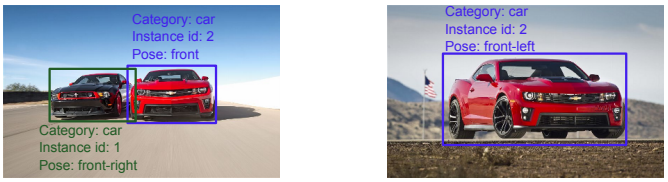
Object Co-detection

Sid Yingze Bao, Yu Xiang, Silvio Savarese

University of Michigan at Ann Arbor, USA
{yingze, yuxiang, silvio}@eecs.umich.edu

Abstract. In this paper we introduce a new problem which we call *object co-detection*. Given a set of images with objects observed from two or multiple images, the goal of co-detection is to detect the objects, establish the identity of individual object instance, as well as estimate the viewpoint transformation of corresponding object instances. In designing a *co-detector*, we follow the intuition that an object has consistent appearance when observed from the same or different viewpoints. By modeling an object using state-of-the-art part-based representations such as [1,2], we measure appearance consistency between objects by comparing part appearance and geometry across images. This allows to effectively account for object self-occlusions and viewpoint transformations. Extensive experimental evaluation indicates that our co-detector obtains more accurate detection results than if objects were to be detected from each image individually. Moreover, we demonstrate the relevance of our co-detection scheme to other recognition problems such as single instance object recognition, wide-baseline matching, and image query.

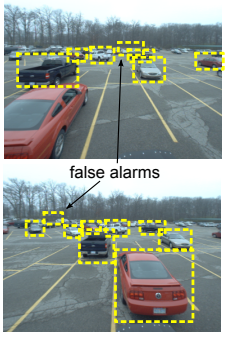
1 Introduction



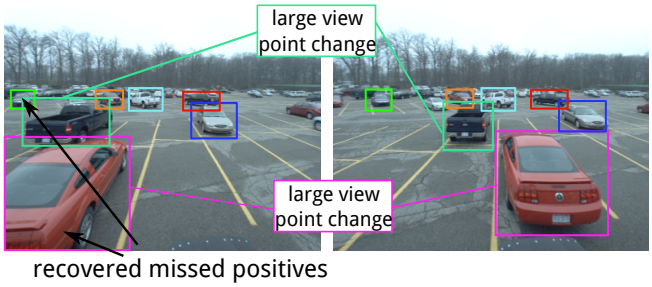
(a) Input Image #1 (b) Input Image #2

Fig. 1: Object co-detection for two images. The goal is to i) detect objects; ii) identify which objects correspond to the same object instance (e.g. the red Camaro); we call these instances *matching objects*; iii) estimate the viewpoint transformation between matching objects.

We introduce a framework for solving a new problem called *object co-detection*. Given multiple images, each of which may contain object instances of a given category observed from different viewpoints, the goal of co-detection is to: 1) detect objects in all images; 2) recognize whether or not objects in different images correspond to the same instance – we refer to these object instances as *matching objects*; 3) estimate the viewpoint transformation between matching



(a) Single image object detection. Notice miss positives and false alarms.



(b) Object co-detection. Different colors correspond to different matching objects. Co-detection recovers missed positives and removes false alarms, compared to single image object detection (Fig. 2a).

Fig. 2: Object co-detection improves object detection and matches objects.

objects. Fig. 1 illustrates co-detection in two images. Fig. 1a shows two instances of the car category: a black Ford Mustang and a red Chevy Camaro. Fig. 1b also contains a red Camaro, which is considered to be the matching object to the Camaro in Fig. 1a. Through the process of co-detection, the two Camaro detections are matched and the viewpoint transformation between the two instances is estimated. The black Mustang is kept as a detection, but it has no matched object in the other image.

An important property that motivates the introduction of the co-detection paradigm is its ability to obtain superior detection results over conventional single-image detection schemes. We argue that, by leveraging on the fact that an object has consistent appearance when observed from the same or different viewpoints, a co-detector is capable of obtaining more accurate detection results than if objects were to be detected from each image individually. Consider the example in Fig. 2a, the red car appears in both images. This car is successfully detected by a state-of-the-art detector [1] in Fig.2a-bottom, but it is not in Fig.2a-top. Our co-detector has the ability to recover the missed detection by leveraging the fact that the same car instance is detected in the other image, and that appearance and shape of the car must be consistent across the two images (up to a viewpoint transformation). If the car instance appears in only one of the images, the co-detector is equivalent to a single image detector. Notice that a co-detector can be applied to an arbitrary number of images (not just two).

Object co-detection is far from being a trivial problem. An object instance may have a dramatically varied appearance due to viewpoint transformations and self-occlusions (parts of the object are only visible from some viewpoints). Moreover, the background surrounding the object may also vary, which makes the naive object matching methods unstable (e.g. by matching bounding boxes via image features). Furthermore, object co-detection requires the simultaneous solution of two already difficult problems: object detection and pose estimation.

State-of-the-art methods that address these problems still have much room for improvement.

In this work, we propose a novel framework for object co-detection. Our method jointly detects and matches objects by their parts. To represent an object category by parts, our model leverages existing part-based object representation models (e.g. [1,2]). One possible object representation is shown in Fig. 4a. We measure appearance consistency between objects by matching their parts (Fig. 4b). Compared with a holistic object representation [3], a part-based object representation is more robust to viewpoint changes and self-occlusions. We combine information from multiple images by introducing an energy based formulation that models both the object's category-level appearance similarity in each image and the instance's appearance consistency across images. We also propose a novel matching potential function to handle large viewpoint transformations and self-occlusions in the part matching process.

The main contributions of this paper include: 1) a general framework for object co-detection, which allows us to detect matching objects from two or multiple images without any knowledge on the viewpoint geometry; 2) a novel energy function and a matching potential function to model the object visual appearances both within images and across images; 3) extensive experimental evaluation on three public datasets – a car dataset [4], a pedestrian dataset [5], and a 3D object dataset [6]. Compared with alternative state-of-the-art methods, the proposed framework can improve both the detection and pose estimation accuracy, as well as match object instances more robustly.

2 Related Work

Co-detection is related with and potentially useful to several other problems in computer vision:

Object detection. Given an object category model, methods such as [3,7,8,1,9,2] identify an object of such category from an input image. Co-detection is a generalization of standard object detection in that it handles multiple input images which contain the same objects. If an object instance is only present in one image, a co-detector degenerates into a standard object detector. Otherwise, a co-detector leverages object appearance and shape consistency to improve object detection accuracy. Furthermore, a co-detector can discover matching instances.

Single instance 3D object detection. Given a 2D or 3D model of an object instance, methods such as [10,11,12,13] detect the same object instance from a query image. Particularly, in [10,11], the object model is just a single training image and the object (which is possibly observed from a different viewpoint) is identified in the query image by matching features or aggregations of features. Object co-detection provides a framework for potentially incorporating the same appearance matching constraints as in [10,11], and it does not require the identification of the object location in the training image (object locations can be unknown)

Image co-segmentation. Given multiple images containing similar foreground objects, methods such as [14,15,16] perform pixel-level segmentation of the shared foreground objects. Most co-segmentation methods only depend on

low-level image appearance information, and hence tend to fail if the object appearance changes because of viewpoint transformations. Furthermore, most co-segmentation methods do not attempt to recognize the object identity and cannot cope with multiple object instances in the same image. On the contrary, a co-detector is designed to detect an arbitrary number of object categories per image and associate a category label to each detection. Moreover, co-detection is designed to handle large viewpoint transformations across images.

Tracking by detection. To solve this problem [17,18,19], correspondences of object detections must be established across frames in order to form tracklets. Unlike co-detection, in these works detections are obtained independently from each frame and subsequently matched. By jointly detecting the same object instance from all the frames, a co-detection framework could potentially improve the tracklet quality and help make tracking by detection more robust.

Semantic structure from motion (SSFM). Given multiple views of a scene, SSFM methods such as [4,20,21] use high level semantic information to help estimate the camera viewpoint changes. In turn, object detection accuracy is improved by leveraging the estimated camera pose geometry. A co-detection method could play a critical role in a SSFM framework in that it can establish matches of objects across views without using camera information (external and internal parameters).

Single instance matching. Given an image of an object instance (e.g a music CD cover), the goal is to retrieve the same object instance from a large collection of images. Methods such as [22,23,24] usually evaluate the similarity based on the whole image and thus require that the image only contains one dominating object. Conversely, our object co-detection is capable of identifying and matching the objects of interest and discarding uninformative background clutter.

Region matching. Methods such as [25,26] match features or regions across views of the same scene. Co-detection is fundamentally different in that it works with high level semantics (i.e. objects). However, co-detection can be helpful for those algorithms since it provides high level contextual information for pruning out false feature or region matches.

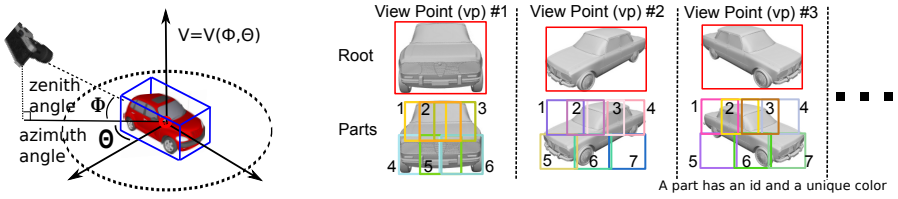
3 Object Co-detection Model

In an object co-detection problem, we are given a total number of K input images $\mathcal{I} = \{I^1, \ldots, I^K\}$. The goal of the co-detector is to detect the matching instances $\mathcal{O} = \{O^1, \ldots, O^K\}$ that simultaneously appear in each of the input image, where O^k is an object instance in image I^k.

3.1 Object Representation

In our co-detection model, we adopt a part-based object representation. An object O in an image is represented by a root r, a number of n parts $\mathcal{P} = \{p_1, \ldots, p_n\}$, and a viewpoint V, i.e., $O = (r, \mathcal{P}, V)$. We explore two types of object representations: 2D part representation and 3D part representation.

In a 2D representation such as [1], the root and parts are specified by rectangles in the image (Fig. 3b). Since different parts are defined for different viewpoints independently, no explicit part correspondence can be established across



(a) The viewpoint V in a 3D part representation. Φ, Θ are zenith and azimuth angles

(b) A 2D part representation, where object parts are represented by 2D rectangles in the image plane. [1] uses 2D part representation.

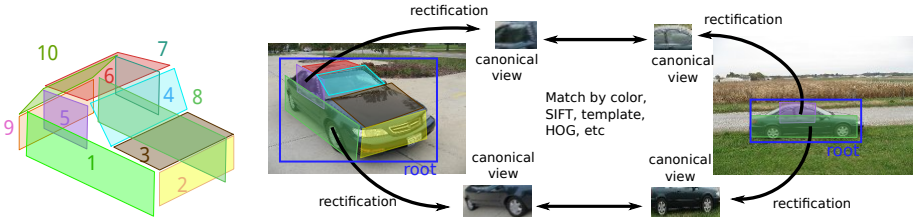
Fig. 3: Viewpoint and 2D part representation.

different viewpoints (Fig. 3b). Thus, a 2D representation is only suitable for matching objects observed from very similar viewpoints (e.g. if images are captured by small-baseline stereo cameras). In such a case, parts association can be easily established.

In a 3D representation such as [6,2,8], the root is specified by a rectangle in the image, and parts are associated to 3D flat surfaces that make up an object (Fig. 4a). The viewpoint is denoted by the azimuth and zenith angle of object pose (Fig. 3a). The canonical view of a part (Fig. 4b) is defined as the most frontal view of the part. If the pose of the object is available, any part in the 2D image can be rectified into its canonical view by using the homography transformation provided by the estimated viewpoint. Such rectification process allows us to compare the normalized appearance of two matching parts when observed from different viewpoints. (Fig. 4b). Moreover, a 3D part representation also enables us to predict if a certain part is occluded by other parts of the object (self-occlusion), which therefore prevents self-occluded parts from being erroneously matched. For all these reasons, a 3D representation is appropriate for matching objects observed from different viewpoints.

3.2 Energy Function for the Model

In formulating the co-detection framework, we follow the key intuition that objects across images are matched by associating corresponding parts. Fig. 5 shows the graphical representation of the model when two images are considered. The linkages between parts model the property that the corresponding parts must



(a) A 3D part representation for a car.

(b) A 3D part representation allows to match objects across images by matching their parts after viewpoint rectification.

Fig. 4: An example of 3D object part representation (a) and the matching process (b). The estimated viewpoint is the key to predicting self-occlusion and matching parts under different viewpoints. The similarity between parts is evaluated based on a bundle of features (Sec. 3.4).

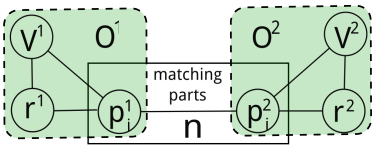


Fig. 5: Object co-detection model when two images are considered. The dashed green box measures the compatibility between an object and its image (E_{unit}). The middle rectangle measures the similarity of parts of different objects (E_{match}).

have similar appearance. Notice that, the model degenerates into a typical part-based object detection model (the green dashed box) if only one image is presented. The model in Fig. 5 can be generalized to the case of K input images and we define the following energy function to measure the likelihood of detecting the matching objects $\{O^1 \cdots O^K\}$ in different images $\{I^1 \cdots I^K\}$:

$$E(\mathcal{O},\mathcal{I}) = \sum_{k=1}^{K} E_{\text{unit}}(O^k, I^k) + \sum_{i=1}^{n} E_{\text{match}}(\{p_i^k\}_{k=1}^K, \{V^k\}_{k=1}^K, \mathcal{I}), \tag{1}$$

where E_{unit} measures the compatibility between the object O^k and the image I^k, and E_{match} models the constraint that the i^{th} part of a matching object should have similar appearance across images.

The term E_{unit} is the *unitary potential* and defined as:

$$E_{\text{unit}}(O^k, I^k) = E_{\text{root}}(r^k, V^k, I^k) + \sum_{i=1}^{n} E_{\text{part}}(p_i^k, V^k, I^k) + \sum_{i=1}^{n} E_{\text{rp}}(r^k, p_i^k, V^k, I^k), \tag{2}$$

where E_{root} and E_{part} are the unary potentials measuring the compatibility between image evidence and the root and the object part respectively; E_{rp} is the pairwise potential that measures the consistency between a part and its root. E_{rp} models the relative location between a root and the part, following a star-model representation. Details of computing E_{unit} are given in Sec. 3.3.

The term E_{match} is the *matching potential* and defined as:

$$E_{\text{match}}(\{p_i^k\}_{k=1}^K, \{V^k\}_{k=1}^K, \mathcal{I}) = \frac{1}{C_K^2} \sum_{k_1,k_2} M(p_i^{k_1}, p_i^{k_2}, V^{k_1}, V^{k_2}, I^{k_1}, I^{k_2}), \tag{3}$$

where $M(p_i^{k_1}, p_i^{k_2}, V^{k_1}, V^{k_2}, I^{k_1}, I^{k_2})$ is a matching function (Eq. 4) which measures the appearance similarity between the i^{th} part of object O^{k_1} and the i^{th} part of object O^{k_2}, and C_K^2 denotes the total number of possible object matches. Details of computing E_{match} are given in Sec. 3.4. Notice that the matching potential for multiple images is in practice expressed as a summation of pair-wise matching potentials.

By using the energy function defined in Eq. 1, a co-detector can boost the score (energy) of true positives if matching objects exist in other images. Therefore, a co-detector is capable of recovering true positives missed by a single-image detector (by threshold cutting).

3.3 Unitary Potential E_{unit}

The unitary potential E_{unit} measures the compatibility between object O^k and the evidence in image I^k. E_{unit} can be evaluated by retaining the score of a detection candidate returned by any standard object detector such as [3,7,1,9,2]. In this paper, we adopt the energy formulation of a typical part-based object detection model (e.g. Sec. 3.1 in [1] and Sec. 3.1 in [2]). In such models, the category-level detection templates, which encode the visual features (e.g. HOG [3]), are trained for both root and parts. Relative locations between a root and parts are also encoded in the models. Given an input image, an object is detected by searching for the optimal locations of the root and parts so that their visual features fit the templates and their relative locations fit the shape model. We define β_{root}, β_{part}, and β_{rp} as the parameters in E_{root}, E_{part}, and E_{rp}. The form of these parameters varies according to the model applied[1]. Sec. 3.6 explains how we learn these parameters.

3.4 Matching Potential E_{match}

The matching potential E_{match} measures the similarity between two objects by matching their corresponding parts. If a part p_i is visible, we can extract its feature ϕ_i from the image. ϕ_i consists of a set of geometrical and visual features. In our experiment, the geometrical feature is: 1) the 3D location of this part w.r.t. the 3D object centroid if a 3D part representation (e.g. [2]) is applied, or 2) the 2D part location w.r.t. the 2D object centroid if a 2D part representation (e.g. [1]) is applied. The visual features include color histogram, point feature [24] and pixel intensity values within image patches. If a 3D part representation is applied, we extract such features after rectifying the part into its canonical view (Fig. 4b).

If a part p_i is visible in both images $I^{k_1}$and I^{k_2}, we compute a vector $\mathbf{s}(\phi_i^{k_1}, \phi_i^{k_2})$ to measure the similarity between its features $\phi_i^{k_1}$ and $\phi_i^{k_2}$:

$$\mathbf{s}(\phi_i^{k_1}, \phi_i^{k_2}) = [s_1(\phi_i^{k_1}, \phi_i^{k_2}), s_2(\phi_i^{k_1}, \phi_i^{k_2}), s_3(\phi_i^{k_1}, \phi_i^{k_2}), s_4(\phi_i^{k_1}, \phi_i^{k_2})],$$

where s_1 is the negative value of the KL-distance between the color histograms, s_2 is the log value of the number of matched SIFT [24] points, s_3 is the inner product of the normalized image patches, s_4 is the inverse value of the distance between their geometrical features. On the other hand, if either part is not visible (self-occluded), we set $\mathbf{s}(\phi_i^{k_1}, \phi_i^{k_2}) = \mathbf{0}$.

To handle object self-occlusions, we associate a visibility indicator v_i^k with part p_i^k, where $v_i^k = 1$ if p_i^k is visible in image I^k and vice versa. v_i^k is a function only of the object shape and viewpoint[2]. After considering the part visibility, we use the following vector to represent the similarity between two parts:

$$\mathbf{d}_i^{k_1 k_2} = [v_i^{k_1} v_i^{k_2} \mathbf{s}(\phi_i^{k_1}, \phi_i^{k_2})^T, 1 - (1 - v_i^{k_1})(1 - v_i^{k_2})]^T.$$

[1] E.g. if the model in [1] is applied, we have $\beta_{root} = F_0'$, $\beta_{part}^i = F_i'$, and $\beta_{rp}^i = d_i$ for each part i, where the right-hand terms are defined in Eq. 2 and 3 in paper [1].

[2] If a 2D part representation is applied, $v_i^k = 1$ for every parts of the object that is seen from the same viewpoint.

Note that $\mathbf{d}_i^{k_1 k_2}$ is a function of part locations, viewpoints and images. The last term of $\mathbf{d}_i^{k_1 k_2}$ accommodates the bias in the case where either part is not visible. We compute the similarity score as

$$M(p_i^{k_1}, p_i^{k_2}, V^{k_1}, V^{k_2}, I^{k_1}, I^{k_2}) = \mathbf{w}_i^T \mathbf{d}_i^{k_1 k_2}, \tag{4}$$

where $\mathbf{w}_i$ is the matching weight to be learned from a training set. Since $\mathbf{d}_i^{k_1 k_2}$ encodes the visibility information, we can learn a universal set of weights $\mathbf{w}_i$ for all the parts under different viewpoints. The procedure for learning $\mathbf{w}_i$ is explained in Sec. 3.6.

3.5 Model Inference

The goal of the inference is to find the optimal matching instances $\mathcal{O}^*$ in the images $\mathcal{I}$ so that:

$$\mathcal{O}^* = \arg\max_{\mathcal{O}} E(\mathcal{O}, \mathcal{I}),$$

where $E(\mathcal{O}, \mathcal{I})$ is defined in Eq.1. The inference outputs the bounding box, part locations, viewpoint and instance ID (which defines matching objects correspondences across images) for each object in the images. Exactly solving the above optimization problem is intractable, since the model contains loops. We propose a two-step inference algorithm to make the problem computationally tractable.

The first step is to predict a candidate pool of object instances consisting of all objects whose unitary potential E_{unit} is larger than a threshold. Fig. 6 illustrates the candidate pool when [1] is applied. Since computing E_{unit} is equivalent to computing the potential score of an object detector, this candidate pool can be obtained by applying category level object detector without non-maximum suppression. Notice that, two resulting candidates may have the same root location but different part locations.

The second step is to identify the best set of co-detections by searching through all across-image matches in this candidate pool. Given K images, suppose the candidate pool of image I^k contains n_k objects ($k = 1 \cdots K$), then there will be $\prod_{k=1}^{K} n_k$ possible matching object candidates. We compute the joint energy $E(\mathcal{O}, \mathcal{I})$ for every matches. Since the unitary potential E_{unit} is already computed during the first step, the additional operation is just to compute the matching potential E_{match}, which is computationally cheap as it only requires the calculation of dot products. Finally, we apply non-maximum suppression to select among the $\prod_{k=1}^{K} n_k$ possible matches the best matching objects. Matching objects are selected based on their energy values – matching objects associated to

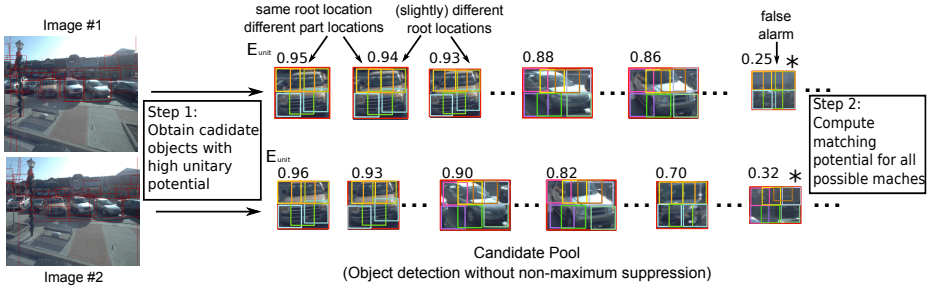


Fig. 6: Two-step inference. In this example, we apply [1] to compute E_{unit}. Two input images are displayed on the left. Each row on the right corresponds to a set of candidate detections extracted from the corresponding image on the left hand side.

high energy values are preferred over those associated with lower energy values. The result of this selection process is the output of the co-detector.

3.6 Model Learning

In order to learn the parameters of the co-detection model, we label the bounding boxes of objects and the ground truth matching objects across images. Given a set of T groups (a group consists of two or more images that include matching objects) of training images $\{\mathcal{I}^t\}$ with labeled matching objects $\{\mathcal{O}^t\}$, the goal is to learn β_{root}, β_{part}, β_{rp}, and $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_n)$. Since the part locations are not labeled, learning can be solved following a latent SVM learning procedure (part locations are latent variables):

$$\begin{aligned} &\{\beta_{\text{root}}, \beta_{\text{part}}, \beta_{\text{rp}}, \mathbf{w}\} \\ &= \arg \min_{\beta_{\text{root}}, \beta_{\text{part}}, \beta_{\text{rp}}, \mathbf{w}} \frac{1}{2}(\|\beta_{\text{root}}\|^2 + \|\beta_{\text{part}}\|^2 + \|\beta_{\text{rp}}\|^2 + \|\mathbf{w}\|^2) + \\ &\lambda \sum_t \max(0, 1 - y_t \max_{\mathcal{P}^t} E(\mathcal{O}^t, \mathcal{I}^t)), \end{aligned} \tag{5}$$

where $\mathcal{P}^t$ represents all possible part locations for the objects $\mathcal{O}^t$, λ is the regularization constant, $y_t \in \{1, -1\}$ indicates if the t^{th} training group is positive or negative. However, exact learning using Eq. 5 is intractable due to the high dimensionality of the unknowns and the presence of loops in the model.

Instead of solving the problem in Eq. 5, we propose a two-step learning procedure. First, we only learn β_{root}, β_{part}, β_{rp} based on individual training images. This is equivalent to learning parameters of a traditional part-based detector (e.g. [1]). By using the learned β_{root}, β_{part}, β_{rp} and labeled root location r^k, the object parts in the training image I^k can be predicted as $\{\bar{p}_i^k\}_{i=1}^n$. Second, we learn $\mathbf{w}$ based on labeled matches, labeled viewpoints, and predicted parts:

$$\mathbf{w} = \arg \min_{\mathbf{w}} \frac{1}{2} \sum_i \|\mathbf{w}_i\|^2 + \lambda \sum_{t=1}^{T} \max(0, 1 - y_t [\sum_{i=1}^{n} E_{match}(\{\bar{p}_i^k\}_{k=1}^K, \{V^k\}_{k=1}^K, \mathcal{I})])$$

where $\mathbf{w}$ can be estimated using a standard support vector machine.

4 Experiments

The experiments are designed in order to demonstrate: 1) an object co-detector is capable of successfully detecting matching objects across images; 2) estimate the viewpoint transformation between matching objects; 3) achieve superior performances than traditional detection methods that work on individual images in isolation; 4) achieve similar performances to traditional detection methods if no matching objects are present in the images; 5) a co-detector can be successfully used to detect an object instance with just one training image (where the same object instance is observed from an unknown and arbitrary viewpoint) and obtain superior results than traditional single instance detectors. Moreover, we present experiments that demonstrate that our co-detection framework can be useful in a number of recognition scenarios so as to: 1) match the same object instances across images where the object location is known but the association and viewpoint transformation is unknown; 2) establish the correct correspondence between images that contain the same (but unknown) object instances seen from different (unknown) viewpoints.

Average Precision (%)		Car (all)	Car (h>80)	Pedestrian (all)	Pedestrian (h>120)
Stereo Pair	[1]	49.8	47.1	59.7	55.4
	Co-detector	**53.5**	**55.5**	**62.7**	**63.4**
Random Pair	[1]	49.8	47.1	**59.7**	55.4
	Co-detector	**50.0**	**49.1**	58.1	**58.1**

Table 1: Object detection results using the car dataset [4] and the pedestrians dataset [5]. "h>X" means we only count the objects with height more than X pixels. The image height of the car / pedestrian dataset is 600 / 480 pixels. "Stereo pair": testing image pairs are obtained from a stereo camera with small baseline; this implies that most images contain matching objects. "Random pair": testing image pairs are randomly selected from the whole data set; this implies that most of these images contain few or none matching objects. The number of testing image pairs are 300 / 200 for the car / pedestrian dataset.

4.1 Object Detection and Pose Estimation

The experiments on object detection and pose estimation are conducted on three publicly available datasets: a car dataset [4] (see Fig. 8a), a pedestrian dataset [5] (see Fig. 8b), and a 3D object dataset [6] (see Fig. 8c and 8d). To evaluate object detection accuracy, we follow the criteria in the PASCAL VOC challenge[3] and report average precision (AP). To evaluate pose estimation accuracy, we follow the criteria in [6]. Tab. 1 shows the object detection results on the car and pedestrian datasets. For both datasets we evaluate the co-detector on image pairs with either small baseline (indicated by stereo pairs) or with large baseline or with no overlap at all (indicated as random pairs). In the former case, the object viewpoint change is not significant, and we apply the model in [1] (which uses a 2D part representation) to represent objects and compute E_{unit}. Tab. 1 shows that, object co-detector achieves higher detection accuracy than a traditional object detector such as [1] when it is applied on each image in isolation. This advantage grows if we only count the large objects in images, since these contain better identifiable parts than small scale objects. Tab. 1 also shows that, if random pairs of images are considered, object co-detection performs on par with single-image detection (e.g. [1]). This result validates the property that if no matching objects are present in the images, a co-detector degenerates into a traditional part-based detector.

Tab. 2 shows the object detection and pose estimation results on the 3D object dataset [6], where significant object viewpoint changes exist. In the following experiments, we use 5 object instances for testing in each category. We enumerate all pairs of images containing matching objects to generate the testing image list. We apply the model in [2]. Examples of a 3D object representations in [2] are shown in Fig. 7. As Tab. 2 shows, object co-detection outperforms [2] in detecting the objects and estimating their pose. The gain may not be substantial

[3] `http://pascallin.ecs.soton.ac.uk/challenges/VOC/`

		Iron	Mouse	Shoe	Car	Cellphone	Stapler	Bike	Toaster	Mean
Object Detection AP (%)	[2]	82.2	52.2	84.1	**98.3**	80.2	**70.5**	**93.8**	97.5	82.3
	Ours	**82.5**	**54.5**	**85.5**	98.0	**81.0**	70.2	93.1	**98.2**	**83.0**
Pose estimation accuracy (%)	[2]	86.0	69.8	86.6	93.1	**86.3**	73.2	90.1	65.4	81.3
	Ours	**89.8**	**72.0**	**88.0**	**95.3**	86.0	**73.9**	**92.3**	**70.3**	**83.5**

Table 2: Object detection and pose estimation results using the 3D object dataset [6].

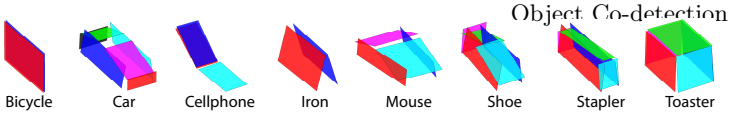


Fig. 7: The 3D part representation for eight categories in [2].

for those categories for which the baseline method [2] already shows very strong performance.

4.2 Detecting Single Object Instances

In this experiment, we demonstrate the ability of the co-detector to detect an object instance from a testing image under the assumption that the same object instance is observed and labeled in one of the training images. The object poses in testing and training are in general different. We compare against a single instance detection method [24], which uses generalized Hough voting and homography validation to detect objects. Tab. 3 shows the detection accuracy for detecting a labeled instance. Notice that our method achieves a significant improvement over [24] in that it leverages the learnt categorical structure of object as opposed to [24] which only relies on low level features and a subsequent geometrical validation step. Tab. 4 summarizes the overall accuracy in detecting objects and estimating their pose. The comparison between Tab. 4 and Tab. 2 allows us to appreciate the superior performance of the co-detector when the object position is available in one of the two images (Tab. 4), as opposed to be unknown in both images (Tab. 2).

4.3 Matching Objects

In this experiment, we demonstrate the ability of the co-detector to discover matching objects. We assume that objects are already correctly detected (i.e., the object bounding box is given for all the images) and the task consists of establishing the correct match between bounding boxes corresponding to same object instances. For each trial, we have 5 candidate object instances and 1 target object instance of the same object category. The goal is to find among the 5 candidates the one that corresponds to the target. We compare the co-detector against a number of baseline methods that are capable of estimating if two object bounding boxes correspond to the same instance or not. These methods use different strategies to compute the matching score. As Tab 5 shows, the co-detector obtains the best performances in all the experiments.

4.4 Matching Images by Objects

In this experiment, the goal is to match images if they contain the same object instance. Unlike the previous experiment, the locations of objects are not given in any of the images. For each trial, we have 5 candidate images and 1 target image. Each image contains one object. The goal is to find among all the image candidates the one that contains the same object instance as in the target image. We compare the co-detector against several possible image matching methods

AP (%)		Iron	Mouse	Shoe	Car	Cellphone	Stapler	Bike	Toaster
Same Pose	[24]	25.4	15.2	37.6	43.2	30.7	25.6	24.6	15.2
	Ours	**90.8**	**56.5**	**86.6**	**98.4**	**88.5**	**72.6**	**93.7**	**98.2**
Different Pose	[24]	2.5	2.2	6.0	3.3	5.6	1.2	5.0	1.3
	Ours	**81.8**	**54.8**	**86.3**	**98.1**	**81.1**	**71.4**	**94.5**	**97.9**

Table 3: Single instance detection result using the 3D object dataset. Same / Different Pose: the azimuth angle (Fig. 3a) of an object in a query image is the same / different as the the azimuth angle of the labeled object.

AP (%)	Iron	Mouse	Shoe	Car	Cellphone	Stapler	Bike	Toaster	Mean
Detection AP.	84.8	55.3	86.3	98.2	83.6	71.7	94.2	98.0	84.0
Pose Est. Acc.	93.2	76.7	90.1	97.9	89.3	79.0	92.1	87.3	88.2

Table 4: Single instance detection results. See Tab. 2 for a comparison.

and report the matching accuracy in Tab. 6. We also apply image matching methods to match the bounding box of the most likely detection returned by [2], and we denote these results as "+Det". If we apply matching methods to match the ground truth bounding boxes of objects, the result will be identical to the experiment reported in Sec. 4.3. Our co-detection model achieves superior performance in all the experiments.

5 Conclusion

In this paper, we have introduced the problem of object co-detection and proposed a novel framework for solving it. We have shown that our framework, by leveraging state-of-the-art part-based object representations, is capable of successfully addressing the co-detection problem in presence of large viewpoint changes and object self-occlusions. We have conducted extensive experimental evaluation on three challenging datasets to demonstrate properties and strengths of our co-detection approach.

Acknowledgments

We acknowledge the support of NSF CAREER #1054127 and NSF CPS grant #0931474.

References

1. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI (2010)
2. Xiang, Y., Savarese, S.: Estimating the aspect layout of object categories. In: CVPR. (2012)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)
4. Bao, S.Y., Savarese, S.: Semantic structure from motion. In: CVPR. (2011)
5. Ess, A., Leibe, B., Gool, L.V.: Depth and appearance for mobile scene analysis. In: ICCV. (2007)
6. Savarese, S., Fei-Fei, L.: 3d generic object categorization, localization and pose estimation. In: ICCV. (2007)
7. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: Workshop on Statistical Learning in Computer Vision, ECCV. (2004)

Accuracy %		Iron	Mouse	Shoe	Car	Cellphone	Stapler	Bike	Toaster
Same Pose	Color	55.4	55.4	40.8	39.2	48.7	53.0	26.8	54.4
	SIFT[24]	46.6	43.7	47.7	58.9	44.9	43.3	40.5	43.2
	SP[27]	46.8	**58.7**	49.2	39.5	42.7	41.3	34.9	66.0
	Ours	**60.0**	55.6	**66.8**	**64.5**	**67.0**	**59.2**	**57.6**	**86.5**
Different Pose	Color	50.1	43.8	38.4	38,3	27.9	43.1	30.2	52.7
	SIFT[24]	26.1	33.4	34.7	27.3	26.2	30.9	27.6	32.4
	SP[27]	29.6	44.8	44.1	29.2	21.3	31.2	30.0	44.5
	Ours	**56.1**	**52.6**	**63.1**	**46.2**	**56.5**	**55.3**	**62.3**	**83.5**

Table 5: Accuracy in matching object instances. Different baseline methods are compared using two different settings: the matching objects have the same / different azimuth pose. In Color, color histograms within the object bounding box (BB) are compared. In SIFT[24], the number of matched SIFT features within the object BB is used. In SP, a spatial pyramid matching method [27] within the object BB is used.

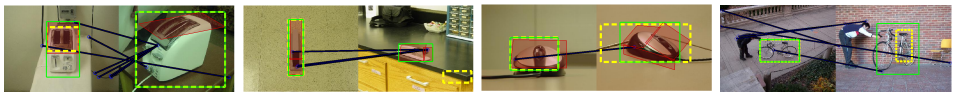
Accuracy %		Iron	Mouse	Shoe	Car	Cellphone	Stapler	Bike	Toaster
Same Pose	BoW[28]	42.2	31.2	37.1	30.7	54.9	31.2	26.9	26.6
	SP[27]	42.7	31.9	39.3	34.1	**56.7**	32.5	31.0	28.6
	Color+Det	52.7	35.5	35.1	39.0	40.8	40.1	26.9	39.6
	SP[27]+Det	40.2	36.3	41.0	38.1	40.5	31.7	32.5	53.9
	SIFT[24]+Det	41.9	39.3	46.4	59.5	40.9	38.5	39.9	41.3
	Ours	**53.6**	**47.6**	**55.1**	**64.7**	53.9	**50.6**	**58.3**	**66.0**
Different Pose	BoW[28]	35.3	32.1	36.6	35.8	30.0	30.3	30.1	31.1
	SP[27]	41.7	33.0	37.1	37.5	29.1	30.5	34.4	31.3
	Color+Det	42.6	36.0	34.6	34.4	20.7	37.6	29.7	40.5
	SP[27]+Det	33.2	29.6	32.3	27.0	22.5	26.6	30.8	39.0
	SIFT[24]+Det	35.8	28.6	33.2	28.1	26.8	27.1	27.3	31.0
	Ours	**48.3**	**44.1**	**45.9**	**44.2**	**40.3**	**44.3**	**64.8**	**59.4**

Table 6: Accuracy in matching images that contain the same object instance. Different baseline methods are compared using two different settings: the matching objects have the same / different azimuth pose. In BoW, bag-of-words model [28] is used to compare images. In SP, a spatial pyramid matching method [27] is used. In Color, color histogram is used. In SIFT[24], the number of matched SIFT features is used. X+Det: matching images by applying method X to match the first detected object by [2]. See Tab. 5 for a comparison.

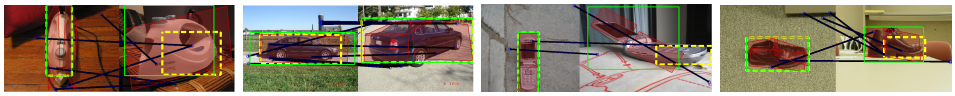
8. Su, H., Sun, M., Fei-Fei, L., Savarese, S.: Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In: ICCV. (2009)
9. Gu, C., Ren, X.: Discriminative mixture-of-templates for viewpoint classification. In: ECCV. (2010)
10. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV. (1999)
11. Ferrari, V., Tuytelaars, T., , Gool, L.V.: Simultaneous object recognition and segmentation from single or multiple model views. IJCV (2006)
12. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. IJCV (2006)
13. Hsiao, E., Collet, A., Hebert, M.: Making specific features less discriminative to improve point-based 3d object recognition. In: CVPR. (2010)
14. Rother, C., Kolmogorov, V., Minka, T., Blake, A.: Cosegmentation of image pairs by histogram matching. In: CVPR. (2006)
15. Batra, D., Kowdle, A., Parikh, D., Luo, J., Chen, T.: icoseg: Interactive co-segmentation with intelligent scribble guidance. In: CVPR. (2010)
16. Hochbaum, D., Singh, V.: An efficient algorithm for co-segmentation. In: ICCV. (2009)
17. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. IJCV (2007)
18. Ess, A., Leibe, B., Schindler, K., , van Gool, L.: A mobile vision system for robust multi-person tracking. In: CVPR. (2008)
19. Choi, W., Savarese, S.: Multiple target tracking in world coordinate with single, minimally calibrated camera. In: ECCV. (2010)
20. Bao, S.Y., Bagra, M., Chao, Y.W., Savarese, S.: Semantic structure from motion with points, regions, and objects. In: CVPR. (2012)
21. Zia, M.Z., Stark, M., Schiele, B., Schindler, K.: Revisiting 3d geometric models for accurate object shape and pose. In: ICCV Workshop on 3D representation and recognition (3dRR-11). (2011)

(a) Car dataset [4].

(b) Pedestrian dataset [5].



(c) The toaster, stapler, mouse, and bike in 3D object dataset [6].



(d) The iron, car, cellphone, and shoe in 3D object dataset [6].

Fig. 8: Anecdotal results on different datasets. Solid bounding boxes: detection results by our object co-detector applied on the image pair. Detected matching instances are shown in different colors. Dashed yellow bounding boxes: detection results by state-of-the-art detector [1] applied on each image individually. Fig. 8c and 8d: detected parts are highlighted in red. The blue lines are SIFT matches obtained by threshold test where the threshold is 0.7.

22. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR. (2006)
23. Berg, A., Berg, T., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: CVPR. (2005)
24. David, G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
25. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing (2004)
26. Toshev, A., Shi, J., Daniilidis, K.: Image matching via saliency region correspondences. In: CVPR. (2007)
27. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. (2006)
28. Fei-Fei, L., Fergus, R., Torralba, A.: Recognizing and learning object categories. CVPR Short Course (2007)