Continuous Pose Estimation with a Spatial Ensemble of Fisher Regressors

Michele Fenzi[1], Laura Leal-Taixé[2], Jörn Ostermann[1], Tinne Tuytelaars[3]

[1] Institut für Informationsverarbeitung (TNT), Leibniz Universität Hannover
{fenzi,ostermann}@tnt.uni-hannover.de
[2] Institute of Geodesy and Photogrammetry, ETH Zurich
leal@geod.baug.ethz.ch
[3] KU Leuven, ESAT - PSI, iMinds
Tinne.Tuytelaars@esat.kuleuven.be

Abstract

In this paper, we treat the problem of continuous pose estimation for object categories as a regression problem on the basis of only 2D training information. While regression is a natural framework for continuous problems, regression methods so far achieved inferior results with respect to 3D-based and 2D-based classification-and-refinement approaches. This may be attributed to their weakness to high intra-class variability as well as to noisy matching procedures and lack of geometrical constraints.

We propose to apply regression to Fisher-encoded vectors computed from large cells by learning an array of Fisher regressors. Fisher encoding makes our algorithm flexible to variations in class appearance, while the array structure permits to indirectly introduce spatial context information in the approach. We formulate our problem as a MAP inference problem, where the likelihood function is composed of a generative term based on the prediction error generated by the ensemble of Fisher regressors as well as a discriminative term based on SVM classifiers.

We test our algorithm on three publicly available datasets that envisage several difficulties, such as high intra-class variability, truncations, occlusions, and motion blur, obtaining state-of-the-art results.

1. Introduction

In this paper, we focus on the problem of continuous pose estimation for object categories. This task takes on a great importance in many applications, where the exact pose of targeted objects is necessary for the accomplishment of broader tasks, such as autonomous driving and scene understanding.

Several works address this problem by exploiting 3D information during training and testing, whereas other works propose 2D-based methods, thus removing the need for 3D CAD models of the classes of interest. Among continuous 2D-based approaches, some employ a classify-and-refine paradigm by letting a refinement step follow a rough viewpoint classification in order to obtain a real-valued pose [12, 15]. More naturally, others argue that pose estimation is inherently a continuous problem and set it in a regression framework [6, 8, 13, 4, 28]. Although regression appears to be a sound approach, most regression-based methods to date are still not able to outperform 3D-based and 2D-based classify-and-refine approaches. In this paper, we start from the promising idea of feature regression [6], we address

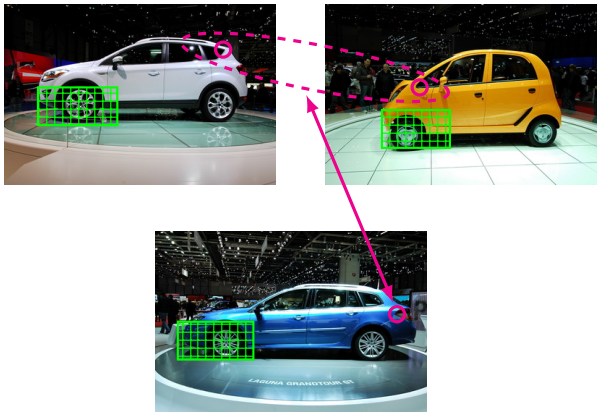

Figure 1. Two shortcomings of [6]: (i) in the top row, two feature regressors (drawn as single feature points in magenta) are wrongly clustered together; (ii) the feature in the test image (bottom row) is wrongly matched to a regression cluster (dashed ellipse). On the contrary, we learn a regressor for each cell according to a fixed grid. For example, we learn a regressor that predicts the Fisher vector representation of the bottom left part of the car (represented in green) for any viewpoint. (Figure best viewed in color.)

its weaknesses, and we successfully show that regression-based pose estimation can actually provide state-of-the-art results.

More specifically in [6], a regressor for each feature track is learned to predict the feature descriptor as a function of the viewpoint, and then regressors are clustered according to feature similarity. During testing, extracted features are matched to model regressors and the pose that minimizes the prediction error is returned. As shown in Figure 1, this formulation has three main limitations that affect its overall performance: (i) clustering is inevitably a noisy process, (ii) feature matching may lead to wrong correspondences, and (iii) geometrical information is either unused or, when used, only pairwise geometrical relations are taken into account [8].

Since Fisher vectors proved to be effective for discrete pose classification [10], we investigate how to integrate the concept of feature regression and Fisher encoding into a probabilistic framework in order to handle the aforementioned shortcomings. However, this integration is not trivial. Given the huge number of feature tracks in [6] and the high dimensionality of Fisher vectors, a simple re-encoding would be unfeasible in terms of memory. As a solution, we propose to build an ensemble of Fisher regressors on the basis of low-level features spatially aggregated from a wide grid. This solution leads to the following advantages:

- Clustering and matching procedures are avoided, as they are automatically induced by the arrangement of the regressors in a grid;
- Relatively small memory cost, as there is only one regressor per cell and each regressor can predict the Fisher descriptor of its corresponding cell for the whole viewpoint range.

The second point we address in this paper is the smoothness assumption on which individual feature regression relies. We show experimentally that this remains valid also for Fisher vectors built upon spatially aggregated features, and this in spite of their high dimensionality.

Finally, we address the problem of the lack of discriminativeness of generative approaches. As object appearance can be very similar in opposite views, generative approaches, like those based on regression, tend to suffer from "flipping" errors in the estimation. We solve this by learning an ensemble of discriminative classifiers that we integrate in our probabilistic framework.

In the next section, we give a review of related works, while in Section 3 we describe feature regression and Fisher encoding. In Section 4, we explain how to build our class representation as an ensemble of Fisher regressors, and how to embed the class representation in a probabilistic framework to estimate the pose of the target object. In Section 5, we present experimental results on three publicly available datasets, and we give our conclusions in Section 6.

2. Related Work

Some works have treated pose estimation as a classification problem by dividing the viewpoint range into discrete bins. In [20], the authors first use a Naive Bayes classifier to find the discrete viewpoint with highest probability, that is later accepted/rejected by an SVM classifier. Similarly, [10] trains a set of viewpoint-based SVM classifiers using Fisher vectors, and increase the classification performance by removing images with similar viewpoint from the training set.

Several works have focused on methods that provide a continuous value for the pose [30, 21, 34, 28, 6, 13, 15]. For this purpose, two different strategies have been explored. In one strategy, 3D training information, often in the form of CAD models, is used to learn a precise arrangement of the object parts in 3D. This permits to mitigate the perspective ambiguity at test time. In the other strategy, that we also follow in this paper, only 2D training information is used to learn a class representation. In the following, we separate the related work according to this categorization.

3D training In [30], the authors build an aspect layout model of the object category using conditional random fields. For a set of manually annotated CAD models, the 3D arrangement of the parts is learned on the basis of rectangular surfaces fit to the models. Appearance is learned from training images by rectifying the corresponding parts into frontal view. Similarly, [21] extends the DPM classifier to 3D by using textured CAD models to learn the three-dimensional part arrangement. In order to obtain a continuous pose inference, the authors employ a parametrized interpolation of the appearance filter coefficients. [18] learns a 2D part model from real images and 3D geometry from synthetic CAD models. The pose is estimated by ranking the likelihood of 2D part detections with respect to the 3D model. In [34], the authors learn a 3D wire-frame model on the basis of manually annotated CAD models and part appearance from non-photorealistic renderings. At test time, a MAP problem is solved by searching over possible projections of the model on the test image. [32] uses images rendered from 3D CAD models to train a deep architecture that learns the most discriminative object parts across different viewpoints. Pose estimation is performed in discrete/continuous fashion by training the network with discrete/interpolated labels.

2D training In [12], a set of classifiers is trained for discrete viewpoints, and the pose is refined by linearly deforming the template as a function of the viewpoint. Sim-

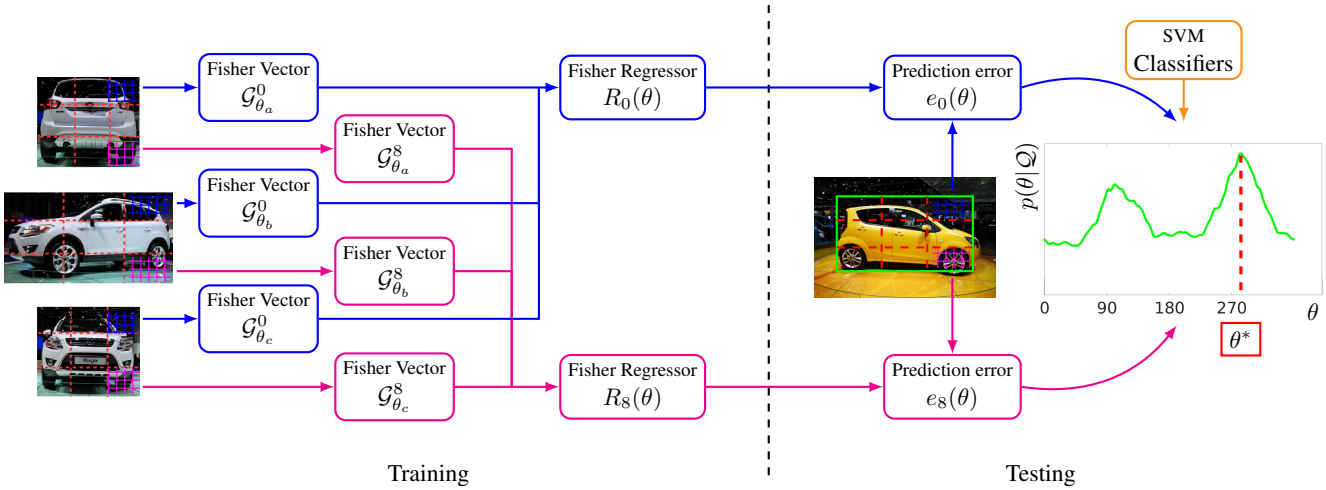

Figure 2. Overview of our method: training and testing stages. We divide each training image by using a wide grid and, for each grid cell, we densely extract low-level features and we encode them in a Fisher vector. Then, we build an ensemble of Fisher regressors to predict the Fisher vector of the corresponding cells for any viewpoint. At test time, we apply the same grid, and we perform Fisher encoding on the extracted low-level features. The SVM-based pose classification and all the prediction errors generated by the Fisher regressors concur in estimating a posterior distribution for the pose, whose maximum is returned as the output pose θ^*. (Figure best viewed in color.)

ilarly, [15] learns a viewpoint-parametrized classifier that first makes coarse viewpoint hypotheses and then performs estimation refinement. In [1], a complex object-viewpoint manifold is built and then untangled by factorizing the manifold in a view-invariant category representation and a category-invariant viewpoint representation, where the latter is used for pose estimation. In [27], the class representation is a probability distribution depending on the image and pose coordinates of extracted edge features. The object pose in the query image is estimated by marginalizing out the product of the query distribution and the class distribution with respect to the spatial coordinates. In [13], pose estimation is carried out by means of a K-ary regression forest, where an optimized K-means clustering step is envisaged at each node.

Among the works that share more similarity to ours, [28] first projects local features on a lower dimensional manifold with feature similarity and geometry constraints in order to learn a class representation. Then, regression is applied to the object features extracted from a full query image in order to estimate the object pose. In our paper, we show that regression on smaller object regions permits to handle the potentially high variability of the class in a better and more flexible way, as parts that belong to different objects can contribute independently to the estimation. Similarly to [28], [6, 8] first learn a class representation by aggregating local features on the basis of their similarity using spectral clustering. Then, they learn one regression model for each cluster and enforce geometrical constraints during pose inference using graph matching. In our paper, we avoid noisy clustering and wrong matches by using a grid structure, whose combination with Fisher encoding permits to inexpensively introduce geometry in the process.

3. Technical background

Here, we introduce the two main technical tools we use in our approach. First, we describe the procedure to build a generic feature regressor using a Radial Basis Function network. Then, we show how Fisher encoding can be applied to feature sets collected from large image regions.

3.1. Feature regressor

Let us consider a set of k-dimensional feature descriptors $\{f_i\}_{i=1}^n$, *e.g.*, HOG [2], or SIFT [19] descriptors, extracted from the same patch under different viewpoints $\{\theta_i\}_{i=1}^n$.

In order to build the feature regressor, we use a Radial Basis Function (RBF) network [14], as it is an effective method to approximate unknown functions given a set of input samples. RBF networks can also be interpreted as simple neural networks with three layers, where the hidden layer is characterized by non-linear RBF activation functions while the output layer is linear in its input. For the RBF network at hand, we use a Gaussian kernel G defined as follows,

$$G(\alpha, \beta) = \exp\left(-\frac{\|\alpha - \beta\|^2}{\sigma^2}\right), \qquad (1)$$

where $\|\cdot\|$ represents a suitable metric for viewpoint distance and σ is the kernel bandwidth.

The descriptor $\hat{f}$ predicted by the RBF network for the

input viewpoint θ is given by

$$R(\theta) = \sum_{i=1}^{n} \mathbf{w}_i G(\theta, \theta_i) = \hat{f}, \tag{2}$$

where the k-dimensional vector coefficients $\mathbf{w}_i$ are estimated during network training. As discussed in [23], if all training viewpoints are used as centers of the RBF function, $\mathbf{w}_i$ are obtained from the following regularized linear least squares problem

$$(\mathbf{G} + \gamma \mathbf{I})\mathbf{W} = \mathbf{Z}, \tag{3}$$

where $\mathbf{G}$ is a $n \times n$ matrix with $G_{ij} = G(\theta_i, \theta_j)$, γ is the regularization parameter, $\mathbf{I}$ is the $n \times n$ identity matrix, and $\mathbf{Z}$ is a $n \times k$ matrix containing the feature descriptors stacked in row fashion. The rows of $\mathbf{W}$ provide the resulting vector coefficients $\mathbf{w}_i$.

3.2. Fisher encoding

Fisher encoding [16] is a state-of-the-art method in object and pose classification tasks [17, 29, 25]. Its superiority over other methods, such as Bag of Words [26], can be attributed to the high discriminativeness that results from encoding the low-level features with the generative model that produces them.

In order to compute the Fisher encoding, we use the procedure proposed in [22]. That is, we first fit a Gaussian Mixture Model (GMM), $u(x) = \sum_{k=1}^{K} w_k u_k(x)$, where K is the number of models, to all training data features. Before fitting the GMM to the input features, we reduce the feature dimensionality by projecting the features on their PCA principal components. Then, we estimate the parameters λ_k of each mixture model, where $\lambda_k = \{w_k, \mu_k, \Sigma_k\}$, *i.e.*, the weight, mean, and diagonal covariance of mixture u_k, respectively.

The encoding for a set of features X is obtained by estimating the first and second order gradient statistics of the features. This is carried out by computing the feature derivatives with respect to the GMM means and variances. First, let $\alpha_i(k)$ be the soft assignment of descriptor f_i to the k-th Gaussian mixture as

$$\alpha_i(k) = \frac{w_k u_k(f_i)}{\sum_{j=1}^{K} w_j u_j(f_i)}. \tag{4}$$

The weighted average of the mean and standard deviation statistics with respect to the k-th mixture, $\mathcal{G}^X_{\mu,k}$ and $\mathcal{G}^X_{\sigma,k}$, are computed as

$$\mathcal{G}^X_{\mu,k} = \frac{1}{m\sqrt{w_k}} \sum_{i=1}^{m} \alpha_i(k) \left(\frac{f_i - \mu_k}{\sigma_k} \right) \tag{5}$$

$$\mathcal{G}^X_{\sigma,k} = \frac{1}{m\sqrt{2w_k}} \sum_{i=1}^{m} \alpha_i(k) \left[\frac{(f_i - \mu_k)^2}{\sigma_k^2} - 1 \right] \tag{6}$$

where m is the number of features in X.

Each feature set X is finally represented by the Fisher vector obtained by stacking $\mathcal{G}^X_{\mu,k}$ and $\mathcal{G}^X_{\sigma,k}$ of all mixtures

$$\mathcal{G}^X = [\mathcal{G}^X_{\mu,1}, \mathcal{G}^X_{\sigma,1}, \ldots, \mathcal{G}^X_{\mu,K}, \mathcal{G}^X_{\sigma,K}], \tag{7}$$

to which we subsequently apply signed square-rooting normalization and L_2 normalization.

In the following section, we show how to combine the concept of feature regression and Fisher encoding in order to create an ensemble of Fisher regressors that will form our class representation, as depicted in Figure 2. We will also discuss the advantages of our choices with respect to other regression-based methods.

4. Class Representation

Let us assume we have a set of different exemplars of the class of interest $\mathcal{O} = \{o_i\}_{i=1}^{M}$, where each instance is depicted in a set of training images $\mathcal{I}_i = \{I_{ij}\}_{j=1}^{N_i}$, where N_i is the number of training images for object o_i. We also assume that we are given a set of bounding boxes that frame the object in each image, and a set of viewpoint labels that indicate the pose of the training instance in the corresponding image $\Theta_i = \{\theta_{ij}\}_{j=1}^{N_i}$.

For each object o_i, we consider only the region contained in each training image I_{ij} defined by the corresponding bounding box. Then, we introduce a grid structure that splits the region into L cells. We densely extract features from each cell c, and we stack the features in a $m \times k$ matrix $\mathbf{M}_{ic}$, where m is the number of features in cell c. Afterwards, we encode the matrix $\mathbf{M}_{ic}$ into a Fisher vector F_{ic}, as described in Section 3.2.

Now, for each cell c and object o_i, we can construct a Fisher regressor $R_{ic}(\theta)$ by using the RBF-based method described in Section 3.1. The advantage of learning a Fisher regressor compared to learning a regressor for each feature track is evident, as we avoid ending up with thousands of regressors making the class representation hard to handle. Besides, by using regressors built upon feature tracks we cannot easily benefit from important spatial cues given by the feature arrangement in the image. On the contrary, our grid-based approach takes geometry automatically into account.

Furthermore, each regressor $R_{ic}(\theta)$ can predict the Fisher vector that encodes the cell information for any viewpoint. For example, if we are given a training set that depicts the object from the whole viewpoint circle, we can estimate a Fisher vector for any viewpoint in [0°, 360°). This is a large advantage with respect to regressors built on individual feature tracks, as they can only work over narrow viewpoint intervals because of the limited visibility of the feature tracks.

Now, for each object o_i we have a set of L Fisher regressors $R_{ic}(\theta)$, each dedicated to a particular cell. We build

our class representation $\mathcal{C}$, as the union of the Fisher regressors constructed from the set of all objects $\mathcal{O}$, *i.e.*,

$$\mathcal{C} = \{R_{ic}(\theta)\}, \text{ where } i = [1, \ldots, M], c = [1, \ldots, L] \quad (8)$$

Thanks to the grid structure, we have a flexible model in which different cells from different models can act separately. This guarantees robustness with respect to high intra-class variability, as an unknown object is often better explained by a combination of separate parts of different training models.

4.1. Pose Inference

Here, we show how we integrate our class representation formed by a spatial ensemble of Fisher regressors in a probabilistic framework in order to estimate the viewpoint of an unknown object.

Given a query image and a bounding box framing the object, we divide the bounding box according to the same grid structure that we used in training. Again, we densely extract features from each cell c, we aggregate them and we encode the resulting matrix in a Fisher vector Q_c. The set of all Q_c is indicated by $\mathcal{Q} = \{Q_c\}_{c=1}^{L}$. Therefore, $p(\theta|\mathcal{Q})$ represents the probability of the object being seen from viewpoint θ when $\mathcal{Q}$ is extracted from the query image.

By applying Bayes' theorem, we obtain

$$p(\theta|\mathcal{Q}) = \frac{p(\mathcal{Q}|\theta)p(\theta)}{p(\mathcal{Q})}. \quad (9)$$

Therefore, the estimated pose can be obtained as the one maximizing Equation (9). Since $p(\mathcal{Q})$ does not depend on θ, our problem reduces to the following maximum a posteriori (MAP) estimation

$$\theta^* = \arg\max_{\theta} \quad p(\mathcal{Q}|\theta)p(\theta). \quad (10)$$

We decompose $p(\mathcal{Q}|\theta)$ into a discriminative and a generative term,

$$p(\mathcal{Q}|\theta) = p(\mathcal{Q}|\text{SVM}_\theta, \theta)p(\mathcal{Q}|\mathcal{C}, \theta). \quad (11)$$

The discriminative term consists of V binary SVM classifiers for opposite viewpoint bins that builds the following probability distribution

$$p(\mathcal{Q}|\text{SVM}_\theta, \theta) = \exp\left(\alpha_\theta \cdot \text{sgn}\left(\sum_{c=1}^{L} \text{SVM}_\theta(Q_c)\right)\right) \quad (12)$$

α_θ is the confidence that we have in each classifier, SVM_θ is the classifier that covers for the tentative viewpoint θ, and $\text{SVM}_\theta(Q_c)$ is the signed distance to the classifier margin when Q_c is input to the classifier. The likelihood of all the tentative poses θ contained in one viewpoint bin is either increased or decreased according to the pooled answers of the corresponding classifier. We would like to stress that this does not turn our approach into a classify-and-refine approach, as the viewpoint bins with negative classification are assigned a smaller, yet non-null probability.

The generative term is based on the prediction error generated by the ensemble of Fisher regressors. Since we use the same grid structure for training and testing, each test cell selects a subset of model regressors $\mathcal{R}_c = \{R_{ic}\}_{i=1}^{M}$, where M is the number of training objects. By assuming that our model regressors can correctly discriminate between different viewpoints, the regression error for each cell and each object $e_{ic}(\theta) = \|Q_c - R_{ic}(\theta)\|$ will be small when the putative pose is similar to the ground truth. Therefore, we can express the generative term as

$$p(Q_c|\mathcal{R}_c, \theta) = \sum_{i=1}^{M} e^{-\|Q_c - R_{ic}(\theta)\|}, \quad (13)$$

where we do an average pooling over the responses of the regressors of each object.

We can easily extend this formulation from one cell to all cells by assuming a mixture model to avoid cancellation problems, which results in

$$p(\mathcal{Q}|\mathcal{C}, \theta) = \sum_{i=1}^{M} \sum_{c=1}^{L} e^{-\|Q_c - R_{ic}(\theta)\|} \quad (14)$$

where $\mathcal{Q} = \{Q_c\}_{c=1}^{L}$ is the set of Fisher vectors extracted from the query image, $\mathcal{C}$ is our class representation as in Equation (8).

Finally, the pose θ^* maximizing the posterior probability $p(\mathcal{Q}|\theta)$ is computed via simulated annealing. The pose prior $p(\theta)$ can be set as an uniform distribution if no additional information on it is available.

5. Experimental Evaluation

In this section, we test our algorithm on three publicly available datasets that permit to evaluate continuous pose estimation methods. We show some image samples of the three datasets in Figure 4.

EPFL Multi-view car dataset The first dataset on which we test our algorithm is the EPFL multi-view car dataset [20]. We choose this dataset to test our algorithm against high intra-class variability, as the cars range from city cars and sedans to Sport Utility Vehicles and concept cars, as shown in Figure 4. This dataset comprises 20 sequences of cars rotating on a platform. Since the shooting time of the images is given, a precise viewpoint label can be computed.

In order to compare our results with state-of-the-art methods, we split the dataset for training and testing according to the classic paradigm used on this dataset. That is, we perform a 50% split by using the images of the first 10 cars for training our model and the images of the second 10 cars for testing. We train our model and we estimate the pose according to the method described in Section 4. We use HOG features densely extracted from patches of size 32×32 pixels with a window stride of 2 pixels as low-level features, we use $L = 9$ cells arranged in a 3×3 configuration, and we use 8 binary SVM classifiers trained with Fisher vectors from opposite viewpoints. We use an off-the-shelf DPM [5] detector in order to determine the object bounding box. Since the detector has 99.3% recall on this dataset (measured according to the Pascal VOC protocol [3]), we evaluate our algorithm on practically all testing images. We show that our algorithm is robust to imprecise bounding boxes, as the average of the Intersection-over-Union (IoU) ratio is around 77% on this dataset.

In Table 1, we compare the performance of our algorithm to state-of-the-art methods. We mark with "(GT)" the approaches that assume the ground truth bounding box of the object as input. We provide results in terms of mean and median absolute error (AE), as these are the two metrics most commonly used on this dataset. The mean AE is the mean of the absolute difference between estimated viewpoint and ground truth, whereas the median AE evaluates the performance of the method after removing very large errors, also known as "flipping" errors. By comparing Mean AE and Median AE for each method in Table 1, it appears that methods that avoid flipping errors (small mean AE) typically have a lower overall accuracy (high median AE), and vice versa. Our method obtains a relative improvement in the Mean AE of approximately 15% with respect to state of the art [15], that uses a 2D-based classify-and-refine approach, whereas we have state-of-the-art performance in terms of Median AE. This means that our algorithm is far less prone to flipping errors without being inferior in terms of overall accuracy.

Compared to regression-based approaches [6, 8, 28, 13], the improvement is even more substantial: more than 40% relative improvement for the mean AE. We present some reason for this large improvement in Figure 3, where we evaluate our algorithm in terms of the number of cells we use to encode the Fisher vector. The two endpoints mimic the approach of [28] (full image encoding) and [6, 8] (single feature encoding). As we can see, a small as well as a large number of cells provide inferior results, thus supporting our claim that an intermediate approach can be simultaneously robust to intra-class variability as well as successfully exploit geometric information.

Finally, in order to evaluate the effect of the discriminative term described in Section 4.1, we have run an additional experiment on the EPFL dataset. Instead of the term described in Equation (12), we replace it with a uniform distribution over [0°, 360°). This results in a mean AE of 19.97°and median AE of 3.4°. As expected, the discriminative term is especially helpful in reducing the mean AE, as many ambiguous situations that lead to "flipping" errors are solved. On the other hand, the median AE is hardly affected, showing that the generative part of our method provides an overall pose estimation that is already accurate.

Table 1. Results on the EPFL dataset in terms of Mean Absolute Error (AE) and Median Absolute Error (AE). "//" indicates that the result is not available in the corresponding paper.

	Mean AE [°]	Median AE [°]
Glasner *et al.* [11]	//	24.8
Pepik *et al.* [21]	//	4.7
Özuysal *et al.* [20]	46.48	//
Redondo *et al.* [24]	39.8	7
Teney *et al.* [27]	34.7	5.2
Torki *et al.* [28] (GT)	33.98	11.3
Fenzi *et al.* [6] (GT)	31.27	//
Hara *et al.* [13] (GT)	24.24	//
Yang *et al.* [32]	24.1	**3.3**
Zhang *et al.* [33] (GT)	24.00	//
Fenzi *et al.* [8] (GT)	23.28	//
He *et al.* [15]	15.8	6.2
Ours	**13.6**	**3.3**

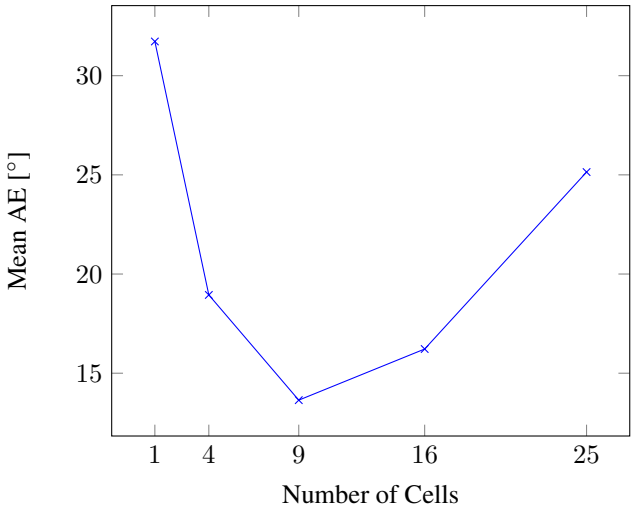

Figure 3. Performance of our algorithm in terms of number of cells.

YouTube & KITTI The second and third datasets on which we test our algorithm are the YouTube and KITTI datasets [31].

- YouTube presents 9 videos of racing cars over three

Table 2. Results for the YouTube dataset in terms of Viewpoint Accuracy / Mean AE, without (left) and with temporal information (right).

	Xiang *et al.* [30]	Ours	Xiang *et al.* [31] with 1st GT	Ours with LP
Race1	**0.52** / 42.62°	0.50 / **34.19**°	**0.67** / 18.73°	0.66 / **15.14**°
Race2	0.53 / 44.30°	**0.62** / **23.03**°	0.77 / 10.83°	**0.84** / **9.03**°
Race3	0.64 / 46.08°	**0.82** / **30.78** °	0.83 / 9.28°	**0.88** / **6.44**°
Race4	**0.79** / **13.37**°	0.61 / 42.15°	0.69 / 15.83°	**0.96** / **3.99**°
Race5	0.54 / 57.79°	**0.68** / **26.82**°	0.71 / 10.75°	**0.90** / **7.23**
Race6	**0.31** / **37.08**°	0.24 / 46.52°	**0.43** / **18.47**°	0.42 / 23.03°
Sedan	**0.79** / **20.84**°	0.72 / 29.70°	0.76 / **9.87**°	**0.94** / 10.09
SUV1	0.47 / 78.38°	**0.75** / **26.25**°	**0.82** / 7.81 °	0.77/ 10.66
SUV2	**0.39** / 63.41°	0.29 / **48.06**°	0.57 / **19.56**°	**0.63** / 19.80°
Mean	0.54 / 47.24°	**0.58** / **34.17**°	0.69 / 13.46°	**0.78** / **11.71**

Table 3. Results for the KITTI dataset in terms of Viewpoint Accuracy / Mean AE, without (left) and with temporal information (right).

	Xiang *et al.* [30]	Ours	Xiang *et al.* [31] with 1st GT	Ours with LP
KITTI01	**0.57** / **44.46**°	0.41 / 47.75 °	0.95 / 6.54°	**1.00** / **3.25**°
KITTI02	**0.33** / 119.54°	0.09 / **56.73**°	**1.00** / **5.40**°	0.82 / 9.82°
KITTI03	**0.50** / **15.99**°	0.30 / 23.78°	0.42 / 15.64°	**1.00** / **4.63**°
KITTI04	0.17 / 58.42°	**1.00** / **5.39**°	0.22 / 27.05 °	**1.00** / **1.31**°
KITTI05	**0.64** / 23.65°	0.57 / **13.23**°	0.36 / 23.59°	**1.00** / **6.68**°
KITTI06	0.59 / 20.29°	**0.69** / **20.06** °	0.31 / 21.63°	**1.00** / **7.60**°
KITTI07	0.70 / 24.50°	**1.00** / **3.72**°	0.96 / 6.86°	**1.00** / **2.40**°
KITTI08	**0.67** / 23.26°	0.57 / **13.66**°	**0.57** / 15.61°	**0.57** / **12.88**
KITTI09	0.50 / 17.60°	**1.00** / **0.76**°	0.50 / 21.63°	**1.00** / **0.60**°
KITTI10	**0.44** / 56.78°	0.35 / **26.01**°	**0.81** / **7.99**°	0.50 / 22.32
KITTI11	**0.68** / **12.29**°	0.52 / 20.92°	**0.88** / **9.33**°	0.57 / 17.90
Mean	0.53 / 37.89°	**0.59** / **21.09**°	0.63 / 14.66	**0.86** / **8.13**

different scenarios: racetrack, snowy road, and desert.

- KITTI is a selection from the KITTI dataset [9] collected by [31]. It comprises 11 videos of urban scenes recorded in the German city of Karlsruhe.

The ground truth viewpoint for each image has been manually annotated by fitting a 3D model [31]. The most difficult aspects of these datasets lie in the size variation of the cars during the sequence, occlusions and truncations, motion blur due to the high car speed, and the presence of smoke, snow and sand that often occlude the car, as shown in Figure 4.

We follow the testing paradigm introduced with these datasets, *i.e.*, we evaluate the pose estimation results only on the frames in which the IoU ratio is above 0.5 (PASCAL VOC criterion). Whereas both approaches of Xiang *et al.* are based on the object detection method proposed in [30], we use an off-the-shelf DPM detector. However, the two detectors obtain practically the same performance on these two datasets in terms of average IoU ratio (0.74 vs. 0.75 in YouTube and 0.54 vs. 0.58 in KITTI, as reported by [31]), so we deem the comparison fair.

We evaluate the performance with respect to two metrics:

- Mean Absolute Error (MAE): mean of the absolute difference between the estimated and the ground truth viewpoint for each sequence.
- Viewpoint Accuracy (VA): percentage of absolute errors smaller than 15°for each sequence.

We perform two different experiments on these datasets. First, we evaluate our method by comparing with [30], as none of the two methods exploit temporal information, *i.e.*, pose estimation is performed separately in each frame. Then, in order to compare with [31], we extend our model with a Linear Programming (LP) formulation to take temporal information into account, similarly to [7]. For this purpose, we exploit the posterior distribution delivered by our method trained on the first 10 sequences of the EPFL dataset. We create a graph by sampling from the posterior at each frame. We set the cost of passing through each node as a function of its pose probability and the cost of transiting from one node to another as a function of the viewpoint

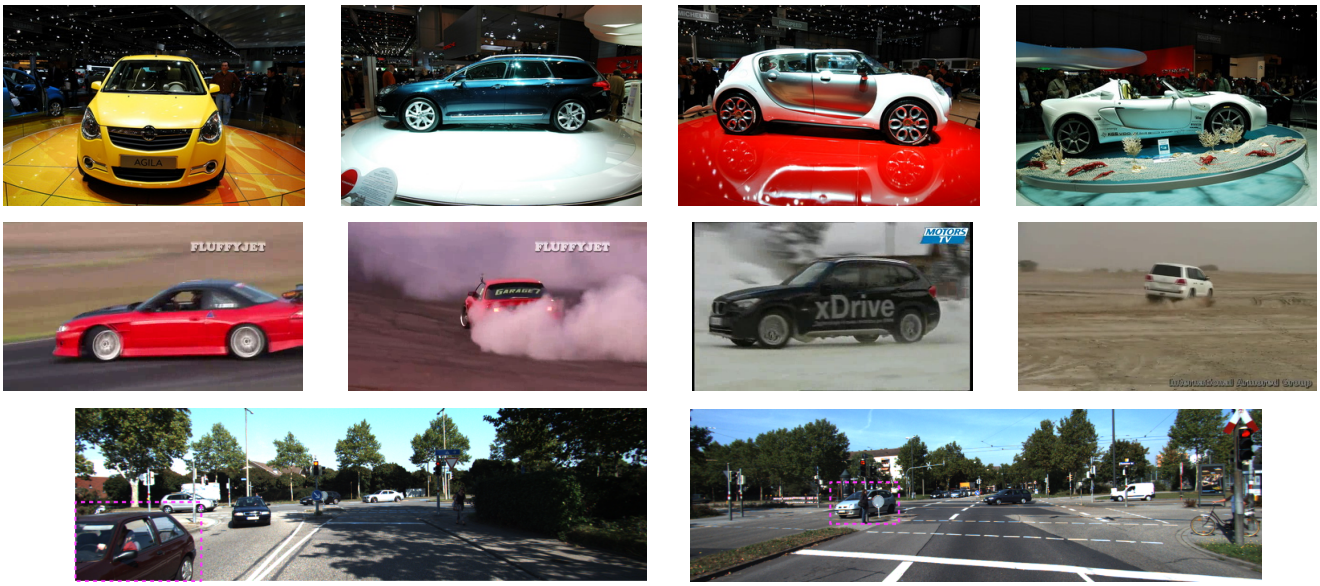

Figure 4. Sample images from the EPFL (top row), YouTube (middle row), and KITTI (bottom row) datasets. For the KITTI dataset, we framed the target car in dashed magenta. (Figure best viewed in color.)

difference. Finally, we find the pose path that minimizes the overall cost, and we compute our results on the nodes contained in this path.

We report the results of both evaluations for each dataset in Table 2 and Table 3, respectively. The left half of each table refers to the first experiment and the right half to the second experiment. Differently from [31], where the tracker is initialized with the ground truth pose of the first frame ("with 1st GT" in the table), we rely only on the strength of the LP tracker in finding the best path.

With regards to the first experiment, our method reduces the Mean AE by more than 25% on YouTube and more than 40% on KITTI with respect to the state of the art, which is a 3D-based pose estimation method. Since our results are superior with respect to both metrics, this implies that our method provides an effectively precise pose estimation.

In the second experiment, when temporal information is used, we prove that the posterior distribution provided by our method actually contains the correct evidence, even if the pose in individual frames is sometimes wrongly evaluated. In fact, this information is retained by the LP tracker that corrects spurious errors, and thus provides a large improvement in the overall accuracy.

6. Conclusion

We propose a novel method to perform continuous pose estimation of object categories on the basis of a spatially arranged ensemble of Fisher regressors. We build our class representation by analyzing the shortcomings of previous regression approaches, *i.e.*, approximate clustering, wrong matches and lack of geometrical context. In order to cope with these limitations, we combined feature regression and Fisher encoding. We work in a grid fashion by building a set of Fisher vectors on the basis of features densely extracted from each cell, so that geometrical information is automatically introduced in the approach. Then, we build an ensemble of Fisher regressors to predict the Fisher vector of the corresponding query cell for any viewpoint. Finally, we estimate the pose as the maximum of the posterior distribution computed on the basis of the regression errors. We introduce discriminativeness into our generative approach by means of an ensemble of SVM classifiers, thus avoiding large "flipping" errors in the classification.

We evaluate our method on three publicly available datasets that envisage different difficulties, such as high intra-class variability, occlusions, truncations, and motion blur. Our method provides results that are superior or comparable to the state of the art, thus showing that a regression-based approach is a valid solution to the continuous pose estimation problem.

Acknowledgments We acknowledge the support of the iMinds MMT department.

References

[1] A. Bakry and A. Elgammal. Untangling Object-View Manifold for Multiview Recognition and Pose Estimation. In *ECCV*, 2014. 3

[2] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005. 3

[3] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 2010. 6

[4] G. Fanelli, J. Gall, and L. V. Gool. Real Time Head Pose Estimation with Random Regression Forests. In *CVPR*, 2011. 1

[5] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *TPAMI*, 2010. 6

[6] M. Fenzi, L. Leal-Taixé, B. Rosenhahn, and J. Ostermann. Class Generative Models based on Feature Regression for Pose Estimation of Object Categories. In *CVPR*, 2013. 1, 2, 3, 6

[7] M. Fenzi, L. Leal-Taixé, K. Schindler, and J. Ostermann. Pose Estimation of Object Categories in Videos Using Linear Programming. In *WACV*, 2015. 7

[8] M. Fenzi and J. Ostermann. Embedding Geometry in Generative Models for Pose Estimation of Object Categories. In *BMVC*, 2014. 1, 2, 3, 6

[9] A. Geiger, P. Lenz, and R. Urtasun. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012. 7

[10] A. Ghodrati, M. Pedersoli, and T. Tuytelaars. Is 2D Information Enough For Viewpoint Estimation? In *ECCV*, 2014. 2

[11] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-Aware Object Detection and Pose Estimation. In *ICCV*, 2011. 6

[12] C. Gu and X. Ren. Discriminative Mixture-of-templates for Viewpoint Classification. In *ECCV*, 2010. 1, 2

[13] K. Hara and R. Chellappa. Growing Regression Forests by Classification: Applications to Object Pose Estimation. In *ECCV*, 2014. 1, 2, 3, 6

[14] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1998. 3

[15] K. He, L. Sigal, and S. Sclaroff. Parameterizing Object Detectors in the Continuous Pose Space. In *ECCV*, 2014. 1, 2, 3, 6

[16] T. Jaakkola and D. Haussler. Exploiting Generative Models in Discriminative Classifiers. In *NIPS*, 1999. 4

[17] J. Krapac, J. Verbeek, and F. Jurie. Modeling Spatial Layout with Fisher Vectors for Image Categorization. In *ICCV*, 2011. 4

[18] J. Liebelt and C. Schmid. Multi-View Object Class Detection with a 3D Geometric Model. In *CVPR*, 2010. 2

[19] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 2004. 3

[20] M. Özuysal, V. Lepetit, and P. Fua. Pose Estimation for Category Specific Multiview Object Localization. In *CVPR*, 2009. 2, 5, 6

[21] B. Pepik, P. Gehler, M. Stark, and B. Schiele. $3D^2PM$ - 3D Deformable Part Models. In *ECCV*, 2012. 2, 6

[22] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher Kernel for Large-scale Image Classification. In *ECCV*, 2010. 4

[23] T. Poggio and F. Girosi. Networks for Approximation and Learning. *Proceedings of the IEEE*, 1990. 4

[24] C. Redondo-Cabrera, R. López-Sastre, and T. Tuytelaars. All Together Now: Simultaneous Object Detection and Continuous Pose Estimation using a Hough Forest with Probabilistic Locally Enhanced Voting. In *BMVC*, 2014. 6

[25] K. Simonyan, O. Parkhi, A. Vedaldi, and A. Zisserman. Fisher Vector Faces in the Wild. In *BMVC*, 2013. 4

[26] J. Sivic. Efcient Visual Search of Videos Cast as Text Retrieval. *TPAMI*, 2009. 4

[27] D. Teney and J. Piater. Multi-view Feature Distributions for Object Detection and Continuous Pose Estimation. *CVIU*, 2014. 3, 6

[28] M. Torki and A. M. Elgammal. Regression from Local Features for Viewpoint and Pose Estimation. In *ICCV*, 2011. 1, 2, 3, 6

[29] K. van de Sande, C. Snoek, and A. Smeulders. Fisher and VLAD with FLAIR. In *CVPR*, 2014. 4

[30] Y. Xiang and S. Savarese. Estimating the Aspect Layout of Object Categories. In *CVPR*, 2012. 2, 7

[31] Y. Xiang, C. Song, R. Mottaghi, and S. Savarese. Monocular Multiview Object Tracking with 3D Aspect Parts. In *ECCV*, 2014. 6, 7, 8

[32] L. Yang, J. Liu, and X. Tang. Object Detection and Viewpoint Estimation with Auto-masking Neural Network. In *ECCV*, 2014. 2, 6

[33] H. Zhang, T. El-Gaaly, A. Elgammal, and Z. Jiang. Joint Object and Pose Recognition Using Homeomorphic Manifold Analysis. In *AAAI*, 2013. 6

[34] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3D Representations for Object Modeling and Recognition. *TPAMI*, 2013. 2