Improved Object Detection and Pose Using Part-Based Models

Fangyuan Jiang, Olof Enqvist, Fredrik Kahl, and Kalle Åström

Centre for Mathematical Sciences, Lund University

Abstract. Automated object detection is perhaps the most central task of computer vision and arguably the most difficult one. This paper extends previous work on part-based models by using accurate geometric models both in the learning phase and at detection. In the learning phase manual annotations are used to reduce perspective distortion before learning the part-based models. That training is performed on rectified images, leads to models which are more specific, reducing the risk of false positives. At the same time a set of representative object poses are learnt. These are used at detection to remove perspective distortion. The method is evaluated on the bus category of the Pascal dataset with promising results.

1 Introduction

Object detection or localization aims at predicting a 2D bounding box with a category label in the image. It is a difficult task not only because the intra-class variation in shape or texture could lead to different appearance of object instances of a certain category, but the changes in lights, viewpoints as well as non-rigid deformation could also account for the difficulty of the task. The dominant approach to object detection is to perform a sliding-window search [1,2,3]. Essentially, this means considering all possible bounding boxes in the image, and much work has focused on reducing the computational load [4,5].

A type of methods that has received lot of attention are the part-based models, e.g. the early work by Fischler and Elschlager [6]. Central for this paper is the work on deformable part-based models (DPM) of Felzenszwalb et al. [3] using a root filter to encode the global appearance, a collection of part filters to capture the local appearance, and a deformation model for the spatial distribution of parts. Zhu et al. [7] reformulate the DPM as a structural SVM and use a hierarchy of parts at different granularities. Vedaldi and Zisserman [8] also propose a structured output model for object detection which implicitly models parts by accounting for alignment of the features representing an object class.

To handle different viewpoints, Felzenszwalb et al. [3] cluster bounding boxes based on their aspect ratios and train separate part-based models for each cluster. This introduces a trade-off: If the number of different models is to small they will not be specific enough and if they are too many a very large training set is required. One way to work around this that also increases efficiency is to allow different models to share parts [9,10].

J.-K. Kämäräinen and M. Koskela (Eds.): SCIA 2013, LNCS 7944, pp. 396–407, 2013.

Other work explicitly incorporates geometric information in the detection or even infer the geometric properties of the objects [11,12,13,14,15,16], e.g. the 3D pose of the object. In order to do so, these methods normally require more detailed manual annotations, both the computational properties of training and the quality of the models are improved. This paper follows along these lines of and use richer annotations of the training dataset. A way to reduce the time required for annotations is to use interactive learning as proposed by [17]. As the detector improves, only the difficult examples require manual annotation.

Similar to [12,18] we will model an object as a number of approximately planar aspects. For each aspect a DPM is built with the method from [3] and the scores from the different aspect detectors are combined to produce the final object detection.

Since the individual aspects are approximately planar and object pose can be computed from the annotations, we can rectify the training images before learning the aspect models. This allows us to learn highly discriminative models in spite of varying viewpoints. At detection, we hypothesize different object poses from a learnt set of *typical poses*, transform the image according to the hypothetical pose and run the detector. In effect, we get not only an accurate bounding box but also roughly the pose of the object. This is appealing as it extracts more information from the 2D images, and enables a richer understanding of the object in its context. For example, a simple bounding box cannot tell in which direction a car is facing.

Modeling an object with a set of approximately planar aspects works well for a large number of categories, see [12] for examples. In the experiments, we work with the bus category from the Pascal VOC challenge.

Overview of the paper. The rest of this paper is organized as follows. Section 2 describes how we model our objects both in terms of appearance and geometry. This includes showing how to compensate for varying viewpoint in training the individual aspect models and a discussion on how to select a set of typical object poses. Section 3 is concerned with the detection pipeline and Section 4, contains quantitative and qualitative experimental results on the bus category of the Pascal dataset. Finally, Section 5 contains a concluding discussion.

2 Modeling Appearance and Geometry

Our object model is defined as a number of roughly planar aspects models together with a set of typical object poses. We will assume that we have annotated training images and a method to estimate the object pose from annotations. Hence we train each aspect model from rectified image patches using a deformable part-based model [3]. One could argue that a deformable model is not quite suited for a rigid object like a bus, but clearly it is still desirable since the position of the parts might vary between different instances of an object.

The fact that aspect models are learnt from rectified image patches, introduce a problem at detection, when the ground-truth object pose is unknown. To resolve this we learn a small set of *typical object poses* from the annotated training

set. At detection we transform the image according to each of the learnt typical poses. If the training set is large enough it will contain most important object poses.

Scores from the aspect detectors are generated by running each aspect model on each of the transformed images. Detections from the different aspect models are combined and thresholded to produce the final object detection. More specifically, once we run the aspect model on the transformed image, a multi-scale score pyramid is generated. Each location in the pyramid defines a score and bounding box indicating the confidence of the object defined by the bounding box occurring at that location. To combine the detection of frontal aspect with the right aspect, we need to find in the side score pyramid the expected position of side given the location of frontal aspect. The size of the frontal basically gives us clues about where to find the side. Also, we enforce the consistency constraints that the edges which both sides share should have the same height.

2.1 Estimating Object Pose

Pose estimation is a crucial building block in training aspect models and learning representative poses. It is not possible to infer the object pose from training images labeled only with a rectangular bounding boxes. Extra information is necessary either from some pre-built model, e.g. a CAD model, or from the manual annotation. Here we manually annotate each visible aspect of the object in training images, as shown in Figure 1.

Fig. 1. Annotation of each visible aspect of the object in training images

The annotations give us a set of known points on the object. The next task is to estimate the pose of the camera relative to the object. We will assume that all internal camera parameters have standard values, except the focal length which we estimate. More precisely, we assume that the principal point lies at the center of the image, that the aspect ratio is 1 and that skew is 0.

If a 3D model of the object is known, four points is sufficient to estimate camera pose and focal length [19,20], but for the case of block-shaped objects,

e.g. buses we use a more specialized approach, which does not require an explicit 3D model. The goal is to estimate a camera matrix $P = (R \,|\, t)$ and the camera focal length f.

Normally, the upper and lower edge of of the bus side are parallel in the 3D world. Hence the corresponding lines in the image intersect at a vanishing point, being the projection of a point at infinity. Let x be a vanishing point in the image and $(X, 0)^T$ the homogeneous coordinates of the corresponding infinity point. Then,

$$\lambda \begin{pmatrix} x \\ f \end{pmatrix} = (R \,|\, t) \begin{pmatrix} X \\ 0 \end{pmatrix} = RX. \tag{1}$$

The same computations for the front (or rear) side of the bus yields

$$\lambda \begin{pmatrix} y \\ f \end{pmatrix} = RY. \tag{2}$$

We note that X and Y represent the front and sideways directions of the bus. This means that they must be perpendicular,

$$x^T y + f^2 \propto X^T R^T R Y = X^T Y = 0 \tag{3}$$

which allows us to estimate the focal length. Knowing the focal length, (1) and (2) also allows us to estimate the camera rotation relative to the orientation of the bus. Finally, we place the origin in the front right corner of the bus and compute the camera translation from its corresponding image projection.

2.2 Training Aspect Models

Rectifying the training images. Detectors trained on the objects under varying viewpoint will tend to be less specific and can lead to high false positive rates. With the method described in the last section, we can estimate the pose of an object and use this to transform the training images such that each visible aspect is rectified. However, at detection we cannot estimate the object pose very accurately, so a model trained on perfectly rectified image patches might not be flexible enough. Hence we add a small perturbation to the exact pose when rectifying the training images.

Assume the pose we estimated for object O is P, where $P = (R, t)$. We generate a small perturbation by picking a rotation angle from a uniform distribution on $[0, 5°]$. Now let $R_x(\theta)$ be a rotation about the x-axis, with rotation angle θ and let S be a random rotation picked from the uniform distribution over all rotations. Then the desired perturbed rotation is $R_p = R_r^T R_x(\theta) R_r R$. We also add a small perturbation to the ground truth translation, $t_p = t + n$, where n is multivariate normal with standard deviation 0.005.

The next step is to transform the image such that one of the planar aspects is rectified. Let us say that we want to transform a bus image such that the side is rectified. Let v_1 and v_2 be a basis for the subspace parallel to the plane and let p

be a point in the plane. Any point in the plane can be written $X_1 v_1 + X_2 v_2 + p$ and its projection

$$\lambda \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = P \begin{pmatrix} v_1 & v_2 & p \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ 1 \end{pmatrix} = H \begin{pmatrix} X_1 \\ X_2 \\ 1 \end{pmatrix}. \tag{4}$$

Thus we have found the homography, H, relating points in the aspect plane with points in the image. By transforming the image according to H^{-1}, we get a rectified image of the aspect, having axis-parallel edges.

Building the Models. We use the standard latent-SVM (LSVM) training of deformable part-based model [3] and will just very briefly review the training. Since the part locations are unknown, they are regarded as latent variables in the training. In latent-SVM, each example x is evaluated by a function of the following form

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \phi(x, z) \tag{5}$$

where β is a vector of model parameters describing the root filter, part filters, 'anchor' positions of parts and deformation coefficents, z is the latent variable specifying the location of root filter and part filters and $\phi(x, z)$ yields the feature vector for a specific configuration.

The goal is to learn the model parameters β from the labeled examples $\langle x_1, y_1 \rangle, \cdots, \langle x_n, y_n \rangle$, where $y_i = -1$, for negative examples and $+1$ for positive examples. This is achieved by minimizing

$$L(\beta) = \mu ||\beta||^2 + \sum_{i=1}^{n} \max\{0, 1 - y_i f_\beta(x_i)\}. \tag{6}$$

Here $\max\{0, 1 - y_i f_\beta(x_i)\}$ is the standard hinge loss and μ determines the relative weight of the regularization term. Figure 2 illustrates the aspect models of the bus category of the Pascal VOC 2011 dataset. One can clearly make out some parts of the bus, especially the wheels.

2.3 Finding Representative Poses

The previous sections have shown how to estimate highly specific models by compensating for varying viewpoint. This introduces a problem since at detection the viewpoint is unknown and cannot be compensated for. To handle this we learn a small set of typical object poses. At detection an image is transformed according to each of these poses and the detector is run on each of the generated views.

To find this set of representative object poses we look again to the annotated training set. Each annotated object yields an object pose that we can use in the learning. Let P_i be the ith pose and let S_i be the set annotated objects which have a similar pose to P_i, where the exact meaning of similarity will be

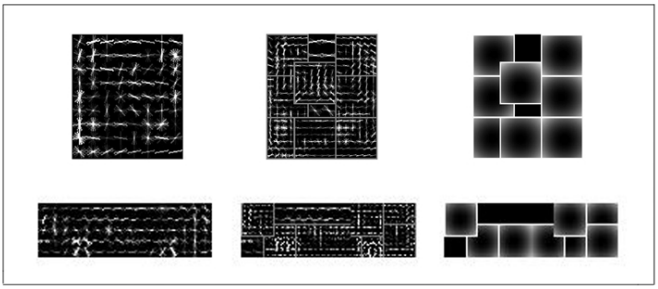

Fig. 2. The two aspect models for the bus category of Pascal VOC 2011. The upper row shows the frontal model and the lower row shows the side model.

described soon. To find k poses which are representative for the training we seek k sets $S_{i_1}, \ldots, S_{i_k}$ such that $\left|\bigcup_j S_{i_j}\right|$ is maximized. This is a maximum k-cover problem and for unbounded k it is NP-hard, but for the size problems that we are considering it is unproblematic to solve with software such as CPLEX.

It remains to define the notion of similar poses. To determine if the pose of an annotated bus is similar to a given pose P, we transform the annotated points according to P. Ideally, both the front and the side of the bus will be transformed to rectangles. Hence as measure of similarity, we measure how much the upper/lower edge of the transformed bus deviates from being horizontal, as illustrated in Figure 3. A certain training example is similar to a pose P if the average angular deviations for the front and side,

$$\frac{1}{2}\left(\theta_{11}+\theta_{12}\right) \quad \text{and} \quad \frac{1}{2}\left(\theta_{21}+\theta_{22}\right). \tag{7}$$

are below a predefined threshold.

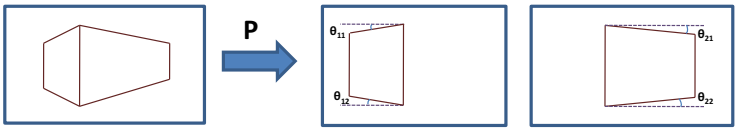

Fig. 3. Measure of angular deviation for pose similarity

3 Detection

Training the aspect models and learning a set of representative poses gives us more specific model representation as well as more flexible choice during detection. For a given test image, we use the set of typical poses to transform it into a number of transformed images. The aspect detectors are run on each of these

images individually. Given that the model consists of M aspects and N typical poses, each pose will define M homographies, each of which is used to rectify the aspect patches. Thus a test image is transformed into $M \cdot N$ candidate images. Still, with small M and using a fast cascade implementation of the detector, see [21], the computational load is no big issue. For buses, we used $M = 2$ and $N = 11$.

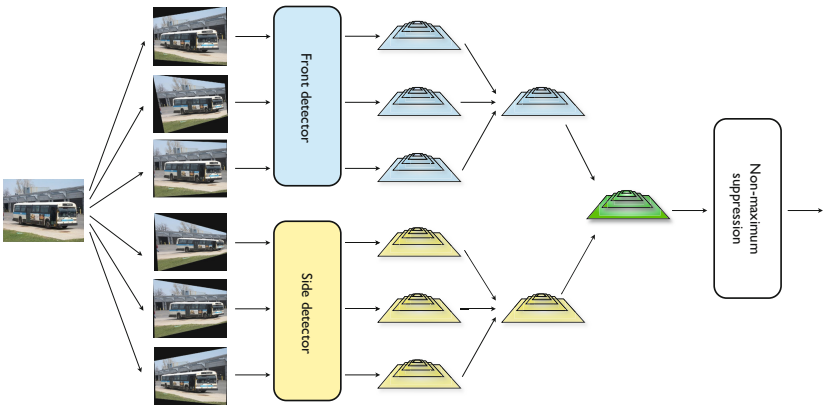

Fig. 4. Overview of the detection pipeline. In the first step the input image is transformed according to each of representative poses. This produces a multiple images that are individually run through the aspect detectors creating a set of score pyramids containing the detector scores at different scales. These are merged into one pyramid per aspect, in the original image coordinate system. Finally, the front and side scores are combined and non-maximum suppression is performed.

3.1 Generating Score Pyramid for a Single Aspect

Now considering a single aspect model $\mathcal{D}$, score pyramids $\mathcal{P}_i^o$, where $i = 1, 2, ..., N$, are generated as the aspect model is run through N transformed image. Different levels in the pyramid corresponds to the different scales with which the image is resized. This is inherited from the original part-based model to enable multi-scale object detection. Each location (x_j, y_j, s_j) in the pyramid defines a rectangular bounding box and a score indicating the likelihood of the object aspect occuring at (x_j, y_j) with scale s_j.

To compare and combine the score pyramids generated from the N different transformed images, we transform all the bounding boxes to the coordinate system of the original image. This yields new score pyramids, where each location implicitly defines a skewed bounding box.

We note that in the original score pyramids, all bounding boxes at a certain level have the same size, but after transformation to the original coordinate frame they are skewed and might change size. When we combine aspects to a object detection we do not want to combine aspects of significantly different

sizes. Hence we only allow combinations where the shared edge of two aspects have approximately the same length. For example, when combining the bounding box of frontal and left side, then we require the right edge of frontal bounding box and the left edge of side bounding box be of the same length. In the next section we will see how this constraint can be enforced in an efficient fashion.

3.2 Combining the Score Pyramids

Let $\mathcal{P}_i^f$ and $\mathcal{P}_i^s$ for $i = 1, 2, ...N$ denote the N score pyramids of frontal aspect and side aspect respectively. To combine the score pyramids from different poses and different aspect models, we need to solve the following problem. For each location (x_j, y_j, s_j) in $\mathcal{P}_i^f$, we need to determine the corresponding location (x_k, y_k, s_k) in $\mathcal{P}_i^s$, of a skewed bounding box BB_j^s for the side aspect. We know that the top-left corner of BB_k^s should coincide with the top-right corner of frontal bounding box BB_j^f, which is implicitly defined at (x_j, y_j, s_j). Let (x_k, y_k), denote the exact location of right-top corner of BB_j^f. We will consider all locations in a small neighborhood of that point.

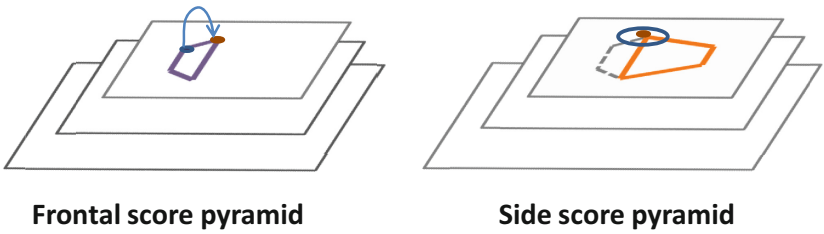

Fig. 5. The illustration on the left shows how to estimate the side location (the brown dot) given the frontal location (blue dot) and the size of skewed frontal bounding box. The illustration on the right shows we search in a small neighborhood (blue circle) of expected location for each level.

To determine s_k, we could search a small neighborhood of (x_k, y_k) for all the levels of $\mathcal{P}_i^s$ for $i = 1, 2, ..., N$ to find the bounding box which fulfills all size constraints. To do so efficiently, we propose a new representation of score pyramid and combine the score pyramids $\mathcal{P}_i^f$ and $\mathcal{P}_i^s$ for $i = 1, 2, ..., N$ into new score pyramids $\mathcal{P}^f$ and $\mathcal{P}^s$.

To make the search more efficient, these score pyramids $\mathcal{P}^f$ and $\mathcal{P}^s$ are created with different levels depending on the length of shared edge. More specifically, we divide the possible bounding box heights into L intervals $l_1, ..., l_L$ and create L corresponding levels in the score pyramid. For each bounding box in the original score pyramids, we check the length of shared edge (the right edge for frontal, the left edge for side) and let this determine where to put it in the new score pyramid. To further simplify the search, we use its top-right corner to represent the front bounding box instead of its top-left corner. Hence this will coincide with the location of side.

Now we have combined the score pyramids from different poses and obtained new score pyramids $\mathcal{P}^f$ and $\mathcal{P}^s$ for the front and side. To get a final detection

we combine these two score pyramids to one score pyramid $\mathcal{P}$. As we use the top-right corner to represent the bounding box in frontal pyramid, we could efficiently combine the two pyramids by directly taking the average of the scores at the same location. To also handle cases when only the front or side is visible, we introduce a threshold s_{thr}. If an aspect has a lower score than s_{thr} we assume that it is not visible. In that case the total score is computed as the average of s_{thr} and the score of the visible aspect. Hence, the score of a combined detection is

$$s(x,y,h) = \begin{cases} \frac{s_f(x,y,h)}{2} + \frac{s_s(x,y,h)}{2} & \text{if } s_f(x,y,h) \geq s_{thr}, s_s(x,y,h) \geq s_{thr} \\ \frac{s_f(x,y,h)}{2} + \frac{s_{thr}}{2} & \text{if } s_s(x,y,h) < s_{thr} \\ \frac{s_s(x,y,h)}{2} + \frac{s_{thr}}{2} & \text{if } s_f(x,y,h) < s_{thr} \end{cases} \tag{8}$$

where $s_f(x,y,h)$ and $s_f(x,y,h)$ denote the score at location (x,y,h) in the combined score pyramids, $\mathcal{P}^f$ and $\mathcal{P}^s$ respectively.

For each location in the final score pyramid $\mathcal{P}$, two skewed bounding boxes are implicitly defined respectively for frontal and side. This gives us a layout estimation of the object. So the layout estimation is inherent in the object detection. Finally, non-maximum suppression is applied on $\mathcal{P}$ to greedily remove detections which has significant overlap; see [3] for details.

4 Experiments

We evaluated our method on the bus category of Pascal VOC 2011 training and validation dataset. The training set has 5717 images of 20 categories among which there are 213 images containing buses. The validation set has 5823 images with 208 containing buses. Since the ground truth annotation is not provided for Pascal VOC 2011 test set, we trained the model on the training set and tested our model on the validation set.

We manually annotated every visible aspect for each positive training example. Pose estimation on annotated training images gives the pose and the actual dimensions of the object up to a scale factor, from which we rectify each aspect of the object. We use both the left and right aspects to train a side detector. Considering the rear and frontal of a bus are quite similar, we trained a frontal detector using both frontal and rear patches. It turned out that this detector worked well for both cases. When determining the similarity of two poses, we set the threshold in (7) as 5 degrees.

In the end, we used 10 different poses to transform the input image. The frontal and side detectors were run on these images but also on the original image so we get 11 score pyramids. Combining the front and side score pyramids, we set the threshold in (8) to $s_{thr} = -1.0$. The experiments are done on a 3.6 GHz Intel Core i7 PC with 64 GB memory, the training takes around 4 hours to train a single aspect model. Detection takes on average 50 seconds per image, but could be speeded up significantly by using the cascade detector from [21].

The precision-recall curve for the bus category is obtained by thresholding all the detection scores at different values, as shown in Figure 6. Average precision is

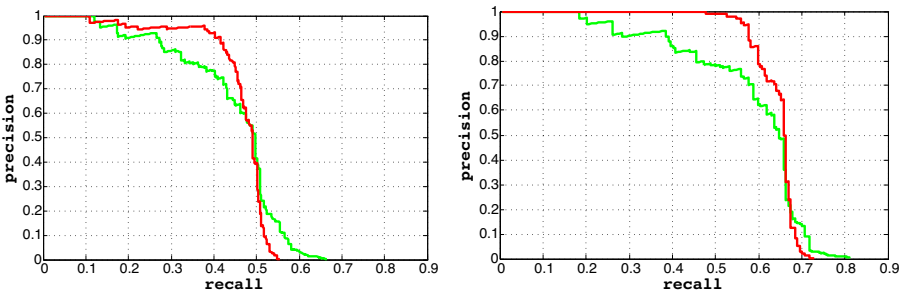

Fig. 6. Precision-recall curves for bus category of Pascal VOC 2011 validation set. The left plot shows the result for the entire set and the right plot shows the result when examples with a lot of occlusion are removed. The green curves are the result using the original part-based model with the average precision AP = 0.450 and 0.587 respectively. The red curves are obtained with our method with AP = 0.468 and 0.644 respectively.

Fig. 7. Example results. Detected bounding boxes are shown in green and their layout in red. The first three rows shows correctly detected objects with roughly correct pose. The method was able to automatically handle cases when only one side is visible. The last row shows the buses of which the pose estimation is less accurate.

calculated by measuring the area under the precision-recall curve using numerical integration. In terms of average precision the improvement over the original part-based model from [3] is limited - from 0.450 to 0.468, but at high precision rates the difference is more significant. We note that Pascal VOC dataset is regarded as a very difficult dataset for detection, occluded and truncated objects usually exist in the dataset, which we do not really expect to handle. To examine this more closely, we removed a third of the bus images that contained largely occluded buses or very distant buses. On the remaining examples we got an average precision of 0.644 compared to 0.587 for [3]. We illustrate some detection results as well as the layout estimation in Figure 7.

5 Conclusions

We described an approach to object detection and layout estimation for objects which can be represented by a number of roughly planar aspects. By training several aspect models and learning a small set of representative poses, we obtained a model with high specificity and flexible choices for detecting objects from various viewpoints. The cost is some extra annotation work as well as increased complexity at detection. We should also mention the limitation of our method which requires the object to be a rigid object with discriminative aspects. The pose estimation in our method is especially well-suited to block-structured objects like the buses.

For the bus category challenging Pascal VOC 2011 dataset, our method achieved better detection results than the original deformable part-based model while keeping a specific and compact model representation. Beyond that, the proposed method also produce geometric information of the detected object, e.g. a pose/layout estimation.

References

1. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Conf. Computer Vision and Pattern Recognition (2001)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Conf. Computer Vision and Pattern Recognition (2005)
3. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Analysis and Machine Intelligence (2010)
4. Lampert, C., Blaschko, M., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. IEEE Trans. Pattern Analysis and Machine Intelligence (2009)
5. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: Int. Conf. Computer Vision (2009)
6. Fischler, M., Elschlager, R.: The representation and matching of pictorial structures. Transactons on Computers (1973)
7. Zhu, L., Chen, Y., Yuille, A.L., Freeman, W.T.: Latent hierarchical structural learning for object detection. In: Conf. Computer Vision and Pattern Recognition (2010)

8. Vedaldi, A., Zisserman, A.: Structured output regression for detection with partial occulsion. In: Advances in Neural Information Processing Systems (2009)
9. Torralba, A., Murphy, K., Freeman, W.: Sharing visual features for multiclass and multiview object detection. IEEE Trans. Pattern Analysis and Machine Intelligence (2007)
10. Ott, P., Everingham, M.: Shared parts for deformable part-based models. In: Conf. Computer Vision and Pattern Recognition (2011)
11. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3d human pose annotations. In: Int. Conf. Computer Vision (2010)
12. Xiang, Y., Savarese, S.: Estimating the aspect layout of object categories. In: Conf. Computer Vision and Pattern Recognition (2012)
13. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., Gool, L.V.: Towards multi-view object class detection. In: Conf. Computer Vision and Pattern Recognition (2006)
14. Savarese, S., Fei-Fei, L.: 3d generic object categorization, localization and pose estimation. In: Int. Conf. Computer Vision (2007)
15. Liebelt, J., Schmid, C.: Multi-view object class detection with a 3d geometric model. In: Conf. Computer Vision and Pattern Recognition (2010)
16. Yang, Y., Ramanan, D.: Articulated pose estimation using flexible mixtures of parts. In: Conf. Computer Vision and Pattern Recognition (2011)
17. Branson, S., Perona, P., Belongie, S.: Strong supervision from weak annotation: Interactive training of deformable part models. In: Int. Conf. Computer Vision (2011)
18. Chiu, H., Kaelbling, L., Lozano-Pérez, T.: Virtual training for multi-view object class recognition. In: Conf. Computer Vision and Pattern Recognition (2007)
19. Triggs, B.: Camera pose and calibration from 4 or 5 known 3d points. In: ICCV 8, pp. 278–284 (1999)
20. Josephson, K., Byröd, M.: Pose estimation with radial distortion and unknown focal length. In: Conf. Computer Vision and Pattern Recognition (2009)
21. Felzenszwalb, P., Girshick, R., McAllester, D.: Cascade object detection with deformable part models. In: Conf. Computer Vision and Pattern Recognition (2010)