for VIPeR data set. It is clear that the proposed deep patch representation (CNN) outperforms all handcrafted representations by a large margin.

Method	VIPeR		CUHK01		iLIDS	
	H-C	CNN	H-C	CNN	H-C	CNN
PML, KISS	33.5	**43.2**	37.4	**61.5**	54.4	**74.8**
PML, XQDA	29.5	**37.6**	39.2	**49.2**	57.8	**73.2**
PML, LSSL	37.1	**47.0**	42.7	**71.3**	60.7	**78.3**

TABLE 1: Performance comparison of different metric learning approaches using handcrafted features (HSV+Lab+ColorSIFT) denoted by **H-C** and deep patches **CNN**. CMC rank-1 accuracies are reported.

When learning the patch representation, we propose to force the CNN to recognize not only the person identity but also the patch location. To evaluate the effectiveness of our approach we also trained the CNN for patches, while neglecting the patch locations (**CNN (-k)**). Fig. 12(b) illustrates that including information on the patch locations allows us to learn more effective features. The CNNs learned with patch locations perform significantly better for both L2 and KISS metric learning.

4.2.4 Patch metric learning

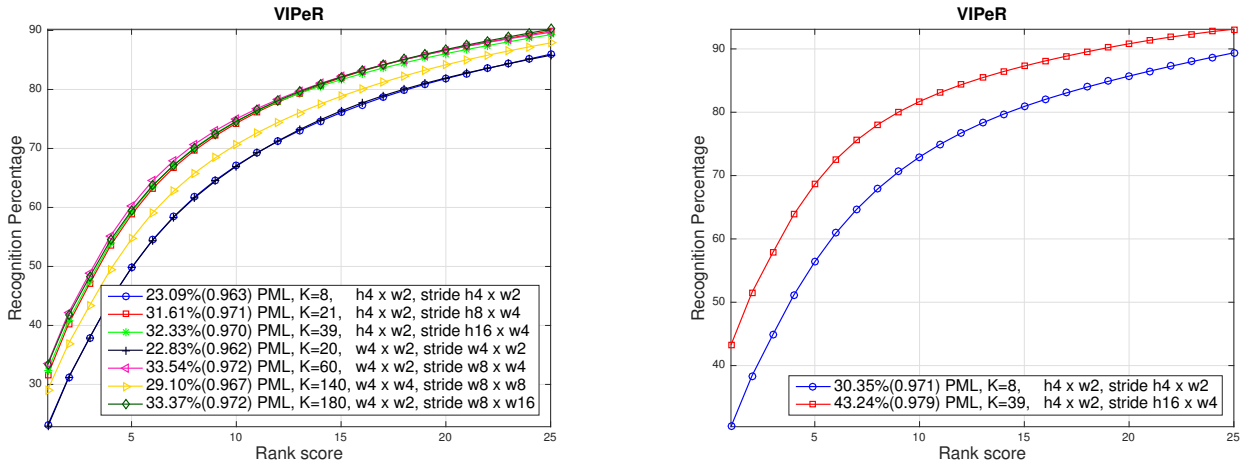
In this section we evaluate our PML model, while employing previously discussed metric learning approaches: KISS metric learning [19], XQDA metric learning [25] and LSSL metric learning [42]. We investigate the performance while employing both handcrafted features (HSV+Lab+ColorSIFT) and deep patch features. As the **i-LIDS** dataset contains a relatively small number of training samples (only 60 subjects available for training) and as indicated in our previous analysis (Section 4.2.1), we learn a single θ for all patches, thus increasing the amount of training examples ($m = 1$). From Fig. 13 it is apparent, that deep patch features significantly improve the recognition accuracy on all datasets. It is also clear that LSSL metric learning consistently achieves the best performance among all metric learning approaches. Surprisingly, XQDA often performs worse than standard KISS metric learning, especially when using deep patches. This discrepancy might be due to the fact that deep patches are already highly discriminative and applying additional discriminative objectives (the Generalized Rayleigh Quotient) may decrease the performance. Table 1 summarizes the results.

Additionally we evaluated the performance of the CNN model while learning on the whole images (equivalent to JSTL model [40]) combined with metric learning approaches. Recall that we only use **CUHK03** and **PRID2011** datasets for training deep features. Each image is represented by 256-dimensional feature vector and we reduce its dimensionality to 50 by PCA before running KISS and LSSL metric learning (found by cross-validation). From Fig. 14 it is apparent that metric learning improves recognition accuracy of global deep features (compare with JSTL, L2). However, it is also clear the best performance of JSTL combined with metric learning is far behind the proposed patch-based learning combined with our deep patch representation.

4.3 Deformable Patch Metric Learning

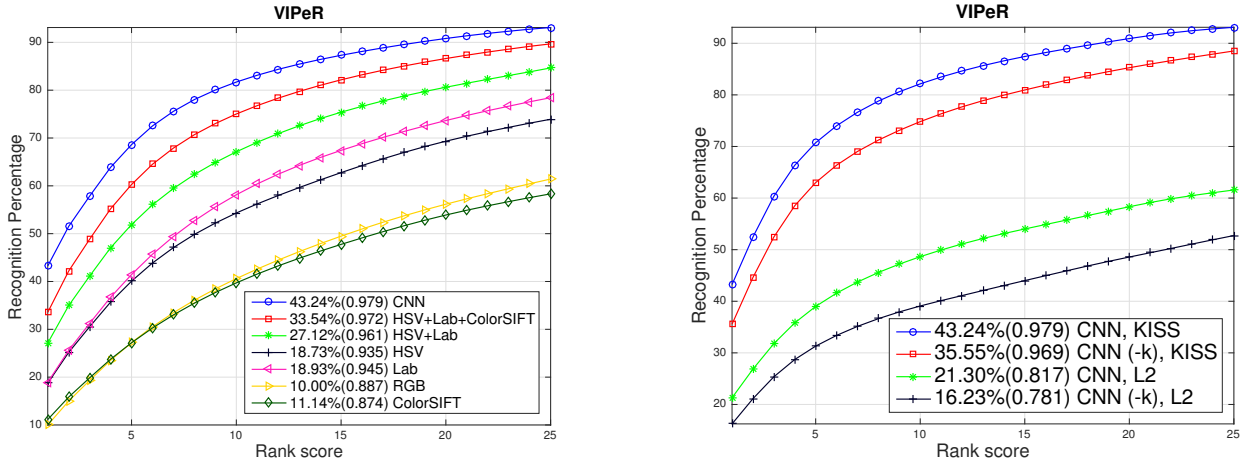
4.3.1 Unsupervised Deformable Model (HPML)

Fig. 15(a) illustrates the impact of our unsupervised deformable model on recognition accuracy. We also compare the effectiveness of different neighborhoods on the overall accuracy. In Eq. (12), we



(a) HSV+Lab+ColorSIFT (b) CNN

Fig. 11: Performance comparison on VIPeR dataset *w.r.t.* different patch layouts; (a) using handcrafted features – HSV+Lab+ColorSIFT; (b) using deep patches – CNN. Overlapping patches yield better performance.



(a) (b)

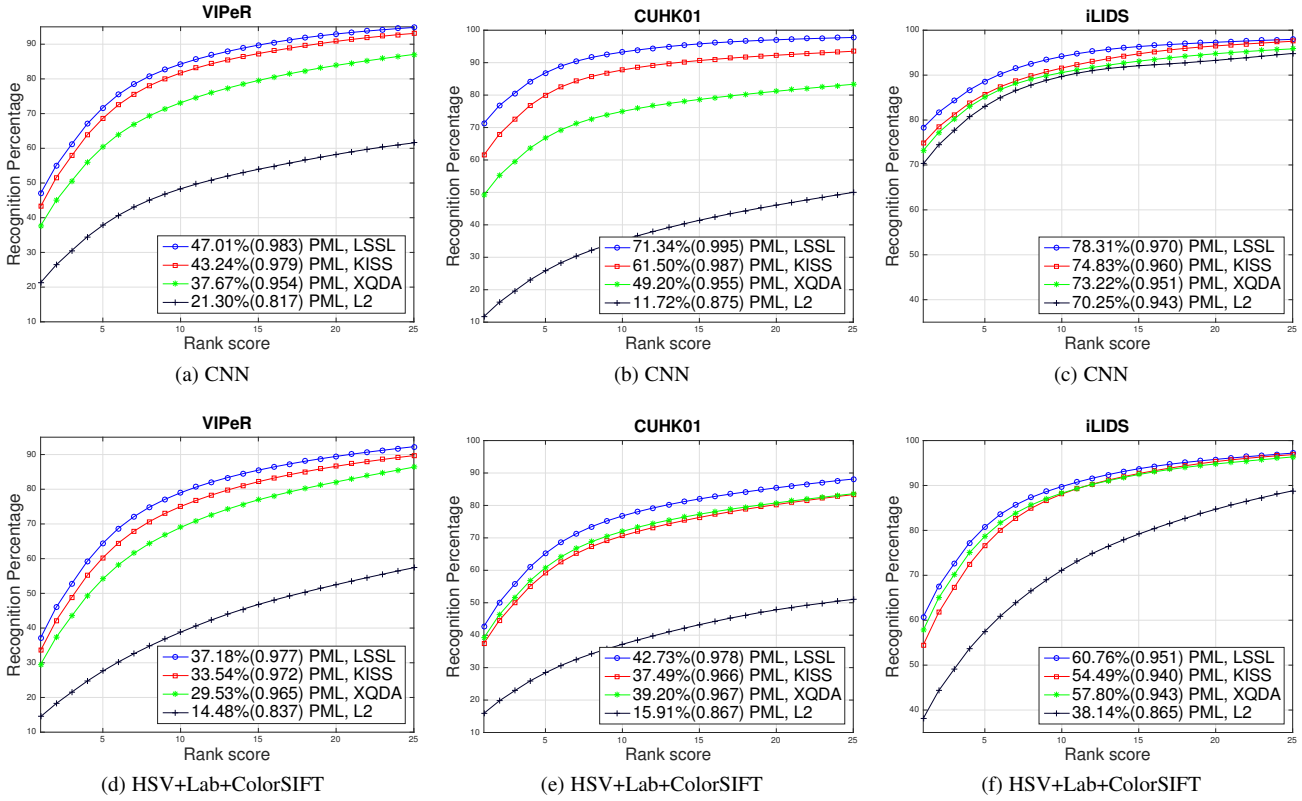
Fig. 12: Performance comparison of different patch descriptors for VIPeR dataset; (a) the best performance is achieved by our deep patch representation (CNN); (b) forcing the CNN to determine the patch locations along with the person identities increases the effectiveness of the deep features: **CNN (-k)** corresponds to the CNN trained without patch locations, and **CNN** was learned with patch locations.

constrain the displacement of patches to $\delta_{\text{horizontal}} \times \delta_{\text{vertical}}$ number of pixels. Interestingly, allowing patches to move vertically ($\delta_{\text{vertical}} > 0$) generally decreases performance. We believe that this is due to the fact that images in all of these datasets were annotated manually and vertical alignment (from the head to the feet) of people in these images is usually correct. Allowing patches to move horizontally consistently improves the performance for all datasets. The highest gain in accuracy is obtained on the iLIDS dataset (3%), which contains inaccurate detections and large amount of occlusions. This indicates that our linear assignment approach provides a reliable solution for pose changes.

4.3.2 *Supervised Deformable model (DPML)*

We simplify Eq. 14 by restricting the number of unique spring constants. Two parameters α_1, α_2 are assigned to patch locations obtained by hierarchical clustering with the number of clusters $m = 2$ (see Fig. 15(b)). α_k encodes the rigidity of the patches at particular locations. We perform an exhaustive grid search iterating through α_1 and α_2 while maximizing Rank-1 recognition rate. Fig 15(b) illustrates the recognition rate map as a function of both coefficients. Interestingly, rigidity (high spring constants) is useful for lower patches (the dark red region in the left-bottom corner of the map) but not so for patches in the upper locations of the bounding box. This might be related to the fact that metrics learned on the lower locations have higher performance (compare with $nAUC$ values in Fig. 9).

Fig. 16 illustrates the performance comparison of different patch integration functions $\mathcal{Z}$. We employ LSSL metric learning together with deep patch features. The results clearly show that introducing deformable models consistently improves the recognition accuracy in all datasets and that the best performance is obtained by the supervised spring model **DPML**.



(a) CNN (b) CNN (c) CNN

(d) HSV+Lab+ColorSIFT (e) HSV+Lab+ColorSIFT (f) HSV+Lab+ColorSIFT

Fig. 13: Performance comparison of different metric learning approaches using **deep patches – CNN** – top row and **handcrafted features - HSV+Lab+ColorSIFT** – bottom row. LSSL metric learning performs the best among all metric learning techniques.

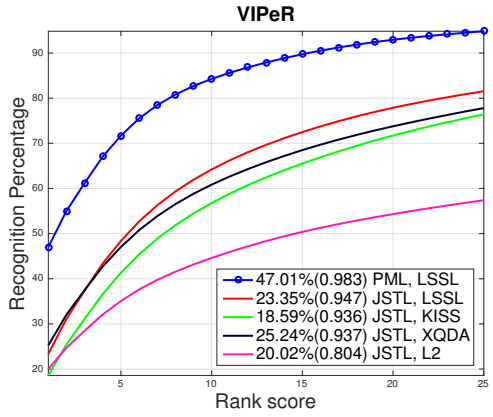
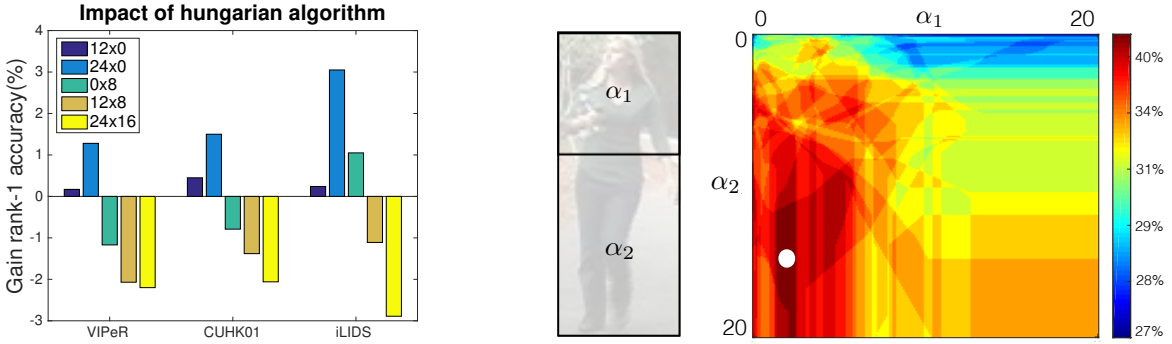


Fig. 14: Performance comparison of global deep features (JSTL model) combined with metric learning approaches *vs.* Patch-based Metric Learning (**PML**) based on the proposed deep patch features. Deep patch features significantly outperform global deep features.

Computational complexity Although the rigid model (**PML**) does not perform as good as deformable models, it is less computationally expensive. It requires only K similarities to be computed to compare two images. However, although **HPML** requires solving Hungarian algorithm (Eq. (12)), in practice the matrix $K \times K$ (see Fig. 2) can be relatively sparse (compare the performance of different neighborhoods in Fig. 15(a)). Given τ non-infinite entries in this matrix, we employed QuickMatch algorithm [28] that runs in linear time $\mathcal{O}(\tau)$. As a result, the deep texture feature extraction is the slowest part and it depends on the GPU architecture (*e.g.* on Tesla K80 VIPeR experiment takes 330s, with 310s spent on deep feature extraction). **DPML** is the slowest model and the same experiment takes around 30min.

4.4 Comparison with Other Methods

Table. 2 reports the performance comparison of our patch-based methods with state-of-the-art approaches across 4 datasets: **VIPeR**, **CUHK01**, **iLIDS** and **CUHK03**-detected. Our methods outperform all state-of-the-art techniques on all datasets. The maximum improvement is achieved on the iLIDS dataset. We improve the state-of-the-art rank-1 accuracy (64.6%) by almost 18% (82.2%). This dataset contains a relatively small number of training samples (we use only 60 subjects for training). Driven by our previous analysis (Section 4.2.1), we learn a single θ for all patches, thus increasing the training set. As a result, PML, HPML and DPML significantly outperform the state-of-the-art approaches. There are three aspects that make our approach more effective on iLIDS: (1) we are able to generate a significantly larger training set using $m = 1$, (2) occlusions in images pollute only a few patch scores in our similarity measure, while in case of full-image based metric learning they might have a global impact on the final dissimilarity measure, (3) misaligned features can be corrected by our deformable models. Notice, that our simplest patch aggregation technique (PML) already achieves very competitive results. This highlights effectiveness of combining patch driven LSSL metric learning with deep patch representation.



(a) **HPML** (b) **DPML**

Fig. 15: Deformable model parameters: (a) **HPML** – comparison of different allowable neighborhoods (horizontal×vertical) when applying Hungarian algorithm for matching patches; (b) **DPML** – exhaustive grid search over α_1 and α_2 coefficients for VIPeR. α_1 and α_2 correspond to patches locations *w.r.t.* to the left image. Grid search map illustrates Rank-1 recognition rate as a function of (α_1, α_2). The white dot highlights the optimal operating point.

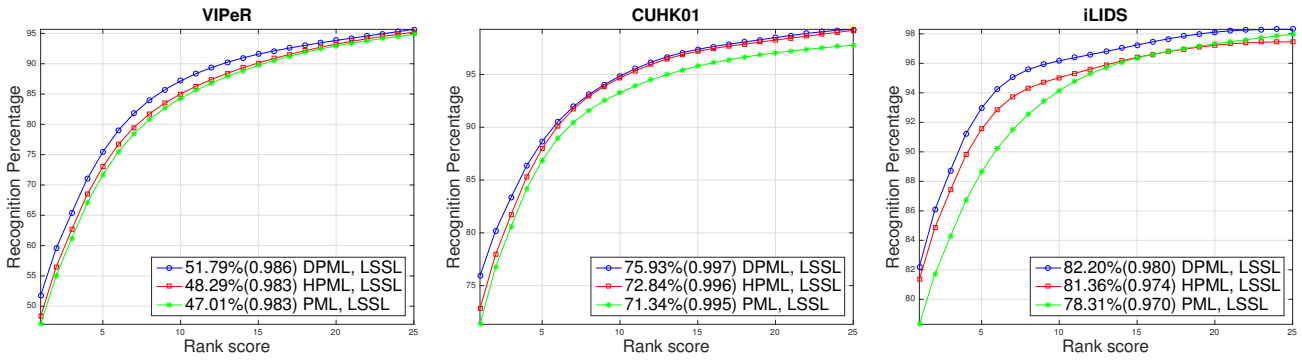


Fig. 16: Performance comparison of Patch based Metric Learning (**PML**) with our deformable models: unsupervised **HPML** and supervised **DPML**. Rank-1 identification rates as well as $nAUC$ values provided in brackets are shown in the legend next to the method name.

METHOD	VIPeR	CUHK01	iLIDS	CUHK03
DPML-CNN	**51.7**	**75.9**	**82.2**	**84.0**
HPML-CNN	48.2	**72.8**	**81.3**	**82.1**
PML-CNN	47.0	71.3	78.3	80.6
DPML [2]	41.4	35.8	57.6	-
PML [2]	33.5	30.6	51.6	-
eSDC [46]	26.7	15.1	36.8	-
SDALF [11]	19.9	9.9	41.7	-
TL [31]	34.1	32.1	50.3	-
Dropout [40]	38.6	66.6	64.6	75.3
KISSME [19]	19.6	16.4	28.4	-
LOMO+XQDA [25]	40.0	63.2	46.3	
Mirror [6]	42.9	40.4	-	
Ensembles [29]	45.9	53.4	50.3	62.1
MidLevel [47]	29.1	34.3	-	-
kLDFA [41]	32.8	-	40.3	-
DeepNN [1]	34.8	47.5	-	45.0
Null Space [44]	42.2	64.9	-	53.7
Null Space (fusion) [44]	**51.1**	69.0	-	54.7
Triplet Loss [7]	47.8	53.7	60.4	-
Gaussian+XQDA [27]	49.7	57.8	-	-
Joint CNN [37]	35.7	71.8	-	52.2
Sample-Specific SVM [45]	42.6	65.9	-	57.0

TABLE 2: Performance comparison on **VIPeR**, **CUHK01**, **iLIDS** and **CUHK03**-detected; CMC rank-1 accuracies are reported. The best scores are shown in red. The second best scores are highlighted in blue. Our approach significantly outperforms the best state of the art approaches.

5 SUMMARY

Re-identification must deal with appearance differences arising from changes in illumination, viewpoint and a person's pose. Traditional metric learning approaches do not address registration errors and instead only focus on feature vectors extracted from bounding boxes. In contrast, we propose a patch-based approach. Operating on patches has several advantages:

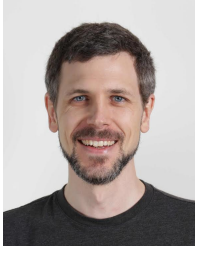
- Extracted feature vectors have lower dimensionality and do not have to be subject to the same levels of compression as feature vectors extracted for the entire bounding box.
- Multiple patch locations can share the same metric, which effectively increase the amount of training data.
- We allow patches to adjust their locations when comparing two bounding boxes. The idea is similar to part-based models used in object detection. As a result, we directly address registration errors while simultaneously evaluating appearance consistency.
- Learning the deep patch features directly from data and forcing the CNN to determine also the patch location results in highly effective patch representation.

Our experiments illustrate how these advantages lead to new state of the art performance on well established, challenging re-identification datasets.

REFERENCES

[1] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *CVPR*, 2015.
[2] S. Bak and P. Carr. Person re-identification using deformable patch metric learning. In *WACV*, 2016.
[3] S. Bak and P. Carr. One-shot metric learning for person re-identification. In *CVPR*, 2017.
[4] S. Bak, E. Corvee, F. Bremond, and M. Thonnat. Person re-identification using spatial covariance regions of human body parts. In *AVSS*, 2010.
[5] D. Chen, Z. Yuan, G. Hua, N. Zheng, and J. Wang. Similarity learning on an explicit polynomial kernel feature map for person re-identification. In *CVPR*, 2015.
[6] Y.-C. Chen, W.-S. Zheng, and J. Lai. Mirror representation for modeling view-specific transform in person re-identification. In *IJCAI*, 2015.
[7] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*, June 2016.
[8] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *BMVC*, pages 68.1–68.11, 2011.
[9] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.
[10] M. Dikmen, E. Akbas, T. S. Huang, and N. Ahuja. Pedestrian recognition with a learned metric. In *ACCV*, pages 501–512, 2010.
[11] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*, 2010.
[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2010.
[13] D. Gray, S. Brennan, and H. Tao. Evaluating Appearance Models for Recognition, Reacquisition, and Tracking. *PETS*, 2007.
[14] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, 2008.
[15] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, 2009.
[16] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *ECCV*, 2010.
[17] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof. Person re-identification by descriptive and discriminative classification. In *SCIA*, pages 91–102, 2011.
[18] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 2009.
[19] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, 2012.
[20] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2), 1955.
[21] I. Kviatkovsky, A. Adam, and E. Rivlin. Color invariants for person reidentification. *TPAMI*, 2013.
[22] W. Li, R. Zhao, and X. Wang. Human reidentification with transferred metric learning. In *ACCV*, 2012.
[23] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014.
[24] Z. Li, S. Chang, F. Liang, T. Huang, L. Cao, and J. Smith. Learning locally-adaptive decision functions for person verification. In *CVPR*, 2013.
[25] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, 2015.
[26] N. Martinel, C. Micheloni, and G. Foresti. Saliency weighted features for person re-identification. In *ECCV Workshops*, 2014.
[27] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato. Hierarchical gaussian descriptor for person re-identification. In *CVPR*, 2016.
[28] J. B. Orlin and Y. Lee. QuickMatch: A very fast algorithm for the Assignment Problem. Technical Report WP 3547-93, Massachusetts Institute of Technology, 1993.
[29] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. Learning to rank in person re-identification with metric ensembles. In *CVPR*, 2015.
[30] S. Pedagadi, J. Orwell, S. A. Velastin, and B. A. Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *CVPR*, 2013.
[31] P. Peng, T. Xiang, Y. Wang, M. Pontil, S. Gong, T. Huang, and Y. Tian. Unsupervised cross-dataset transfer learning for person re-identification. In *CVPR*, June 2016.
[32] B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by support vector ranking. In *BMVC*, 2010.
[33] Y. Shen, W. Lin, J. Yan, M. Xu, J. Wu, and J. Wang. Person re-identification with correspondence structure learning. In *ICCV*, 2015.
[34] H. Sheng, Y. Huang, Y. Zheng, J. Chen, and Z. Xiong. Person re-identification via learning visual similarity on corresponding patch pairs. In *KSEM*, 2015.
[35] H. Shi, Y. Yang, X. Zhu, S. Liao, Z. Lei, W. Zheng, and S. Z. Li. Embedding deep metric for person re-identification: A study against large variations. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *ECCV*, 2016.
[36] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, June 2014.
[37] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
[38] G. Wang, L. Lin, S. Ding, Y. Li, and Q. Wang. DARI: distance metric and representation integration for person verification. In *AAAI*, 2016.
[39] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2006.
[40] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016.
[41] F. Xiong, M. Gou, O. Camps, and M. Sznaier. Person re-identification using kernel-based metric learning methods. In *ECCV*, 2014.
[42] Y. Yang, S. Liao, Z. Lei, and S. Z. Li. Large scale similarity learning using similar pairs for person verification. In *AAAI*, 2016.
[43] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *TPAMI*, 2013.
[44] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *CVPR*, 2016.
[45] Y. Zhang, B. Li, H. Lu, A. Irie, and X. Ruan. Sample-specific svm learning for person re-identification. In *CVPR*, 2016.
[46] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *CVPR*, 2013.
[47] R. Zhao, W. Ouyang, and X. Wang. Learning mid-level filters for person re-identification. In *CVPR*, 2014.
[48] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian. MARS: A video benchmark for large-scale person re-identification. In *ECCV*, 2016.
[49] W.-S. Zheng, S. Gong, and T. Xiang. Associating groups of people. In *BMVC*, 2009.
[50] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*, 2011.
[51] W.-S. Zheng, X. Li, T. Xiang, S. Liao, J. Lai, and S. Gong. Partial person re-identification. In *ICCV*, 2015.

Sławomir Bąk is an Associate Research Scientist at Disney Research Pittsburgh. His research focuses on recognition (person re-identification), visual tracking, Riemannian manifolds and metric learning. Before joining Disney Research, he spent several years as a Postdoctoral Fellow and PhD student at INRIA Sophia Antipolis. He received his PhD from INRIA, University of Nice in 2012 for a thesis on *person re-identification*. Sławomir obtained his Master's Degree in Computer Science (Intelligent Decision Support Systems) from Poznan University of Technology in 2008.



Peter Carr is a Research Scientist at Disney Research, Pittsburgh. His research interests lie at the intersection of computer vision, machine learning and robotics. Peter joined Disney Research in 2010 as a Postdoctoral Researcher. Prior to Disney, Peter received his PhD from the Australian National University in 2010, under the supervision of Prof. Richard Hartley. Peter received a Master's Degree in Physics from the Centre for Vision Research at York University in Toronto, Canada, and a Bachelor's of Applied Science (Engineering Physics) from Queen's University in Kingston, Canada.