Appearing in the proceedings of the 6th Iberian Conference on
Pattern Recognition and Image Analysis (IbPRIA), 2013.

Modeling Pose/Appearance Relations for Improved Object Localization and Pose Estimation in 2D images

Damien Teney
University of Liège, Belgium
Damien.Teney@ULg.ac.be

Justus Piater
University of Innsbruck, Austria
Justus.Piater@UIBK.ac.at

Abstract. We propose a multiview model of appearance of objects that explicitly represents their variations of appearance with respect to their 3D pose. This results in a probabilistic, generative model capable of precisely synthesizing novel views of the learned object in arbitrary poses, not limited to the discrete set of trained viewpoints. We show how to use this model on the task of localization and full pose estimation in 2D images, which benefits from its particular capabilities in two ways. First, the generative model is used to improve the precision of the pose estimate much beyond nearest-neighbour matching with training views. Second, the pose/appearance relations stored within the model are used to resolve ambiguous test cases (e.g. an object facing towards/away from the camera). Here, changes of appearance as a function of incremental pose changes are detected in the test scene, using a pair or triple of views, and are then matched with those stored in the model. We demonstrate the effectiveness of this method on several datasets of very different nature, and show results superior to state-of-the-art methods in terms of accuracy. The pose estimation of textureless objects in cluttered scenes also benefits from the proposed contributions.

1 Introduction and related work

We focus on the problem of 3D pose estimation of known objects in 2D images, using multiple registered images of the objects as training examples. Pose estimation, which is closely coupled to the related tasks of object recognition and localization, is a fundamental problem in computer vision and has naturally received great interest over the years. The main contribution of this paper is to explicitly include, in an existing multiview model of appearance [14], the possible changes of appearance undergone by the object as it pose varies between the trained viewpoints. With the exception of [9], this is, to our knowledge, the only work to include such information within a model of appearance in the context of pose estimation. We make use of this additional information in two different ways to improve the precision and accuracy of pose estimation. In the following we relate our approach to related work.

Multiview models of appearance. The traditional methods for object recognition using 2D images alone, known as appearance-based, typically use

specific models for individual viewpoints, e.g. a model for cars seen from the front, and another for cars seen from the side. Recent contributions in object recognition have introduced more and more models of appearance that include different viewpoint and that are also relevant to pose estimation. Some methods still treat those different viewpoints somewhat independently [14,18], while others try to match and link features across viewpoints [5,10,16]. Savarese *et al.* [10], for example, model an object as a collection of planar parts that can appear in different views. We follow an intermediate approach, by storing independently the image features that make up the different views, but we also store, along with every each image feature, how its appearance varies with respect to the pose of the object. The multiview models mentioned above were mainly used on the task of localization and recognition, without recovering a 3D pose explicitly, or only as rough estimate such as "frontal view" or "side view". We rather focus on *continuous* pose estimation, to recover precise 3D position and orientation, as is needed, e.g. for robotic applications [7,18].

Continuous pose estimation. The classical approach to pose estimation using 2D training examples is to match highly discriminative features between the test and training views. These matches then vote for the most similar training example, yielding a nearest-neighbour classification of limited precision. Some authors have proposed averaging [18] and probabilistic smoothing [14,15] schemes to increase precision beyond the resolution of the training examples on the viewing sphere. While these procedures basically perform some averaging between trained viewpoints, we rather explicitly detect, and include in the model, the deformations and the transformations of appearance between the discrete viewpoints seen during training. We then use this information in our generative model to finely optimize the 3D pose, starting from a rough nearest-neighbour estimate. Another, radically different approach was proposed by Torki and Elgammal [17], who learn a regression from local image features to the pose of the object. This original approach recovers a precise pose, but cannot handle significant clutter or occlusions, and the accurate pose estimation depends on the (supervised) enforcement of a one-dimensional manifold constraint (corresponding to the 1D rotation of the object in the training examples). It is not clear how that approach would extend to the estimation of the full 3D pose of an object.

New view synthesis. Our approach uses dense optical flow to identify the deformations between pairs of neighbouring training views. Only those parts of this dense information are then retained that correspond to the sparse image features actually stored in the model. This information can then be used in a generative manner, to synthesize the appearance of the object in a new, unseen pose, by transforming the image features of nearby trained viewpoints according to those stored deformations. The problem of new view synthesis has been studied in the field of computer graphics through the technique of *morphing* [1,2,11]. Most methods only consider pairs or triples of views, whereas we are interested in modeling and using transformations over the whole viewing sphere. Morphing algorithms also often rely on established correspondences between specific

image features of the input views [11], whereas we use dense optical flow to identify deformations between neighbouring views, before applying them to sparse features. As an advantage, our approach readily applies to difficult-to-match features (as opposed to the competing method of Savarese *et al.* [9]). This practically allows handling non-textured objects containing little detail. Although some global consistency in the detected deformations is enforced by optical flow algorithm, each image feature independently stores its possible deformations. This does not limit the model to a particular class of overall transformations. On the contrary, Savarese *et al.* [9] specifically models affine transformations of object parts, assuming that large planar parts can be identified (which is not a universal property of objects). Note also that we use a sparse set of training views (typically spaced about 20° apart on the viewing sphere) and do not require videos or dense sequences of images to track features between frames, as opposed to Sun *et al.* [13]. One may also note a similarity in spirit with the classical active appearance models used mainly for object tracking; they are however based on and limited by point-wise matches of specific landmarks.

Active vision. In addition to the generative model we use to refine pose estimates, we show how to use the deformations stored in the model to resolve ambiguous test cases (Section 4). In such scenes, the 2D appearance of the object can equally correspond to several 3D poses (see Fig. 4 for an example). We propose to also identify the changes of appearance with respect to the pose in the *test scene*. The camera is therefore allowed to move slightly in two orthogonal directions on the viewing sphere (around the test scene). The changes of appearance are detected — as with the training views — and used as extra dimensions in the descriptor of the image features. The features of the test scene can then be matched more discriminatively with those of the training data, and effectively identify the single correct pose unambiguously. This procedure, which can prove crucial in robotic applications, has been proposed in the field of active vision [3,12], but not integrated, to our knowledge, in such a straightforward manner within a pose estimation method. It resembles the way humans themselves resolve such perception ambiguities (in addition to stereo vision) by moving around the scene. Note that a similar effect can be obtained by fusing the result of independent pose estimations from multiple 2D images [14,18]. Our integrated procedure is however arguably more efficient, the displacement (change of camera position) does not need to be precisely know, and can also be very small (theoretically infinitesimal small, even though image noise and resolution dictate minimum displacements in practice).

2 Representation of training and test views

2.1 Notations for image features and object poses

The contributions of this paper are integrated with the method proposed in earlier work [14]. That method performs object recognition and pose estimation in 2D images, and is applicable to various types of image features. This includes features that cannot easily be matched between training and test views, such as

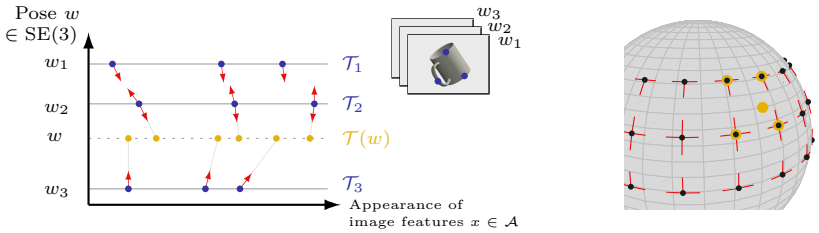

Fig. 1: Left: schematic representation of training data, in the dimensions of pose and appearance. Image features (blue points) are extracted from training images, and their possible changes of appearance (red arrows) are identified between neighbouring views. The appearance of the model at a novel view w (orange) is generated by adjusting the features of close-by views, according to those stored deformations. Right: representation of a set of training viewpoints (black dots) on the viewing sphere. The changes of appearance are detected between pairs of neighbouring views (red links). The appearance of a novel view (orange dot) is generated using the closest training viewpoints, four in this case (orange circles).

the edge points we use in our implementation. We will first review the notation for representing the test and training views, followed by the generative model, capable of synthesizing novel views.

The test data corresponds to a single 2D image of a scene, from which we extract image features. In general, an image feature x is defined by its localization in the image x^{p}, and an optional appearance descriptor x^{a}. Together, they are defined on the *appearance* space $\mathcal{A}$. Practically, we use points identified along edges, combined with the local orientation of the edge (see Fig. 3), so that $\mathcal{A} = \mathbb{R}^2 \times S_1^+$ (the 2D localization plus an orientation without direction). The image features from the test view form a set of *observations* $\mathcal{O} = \{x_i\}_i$, with $x_i \in \mathcal{A}$. The training data correspond to a series of K images of the object of interest, in different poses $w_k \in \mathrm{SE}(3)$ with $k = 1, \ldots, K$. Image features are extracted from each training view k to form a set $\mathcal{T}_k = \{x_i\}_{i=1}^{M_k}$, with $x_i \in \mathcal{A}$.

A pose $w \in \mathrm{SE}(3)$, which defines a 3D location together with a 3D orientation, conveniently decomposes into separate sets of dimensions w^{v} and w^{t}. We call w^{v} the *viewpoint transformations* (defining which side of the object is facing the camera, i.e. an element of S^2) and w^{t} the set of *in-plane transformations* (i.e. translations and rotations parallel to the image plane, and depth/scale changes). In-plane and viewpoint transformations are considered separately, since the changes of appearance induced by the former are fixed by the calibration of the camera. The calibration is assumed to be known, and those transformations can thus be formally hard-coded in the function $\mathrm{transformInPlane}_{w^{\mathsf{t}}}(\mathcal{T}) = \mathcal{T}'$, which transforms a set of image features $\mathcal{T}$ according to the in-plane transformations w^{t}. Without loss of generality, the following discussion will assume that the training views have been normalized for in-plane transformations, that is, centered and set to a similar scale/rotation[1].

[1] Formally, $\mathcal{T}_k \leftarrow \mathrm{transformInPlane}_{-w_k^{\mathsf{t}}}(\mathcal{T}_k)$ and $w_k^{\mathsf{t}} \leftarrow 0, \ \forall k$.

2.2 Generative model of training data

The training data, as presented above, defines the appearance of the object of interest at a set of discrete trained viewpoints. The goal of our generative model is to fill in the gaps between those viewpoints. Although it may be possible to establish explicit correspondences between image features of nearby training images, this approach may not always be reliable, and it does not generalize to dense or non-discriminative image features such as our edge points. Therefore, we choose to identify *dense* deformations between pairs of adjacent training views, using an optical flow algorithm. Those deformations will then be combined to deform the image features of the training images into the novel viewpoint.

More precisely, for an arbitrary viewpoint w^{v}, we identify its closest training viewpoints $\mathrm{nb}(w^{\mathsf{v}}) = \{k : d(w^{\mathsf{v}}, w_k^{\mathsf{v}}) \leq t\}$, where $d(\cdot,\cdot)$ measures the angular distance between two viewpoints. The threshold t is chosen similar to the typical angular distance between neighbouring viewpoints in the training data. We also identify the set of all neighbouring training viewpoints as $\mathrm{NB} = \{(k, k') : k' \in \mathrm{nb}(w_k^{\mathsf{v}}), k \neq k'\}$. During an off-line training phase, an optical flow algorithm [6] is applied on all pairs of views $(k, k') \in \mathrm{NB}$. Each pair produces a dense flow map $\mathrm{UV}_{k\to k'}(x)$ that corresponds, in our case, to the local deformation (translation in the image plane) undergone at an image location x when moving from viewpoint k to k'. Although we compute a *dense* optical flow, we only need to store the actual values of the maps UV for the positions of the few image features of each view. We can now define our generative model $\mathcal{T}(w)$, which produces a set of image features corresponding to the appearance of the object in an arbitrary pose w. Its definition combines the image features of all nearby training views, individually translated using the stored deformations, then adjusted for in-plane transformations. Formally,

$$\mathcal{T}(w) \;=\; \bigcup_{k\in\mathrm{nb}(w^{\mathsf{v}})} \mathrm{transformInPlane}_{w_k^{\mathsf{t}}\to w^{\mathsf{t}}}\big(\;\mathrm{deform}_{w_k^{\mathsf{v}}\to w^{\mathsf{v}}}(\mathcal{T}_k)\;\big)\,. \tag{1}$$

The functions transformInPlane() and deform() adjust a set of image features respectively for in-plane and out-of-plane transformations. While the definition of the former is trivial (it just applies the translation, scaling and rotation of its parameter), the latter is more complex. It uses a linear combination of two available deformations to translate each image feature. We denote those two deformations by the indices of the viewpoints that generated them, and call them (k, k') and (k, k''). They are chosen from NB so that the novel viewpoint can be reached (on the viewing sphere) by a positive linear combination of them. Consequently, $\exists\ \alpha, \beta \in \mathrm{R}^+ : w^{\mathsf{v}} = w_k^{\mathsf{v}} + \alpha(w_{k'}^{\mathsf{v}} - w_k^{\mathsf{v}}) + \beta(w_{k''}^{\mathsf{v}} - w_k^{\mathsf{v}})$. Practically, this means that the viewpoints k, k' and k'' cannot be collinear on the viewing sphere. With training viewpoints spaced on a grid, as in our experiments, we simply choose k' and k'' respectively along the changes in azimuth and elevation. It is now straightforward to define the deform() function that combines those two chosen deformations:

$$\begin{aligned}\mathrm{deform}_{w_k^{\mathsf{v}}\to w^{\mathsf{v}}}(\mathcal{T}_k) = \{x_i' :\;\; & x_i'^{\mathsf{p}} = x_i^{\mathsf{p}} + \alpha\,\mathrm{UV}_{k\to k'}(x_i^{\mathsf{p}}) + \beta\,\mathrm{UV}_{k\to k''}(x_i^{\mathsf{p}}) \\ & \text{and}\quad x_i'^{\mathsf{a}} = x_i^{\mathsf{a}}\;,\quad \forall x_i \in \mathcal{T}_k\;\}\,.\end{aligned} \tag{2}$$

3 Refinement of pose with generative model

3.1 Method

The proposed generative model readily integrates with the method proposed in [15]. That method for pose estimation relies on continuous distributions of image features in the appearance space to represent the training and test views, using kernel density estimation. These distributions are simply built using the elements of $\mathcal{O}$ and $\mathcal{T}(w)$ as particles, giving respectively $\phi_{\mathcal{O}}(x) = \frac{1}{|\mathcal{O}|} \sum_{x_i \in \mathcal{O}} \mathsf{K}(x, x_i)$ and $\phi_{\mathcal{T}(w)}(x) = \frac{1}{|\mathcal{T}(w)|} \sum_{x_i \in \mathcal{T}(w)} \mathsf{K}(x, x_i)$, with $\mathsf{K}(\cdot, \cdot)$ a kernel on $\mathcal{A}$.

We reuse the base method proposed in [15] to obtain initial proposals for the 3D pose of the object in the new scene. That method iterates over the training viewpoints and some discrete values of scale and in-plane rotation, then uses a probabilistic voting scheme between matching training and test features to identify the most probable image location. The peaks with high voting scores are then retained as initial pose estimates. This method basically corresponds to a nearest-neighbour identification of the training views in the test scene, and gives us initial estimates to refine by local optimization.

Using the two distributions of image features presented above, and a cross-correlation between them as a measure of similarity [14,15], we now have a likelihood function that can be used to evaluate any arbitrary pose w:

$$\begin{aligned} p(w) \;&=\; \int_{\mathcal{A}} \phi_{\mathcal{O}}(x)\, \phi_{\mathcal{T}(w)}(x)\, \mathrm{d}x \;, \quad \text{approximated with} \\ p(w) \;&\approx\; \frac{1}{n} \sum_{i}^{n} \phi_{\mathcal{O}}(x_i) \quad \text{where} \;\; x_i \sim \phi_{\mathcal{T}(w)} \;, \end{aligned} \tag{3}$$

using Monte Carlo integration, which gives a convenient expression, relatively inexpensive to evaluate. This function $p(w)$ constitutes the objective function that we seek to maximize when optimizing a pose estimate. In practice, it is generally smooth in the neighbourhood of the global optimum, but no assumption can be made about its convexity, and its definition on the 6-dimensional pose space SE(3) makes the evaluation of its gradient difficult. Fortunately, our initial pose estimate can be assumed to be a close approximation of the global optimum. All those conditions motivated the use of a simple hill-climbing algorithm. We iteratively optimize pairs of dimensions at a time, namely the 2 viewpoint angles, the image location, then the scale and in-plane rotation. We empirically observed that a close approximation of the global optimum can be reached in this way after only a few iterations (see Fig. 5, bottom right).

3.2 Results

Rotating car. We first evaluate our method on the first sequence of the "rotating car" dataset [8]. It consists of 118 images of a car on a rotating platform, shot at a motor show. Although it only includes a single degree of freedom (the rotation around the vertical axis), this dataset is interesting as it was shot in real

conditions, features a highly textured object of complex structure and allows a comparison of precision with two state-of-the-art methods. We report in Fig. 2 the precision (error of the estimated rotation angle of the car) of our initial pose estimate and of the pose estimates optimized using our generative model (Section 3.1). We show a clear advantage, with different sizes of training sets (which contain uniformly spaced images from the sequence).

Number of training views	**15**	**30**	**40**
Baseline comparison 1 [17]	5.47	1.93	1.84
Baseline comparison 2 [15]	4.42	1.62	1.49
Initial estimate (nearest neighbour)	6.79	3.78	3.00
Optimized w/ generative model	**1.99**	**0.92**	**0.78**

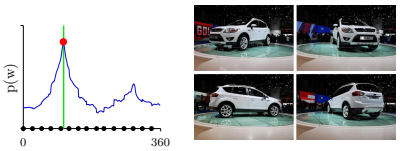

Fig. 2: Results of pose estimation on the rotating car dataset; mean error in degrees. Center: for one test image, we verify that our objective function (blue; evaluated over the whole range of the 1D rotation for demonstration) presents its global optimum near the ground truth (green line). Also represented: training poses (black dots) and our optimized result (red dot). Right: samples test images.

3D pose dataset. We now evaluate our method using the "3D pose dataset" of Viksten *et al.* [4,19]. It is one of the few public datasets available with views spanning more than a 1D rotation, and precisely annotated (in this case with the azimuth/elevation angles of the viewpoint). We use the only object (Volvo car) that was evaluated individually [4], using the same experimental conditions. This allows a comparison with a classical method [4], which uses discriminative image descriptors with a voting and averaging scheme; this constitutes the classical approach for robust 3D pose estimation. The small and large training sets contain views spaced respectively 20° and 10° apart (on both azimuth and elevation angles), with test views in between. With the larger training set, we obtain results superior to [4] in terms of accuracy (Fig. 3). The smaller training set is more challenging for detecting deformations between views, reaching the limits of the optical flow algorithm we use. Our large mean error is caused by two views that yield wrong initial pose estimates, with a large error of almost 180°, due to the similar aspect of the front and rear of the car.

Spacing between train. views	**20°**	**10°**
Baseline comparison [4]	4.21° / 2.66°	1.25° / 1.06°
Initial estimate (near. neigh.)	34.44° / 10.00°	5.14° / 5.00°
	5.00° / 5.00°	1.23° / 1.23°
Optimized w/ gen. model	27.22° / **1.78°**	**0.92°** / **0.88°**
	2.00° / 2.00°	1.00° / 1.00°

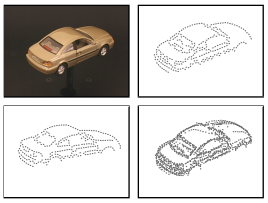

Fig. 3: Results on 3D pose dataset; mean (median) error of azimuth/elevation angles. Clockwise: one test image, its image features, features produced by our generative model at the optimized pose, and features of closest training view; the generative model closely approximates the appearance of the unseen view.

4 Matching pose/appearance relations in ambiguous test scenes

4.1 Method

We propose to make use of the pose/appearance relations identified and stored within the model in a second manner, as extra dimensions of the descriptor of the image features. In this context, the same changes of appearance with respect to the pose are identified in the test view, using additional images obtained by moving the camera slightly around the test scene (which effectively changes the relative pose between the camera and the object of interest). Those additional images are only used to identify the deformations (as in Section 2.2); only the features of the original test image are actually used. Each image feature $x \in \mathcal{A}$ of both the training set and the test image is then complemented with an extra information x^{d}, a first-order approximation of the derivative of its position in the image with respect to the viewpoint, i.e. $x^{\mathsf{d}} \approx \frac{\partial x^{\mathsf{p}}}{\partial w^{\mathsf{v}}}$ ($\in \mathcal{R}^2 \times \mathcal{R}^2$). This conveniently constitutes a compact representation of the local deformations. In practice, considering an image feature x of a training view k, we approximate x^{d} by averaging the deformations identified with the neighbouring views:

$$x^{\mathsf{d}} \;=\; \text{average}_{k' \in \text{nb}(k)} \left(\frac{\text{UV}_{k \to k'}(x^{\mathsf{p}})}{w^{\mathsf{v}}_{k'} - w^{\mathsf{v}}_{k}} \right) . \tag{4}$$

Using azimuth and elevation angles to parametrize a viewpoint on the 2-sphere, this expression gives us two vectors (each $\in \mathcal{R}^2$), corresponding to the translation in the image place relative resp. to azimuth and elevation changes of the viewpoint. These extra feature descriptors are similarly extracted in the test view. We then use them, both when matching observations between the training and test views for voting for an initial pose estimate, and when measuring the similarity between the test view and a generated view (Section 3.1). In both cases, we set a hard threshold for classifying two features x_1 and x_2 as similar[2]:

$$\text{angle}(x^{\mathsf{d}}_1, x^{\mathsf{d}}_2) < 135^\circ \quad \textit{or} \quad \|x^{\mathsf{d}}_1\| < t' \quad \textit{or} \quad \|x^{\mathsf{d}}_2\| < t' \; . \tag{5}$$

The threshold t' on the magnitude of the deformations discards small and insignificant deformations, which cannot be identified reliably. The function $\text{angle}(\cdot, \cdot)$ measures the difference in direction between the two deformations. The maximum value of 135° discards matches of *truly opposite* directions (as is the case with the ambiguous situations we are interested in), while still keeping most (even uncertain) matches (maybe simply due to noise), which is important in the voting algorithm [15] for the initial pose estimate.

4.2 Results

We finally evaluate the second proposed use of image deformations, for resolving ambiguous test cases by matching them. No dataset suitable for this very

[2] To shorten notations, we express the condition on a single vector, but it must be verified by both parts of x^{d}.

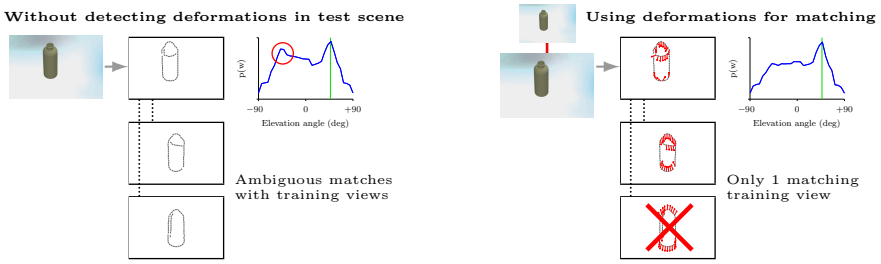

Fig. 4: Left: ambiguous test scene. The extracted edges correspond equally well to training images of the bottle facing towards/away from the camera, and our pose likelihood function $p(w)$ presents two strong peaks (incorrect one in red, ground truth in green). Right: using a second image taken after moving the camera slightly to the top, we detect deformations of image features (red arrows), and match them with the training examples; only the correct peak of $p(w)$ remains.

Objects					
No deform. + near. neigh.	55% 6.3°	80% 5.0°	68% 6.5°	69% 5.2°	51% 16.7°
Match. def. + near. neigh.	66% 6.2°	82% 16.6°	81% 6.2°	79% 5.4°	60% 17.5°
Match. def. + gen. model	70% 5.0°	82% 14.0°	92% 4.6°	77% 1.4°	60% 14.7°

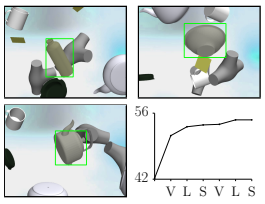

Fig. 5: Left: results on synthetic scenes without/with matching of deformations; success rate (localization error < 20px and pose error $< 20°$) and mean pose error of successes. Right: sample test images with localization results as bounding boxes. Bottom right: typical evolution of our objective function after successive optimizations for viewpoint, image localization and scale/in-plane rotations.

particular problem is currently available, and we resorted to synthetic images, featuring simple objects. Although simple in appearance, they actually prove challenging for pose estimation due to their lack of detail and texture. The test data is now a "central" 2D image, complemented by 2 additional views obtained by moving the camera slightly to the right and to the top. This allows recovering the changes of appearance of the scene with respect to the pose, and matching them with the training data (Fig. 4). As reported in Fig. 5, this eases the localization and improves the precision of the pose estimation.

5 Conclusions

We integrated, within a multiview model of appearance, the explicit transformations undergone by the image features between the training viewpoints. The deformations between example images are detected with dense optical flow and stored for the discrete image features. First, we used this information in a generative model, to refine an initial estimate of the 3D pose of the object in a

new scene. Second, we showed how to match deformations between training and test data, in order to resolve the pose in ambiguous test images. We clearly demonstrated the advantage of those contributions on several datasets, and over existing methods. As future work, it will be interesting to integrate and evaluate these principles within the context and practical conditions of robotic applications.

Acknowledgments The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement no. 270273, Xperience. Damien Teney is supported by a research fellowship of the Belgian National Fund for Scientific Research.

References

1. Avidan, S., Shashua, A.: Novel view synthesis in tensor space. In: CVPR (1997) II
2. Chen, S.E., Williams, L.: View interpolation for image synthesis. In: SIGGRAPH (1993) II
3. Chen, S., Li, Y., Kwok, N.M.: Active vision in robotic systems: A survey of recent developments. Int. J. of Rob. Res. 30(11), 1343–1377 (2011) III
4. Johansson, B., Moe, A.: Patch-duplets for object recognition and pose estimation. In: CRV (2005) VII
5. Kushal, A., Schmid, C., Ponce, J.: Flexible object models for category-level 3D object recognition. In: CVPR (2007) II
6. Liu, C.: Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. Ph.D. thesis, MIT (2009) V
7. Martinez Torres, M., Collet Romea, A., Srinivasa, S.: MOPED: A scalable and low latency object recognition and pose estimation system. In: ICRA (2010) II
8. Ozuysal, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: CVPR (2009) VI
9. Savarese, S., Fei-Fei, L.: View synthesis for recognizing unseen poses of object classes. In: ECCV. Marseille, France (2008) I, III
10. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: IEEE Int. Conf. on Comp. Vis. (2007) II
11. Seitz, S.M., Dyer, C.R.: Toward image-based scene representation using view morphing. In: Int. Conf. on Patt. Rec. pp. 84–89 (1996) II, III
12. Sipe, M.A., Casasent, D.: Best viewpoints for active vision classification and pose estimation. In: Intelligent Robots and Comp. Vis. pp. 382–393 (1997) III
13. Sun, M., Su, H., Savarese, S., Fei-Fei, L.: A multi-view probabilistic model for 3D object classes. In: CVPR (2009) III
14. Teney, D., Piater, J.: Generalized Exemplar-Based Full Pose Estimation from 2D Images without Correspondences. In: DICTA (2012) I, II, III, VI
15. Teney, D., Piater, J.: Continuous pose estimation in 2D images at instance and category levels. Submitted (2013) II, VI, VII, VIII
16. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiel, B., Van Gool, L.: Towards multi-view object class detection. In: CVPR (2006) II
17. Torki, M., Elgammal, A.M.: Regression from local features for viewpoint and pose estimation. In: ICCV (2011) II, VII
18. Vikstén, F., Soderberg, R., Nordberg, K., Perwass, C.: Increasing pose estimation performance using multi-cue integration. In: ICRA (2006) II, III
19. Vikstén, F., Forssén, P.E., Johansson, B., Moe, A.: Comparison of local image descriptors for full 6 degree-of-freedom pose estimation. In: ICRA (2009) VII