Embedding Geometry in Generative Models for Pose Estimation of Object Categories

Michele Fenzi
http://www.tnt.uni-hannover.de/staff/fenzi
Jörn Ostermann
http://www.tnt.uni-hannover.de/staff/ostermann
Institut für Informationsverarbeitung (TNT)
Leibniz Universität Hannover
Hannover, Germany

Abstract

Regression-based models built on local gradient-based feature descriptors have showed to be successful for continuous pose estimation of object categories. Nonetheless, a crucial weakness of these methods is that no geometric information is taken into account. Therefore, geometrically inconsistent poses may be preferred, and this forces to employ a coarse-grained pose estimator as a pre-processing step to avoid potentially large estimation errors. In this paper, we propose a method that combines generative feature models and graph matching techniques in a unified probabilistic formulation of the continuous pose estimation problem. Our approach retains the lightness and generality of generative feature modeling, while favoring geometrically consistent results. Experiments show that pose pre-processing steps are not needed if geometry is embedded in the matching stage. We evaluated our approach on two different car datasets and we experimentally show that our algorithm outperforms state-of-the-art methods by 25%.

1 Introduction

Pose estimation for object categories is becoming increasingly important and of interest for the Computer Vision community, both as a fundamental part of larger tasks or as a standalone challenge. Estimating the pose of an unknown object of a given class is usually addressed similarly to the case in which the specific object is known. However, a straightforward application of the latter approaches is strongly impaired by several difficulties, and compels to develop solutions that take into account the intrinsic differences between the two problems. On the one hand, intra-class variability works against a modeling tailored to individual objects, while, on the other hand, inter-class variation calls for a sufficient amount of discriminativeness between class models.

Among the many approaches proposed in literature, those based on local features have shown to work effectively for the solution of pose estimation problems for object categories. While some use explicit 3D information obtained through CAD models [14] or 3D reconstruction procedures [6], others have shown that the coupling of feature regression and view labeling is enough to solve this task [5, 17]. However, these methods rely solely on the discriminative power of gradient-based features without taking into account other cues. This has shown to be problematic when objects have similar appearance in different views, *e.g.*, the two side views of a car. Approaches that treat features as independent units, without

Figure 1: Features of a query image (left) are matched to a class generative model, visually represented by the two rightmost images. Matches are indicated by color. If matching is based only on the distance between feature descriptors, then it is impossible to disambiguate which view is correct. Even if absolute spatial distances between features are considered, ambiguity still remains. Only if oriented distances are taken into account, the correct configuration is favored. (Figure best viewed in color.)

taking into account any inter-relation among them, fail to solve these situations, and need to resort to external coarse-grained pose estimators for disambiguation [11].

The method we propose here integrates in a unified probabilistic framework

- A feature regression-based approach for pose estimation
- A graph matching-based approach to enforce geometric constraints on the solution.

Thereby, it retains the benefits of regression-based methods, like generality and lightness, while favoring geometrically consistent results. As an important consequence, our approach has no need to resort to any external pose pre-processing thanks to the integration of geometric cues in the formulation.

1.1 Contribution and Motivation

Contribution: We propose a method that solves the problem of pose estimation for object categories by integrating a feature regression-based approach and a graph matching technique in a unified probabilistic framework. We take geometric cues into account when matching test features to model features by using graph matching. Graph matching evaluates the geometrical consistency between pairs of candidate matches, and yields an affinity score that we interpret as a probability on the correctness of each match.

Another merit of our contribution is that our approach has no need to resort to a coarse pose pre-processing step. This is doubly important. On the one hand, our approach does not rely on the hard decision yielded by the coarse pose estimator, but on a soft matching that allows to postpone the problem solution to a later stage when more data is available. On the other hand, the treatment of the pose estimation problem is compact, and the claim of using a pure feature-based approach is only now totally correct.

Motivation: By using a regression-based approach, we treat pose estimation as a continuous problem unlike most methods that provide only discrete values for the pose [9, 13, 14]. We couple it with a graph matching-based approach to enforce geometric constraints on the solution, instead of relying only on the discriminative power of local feature descriptors [5]. In this regard, the motivation for our contribution is that local features alone are not effective when the object appearance is very similar in different views. For example, two symmetric views are nearly identical in terms of their features, thus there is no unique way to determine

the object pose by considering only appearance, as shown in Fig. 1. On the contrary, by taking geometry into account, these ambiguities are solved by exploiting the feature spatial structure.

The choice of graph matching is motivated by the fact that we deal with unknown objects and class models, and thus no rigid geometric transformation can easily be applied to align them. Graph matching not only solves pose ambiguities, but it permits to compute a *soft* geometric match between the test image and the model, bringing additional consistency and precision to the solution, even when a pure feature-based approach would suffice.

In Section 2, a review of related works is given. Section 3 introduces the feature regression approach, the graph matching formulation and the unified probabilistic framework in which they are coupled. In Section 4, we present experimental results on two public car datasets, and we give our final conclusions in Section 5.

2 Related Works

In literature, most of the works follow two main strategies to tackle the problem of pose estimation for object categories. One is based on 3D class models, *e.g*, obtained through Structure-from-Motion [6, 18] or synthetic CAD modeling [9, 10, 14]. In the former, a fused class model or individual training models vote for the pose either in image or in pose space. In the latter, 3D models are used in a refinement step after a coarse hypothesis has been generated by 2D-based SVM classifiers [4].

Our contribution follows the other direction, *i.e.*, to exploit only 2D data [15, 16] or to combine 2D information and viewpoint labeling [13]. The method proposed by [13] returns a quantized pose value as output of a SVM classifier bank trained for each discrete pose. In [2], starting from a similarly quantized pose, the output is refined by finding the maximum of a score function using view-deformed templates.

The two works that are most similar to ours are [17] and [5], as they both rely on a feature regression-based approach. While in [17] regression is applied to each set of object features as a whole, [5] advocates that performing regression separately on each local patch and combining patches of different training images better explains a query patch. We agree with this argument and we adopt this paradigm too, but [5] shows to be weak when different training views have many similar patches, as no spatial structure is considered. [17] introduces a distance term when driving the projection of the object features on a smaller dimensional manifold. Nonetheless, the method relies only on the absolute distance between each pair of features, which does not suffice to solve the pose ambiguity, as Figure 1 confirms.

The popularity of graph matching techniques in Computer Vision has increased over the years, thanks to their applicability to tasks such as object recognition [8], shape matching [2] or image matching [19]. Our formulation is inspired by [8], where graph matching is applied to object recognition and object categorization. [1] provides a closed-form solution for graph matching with one-to-one matching constraints. Graph matching extends to hypergraph matching when n matches at a time are considered, instead of pairs. Even though hypergraph matching-based approaches have shown to give better results [2, 19], the price to pay is a much higher computational burden due to the introduction of tensors and a greater implementation complexity.

3 Method

In the following, we first describe the basis of our method, the generative feature model, and how to build a generative class model from individual models. Then, we show our graph matching-based formulation to match a query image to a class model. Finally, we describe how the query pose can be estimated in a probabilistic framework, and how we integrate the results of graph matching therein.

3.1 Generative Feature Modeling

In this section, we briefly describe the regression-based method proposed in [5] that we leverage in our paper. We use a set of regression functions to model the behavior of gradient-based feature descriptors as a function of the pose. Each regressor predicts the descriptor of a certain patch in a query pose. This modeling relies on the smoothness in the amplitude variation of each descriptor component when the viewing angle changes.

Given a patch i, let $t^i = \{(f_1^i, \alpha_1^i), (f_2^i, \alpha_2^i), \ldots, (f_n^i, \alpha_n^i)\}$, *i.e.*, a set of n pairs, each composed of a feature descriptor f_j^i describing patch i under the corresponding viewing angle α_j^i and the viewing angle itself. For each feature track t^i, we create a generative feature model F^i as a linear combination of Gaussian kernels centered at the training poses,

$$F^i(\alpha) = \sum_{j=1}^{n} G(\alpha, \alpha_j^i)\mathbf{w}_j^i, \tag{1}$$

where G is an exponential function measuring the angular distance between two viewing angles. $\mathbf{w}_j^i$ are vector coefficients estimated from t^i by solving the following regularized linear least squares problem

$$(\mathbf{G}^i + \lambda \mathbf{I})\mathbf{W}^i = \mathbf{Z}^i \tag{2}$$

where $\mathbf{G}^i$ is a $n \times n$ matrix such that $\mathbf{G}_{lm}^i = G(\alpha_l^i, \alpha_m^i)$, $\mathbf{I}$ is the identity matrix, $\mathbf{W}^i$ and $\mathbf{Z}^i$ are matrices containing the unknown coefficients and the feature descriptors of t^i stacked in row order, respectively.

In order to have a unique class representation stemming from different training instances, we cluster all the tracks collected during training on the basis of their similarity in descriptor and pose space using spectral clustering. Each entry in the $N \times N$ similarity matrix used for spectral clustering is the alignment score of a pair of tracks.

At run time, query features must be matched against a set of feature cluster representatives, which are the centers of the corresponding cluster in descriptor space. The simple nearest neighbor matching proposed by [5] is prone to the intrinsic ambiguity occurring with similar views and, in any case, geometrical context is not taken into account. In the following, we introduce our contribution based on graph matching to exploit the inherent spatial ordering of the features to improve matching quality.

3.2 Graph Matching

In this section, we describe the graph matching-based approach we use in order to favor geometrically consistent poses. Graph matching is a commonly used technique to solve set correspondence problems when the two sets to be matched have some internal structure that should be respected, *e.g.*, sets of contour points. In the graph matching paradigm, the two sets are considered as two separate graphs and the correspondence problem is thus interpreted as

a graph matching problem. More specifically, each set is interpreted as an attributed graph defined by the triple $G = (V, E, A)$, where V is the set of vertices, E is the set of edges and A is an attribute matrix. Each entry A_{ij} is a multi-dimensional attribute for the edge $e_{ij} \in E$ and represents some *relationship* between vertices $i, j \in V$. Attributes are also defined for loops e_{ii}, and can be defined differently with respect to off-diagonal attributes.

Therefore, given two feature sets and their corresponding attributed graphs $G = (V, E, A)$ and $G' = (V', E', A')$, we are interested in a mapping $M = \{(i, i') | i \in V, i' \in V'\}$ of the vertices of the two sets that best respects the original attributes by maximizing the graph matching score S,

$$S = \sum_{(i,i') \in M, (j,j') \in M} g(A_{ij}, A'_{i'j'}), \tag{3}$$

where g is a function that evaluates the similarity between two attributes. M can also be rewritten as a binary vector $\mathbf{x} \in \{0,1\}^{nn'}$, where $n = |V|$, $n' = |V'|$, and $x_{ii'} = 1$ if the correspondence $(i, i') \in M$. Therefore, the set correspondence problem is solved by finding the vector $\mathbf{x}^*$ that maximizes the matching score,

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} S = \arg\max_{\mathbf{x}} \mathbf{x}^T \mathbf{W} \mathbf{x}, \qquad \text{s.t . } \mathbf{x} \in \{0,1\}^{nn'} \text{ and } \mathbf{C}\mathbf{x} = \mathbf{b}, \tag{4}$$

where $\mathbf{W}$ is a $nn' \times nn'$ matrix with $W_{ii',jj'} = g(A_{ij}, A'_{i'j'})$. $\mathbf{C}\mathbf{x} = \mathbf{b}$ is a set of linear constraints that may be imposed on the solution, *e.g.*, to guarantee a one-to-one matching.

Since Integer Quadratic Problems are NP-hard, we adopt an approximate solution. We relax the problem by admitting multiple matches and dropping the integral constraint for a solution that can take real values in $[0, 1]$. In order to apply the Raileigh's ratio theorem, we need to fix the norm of $\mathbf{x}$. By fixing $\|\mathbf{x}\| = 1$, our relaxation is satisfied. In addition, if $\mathbf{W}$ has only non-negative entries, the Perron-Frobenius' theorem guarantees that all entries in $\mathbf{x}^*$ are in the interval $[0, 1]$, thus providing the advantage that the solution can be directly interpreted in probabilistic terms. Thereby, the IQP problem reduces to the following form,

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} S = \arg\max_{\mathbf{x}} \mathbf{x}^T \mathbf{W} \mathbf{x} \qquad \text{s.t . } \|\mathbf{x}\| = 1. \tag{5}$$

According to Raileigh's ratio theorem, the solution $\mathbf{x}^*$ is the principal eigenvector of $\mathbf{W}$, *i.e.*, the eigenvector with the largest associated eigenvalue.

Now, we show how we integrate graph matching in our framework. We consider all test features as nodes of the test graph G and a subset of the model features $\mathcal{C} = \{c\}_1^N$ as nodes of the model graph G'. The reduction of the model size is necessary in order to avoid an intractably large problem. For each test feature, we only consider the K nearest neighbors in the model as model features. This pruning removes matches that are very far in descriptor space and permits to focus on disambiguating candidate matches on a geometric basis.

With regard to the attribute matrix A, the following assignment is considered

$$A_{ij} = \begin{cases} f_i & \text{for } i = j \\ (\alpha_{ij}, r_{ij}) & \text{for } i \neq j \end{cases} \tag{6}$$

where α_{ij} is the angle between the x-axis and the directed segment P_{ij} connecting the two test feature point locations, r_{ij} is the length of P_{ij}, f_i is the test feature descriptor. The model attribute matrix A' is defined similarly, by considering the 2D location of the model feature as the average location of all the points in its cluster and the model feature descriptor as the cluster representatives.

Correspondingly, each entry $W_{ii',jj'}$ is defined as follows:

$$W_{ii',jj'} = \begin{cases} \log_{10}(m - d_{ii'} + 1) & \text{if } i = j \text{ and } i' = j' \\ m\left(1 - \frac{\beta}{\tau_1}\right)^2 \left(\frac{\tau_2 - \rho}{\tau_2 - 1}\right)^2 & \text{if } \beta \leq \tau_1 \text{ and } 1 \leq \rho \leq \tau_2 \text{ and } (i \neq j \text{ or } i' \neq j') \\ m\left(1 - \frac{\beta}{\tau_1}\right)^2 \left(\frac{\tau_2 \rho - 1}{\tau_2 - 1}\right)^2 & \text{if } \beta \leq \tau_1 \text{ and } \frac{1}{\tau_2} \leq \rho < 1 \text{ and } (i \neq j \text{ or } i' \neq j') \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The first line refers to the diagonal entries of the attribute matrix, and it takes only appearance into account as in standard feature matching. Since $d_{ii'} = \|f_i - f_{i'}\|$ is the distance in descriptor space and $m = \max_{ii'} d_{ii'}$, a high entry is assigned to feature pairs that are close in descriptor space. The remaining three lines involve the enforcement of the geometric structure, where $\beta = |\alpha_{ij} - \alpha_{i'j'}|$ is the absolute angular distance and $\rho = \frac{r_{ij}}{r_{i'j'}}$ is the Euclidean distance ratio. The absolute orientation difference and the length ratio of the two segments are compared to two thresholds, τ_1 and τ_2, and a matching score is defined accordingly. A high entry is assigned to feature pairs whose locations are geometrically consistent, both in orientation and length.

All entries of $\mathbf{x}^*$ are in $[0,1]$. Each entry reflects the degree of association of that match to the main cluster in terms of appearance and geometry. In other words, it represents a confidence measure about the correctness of that match.

3.3 Pose Estimation in a Probabilistic Framework

After extracting a set of features $\mathcal{F} = \{f\}_{m=1}^{M}$ from the query image, we match each test feature against the model as we have described in Sec. 3.2, where the model is represented by a set $\mathcal{C} = \{c\}_1^N$ of feature clusters. After the graph matching step, for each feature f we can define a probability $p(\alpha, c|f)$ that expresses the likelihood of observing the object from the viewpoint α and that c is a correct match for f. We can also write this probability as

$$p(\alpha, c|f) = p(\alpha|f,c)p(c|f). \tag{8}$$

The returned pose and final matching (α^*, c^*) are those maximizing the latter probability,

$$(\alpha^*, c^*) = \arg\max_{(\alpha,c)} p(\alpha|f,c)p(c|f) \tag{9}$$

The first factor $p(\alpha|f,c)$ can be expressed in terms of the generative feature model as

$$p(\alpha|f,c) = \sum_{i:i^i \in c} \frac{u_i}{U} \exp\left(-\frac{(e^i)^T R^i e^i}{2}\right) G(\alpha, \beta_i), \tag{10}$$

where U is a normalization constant and $u_i = \min_j \|f - f_j^i\|$ weighs the contribution of the i-th regressor. $e = f - F^i(\alpha)$ is the prediction error made by the i-th regressor in cluster c to the test descriptor f, R_i is the covariance matrix of the i-th regressor estimated during training. $\beta_i = \arg\min_j |\alpha - \alpha_j^i|$ weighs the view consistency of the i-th regressor in cluster c to the tentative pose α.

In the formulation in [5], the term $p(\alpha)$ is used to introduce a pre-processing pose classifier that acts as a support in estimating the pose. The classifier generates a pose prior that is

uniformly distributed over the output bin and is null elsewhere. This hard decision proves to be disadvantageous, as oftentimes the output is wrong, and the class generative model cannot revert this. On the contrary, in our approach no pose is favored over others, and we let the graph matching results drive the maximization. Experimentally, this proves to be beneficial, as we are able to decrease the mean absolute error by 25%, as shown in Secrion 4.

We derive the term $p(c|f)$ straightforwardly from the graph matching results. Since $\|\mathbf{x}\| = 1$ and $x^*_{fc} \in [0,1]$, we can interpret the square of each score as a probability, so that

$$p(c|f) = \frac{(x^*_{fc})^2}{\sum_{c:f\sim c}(x^*_{fc})^2}, \tag{11}$$

where $\sim$ indicates a candidate match.

Now we consider all query descriptors $\mathcal{F} = \{f\}_{m=1}^{M}$ and we assume a mixture model where each feature contributes equally to avoid cancellation due to outliers. Thus, we obtain the estimation for the pose and the final matching by maximizing the following

$$(\alpha^*, c^*) \approx \arg\max_{(\alpha,c)} \sum_m p(\alpha|f_m, c)p(c|f_m) \tag{12}$$

4 Experimental Results

In this section, we show that our algorithm performs better than other state-of-the-art methods based on feature regression [5, 17], as the introduction of spatial context proves to be beneficial in solving the pose estimation problem.

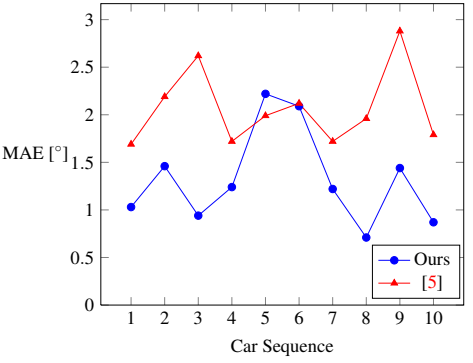
Vehicles are common dataset objects with a strong symmetry, and they proved to be very challenging for the aforementioned methods. Thus, we tested our approach on two car datasets: the EPFL multi-view car dataset [13] and the PASCAL VOC 2006 dataset [3]. The first dataset encompasses a set of car sequences rotating on a platform. Each sequence is provided with snapshot times, so that an orientation label can be assigned to each image. The challenge of this dataset is to build a model that is representative enough given the highly variable training set, which ranges from city cars and station wagons to car prototypes and racing cars. The second dataset consists of many different classes, of which we considered only cars. Sample images of both datasets are shown in Figure 2b.

Before showing the better performance of our method in estimating the pose for object categories, we compare it to [5] in a single instance pose estimation experiment.

4.1 Single Instance Pose Estimation

We test our algorithm on the first 10 car sequences of the EPFL dataset with a 33% split between training and testing, *i.e.*, one image every three for learning and the rest for testing. For each sequence, we track features over the training images and we compute a regression function for each track, as described in Section 3.1. The class model in this case coincides with the exemplar model itself and no clustering is performed.

For each test image, we extract a set of query features and, for each feature, we find the $K = 2$ nearest neighbors in the model. Then, we apply our graph matching-based approach to score the candidate matches, as explained in Section 3.2. Finally, we obtain the estimation of the pose by maximizing Eq. (12).



(a) Individual case. Our method outperforms [5] by 36% on average.

(b) Sample images from the EPFL (top row) and Pascal VOC 2006 (bottom row) car dataset.

Method	MAE [°] 90th percentile	MAE [°] 95th percentile	MAE [°]
Ozuysal *et al.* [13] (Baseline)	-	-	46.48
Torki *et al.* [17] - 50% split	19.4	26.7	33.98
Fenzi *et al.* [5] - 50% split	14.51	22.83	31.27
Ours - 50% split	**12.67**	**17.77**	**23.38**
Torki *et al.* [17] - LOO split	23.13	26.85	34.90
Fenzi *et al.* [5] - LOO split	**14.41**	22.72	31.16
Ours - LOO split	15.53	**19.27**	**24.53**

Table 1: EPFL dataset. Our method compared to [13], [17] and [5].

We evaluated the performance of our method using SIFT features [12], as [5] shows they give better results with respect to lower-dimensional features. We compared the mean absolute error (MAE) in degrees between the ground truth orientation and the result returned by each algorithm.

In comparison to [5], that reports an average error of 2.06°, our method achieves a MAE over the 10 sequences of 1.31°, providing thus an improvement in accuracy of approximately 36%, as shown in Figure 2a. Given that the distance in pose space of each training sample ranges from 7.5° to 18°, our method proves to be very accurate.

4.2 Class Pose Estimation - EPFL Dataset

In this section, we compare our method to [5, 13, 17] on the EPFL multi-view car dataset for class pose estimation. We used the same testing framework, *i.e.*, two different splits between training and testing. *(i) 50% Split*: training the model on the first 10 sequences and testing it on the remaining 10; *(ii) Leave One Out (LOO)*: training the model on 19 sequences and testing it on the remaining one.

We build our model according to Section 3. Then, we extract a set of features $\mathcal{F}$ from each query image and, for each feature, we find its $K = 5$ nearest neighbors in the model. Then, we use our graph matching-based approach to assign a score to the candidate matches,

Method	MAE [°]
Fenzi *et al.* [5]	28.50
Fenzi *et al.* [5] with pose pre-processing [11]	14.70
Ours	**14.49**

Table 2: PASCAL VOC 2006 dataset. Our method compared to [5], without and with [11].

as explained in Section 3.2. With respect to the single instance case, the number of potential matches per test feature K is increased to take imperfect clustering into account. Finally, we estimate the car pose by maximizing Eq. (12).

We compared against three other methods: [13] as they introduced the dataset, [17] and [5] which are state-of-the-art methods using a regression-based approach. In Table 1, we can see that our method outperforms all others. In particular, our absolute MAE is 25% smaller with respect to the results published in [5]. This experimentally shows that introducing a soft geometric match is beneficial with respect to a hard decision based on a pre-processing pose estimator [11]. In the first two columns of Table 1, we also provide results of our method in terms of the 90^{th} and the 95^{th} percentile. Even by discarding most of the large errors due to 180° flipped estimations, our method still obtains a better accuracy. As in [17] and [5], the performance of our method with 50% and LOO splits is similar, showing that the model relies on the first 10 sequences to estimate the final pose, while the second 10 sequences seem to introduce a small amount of noise in the model.

4.3 Class Pose Estimation - PASCAL VOC 2006 Dataset

In this section, we compare our method to [5] without and with the pre-processing 4-bin pose classifier [11]. We consider the car subset in the PASCAL VOC 2006 test dataset. More precisely, we consider all pictures in which the car is in one of the four annotated orientations: front, rear, left side, right side. All three methods are trained with the first 10 sequences of the EPFL dataset.

As shown in Table 2, our approach performs better than [5] by a factor of 2, when [5] is used without any pre-processing step. More importantly, the performance is still better even when the pre-processing step is used. Unlike in the EPFL experiment, where the DPM-based pose classifier has a non-negligible error, its performance for this dataset is almost perfect (96% accuracy). Therefore, our method not only recovers the correct orientation over the whole pose range (360°) instead of the smaller (90°) correct interval given by the classifier, but it is also more accurate.

5 Conclusions

We proposed a method that combines an approach for continuous pose estimation for object categories based on feature regression and a graph matching strategy that helps disambiguating the pose solution. Our method exploits the advantages of class generative models to predict the behavior of gradient-based feature descriptors as a function of a given view. In addition, it takes the feature spatial ordering into account during the matching stage on the basis of a graph matching strategy that enforces geometric constraints on the solution. Experiments show that our approach outperforms state-of-the-art algorithms by 25% for class pose

estimation tasks, as the introduction of geometric context permits to solve view-problematic situations as well as to provide an overall additional accuracy.

References

[1] T. Cour, P. Srinivasan, and J. Shi. Balanced Graph Matching. In *NIPS*, 2006.

[2] O. Duchenne, F. R. Bach, I.-S. Kweon, and J. Ponce. A Tensor-based Algorithm for High-order Graph Matching. In *CVPR*, 2009.

[3] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[4] P. F. Felzenszwalb, R. B. Girshick, and D. A. McAllester. Cascade Object Detection with Deformable Part Models. In *CVPR*, 2010.

[5] 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.

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

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

[8] M. Leordeanu and M. Hebert. A Spectral Technique for Correspondence Problems Using Pairwise Constraints. In *ICCV*, 2005.

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

[10] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent Object Class Detection Using 3D Feature Maps. In *CVPR*, 2008.

[11] R. J. López-Sastre, T. Tuytelaars, and S. Savarese. Deformable Part Models Revisited: A Performance Evaluation for Object Category Pose Estimation. In *ICCV Workshops*, 2011.

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

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

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

[15] S. Savarese and L. Fei-Fei. 3D Generic Object Categorization, Localization and Pose Estimation. In *ICCV*, 2007.

[16] S. Savarese and L. Fei-Fei. View Synthesis for Recognizing Unseen Poses of Object Classes. In *ECCV*, 2008.

[17] M. Torki and A. M. Elgammal. Regression from Local Features for Viewpoint and Pose Estimation. In *ICCV*, 2011.

[18] J. Xiao, J. Chen, D.-Y. Yeung, and L. Quan. Structuring Visual Words in 3D for Arbitrary-View Object Localization. In *ECCV*, 2008.

[19] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *CVPR*, 2008.