Exploiting View-Specific Appearance Similarities Across Classes for Zero-shot Pose Prediction: A Metric Learning Approach

Alina Kuznetsova
Leibniz University Hannover
Appelstr 9A, 30169
Hannover, Germany

Sung Ju Hwang
UNIST
50 UNIST-gil, 689798
Ulsan, Korea

Bodo Rosenhahn
Leibniz University Hannover
Appelstr 9A, 30169
Hannover, Germany

Leonid Sigal
Disney Research
4720 Forbes Avenue, 15213
Pittsburgh, PA, US

Abstract

Viewpoint estimation, especially in case of multiple object classes, remains an important and challenging problem. First, objects under different views undergo extreme appearance variations, often making within-class variance larger than between-class variance. Second, obtaining precise ground truth for real-world images, necessary for training supervised viewpoint estimation models, is extremely difficult and time consuming. As a result, annotated data is often available only for a limited number of classes. Hence it is desirable to share viewpoint information across classes. Additional complexity arises from unaligned pose labels between classes, i.e. a side view of a car might look more like a frontal view of a toaster, than its side view. To address these problems, we propose a metric learning approach for joint class prediction and pose estimation. Our approach allows to circumvent the problem of viewpoint alignment across multiple classes, and does not require dense viewpoint labels. Moreover, we show, that the learned metric generalizes to new classes, for which the pose labels are not available, and therefore makes it possible to use only partially annotated training sets, relying on the intrinsic similarities in the viewpoint manifolds. We evaluate our approach on two challenging multi-class datasets, 3DObjects and PASCAL3D+.

Introduction

One of the fundamental challenges in visual object recognition is dealing with the appearance variation of the objects due to the viewpoint changes. A *bicycle* and a *horse* might look very different depending on whether they are seen from the frontal or the side view. Therefore, multi-view recognition, or joint pose estimation and object recognition, has been an important topic in computer vision (Savarese and Fei-Fei 2007, Sun et al. 2009, Zhang et al. 2013) with recently resurgent interest (Bakry and Elgammal 2014, He, Sigal, and Sclaroff 2014, Tulsiani and Malik 2015) [1].

However, obtaining labels for object pose is very difficult. While one can potentially obtain (weak) class labels from the web (e.g., through Google/Flickr searches), pose data is not available through such mediums. Therefore pose labeling almost exclusively requires manual annotation, but even that is not straightforward as people are not consistent in their definition of canonical viewpoints (e.g., some may define the side view of a bicycle to be zero-degrees, others perhaps ninety-degrees) and notoriously bad at fine grained viewpoint estimation.

In more extreme case of symmetric objects, such as a ball or a vase, finding object pose is an ill-posed problem since one cannot make visual distinction between different pose orientations of the object. Even if we provide canonical pose for each object and do not consider such ill-posed cases, the fact that people are bad at the task, requires expensive alignment of templates to find the precise pose (Xiang et al (2014)). Another practical problem is that given photographer bias, in many real-world datasets, the observed pose variations might not be large enough to cover the entire view-based appearance space of the object (Chen and Grauman 2014).

To cope with such difficulties for constructing supervised pose datasets, it would be convenient to minimize such labeling effort by possibly transferring knowledge about pose from one class to another. To this end, we propose a method to exploit common appearances across classes for the task of joint categorization and pose estimation. Our idea is based on the intuition that in many cases, an object could appear more similar to another object from a different class in the same viewpoint, than to an object from the same class in a different viewpoint. For example, a side view of a motorcycle resembles a side view of a bicycle more closely than a frontal view of a motorcycle, and a side view of a horse resembles a side view cow more closely than a frontal view of a horse. This suggests that there exist some common appearances shared across classes for the same viewpoint for us to exploit.

Specifically, we propose a multi-task metric learning approach (see Figure 1), which shares a common metric among the classes to capture shared view-specific components, to solve this joint pose and class recognition problem. We resort to metric learning because modeling the pose variation with similarity constraints is a natural way to express on one side continuity of the appearance variation due to pose changes (unlike classification with discrete labels) and on another side allows to easily express the homeomorphism

[1] We will use the terms viewpoint and pose estimation interchangeably.

between the object's pose manifold and the unit sphere (Zhang et al (2013)).

Contributions: 1) We explore metric-learning-based approaches for simultaneous pose and class prediction, which are flexible with respect to the type of the viewpoint labels, and are also scalable to a large number of categories. We also show how to extend these methods for detection. 2) We further propose a novel multi-task metric learning approach, which shares a common metric among the classes to capture shared view-specific components, while still allowing to capture class-specific individual aspects of pose-parametrized appearance. 3) We show that models learned using the multi-task approach are capable of performing zero-shot pose estimation, which, to our knowledge, is a novel task not addressed by any existing models. 4) We obtain state-of-the-art performance on both pose and class recognition in 3DObjects and PASCAL3D+ datasets.

Related Work

Joint pose estimation and classification: Joint pose estimation and instance/class recognition is a well-studied topic in computer vision. Most prior works pose the problem as classification (Savarese and Fei-Fei 2007, Sun et al. 2009, Xiang, Mottaghi, and Savarese 2014), where the task is to classify each instance as belonging to a class in a specific discretized viewpoint. He et al. (2014) uses a kernel-based approach to jointly model localization and pose estimation using a product of two kernels. More recently, researchers have been exploring the power of deep learning models, which have shown state-of-the art performance for classification and pose estimation. Ghodrati et al (2014) used activation features from the fifth layer of a convolutional neural network (Jia et al. 2014), and Tusiani and Malik (2015) fine-tuned a convolutional neural network by treating a combination of an object class and a specific angle as an output.

Other works formulate the pose estimation task as a regression problem (Torki and Elgammal 2011) from the whole image to the continuous view space. In (Fenzi et al. 2013, Redondo-Cabrera, Lopez-Sastre, and Tuytelaars 2014) a set of votes is produced using regression from local image patches (or features) and aggregated into a final viewpoint prediction. Zhang et al (2013) proposed a generative model, which assumes that each instance is generated by a view-transformation followed by a style-transformation. This generative framework is further extended in (Bakry and Elgammal 2014) to consider view- or class- specific projections.

Our metric learning approach, unlike classification or regression methods, can handle both discrete and continuous labels. Further, the challenging task of zero-shot pose recognition, unexplored by any of the introduced methods, can be performed within the same framework.

Metric learning: Our classification and viewpoint prediction is based on the k-nearest neighbor search in the learned metric space that maximizes class separation and preserves the view manifold. For base metric learning, we use large margin formulation of Weinberger and Saul (2009). Specifically, we build upon the large-margin multi-task metric learning introduced in Parameswaran et al (2010), which parameterizes the distance between the two points using both the task-specific metric, and a shared metric among all tasks. However, in sharp contrast to Parameswaran et al (2010), in our case the task (pose estimation) depends on a class, and is therefore unknown. A key strength of metric learning is that it can generalize to novel tasks. This has been explored for the case of zero shot class recognition in (Mensink et al. 2013), using a single metric. We, on the other hand, explore effectively a hierarchy of leaned metrics for zero shot pose estimation.

Figure 1: We learn a global metric Q_0 to discriminate classes and preserve global view-specific appearance, as well as class-specific pose estimation metrics Q_{car} and Q_{bus}. This joint learning allows to predict the pose for instances of novel object classes. For example, we can estimate the pose of the class *bus* by utilizing the view labels for class *car*, which is its neighbor in the class space.

Transfer learning and zero-shot recognition: Our method can also be viewed as a transfer learning approach since it performs zero-shot pose prediction for a novel class leveraging the view-specific appearance of existing classes. The most dominant method for zero-shot recognition is attribute-based recognition, where global properties, such as attributes, are used to transfer information from a set of source classes to target classes (Farhadi et al. 2009, Lampert, Nickisch, and Harmeling 2009).

There is not much work on transfer learning or zero-shot recognition for pose estimation. The most relevant work to our method is (Chen and Grauman 2014), which presents a method to infer unseen views of people using tensor completion. Our method is similar in the sense that we perform zero-shot pose estimation; however, our method can transfer the knowledge about view-specific appearance across categories, while (Chen and Grauman 2014) focused on a single (person) class.

Metric multi-view object recognition

Traditionally, pose estimation is solved for each class individually. However, since different classes actually share similar visual elements that change in the similar ways with the pose, solving pose estimation problem jointly, considering all classes, can be beneficial (He, Sigal, and Sclaroff 2014, Zhang et al. 2013). Moreover, learning such shared elements among classes could potentially allow pose prediction for new classes, for which the viewpoint labels are not available.

Metric learning for pose estimation

Past work has shown that the instances in different viewpoints form a continuous low-dimensional manifold in the original feature space (Murase and Nayar 1995). Therefore our goal is to preserve such manifold structure with distance

constraints. More specifically, we want to learn a Mahalanobis distance matrix Q, such that a sample has a smaller distance to another sample in a similar pose, compared to the distance to a sample that has a very different pose. Given two points $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$, the distance between these two points is defined as:

$$d_Q(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T Q (\mathbf{x}_i - \mathbf{x}_j). \quad (1)$$

Given a set of N_c training samples from the class c, $\mathcal{D}_c = \{(\mathbf{x}_i, \mathbf{p}_i)\}_{i=1}^{N_c}$, where $\mathbf{x}_i \in \mathbb{R}^D$ is a D-dimensional feature descriptor for image i and $\mathbf{p}_i \in \mathbb{R}^P$ is a P-dimensional pose label, the problem of metric learning for pose estimation can be formulated as following:

$$\min_{Q_c} \sum_{ijl} \xi^+_{ijl} + \lambda \text{tr}(Q_c), \quad Q_c \succeq 0 \quad (2)$$

$$d_{Q_c}(x_i, x_j) + m \leq d_{Q_c}(x_i, x_l) + \xi_{ijl}, \quad (3)$$
$$d^p(p_i, p_j) \leq t_l, d^p(p_i, p_l) \geq t_u$$

where $\text{tr}(Q_c)$ is the trace of Q_c and $Q_c \succeq 0$ requires Q_c to be positive semidefinite, $\xi^+ = \max(\xi, 0)$, and $d^p(\cdot, \cdot)$ is the distance in the pose space, specific for the annotations provided. Further, t_l, t_u are similarity and dissimilarity thresholds and m is the margin. Since the view manifold is low-dimensional, it is reasonable to require Q_c to be low-rank. Minimizing the rank is in turn approximated by the nuclear norm $\|Q_c\|_*$, which is equivalent to $\text{tr}(Q_c)$, that we minimize in (2), for a positive semidefinite matrix Q_c.

Metric learning for joint pose estimation and class prediction

Training per-class pose estimators has several drawbacks. First, the performance of the viewpoint/pose estimation heavily depends on the performance of the classification algorithm when the class is unknown. Second, some classes may share similar traits in their viewpoint/pose changes, which is completely ignored by the independent training of the classifiers.

Therefore, we propose to jointly learn a metric for classification and pose estimation. Now, the training set $\mathcal{D} = \{(\mathbf{x}_i, y_i, \mathbf{p}_i)\}_{i=1}^{N}$ contains samples from classes $c = 1 \dots C$ together with their class labels y_i. This problem of joint metric learning is formulated as follows:

$$\min_{Q} \sum_{ijl} (1-\mu)\xi^+_{ijl} + \sum_{ijl} \zeta^+_{ijl}\mu + \lambda \text{tr}(Q), \quad (4)$$

$$d_Q(x_i, x_j) + m_c \leq d_Q(x_i, x_l) + \zeta_{ijl}, \quad (5)$$
$$y_i = y_j, y_i \neq y_l$$

$$d_Q(x_i, x_j) + m_v \leq d_Q(x_i, x_l) + \xi_{ijl}, \quad (6)$$
$$d^p(p_i, p_j) \leq t_l, d^p(p_i, p_l) \geq t_u,$$
$$y_i = y_j = y_l, \quad Q \succeq 0$$

where $\mu \in [0, 1]$ defines the trade-off between the classification and the pose estimation, m_c is the classification margin, and m_v is the view-similarity margin. The relative scale of m_c and m_v is crucial for learning. We found through cross validation, that $m_v = m_c/C$ gives good results. By formulating the metric learning optimization jointly for all classes, we ensure that if some classes share a pose metric, this will be incorporated into the resulting matrix Q.

However, different classes may not share identical pose metrics. Moreover, the classification task differs significantly from the viewpoint estimation task, and therefore the requirements imposed on the metric by Eq. (5)-(6) can even be contradictory, when only a single metric Q is learned. We resolve this issue by introducing a global shared metric Q_0 that discriminates classes as well as preserves common manifold for view estimation. We then enable each class to have its own pose metric Q_c, which should account for unique viewpoint-related variation for the corresponding class c. We propose the following multi-task formulation:

$$\sum_{ijl} \xi^+_{ijl}(1-\mu) + \sum_{ijl} \zeta^+_{ijl}\mu + \lambda \text{tr}(Q_0) + \sum_c \gamma \text{tr}(Q_c) \quad (7)$$

$$d_{Q_0}(x_i, x_j) + m_c \leq d_{Q_0}(x_i, x_l) + \xi_{ijl}, \quad (8)$$
$$y_i = y_j, y_i \neq y_l$$

$$d_{Q_0+Q_c}(x_i, x_j) + m_v \leq d_{Q_0+Q_c}(x_i, x_l) + \zeta_{ijl}, \quad (9)$$
$$d^p(p_i, p_j) \leq t_l, d^p(p_i, p_l) \geq t_u,$$
$$y_i = y_j = y_l = c, \quad Q_0 \succeq 0, Q_c \succeq 0, c = 1 \dots C$$

In the above formulation, the pose similarity between two instances is parametrized by the sum of the global metric and the per-class metric, $Q_0 + Q_c$. The constraints of type (8) encourage Q_0 to push away the samples from different classes, while the constraints of type (9) require the samples from the class c to form a manifold with respect to the metric $Q_0 + Q_c$ having continuous structure w.r.t. pose.

Optimization

The optimization problems (4)-(6) and (7)-(9) are instances of semidefinite programming. We use a variant of stochastic projected gradient descend which subsamples active constraints. After each update step on matrices $Q_0, Q_1, \dots, Q_c$ we project them back to the cone of positive semidefinite matrices, using SVD decomposition. To further speed-up the optimization process for large-scale datasets optimized gradient computation can be used (Weinberger and Saul 2008).

Pose estimation and class prediction

While training multiple metric is intuitive, using them for pose estimation is not straightforward. Unlike multi-task metric concept of Parameswaran et al (2010), where the task is known at test time, in our case the task (pose estimation) depends on a class, and is therefore unknown. The key question is how to infer the pose label in this formulation. An intuitive way to predict pose is to first predict a class c according to metric Q_0 and then use the corresponding metric $Q_0 + Q_c$ for pose prediction. However, this would introduce errors in pose prediction whenever the class is incorrectly predicted. Instead, we produce pose prediction for each class and then choose the most confident estimate.

Given a set of training triplets $\mathcal{D}$ and a set of learned metrics $Q_0, Q_c, c = 1 \dots C$, for a new sample $\mathbf{x}^*$, the k nearest neighbors $\{\mathbf{x}_i\}_{i \in I_k}$ from the training set are selected using the set of learned metrics, such that distance to the sample $\mathbf{x}_i$

is measured as $d_i = d_{Q_0+Q_{y_i}}(\mathbf{x}^*, \mathbf{x}_i)$. The final pose prediction $\mathbf{p}$ is formed by finding the modes of the pose predictions $\mathbf{p}_i$ of the samples coming from the same class, weighted by the prediction confidence of a single sample d_i^{-1}, and selecting the most confident mode; the confidence of the mode is defined as $r^{\mathbf{p}} = \sum_{j \in I(\mathbf{p})} d_j^{-1}$, where $I(\mathbf{p}) \subset I_k$ is a subset of the nearest neighbors contributing to the mode.

Class label prediction is done by performing k nearest neighbor search using the learned metric Q_0, and choosing the weighted mode of their class labels as the final prediction; the confidence for the class c is then computed as $r^c = \sum_{j \in I(c)} d_j^{-1}, \quad I(c) = \{j : j \in I_k, y_j = c\}$.

Zero-shot pose prediction

The proposed algorithm for pose estimation can be extended for pose prediction for the classes without any pose labels. To do so, we train the model using (7)-(9) (or (4)-(6)) without imposing view-preserving constraints on the classes that do not have viewpoint labels. Then, for zero-shot pose estimation, we only consider the samples that have pose labels as potential nearest neighbors.

Since different classes might have different, unaligned, pose labels, the prediction for a sample from a class without a pose label C_z is formed as a set $\tilde{p} = \{\tilde{\mathbf{p}}^c \in \mathbb{R}^P\}_{c \in \mathcal{C}}$, where $\tilde{\mathbf{p}}^c$ is the prediction of the class c and $\mathcal{C}$ denotes different classes found among k nearest neighbors. In the experiments we observed, that only a small subset of all classes participate in the prediction formation for all samples of the class C_z.

Detection

The proposed approach can be integrated into existing detection frameworks. We propose to couple the proposed method with the pre-trained R-CNN detector (Girshick et al. 2014). In the experiments, we show that the combined model allows us to improve the performance of the detector and, in addition, estimate the viewpoint.

Detection using R-CNN detector is performed as follows: first, object proposals are extracted, using selective search (Uijlings et al. 2013); then, each proposal is evaluated based on the pre-trained SVMs, and the detection score s_i^c is computed; as a next step, the most confident object proposals are chosen, i.e. such that $s_i^c > \tau$, where τ is the detection threshold; finally, the bounding box regression and the non-maxima suppression is applied.

We introduce the changes in the detection process by combining the detection score of the SVMs with the confidence score, produced by our model, thus, re-ranking the proposals. Assume a proposal i received a detection score s_i^c from the SVM corresponding to the class c and the confidence r_i^c of the trained model. Then, the confidences for all object proposals on a single image are normalized to the interval $[0, 1]$ and the final score is computed as $(s_i^c - \tau) r_i^c + \tau$. In all experiments, we use RCNN detector, pre-trained on the PASCAL dataset.

Experiments

We evaluate our approach on two datasets. To stress that our approach is independent of the type of labeling provided as pose annotations (i.e., continuous or discrete labels), we chose one dataset containing discrete labels (3DObjects (Savarese and Fei-Fei 2007)) and one with continuous labels (PASCAL3D+ Xiang et al (2014)).

	class	Acc^{ϕ}	Acc^{θ}	$Acc^{(\phi,\theta)}$
OVM	75.7	57.2	59.8	–
3DOCM	90.53/83.07	80.34/81.86	–	–
KNN-VC	95.17	84.94	85.20	71.68
J-VC	97.35	89.92	91.65	80.84
MM-VC	96.14	89.87	91.69	**82.79**
MMJ-VC	**97.36**	**90.15**	**91.82**	82.00

Table 1: 3DObjects: class recognition and pose estimation accuracy compared with OVM (Bakry and Elgammal 2014) and 3DOCM (Savarese and Fei-Fei 2007) (%)

In all experiments, we use *pool-5* Caffe features (Jia et al. 2014), since they better preserve viewpoint variations, as verified both by our experiments and by Ghodrati et al (2014). We first reduce the dimensionality of the feature space using principal component analysis and project the 9216-dimensional features into $D = 500$ dimensional space. We use $k = 50$ nearest neighbors in all experiments to form predictions, where k is found by cross-validation.

We compare our method with the state-of-the-art methods, as well as provide our baselines to show the advantage of our final formulation in various tasks. We use the following variants of our model for comparison:

KNN-VC: a simple k-nearest neighbors baseline to show the improvement due to learned metric in comparison to the original metric in the feature space.

MM-VC: we learn one metric per class for pose, as well as a separate metric for class prediction.

J-VC: we learn a single joint metric for both class and viewpoint prediction, as defined in Eq.(4)-(6).

MMJ-VC: we learn the multi-metric model, described in Eq.(7)-(9).

We evaluate performance of our method in two main experiments: 1) the fully supervised case; 2) the zero-shot learning experiment, where we exclude ground truth pose labels from training for one class and evaluate the performance of the model for the same class. We perform this experiment for all classes. We propose to measure the performance of zero-shot prediction by measuring distance in the pose space between pairs of samples instead of directly comparing predicted poses, since the ground truth labeling might not correspond to the pose labeling induced by the neighbor classes. Note, that relative pose prediction can be transformed into the ground truth pose prediction by calculating the pose relative to fixed samples with known pose labels.

We define the predicted pose distance between two samples i and j as follows: first, for each class that participates in the pose prediction of both samples i and j the distance $d^p(\tilde{\mathbf{p}}_i^c, \tilde{\mathbf{p}}_j^c)$ in the pose space is computed. Afterwards, the

	bicycle		car	
	AP/AVP	MPPE	AP/AVP	MPPE
3D2PM	95.8/–	94.1/–	95.8/–	**99.6**/–
BnB	95.1/–	94/–	98.2/–	87.9/–
VDPM	91/–	90/–	96/–	89/–
3DCAD	87.0/–	–/87.7	94.9/–	–/82.6
ours-sep	98.57/72.09	93.0/83.1	99.07/86.61	92.3/90.3
ours-comb	**99.06**/**83.61**	**96.7**/**89.5**	**99.06**/**88.88**	95.5/**91.8**

Table 2: 3DObjects: detection and pose estimation performance (AP/AVP/MPPE) of MMJ-VC model, combined with R-CNN detector, where *ours-sep* denotes the results without object proposal rescoring, and *ours-comb* denotes the results with rescoring; we compare against 3D2PM (Pepik et al. 2012a), BnB (He et al (2014)), VDPM (Lopez-Sastre et al (2011)), 3DCAD (Schels et al (2012))

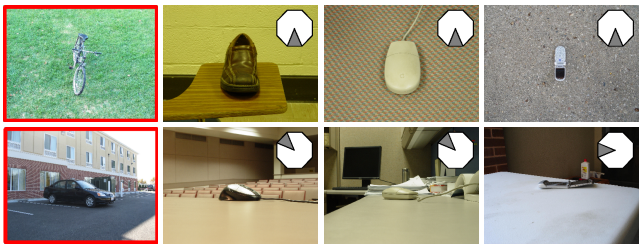

Figure 2: Zero-shot pose estimation examples: the first column shows the input image (denoted by the red boundary) and the remaining columns show samples selected for pose prediction.

predicted pose distance is averaged across all the classes:

$$d(\tilde{p}_i, \tilde{p}_j) = \frac{1}{|\mathcal{C}_{act}|} \sum_{c \in \mathcal{C}_{act}} d^p(\tilde{\mathbf{p}}_i^c, \tilde{\mathbf{p}}_j^c), \qquad (10)$$

where $\tilde{\mathbf{p}}_i^c$ corresponds to the prediction of the class c and $\mathcal{C}_{act} = \mathcal{C}_i \cap \mathcal{C}_j$ is the set of the classes, that formed prediction for for both samples i and j; if two samples have a non-intersecting set of predicting classes, $d(\tilde{p}_i, \tilde{p}_j)$ is set equal to the maximal distance in the pose space.

3DObjects dataset

This dataset contains 10 object classes, where each class has 10 instances that are presented in different views and scales. In total, 3 scales are used; the view space is discretized by azimuth angle ϕ into 8 intervals, and by elevation angle θ into 3 intervals. The pose vector is then defined as $\mathbf{p} = (\phi, \theta)^T \in \mathbb{N}^2$.

We define the distance in the pose space $d^p(\mathbf{p}_i, \mathbf{p}_j) = \min(8 - |\phi_i - \phi_j|, |\phi_i - \phi_j|) + |\theta_i - \theta_j|$, where $\phi \in 1 \dots 8$ denotes the azimuth interval number and $\theta \in 1 \dots 3$ denotes the elevation interval number; we set $t_l = 0$ and $t_u = 1$. We following the protocol of (Savarese and Fei-Fei 2007) in our experiments and measure accuracy for azimuth Acc^{ϕ}, elevation Acc^{θ} and total accuracy $Acc^{(\phi,\theta)}$.

Fully supervised case: In Table 1, the results for pose and class recognition are presented. First, we outperform both prior works, (Savarese and Fei-Fei 2007) and (Bakry and Elgammal 2014), significantly. Second, the learned metric outperforms a simple KNN-VC baseline both in recognition and in pose estimation. Furthermore, both MM-VC and MMJ-VC outperform J-VC for pose prediction. This validates the importance of separate class-specific and pose-specific metrics employed in MM-VC and MMJ-VC, as compared to J-VC that has a single metric for both tasks. To compare our results with the other published works, we also provide detection results for two classes (*car* and *bycicle*) in Table 2. We outperform most of the previous works (Pepik et al. 2012a, He, Sigal, and Sclaroff 2014, López-Sastre, Tuytelaars, and Savarese 2011, Schels, Liebelt, and Lienhart 2012), both for detection and pose estimation for all baselines, and perform on par with (Pepik et al. 2012a) on *car* class. Note the increase of in both AP and AVP metrics due to the object proposals rescoring. Our results are slightly worse then the ones reported in (Pepik et al. 2012b), however, the method presented in (Pepik et al. 2012b) requires 3D geometric CAD models for each class and large set of synthetic data for training, while our model does not.

	class	$MedError(\circ)$	$Acc_{\pi/6}$
KNN-VC	61.70/62.72	35.74/37.69	49.76/50.76
J-VC	71.49/82.23	**31.93**/31.54	51.31/55.05
MM-VC	70.35/**85.12**	36.61/38.48	48.55/47.35
MMJ-VC	**71.75**/83.06	32.81/**29.67**	**51.84**/**55.20**

Table 5: PASCAL3D+: class recognition and pose estimation accuracy (the first number shows the results on the whole dataset, while the second - in case of non-truncated and non-occluded images only).

Zero-shot pose: We evaluate the performance in the zero-shot pose estimation experiment using the relative pose given by Eq. (10). The results are presented in Table 3. Since objects in 3DObject dataset are very distinct, only general features, such as rectangular form, can be transferred between categories. However, we still are able to predict the pose for the objects from the novel category about 3 times better than random. Our full multi-metric model (MMJ-VC) gives the best performance, since it contains both the joint multi-task learning objective and combination of shared and class-specific metrics. Notably, MM-VC performs slightly worse than simple KNN-VC baseline, which points to the key importance of joint multi-task learning for zero-shot prediction. The visual results for zero-shot pose estimation are presented in Figure 2, where the samples for zero-shot prediction and the first three nearest neighbors with respect to the learned model are shown (for MMJ-VC model). The way the zero-shot prediction is formed makes pose estimation robust against unaligned pose labels.

PASCAL3D+ dataset

The dataset contains images of 12 different categories from PASCAL VOC 2012 training and validation sets. The annotation include continuous labels of the azimuth and elevation angles, as well as information about occlusion and truncation of the objects. Following (Xiang, Mottaghi, and Savarese 2014), we train the model on the images from the training set and test on the images from the validation set. For PASCAL3D+ dataset, we use the distance in the pose space $d^p(\mathbf{p}_1, \mathbf{p}_2)$, as well as two performance metrics, proposed in (Tulsiani and Malik 2015): first, we estimate median of the distance in the pose space $d^p(\mathbf{p}_{pred}, \mathbf{p}_{gt})$ (we

	bicycle	car	cell	iron	mouse	shoe	stapler	toaster	mean
KNN-VC	47.0/17.1	**47.3**/**25.0**	45.6/20.7	45.6/19.1	43.2/20.8	**48.5**/22.7	**47.2**/20.8	41.9/19.6	45.7/20.7
J-VC	48.4/20.5	44.6/23.5	**46.3**/**22.5**	45.5/20.6	44.9/23.9	46.1/25.2	46.4/22.8	42.0/19.2	45.5/22.3
MM-VC	47.3/19.7	37.9/19.9	45.5/21.6	44.7/19.2	43.1/21.0	44.6/24.9	45.8/21.8	40.1/18.0	43.6/20.7
MMJ-VC	**49.0**/**20.6**	45.6/24.1	45.7/22.2	**46.3**/**20.8**	**45.1**/**23.2**	48.1/**26.8**	46.5/**22.5**	**43.3**/**20.3**	**46.2**/**22.6**

Table 3: 3DObjects: zero-shot pose estimation accuracy (Acc^{ϕ}/$Acc^{(\phi,\theta)}$).

	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	mean
KNN-VC	**37.75**	**39.98**	36.55	29.34	31.17	33.14	39.68	48.12	**39.90**	48.37	28.40	47.71	38.34
J-VC	35.65	36.62	35.57	**57.72**	33.71	33.03	37.08	**49.90**	36.77	**55.07**	34.66	55.13	41.74
MM-VC	36.60	40.30	35.34	37.44	**40.71**	**33.97**	39.43	48.07	35.16	51.09	36.22	49.88	40.35
MMJ-VC	34.42	37.79	**36.66**	56.42	36.11	32.47	36.32	49.81	37.81	54.32	**38.49**	**57.40**	**42.33**

Table 4: PASCAL3D+: zero-shot pose estimation accuracy ($Acc_{\pi/6}$) for the whole dataset.

denote this metric as $MedErr$); second, we use a discrete accuracy metric Acc_{θ}, that measures the fraction of the samples, for which $d^p(\mathbf{p}_{pred}, \mathbf{p}_{gt}) < \theta$. During training, we set t_l and t_u to 5% and 95% quantiles of the per class pose distances distribution.

Fully supervised case: The results are presented in Table 5. As in the previous experiment, J-VC and MMJ-VC baselines perform better than KNN prediction, however, MM-VC baseline performs poorly this time. MMJ-VC baseline slightly outperforms J-VC baseline. Note, that the performance drop between the case of fully visible subset of PASCAL3D+ dataset and the whole dataset is not as great as expected. This might be because such occluded instances are present in both train and test sets, since most of the occlusions in the datasets are typical for a given object category. Therefore, we are still able to estimate the pose correctly. We do not directly compare our results with the ones presented in (Tulsiani and Malik 2015), since we use generic object features from (Jia et al. 2014), while in their work they use object features fine-tuned for the particular set of classes in the dataset, as well as for pose estimation, making comparison unfair. We provide the comparison of MMJ-VC method

	aero	boat	mean
VDPM	40.0/34.6	3.0/1.5	26.8/19.5
3DDPM	41.5/37.4	0.5/0.3	27.0/23.8
ours	**71.1**/**53.1**	**32.3**/**12.2**	**44.7**/**33.4**
VDPM	39.8/23.4	5.8/1.0	29.9/18.7
3DDPM	40.5/28.6	0.5/0.2	28.3/21.5
ours	**71.1**/**32.8**	**32.3**/**6.3**	**44.7**/**24.7**

Table 6: PASCAL3D+: detection and pose estimation performance of our model (MMJ-VC + RCNN), compared against VDPM (Xiang, Mottaghi, and Savarese 2014) and 3DDPM (Pepik et al. 2012b) using (AP/AVP) with 4 views (upper Table), 8 views (lower Table). We do not provide results with and without rescoring, since the detector is already trained on the PASCAL dataset.

in the detection task with two baselines (Xiang, Mottaghi, and Savarese 2014, Pepik et al. 2012b) (for (Pepik et al. 2012b), we use the results reported in the (Xiang, Mottaghi, and Savarese 2014))) using AP and AVP metric. The results are presented in Table 6. Note, our method outperforms on average both (Xiang, Mottaghi, and Savarese 2014, Pepik et al. 2012b) and (Pepik et al. 2012b), although both of them are discriminative methods, and (Pepik et al. 2012b) requires 3D CAD models for each class, while our method is not explicitly aware of 3D geometry of the objects.

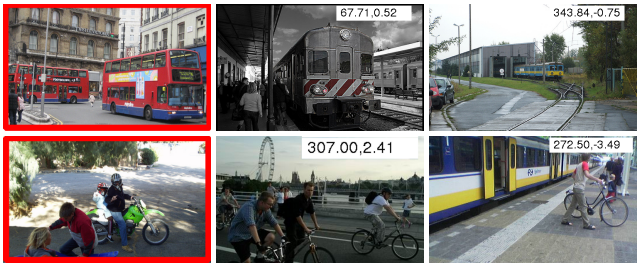

Figure 3: Zero-shot pose estimation examples on the PASCAL 3D+ dataset; **top row:** test samples from class *bus* (with red boundary) and the nearest neighbors retrieved; **bottom row:** nearest neighbors found for the sample from the class *motorbike*.

Zero-shot pose: We achieve higher improvement compared to the results we have on the 3DObjects dataset. We attribute this to the fact, that PASCAL3D+ dataset contains many categories that have similar appearance variations due to viewpoint change, such as *bike* and *motorbike* or *car* and *bus*. MMJ-VC baseline outperforms all other baselines in that case as well, while MM-VC baseline, which models view-similarity separately, performs poorly as on 3DObjects dataset. Figure 3 shows examples of the nearest neighbors selected for zero-shot pose prediction.

Conclusion

We have presented a method for simultaneous pose estimation and class prediction using learned metrics. Our metric learning-based approach encodes the pose information as relative distances between points, and can handle both discrete and continuous labels unlike existing classification or regression-based solutions. Further, it can generalize to the pose estimation task for a novel class, at almost no cost. By jointly training the classification metric with pose metric, we are able to learn shared visual components across categories for class separation and model view-specific appearance. We have validated our method on two datasets, and have shown that jointly learned metric outperforms separately learned metrics for the fully supervised pose estimation as well as generalizes pose estimates for a novel category without pose labels. Furthermore, we showed the multi-task joint formulation further outperforms a single-metric formulation (especially for zero-shot).

References

Bakry, A., and Elgammal, A. 2014. Untangling object-view manifold for multiview recognition and pose estimation. In *ECCV*.

Chen, C.-Y., and Grauman, K. 2014. Inferring unseen views of people. In *CVPR*.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing objects by their attributes. In *CVPR*.

Fenzi, M.; Leal-Taixé, L.; Rosenhahn, B.; and Ostermann, J. 2013. Class generative models based on feature regression for pose estimation of object categories. In *CVPR*.

Ghodrati, A.; Pedersoli, M.; and Tuytelaars, T. 2014. Is 2d information enough for viewpoint estimation? In *BMVC*.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*.

He, K.; Sigal, L.; and Sclaroff, S. 2014. Parameterizing object detectors in the continuous pose space. In *ECCV*.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. In *MM*.

Lampert, C.; Nickisch, H.; and Harmeling, S. 2009. Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. In *CVPR*.

López-Sastre, R. J.; Tuytelaars, T.; and Savarese, S. 2011. Deformable part models revisited: A performance evaluation for object category pose estimation. In *ICCV Workshops*.

Mensink, T.; Verbeek, J.; Perronnin, F.; and Csurka, G. 2013. Distance-based image classification: Generalizing to new classes at near-zero cost. *TPAMI*.

Murase, H., and Nayar, S. K. 1995. Visual learning and recognition of 3-d objects from appearance. *IJCV*.

Parameswaran, S., and Weinberger, K. Q. 2010. Large margin multi-task metric learning. In *NIPS*.

Pepik, B.; Gehler, P.; Stark, M.; and Schiele, B. 2012a. 3d 2pm - 3d deformable part models. In *ECCV*.

Pepik, B.; Stark, M.; Gehler, P.; and Schiele, B. 2012b. Teaching 3d geometry to deformable part models. In *CVPR*.

Redondo-Cabrera, C.; Lopez-Sastre, R.; and Tuytelaars, T. 2014. All together now: Simultaneous detection and continuous pose estimation using a hough forest with probabilistic locally enhanced voting. In *BMVC*.

Savarese, S., and Fei-Fei, L. 2007. 3d generic object categorization, localization and pose estimation. In *ICCV*.

Schels, J.; Liebelt, J.; and Lienhart, R. 2012. Learning an object class representation on a continuous viewsphere. In *CVPR*.

Sun, M.; Su, H.; Savarese, S.; and Fei-Fei, L. 2009. A multi-view probabilistic model for 3d object classes. In *CVPR*.

Torki, M., and Elgammal, A. 2011. Regression from local features for viewpoint and pose estimation. In *ICCV*.

Tulsiani, S., and Malik, J. 2015. Viewpoints and keypoints. In *CVPR*.

Uijlings, J.; van de Sande, K.; Gevers, T.; and Smeulders, A. 2013. Selective search for object recognition. *IJCV*.

Weinberger, K., and Saul, L. 2008. Fast solvers and efficient implementations for distance metric learning. In *ICML*.

Weinberger, K. Q., and Saul, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *JMLR*.

Xiang, Y.; Mottaghi, R.; and Savarese, S. 2014. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*.

Zhang, H.; El-Gaaly, T.; Elgammal, A. M.; and Jiang, Z. 2013. Joint object and pose recognition using homeomorphic manifold analysis. In *AAAI*.