Constructing Implicit 3D Shape Models for Pose Estimation

Mica Arie-Nachimson Ronen Basri*
Dept. of Computer Science and Applied Math.
Weizmann Institute of Science
Rehovot 76100, Israel

Abstract

We present a system that constructs "implicit shape models" for classes of rigid 3D objects and utilizes these models to estimating the pose of class instances in single 2D images. We use the framework of implicit shape models to construct a voting procedure that allows for 3D transformations and projection and accounts for self occlusion. The model is comprised of a collection of learned features, their 3D locations, their appearances in different views, and the set of views in which they are visible. We further learn the parameters of a model from training images by applying a method that relies on factorization. We demonstrate the utility of the constructed models by applying them in pose estimation experiments to recover the viewpoint of class instances.

1. Introduction

3D objects may appear very different when seen from different viewing positions. Computer vision systems are expected to consistently recognize objects despite this variability. Constructing representations of 3D objects and their appearances in different views therefore is an important goal of machine vision systems. Recent years have seen tremendous effort, and considerable success, in using statistical models to describe the variability of visual data for detection and classification. This work, however, has focused primarily on building deformable 2D representations of objects, while very few studies attempted to build class models that explicitly account for viewpoint variations. This paper presents an effort to construct a statistical voting procedure for classes of rigid 3D objects and use it to recover the pose of class instances in single 2D images.

Constructing 3D models from 2D training images is challenging because depth information is difficult to infer from single images. Training methods must therefore relate the information in pairs, or larger subsets of images, but this requires a solution to a problem of correspondence, which is aggravated by the different appearances that features may take due to both viewpoint and intra-class variations. However, once correspondence is resolved one may be able to construct models that can generalize simultaneously to novel viewpoints and class instances and allow estimating geometric properties of observed class instances such as viewpoint and depth.

Here we present a system that constructs 3D models of classes of rigid objects and utilizes these models to estimating the pose of class instances. Our formulation relies on the "implicit shape models" [13], a detection scheme based on weighted voting, which we modify to allow for 3D transformations and projection and to account for self occlusion. The class models we construct consist of a collection of learned features, their 3D locations, their appearances in different views, and the set of views in which they are visible. We further learn the parameters of these models from training images by applying a method that relies on factorization [23]. Note that unlike most prior work, which used factorization to reconstruct the shape of specific objects, here we use factorization to construct voting models for classes of rigid objects whose shapes can vary by limited deformation (see however extensions of factorization to more general linear models in, e.g., [18, 17]). Finally, we demonstrate the utility of the constructed models by applying them in pose estimation experiments.

Most recognition work that deals with class variability either ignores variability due to viewpoint altogether (e.g. [6, 13]) or builds a separate independent model for a collection of distinct views [3, 5, 16]. Several recent methods construct "multiview models," in which models of nearby views are related either by sharing features or by admitting some geometric constraints (e.g., epipolar constraints or homographies) [12, 19, 20, 21, 22, 24]. 3D class models are constructed in [14] by training using synthetic 3D models and their synthesized projections. [9, 26] introduce multiview models in which features from different

*Research was conducted in part while RB was at TTI-C. At the Weizmann Inst. research was supported in part by the Israel Science Foundation grant number 628/08 and conducted at the Moross Laboratory for Vision and Motor Control.

views are associated with a 3D volumetric model of class instances.

Our paper is divided as follows. Our implicit 3D shape model is defined in Section 2. An approach to constructing the model is described in Section 3. A procedure for estimating the pose of class instances is outlined in Section 4. Finally, experimental results are provided in Section 5.

2. Implicit 3D Shape Model

Implicit shape models were proposed in [13] as a method for identifying the likely locations of class instances in images. Object detection is achieved in this method by applying a weighted voting procedure, with weights determined by assessing the quality of matching image to model features. The method was designed originally to handle 2D patterns whose position in the image is unknown. Here we modify this procedure to allow handling projections of 3D objects taking into account visibility due to self occlusion.

We define our implicit 3D shape model as follows. We represent a class as a set of features embedded in 3D space, where each feature point represents a "semantic part" of an object. Then, given a 2D image we consider transformations that map the 3D feature points to the image plane and seek the transformation that best aligns the model features with the image features.

Specifically, we assume we are given a codebook of J model (class) features, denoted $F_1, ..., F_J$, where every feature F_j represents such a semantic element. Features are associated with some 3D location L_j and an appearance descriptor E_j. Given an image I, we obtain a set of image features, denoted $f_1, ...f_N$, with f_i associated with a 2D location l_i and an appearance descriptor e_i. We further define the discrete random variables F and f to respectively take values in $\{F_1, ..., F_J\}$ and $\{f_1, ...f_N\}$ and use the notation $P(F_j, f_i)$ to denote $P(F = F_j, f = f_i)$. Let $\mathcal{T}$ denote a class of transformations that map 3D locations to the image plane, and let $T \in \mathcal{T}$ denote a transformation in this class. We seek to maximize the score $P(T|I)$ over all $\mathcal{T}$. This score can be written as follows.

$$P(T|I) = \sum_i \sum_j P(F_j, f_i, T|I).$$

As the dependence on I is implicit in the selection of the image features $f_1, ..., f_N$ we henceforth simplify notation by dropping this dependence. We further assume that the joint probability $P(F_j, f_i, T)$ can be written as a product

$$P(F_j, f_i, T) = \frac{P(L_j, l_i, T)P(E_j, e_i, T)}{P(T)},$$

and use a uniform prior for $T \in \mathcal{T}$. The first term in the numerator, $P(L_j, l_i, T)$, models the effect of feature location and is defined as a function of the displacement $l_i - T(L_j)$. Specifically, we currently use a uniform distribution over a square window around l_i (similar to the choice made in [13]). The second term in this product, $P(E_j, e_i, T)$, models the effect of appearance, and will be determined by the quality of the match between e_i and E_j. We further write this term as

$$P(E_j, e_i, T) = P(E_j, T|e_i)P(e_i),$$

and assume a uniform distribution for $P(e_i)$. To define the term $P(E_j, T|e_i)$ we use the score function $\mathrm{match}(e_i, E_j)$ (defined below), which evaluates the quality of a match between appearance descriptors. We set this term to

$$P(E_j, T|e_i) \propto \begin{cases} \frac{\mathrm{match}(e_i, E_j)}{\sum_{j \in \mathcal{V}(T)} \mathrm{match}(e_i, E_j)} & j \in \mathcal{V}(T) \\ 0 & j \notin \mathcal{V}(T), \end{cases}$$

where $\mathcal{V}(T)$ denotes the set of model features that are visible under T.

Our model has the following parameters:

Appearance: the appearance of an image feature e_i is represented by a SIFT descriptor [15] computed over a 16×16 pixel patch centered at l_i. A model feature F_j is associated with a set of 2D appearances, $E_j = \{e_{j,1}, e_{j,2}, ...\}$, representing the appearances of the feature in different viewpoints and in different class instances. We compare a model feature E_j with an image feature e_i by the measure $\mathrm{match}(e_i, E_j) \stackrel{def}{=} \max_k d(e_i, e_{j,k})$ over all appearances associated with E_j, where $d(e_i, e_j) = \exp(-||e_i - e_j||_2^2)$.

Location: a model feature is associated with a 3D location $L_j = (X_j, Y_j, Z_j)^T$, and an image feature is associated with a 2D location $l_i = (x_i, y_i)^T$.

Transformation: we allow for a 3D similarity transformation followed by an orthographic projection. A similarity transformation has 6 degrees of freedom and is defined by a triplet $< s, R, \mathbf{t} >$, where s denotes scale, R is a 2×3 matrix containing the two rows of a rotation matrix with $RR^T = I$, and $\mathbf{t}$ is a 2D translation. We limit the allowed scale changes to the range $[0.4, 2.5]$.

Visibility: we allow each feature to be visible in a restricted subset of the viewing sphere. We model this subset by a disk (as in [2]), defined by the intersection of a (possibly non-central) half space and the viewing sphere. Such a choice of visibility region is reasonable when the object is roughly convex and the appearance of features changes gradually with viewpoint.

The following section describes how we construct this model from training images.

3. Model Construction

Constructing a 3D class model from a training set of images involves computing statistics of feature locations in a 3D object centered coordinate frame. As depth information is not directly available in single images we need a method that can integrate information from multiple images. Moreover, the recovery of depth values requires accurate knowledge of the pose of the object in the training images, and it is arguably desirable not to require such detailed information to be provided with the training images. The task of constructing a 3D class model is therefore analogous to the problem of shape reconstruction in which the 3D structure of a rigid object is determined from an uncalibrated collection of its views. Below we follow up on this analogy and use a procedure based on Tomasi and Kanade's factorization method [23] (abbreviated below to *TK-factorization*) to construct a 3D class model. For completeness we briefly review this procedure below.

Given a set of images taken from different viewpoints around an object, along with the 2D locations of corresponding points across these images, the TK-factorization recovers the camera motion and the original 3D locations of the points. We begin by constructing a $2f \times p$ matrix M, which we call the *measurement matrix*. M includes the x and y coordinates of the p feature points in the f images organized such that all corresponding locations of a feature form a column in M. The coordinates are first centered around a common origin, e.g., by subtracting the mean location in each image. We then express M as a product $M \approx RS$ where the $3 \times p$ matrix S, called the *shape* matrix, includes the recovered 3D positions of the feature points, and the $2f \times 3$ matrix R, called the *transformation* matrix, includes the transformations that map these 3D positions to their projected locations in each of the images. We set those matrices to $R = U_3\sqrt{\Delta_3}A$ and $S = A^{-1}\sqrt{\Delta_3}V_3^T$, where $M = U\Delta V^T$ is the Singular Value Decomposition of M, Δ_3 is a 3×3 diagonal matrix containing the largest three singular values of M, U_3 and V_3 include the three left (respectively right) dominant singular vectors of M, and A is an invertible 3×3 ambiguity matrix. The components of A are determined by exploiting the expected orthogonal structure in R. Specifically, let $\mathbf{r}_x$ and $\mathbf{r}_y$ denote a pair of rows in $U_3\sqrt{\Delta_3}$ that correspond to a single image. We seek A that solves

$$\begin{pmatrix} \mathbf{r}_x \\ \mathbf{r}_y \end{pmatrix} AA^T \begin{pmatrix} \mathbf{r}_x \\ \mathbf{r}_y \end{pmatrix}^T = sI_2$$

simultaneously for all images, where I_2 representing the 2×2 identity matrix, and $s > 0$ is an unknown scaling factor.

There are several difficulties in applying the TK-factorization to our problem. Perhaps the most important problem concerns the strict rank assumption made by the TK-factorization, which is appropriate when the images include orthographic views of a single, rigid object, but may break in the presence of significant intra-class variations. This concern limits the applicability of the TK-factorization to classes of fairly tight shapes. Nevertheless, we demonstrate in our experiments that the TK-factorization can indeed be used to construct 3D models for common classes of objects sustaining reasonable shape variations. Applying the TK-factorization method is difficult also because it requires a solution to a correspondence problem, which is generally hard, particularly when the objects appearing in the training images differ by both viewpoint and shape. Finally, the TK-factorization must deal with significant amounts of missing data, due to self occlusion. Below we introduce an attempt to overcome these difficulties.

In the sequel we assume we are given a training set that includes segmented images of class objects seen from different viewpoints. The training set is comprised of two subsets. The first subset, which we call the *initial set*, includes images of a single class instance (i.e., a specific car model) seen from different viewpoints. The initial set will be used to construct a 3D model for this specific class instance that will serve as an initial model for the class. The remaining set includes images of other class instances seen from different viewpoints, where in particular each pair of images may differ simultaneously in both instance and viewpoint. Those images will be used to refine the initial model.

3.1. Model initialization

Our first goal at this stage is to use the initial set to construct a 3D model for a specific class instance. We assume we can roughly center the images of the initial set around the center of their segmented region. Then, to initialize the reconstruction process we select a small collection of feature points and manually mark their locations in all the images in the initial set. This produces a measurement matrix M_I of size $2f_I \times p_I$, where f_I denotes the number of images in the initial set and p_I the number of selected feature points. M_I includes the x and y coordinates of the marked points, along with many missing entries due to self occlusion.

To recover the 3D position of the marked points we proceed by recovering a rank 3 approximation to M_I using [25] (initialized with the algorithm of [11]), which accounts for the missing entries. We seek a rank 3 approximation in this case since the images are roughly centered. We then use the TK-factorization [23] to recover the corresponding transformation and shape matrices R_I and S_I. Finally, we extend the model by adding many more points from each image of the initial set. For each image we apply the Harris corner detector [8] and associate with the detected points a depth value by interpolating the depth values of the manually marked points. By using interpolation we avoid the

need to resolve correspondences between features in different images. This step provides us with a rich (albeit approximate) 3D model of a specific class instance.

3.2. Constructing a 3D class model

Given the initial model we turn to constructing a 3D class model. Our aim at this stage is to extend the measurement matrix M_I by appending additional rows for all the remaining training images. This we can do if we solve a correspondence problem between each of the remaining training images and (at least one) of the initial images.

We match a training image to the initial images as follows. Given a training image I_t we seek the most similar image in the initial set, denoted $I_{I(t)}$, using the method of implicit 2D shape models [13]. For every image I_i of the initial set we produce a voting map V_i as follows. We begin by overlaying a grid with an inter-distance of 4 pixels and use every grid point as a feature. We then compare every pair of features $(e_t, l_t) \in I_t$ and $(e_i, l_i) \in I_i$ and vote for a center location at $l_t - l_i$ with weight proportional to $|e_t - e_i|$. The sought image $I_{I(t)}$ is then selected to be the image of the initial set that gave rise to the highest peak.

Next we assign correspondences to the pair I_t and $I_{I(t)}$. We consider the location of the highest peak in $V_{I(t)}$ and identify pairs of features that voted for that location. To obtain additional correspondences we repeat this process by adding the correspondences found between I_t and the views nearest to $I_{I(t)}$ (we used four nearest views in our experiments).

Following this process we obtain a new measurement matrix $\tilde{M}$ that includes the locations of feature points in the initial set (M_I) as well as new correspondences extracted from the training images. However, $\tilde{M}$ may still contain significant amounts of missing data due to occlusions and lack of sufficient matches. Solving for the missing data can be difficult in this case since $\tilde{M}$ may deviate from low rank due to intra-class variations. We approach this problem by initializing the search for a low rank approximation of [25] by filling in the missing values in $\tilde{M}$ with corresponding values from $R_I S_I$. Once a low rank approximation, denoted M, is found, we proceed by applying the TK-factorization algorithm to M, recovering a transformation matrix R and a shape matrix S. Note that this step of factorization is meant to recover the 3D information of the class. Finally, we use the recovered shape matrix to fill in the location parameters L_j of the model.

3.3. Determining visibility

The transformation matrix R produced by the TK-factorization for the entire training data can be used to infer the viewpoint from which each object is viewed in each image. This information, along with the missing entries in the measurement matrix, can be used to determine the visibility model.

To determine the viewpoint of each training image from R we first note that although the TK-factorization uses the anticipated orthogonal structure of R to remove ambiguities, it does not *enforce* such a structure, and so the obtained rows may not be orthogonal as desired. Therefore, to associate a similarity transformation with each training image we apply an orthogonalization process. Given two rows of R, denoted $\mathbf{r}_x$ and $\mathbf{r}_y$, that correspond to a single training image we seek the nearest pair of vectors $\tilde{\mathbf{r}}_x$ and $\tilde{\mathbf{r}}_y$ that are both orthogonal and have equal norms, i.e.,

$$\min_{\tilde{\mathbf{r}}_x, \tilde{\mathbf{r}}_y} \|\mathbf{r}_x - \tilde{\mathbf{r}}_x\|^2 + \|\mathbf{r}_y - \tilde{\mathbf{r}}_y\|^2$$

such that

$$\|\tilde{\mathbf{r}}_x\| = \|\tilde{\mathbf{r}}_y\| \text{ and } \tilde{\mathbf{r}}_x^T \tilde{\mathbf{r}}_y = 0.$$

Such a pair is expressed in closed form [1] by

$$\begin{pmatrix} \tilde{\mathbf{r}}_x \\ \tilde{\mathbf{r}}_y \end{pmatrix}^T = \frac{1}{2\Delta} \begin{pmatrix} 2\Delta + \|\mathbf{r}_x\|^2 & -\mathbf{r}_x^T \mathbf{r}_y \\ -\mathbf{r}_x^T \mathbf{r}_y & 2\Delta + \|\mathbf{r}_y\|^2 \end{pmatrix} \begin{pmatrix} \mathbf{r}_x \\ \mathbf{r}_y \end{pmatrix}^T,$$

with $\Delta = \sqrt{\|\mathbf{r}_x\|^2 \|\mathbf{r}_y\|^2 - (\mathbf{r}_x^T \mathbf{r}_y)^2}$. The pair of vectors $\tilde{\mathbf{r}}_x$ and $\tilde{\mathbf{r}}_y$ obtained represent the first two rows of a scaled rotation in 3D. The viewpoint corresponding to this rotation is given by

$$\mathbf{v} = \frac{\tilde{\mathbf{r}}_x}{\|\tilde{\mathbf{r}}_x\|} \times \frac{\tilde{\mathbf{r}}_y}{\|\tilde{\mathbf{r}}_y\|}.$$

Finally, to determine the visibility model, for each feature (column of M) we consider all its non-missing entries, corresponding to the training images in which it was observed. We then determine the visibility region by selecting the half space whose intersection with the viewing sphere creates the minimal region that includes all the viewpoints in which the feature was observed from. For a feature (E_j, L_j) let $\mathbf{v}_1, \mathbf{v}_2, ...$ denote the set of viewpoints from which it was observed. Denote by $\bar{\mathbf{v}}$ the viewpoint in the direction of the mean of those viewpoints. Then we set the visible region to include all the viewpoints $\mathbf{v}$ in the set $\{\mathbf{v} | \bar{\mathbf{v}}^T \mathbf{v} \geq \min_k \bar{\mathbf{v}}^T \mathbf{v}_k - \epsilon\}$ for some constant $\epsilon > 0$.

4. Pose estimation

Given a test image we can use the model to estimate the pose of an object by evaluating $P(T|I)$. Computing $P(T|I)$, however, can be demanding since T is a 3D similarity transformation, and so this probability distribution is defined over a 6-dimensional parameter domain. Rather than discretizing this domain we chose to evaluate the probability in regions of this domain suggested by the data. This we do by applying a RANSAC procedure [7, 10] as follows.

Given a test image I we first overlay a grid over the image with distance of 4 pixels between grid points. Using

Figure 1. Example images from the initial set.

Figure 2. Example images from the remaining training set.

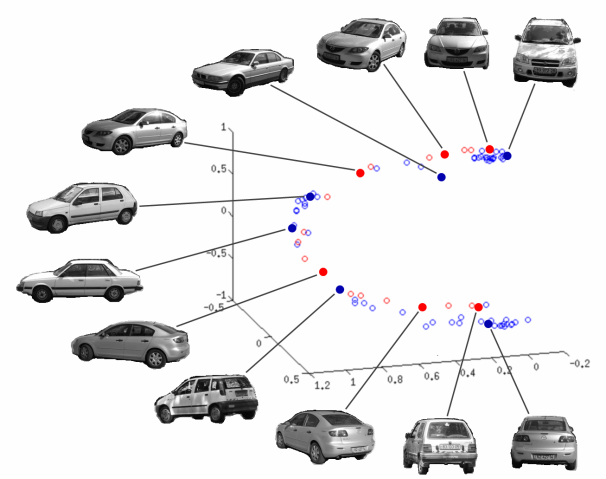

Figure 3. Recovery of camera positions for the training set. Camera positions recovered for the initial set are marked in red, and positions recovered for the remaining training images are marked in blue.

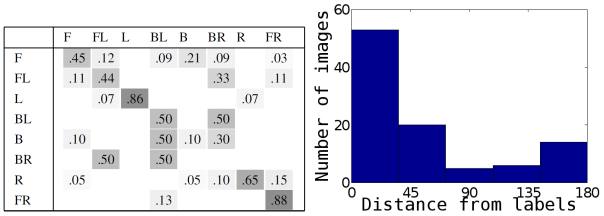

Figure 4. Pose estimation results for the car dataset of [19]. For the eight views labeled in the dataset (frontal, left, back, right, and intermediate views) we show a view confusion matrix (left) and a histogram of view error (right).

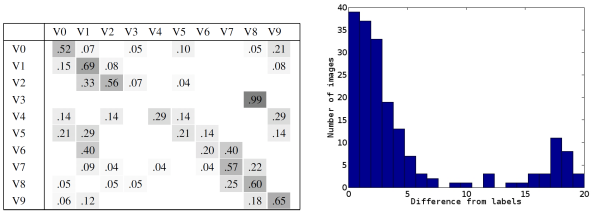

Figure 5. Pose estimation results for the PASCAL VOC 2007 car dataset [4]. Here we compare our pose estimations with a manual labeling relating each of the cars in the dataset to its nearest view among the 21 images of the initial set and their mirror (left/right) reflections. Left: a confusion matrix. For the sake of presentation each 4 images are placed in a single bin, obtaining an average bin size of 36°. Right: a histogram of view errors. The peak around zero indicates that most images were matched either to or near their most similar pose. The small peak near 20 signifies the common 180° confusion.

every grid point as a feature (e_i, l_i) we compare its appearance (normalized SIFT descriptor) to each of the model features. We then select the best matching model feature, i.e., $\max_j \text{match}(e_i, E_j)$. Next, we go through the list of best matches and enumerate triplets of k best matching pairs. For each such triplet we compute a 3D-to-2D similarity transformation and evaluate $P(T|I)$.

5. Experiments

We used our method to construct an implicit shape model for sedan cars. Our training set consisted of 86 segmented images of cars. 21 of the 86 training images, which included images of the same car (Mazda3 model) seen from different viewpoints, were used as the initial set. The remaining 65 images of the training set included images of different car models seen in different viewpoints. Very few of the remaining images were of the same car model. Some of the training images (available in [27]) can be seen in Figures 1 and 2. We initialized the training by selecting 18 identifiable features and marking their location manually in each of the 21 images of the initial set. We then applied the procedure described in Section 3 constructing first a 3D model for the Mazda3 car, and then extending it by constructing a model for the full training set. Following this procedure we obtained a model with about 1800 features covering different viewpoints and class instances, each associated with a 3D location and a collection of appearances. We further recovered a viewpoint for each training image and used those viewpoints to construct a visibility region for each feature. Fig. 3 shows the viewpoints recovered for the training images. It can be seen that the viewpoints are roughly coplanar, spanning a range of 180° from frontal to rear views.

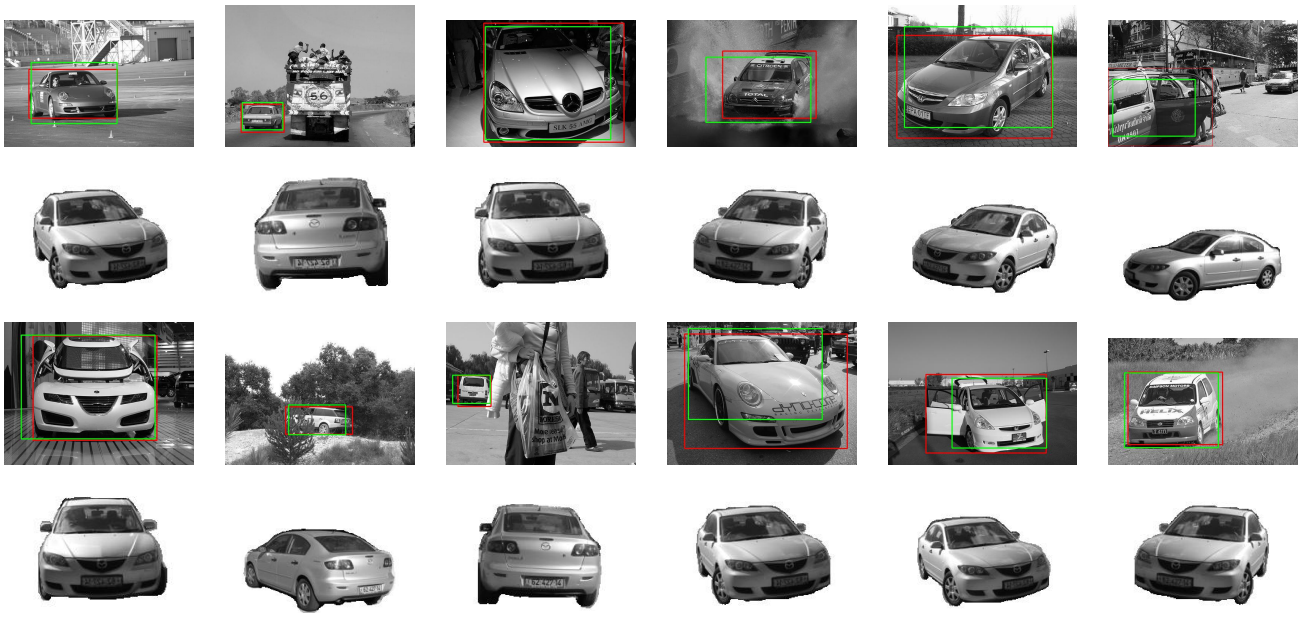

Figure 6. Pose estimation examples. The figure shows green bounding boxes around cars identified by our method. Ground truth bounding boxes are shown in red. In addition, below each image we show the car from the initial set whose view was found closest by our method.

Figure 7. Pose estimation errors. The figure shows green bounding boxes around cars identified by our method. Ground truth bounding boxes are shown in red. Below each image we show the car from the initial set whose view was considered closest by our method.

We used our model to estimate the pose of cars in the dataset of 3D object categories of [19] and the PASCAL VOC 2007 car dataset [4]. For the first dataset only we enriched the model by adding the images of 5 out of the 10 cars in the dataset to the training set. Then, for each image of the remaining 5 cars, we used our model to detect the single most likely transformation according to our voting scheme. We first produced four scalings for each test image I, by factors of 0.5, $1/\sqrt{2}$, 1, and $\sqrt{2}$, and flipped the image about the vertical axis to account for right/left reversals, producing overall 8 copies of I. For each of these 8 copies we overlaid a grid of distance 4 pixels between grid points and produced for each grid point a SIFT descriptor [15]. We further normalized each descriptor by dividing its entries by their sum (adding $\epsilon = 0.2$ to avoid noisy responses in uniform regions).

We next produced a collection of correspondences of triplets of points, as we describe in Section 4, with $k = 100$, and recovered for each triplet a 3D-to-2D similarity transformation. We excluded near singular transformations, transformations with scale outside the range $[0.4, 2.5]$, and transformations for which either one of the basis pair or more than 80% of the model features were not visible. For each of the remaining transformations we then evaluated $P(T|I)$ and selected the single transformation that gave the highest score over all four scalings. Finally, we placed a bounding box by fitting the smallest box that included more than 95% of the entire transformed and projected 3D model points. Of the 160 test images (5 objects, 16 viewpoints, 2 scalings) our method detected 98 cars (61.25%) (compared to classification rates between 45-75% reported in [19, 20]). Fig. 4 shows a histogram of view errors and confusion matrix relative to ground truth labeling of 8 different car directions. Most of the detected cars were classified correctly or fell into the neighboring bin. A few additional mistakes resulted in 180° errors (e.g., front/back confusion or right/left

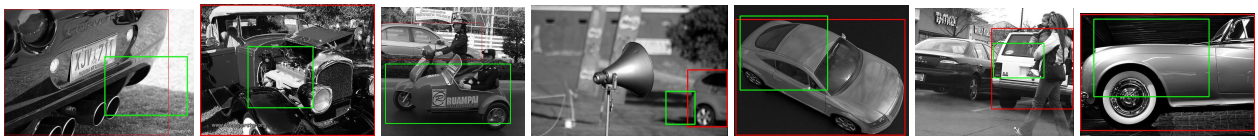

Figure 8. Detection errors. Our detection in green vs. ground truth in red.

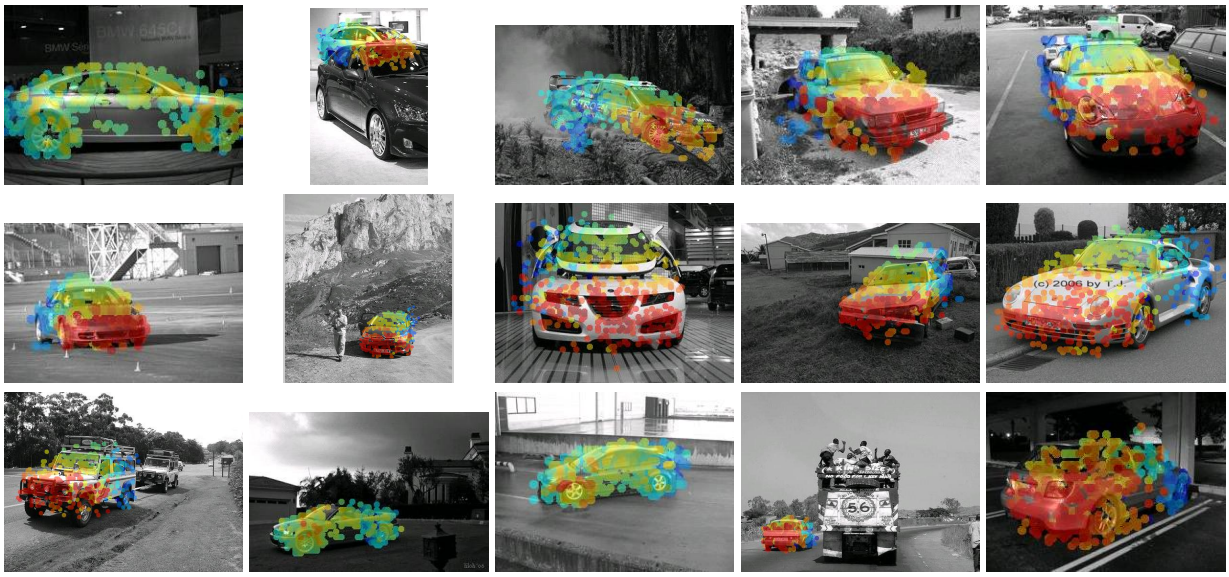

Figure 9. Depth values of features predicted by the model. Color values vary from red (closer to the camera) to blue (further from the camera).

Figure 10. Part labels propagated from the model based on the voting.

reversal).

We next used our model (trained only with the original 86 images) to estimate the pose of cars in the PASCAL 2007 VOC car database. We used the car test set presented for the detection challenge, excluding test images in which there was no car of more than 7000 pixels in size, obtaining 490 images. We used the same procedure to detect the single most likely transformation in each image according to our voting scheme. Overall our method found 188 cars in the 490 images (38.3%) with average precision rate of 0.261. We then evaluated the accuracy of our pose estimation by automatically selecting, for each of the detected cars, the image from the initial set with pose closest to the one found by our method. Quantitative results are shown in Fig. 5 and some examples of the results obtained with our pose estimation procedure are shown in Figures 6-8. For the quantitative assessment we manually labelled each of the test images with the image of the initial set that appear visually to be pictured from the nearest viewing angle. This data is available in [27]. As our initial set consists of 21 images spanning 180 degrees, this labelling represents an average accuracy of 9°. We then compared our pose esti-

mation results with the manual labellings. The results are shown in the histogram and confusion matrix in Fig. 5. As can be seen, most car images were associated with one of the 3 nearest viewing angles, with most errors occuring by 180° rotation (e.g., confusing front and rear views of cars or left/right reversals). Note that achieving highly accurate pose estimations can be difficult since viewpoint variations can often be traded with intra-class variations.

To further demonstrate the utility of our 3D model we estimated the depth values of image features predicted by our 3D model. These depth values can be seen in Fig. 9. Finally, we manually labeled features in the 3D model that correspond to semantic parts of cars, e.g., the wheels, windows, headlights, mirrors, etc. Then, for the test images we set the 2D location of these labels based on the votings. Fig. 10 shows the location of parts identified using the projected models. It can be seen that both the depth values and part labels are in fairly good agreement with the image.

6. Conclusion

Computer vision systems are expected to handle 3D objects whose appearance can vary both due to viewing direction and to intra-class variations. In this paper we have presented a system for pose estimation of class instances. Relying on the framework of implicit shape models, we have introduced a voting procedure that allows for 3D transformations and projection and accounts for self occlusion. We have further used factorization to construct a model from training images. Our results indicate that, despite significant intra-class variations, our voting procedure in many cases is capable of recovering the pose of objects to a reasonable accuracy. Our future objectives include constructing generative class models of rigid 3D objects and enhancing these models to allow articulations or deformations.

Acknowledgment: We thank Gregory Shakhnarovich, Nathan Srebro, and David Jacobs for useful discussions.

References

[1] R. Basri, D. Weinshall, Distance metric between 3D models and 2D images for recognition and classification, *IEEE TPAMI*, **18**(4), 1996.

[2] R. Basri, P. Felzenszwalb, R. Girshick, D. Jacobs, C. Klivans, Visibility constraints on features of 3D objects, *CVPR*, 2009.

[3] O. Chum, A. Zisserman, An exemplar model for learning object classes, *CVPR*, 2007.

[4] M. Everingham, L. Van Gool, CKI. Williams, J. Winn, A. Zisserman, *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,* http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

[5] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Discriminatively trained mixtures of deformable part models, *PASCAL VOC Challenge*, 2008.

[6] R. Fergus, P. Perona, A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR*, 2003.

[7] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Com. of the ACM*, **24**(6), 1981.

[8] C. Harris, M. Stephens, A combined corner and edge detector. *Alvey Vision Conference*, 1988.

[9] D. Hoeim, C. Rother, J. Winn, 3D layoutcrf for multi-view object class recognition and segmentation, *CVPR*, 2007.

[10] D.P. Huttenlocher, S. Ullman, Recognizing solid objects by alignment with an image, *IJCV*, **5**(2), 1990.

[11] D. Jacobs, Linear fitting with missing data for structure-from-motion, *CVIU*, **82**, 2001.

[12] A. Kushal, C. Schmid, J. Ponce, Flexible object models for category-level 3d object recognition, *CVPR*, 2007.

[13] B. Leibe, B. Schiele, Combined object categorization and segmentation with an implicit shape model, *Workshop on Statistical Learning in Computer Vision, Prague*, 2004.

[14] J. Liebelt, C. Schmid, K. Schertler, Viewpoint-independent object class detection using 3D feature maps, *CVPR*, 2008.

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, **60**(2), 2004.

[16] M. Ozuysal, V. Lepetit, P. Fua, Pose Estimation for Category Specific Multiview Object Localization, *CVPR*, 2009.

[17] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, L. Agapito, Factorization for Non-Rigid and Articulated Structure using Metric Projections, *CVPR*, 2009.

[18] V. Rabaud, S. Belongie, Linear Embeddings in Non-Rigid Structure from Motion, *CVPR*, 2009.

[19] S. Savarese, L. Fei-Fei, 3D generic object categorization, localization and pose estimation, *ICCV*, 2007.

[20] S. Savarese, L. Fei-Fei, View synthesis for recognizing unseen poses of object classes, *ECCV*, 2008.

[21] M. Sun, H. Su, S. Savarese, L. Fei-Fei, A Multi-View Probabilistic Model for 3D Object Classes, *CVPR*, 2009.

[22] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, L. Van Gool, Towards multi-view object class detection, *CVPR*, 2006.

[23] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization method, *IJCV* **9**(2), 1992.

[24] A. Torralba, K. Murphy, and W. Freeman. Sharing features: efficient boosting procedures for multiclass object detection, *CVPR*, 2004.

[25] T. Wiberg, Computation of principal components when data are missing, Proc. Second Symp. Computational Statistics, 1976.

[26] P. Yan, D. Khan, M. Shah, 3d model based object class detection in an arbitrary view, *ICCV*, 2007.

[27] http://www.wisdom.weizmann.ac.il/~vision/ism3D/