Computer Vision and Image Understanding 139 (2015) 73–87

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Part-based deformable object detection with a single sketch☆

Sreyasee Das Bhattacharjee[a,b,*], Anurag Mittal[a]

[a] Department of Computer Science & Engineering, Indian Institute of Technology, Madras, India
[b] Department of Electrical & Electronic Engineering, Nanyang Technological University, Singapore, Singapore

ARTICLE INFO

Article history:
Received 3 July 2014
Accepted 13 June 2015
Available online 19 June 2015

Keywords:
Hand-drawn sketches
Contour-based object detection
Part-based models
Dynamic Programming

ABSTRACT

Object detection using shape is interesting since it is well known that humans can recognize an object simply from its shape. Thus, shape-based methods have great promise to handle a large amount of shape variation using a compact representation. In this paper, we present a new algorithm for object detection that uses a single reasonably good sketch as a reference to build a model for the object. The method hierarchically segments a given sketch into parts using an automatic algorithm and estimates a different affine transformation for each part while matching. A Hough-style voting scheme collects evidence for the object from the leaves to the root in the part decomposition tree for robust detection. Missing edge segments, clutter and generic object deformations are handled by flexibly following the contour paths in the edge image that resemble the model contours. Efficient data-structures and a two-stage matching approach assist in yielding an efficient and robust system. Results on ETHZ and several other popular image datasets yield promising results compared to the state-of-the-art. A new dataset of real-life hand-drawn sketches for all the object categories in the ETHZ dataset is also used for evaluation.

1. Introduction

Object detection is an important problem in Computer Vision. The basic idea is to search for an object in an image and find its boundary or bounding box given some information about the object. Such information may be in the form of a single or a set of images of the object, or even a hand-drawn sketch. There are two main approaches to this problem that have been considered in the literature: feature-based [1–5] and contour-based [6–9], although some authors have also tried to combine both these ideas [10,11]. While feature-based methods are currently most popular, they have the limitation that the different appearances and articulations of the object are difficult to model without a very large training set. For many texture-less objects (like brand logos), shape is the sole dominant discriminative feature. Additionally, as observed by Dickinson [12], though humans use many visual cues to recognize an object, shape information retrieved from image contours is sufficient for recognition; other features like color, texture, shading, and depth information are not so essential for the task. Hence, it can lead to a robust, flexible and efficient system, invariant to object appearance such as color, texture and illumination, for object detection without the need for learning from a large set of images.

However, challenges are also manifold. First, it is not so easy to distinguish the object (shape) contours from other edges due to the presence of object and/or background texture. Second, the gradient along the object boundary may become very weak in certain portions due to the presence of a matching background. Third, the object may undergo some deformations due to articulations or a change in the viewpoint. Last but not the least, there are variations within the input sketches provided and between different objects in the same object category.

In this work, we address several challenges of a contour-based methodology in a pictorial structure-like framework, which we believe have not been fully addressed before. In order to make the problem tractable, we require the user to have a rough understanding of the general structure of the object and be able to draw in a manner which is not very sloppy and indeed draw a sketch that resembles the object silhouettes. This will be discussed later at length in the paper. Our system does not solve the general problem of learning object models, but the proposed system offers a fast and flexible sketch-based class-specific modeling with minimal training. The sketch should actually be recognizable as the object silhouettes, tailored to be good for recognition by a shape matching system (as opposed to real-life sketches [13]). Apart from the application of sketch-based image retrieval using input from touch-based devices, the method can be a component of a complete contour-based

☆ This paper has been recommended for acceptance by Vittorio Ferrari.
* Corresponding author.
E-mail address: sreya.iitm@gmail.com, dbhattacharjee@ntu.edu.sg (S. Das Bhattacharjee).

http://dx.doi.org/10.1016/j.cviu.2015.06.005
1077-3142/

learning-detection system if one or multiple sketches of the object can be learnt from training data [14,15].

The rest of the paper is organized as follows: Section 2 discusses some related works in this problem domain. Section 3 describes our part decomposition algorithm. The *Coarse Matching* strategy to detect the objects in a deformable and locally affine-invariant way is described in Section 4. *Contour tracing* to verify the detection and trace the object contour is described in Section 5. Finally, Section 6 shows the results of experiments on several datasets.

2. Related work

The entire spectrum of contour-based methods is huge and can be roughly divided into two groups; sketch-based [6,16–22] and learning-based [9,23–27], although learning may also output a single or a few sketches as output and thus the two methods are not unrelated. In this paper, our focus in literature survey will primarily be restricted to sketch-based methods and a detailed study on learning-based methods is beyond the scope of this paper. Several contour-based methods [6,18,19] use a single or a few sketches as input models for object detection. Ferrari et al. [18] process an image to create a contour segment network (CSN) and find paths in an image resembling the contour chains in the object sketch. Bai et al. [19] search for image contours within a certain bandwidth of the object contours using a shape-specific window called 'ShapeBand'.

Various shape descriptors, especially the edge-based Shape Context [28], have also been used for this problem for faster retrieval. Thayananthan et al. [6] use an improved version of Shape Context [28] to incorporate information regarding edge orientations and continuity in a Dynamic Programming framework that finds the best path in the contour network. Lu et al. [29] have developed a shape descriptor based on a 3D histogram of angles and distances for pairs of three consecutive sample points along object contours. Given a model sketch, the proposed method defines a 'Part Bundle' to represent the different poses of an object part. Such an AND/OR model of object representation can be used with a variety of matching methods including ours. Recently, Donoser et al. [16] and Riemenschneider et al. [17] have proposed a shape descriptor by analyzing the angles created by pairs of three sample points along the contour. In another recent work, Ma and Latecki [21] have proposed a model-based partial shape matching scheme based on a shape descriptor similar to Shape Context, where the inference is drawn from a maximum clique on a weighed graph. Toshev et al. [22] propose a global boundary-based shape descriptor called 'Chordiogram', capturing the geometric relationship between every pair of boundary edges called the 'chord' of a segmented test image. Object boundary information in terms of the orientation of boundary normals with respect to the interior is also captured for attaining better discriminability. However, the descriptor is sensitive to deformations, clutter and the performance is subject to the correctness of the initial segmentation. A set of recent papers [30,31] concentrates on ameliorating this drawback, proposing a technique using multiple segmentations.

While most of the above mentioned methods use only a single-level object model and the deformation is modeled with respect to a global positional co-ordinates such as the centroid, this limits the amount of deformation that can be handled. Thus, there is a growing interest in part-based object representation [10,32,33], which can handle more local shape variations within a model structure. Each part can be matched separately, while retaining the global shape description using more sophisticated representation schemes. Although such representations have been used so far mostly for feature-based approaches only, we inherit the concept of Pictorial Structures in sketch-based models using contour-based cues. The original Pictorial Structure (PS) was proposed by Fischler and Elschlager [34] and later its efficient computation method was proposed by Felzenszwalb and Huttenlocher [35]. Ronfard et al. [36] extended the original approach without needing background subtraction by relying on a discriminative appearance model. Ramanan [37] extended the above approach with a concept of *Image Parsing*. Robust part templates are discriminatively learnt [38] using an iterative parsing approach. In the first iteration, inference is drawn using only generic edge models as unary potentials. The resulting pose is used to build case specific appearance models and inference is drawn repeatedly using both edges and appearance terms. In another recent work, Ferrari et al. [39] proposed an extension by integrating features from an automatic foreground segmentation step (called *Foreground Highlightening*) for improved performance. Many extensions of the original PS-based method [34] have also been considered before [6,35,40], which typically use various features like shape context [41], HOG [42] or image intensity information [37] to determine the object pose. Boosted classifiers or SVMs are then typically used for learning the models.

In the proposed work, one or a handful few (possibly) hand-drawn sketches are used as the model(s), which can preserve the significant amount of discriminative object structural information required for a reliable detection. The given sketch model is segmented into multiple parts, which are expected to capture certain genuine object parts and thus having higher semantic significance within the part structures. By semantically significant parts we plan to identify those parts, which are likely describing certain actual parts of an object and therefore is expected to be aligned to the choices of a human observer. The proposed part-based object representation scheme is inspired by the part-segmentation technique proposed by Gopalan et al. [43] and proposes an improvement toward automation. Each part is segmented into contour fragments and represented using affine-deformable *connected segment pairs* (*CSP*) for robust matching in an image. A bottom-up approach allows us to handle more amount of deformation, especially at the *joint points* that connect two neighboring parts and about which the parts are allowed to rotate. The initial *Coarse Matching* stage, which works as a rough hypothesis generator, identifies each of the parts in a locally affine-invariant way. The constituent *CSP*s representing a part, contribute their individual estimates which are accumulated using Hough-style voting to identify candidate part locations (*peripherals* or *Root*). This allows us to handle occlusions better. The initiatory coarse level search process is further followed by a more extensive *Contour Tracing* stage that allows for an exhaustive verification at a finer level by following the maximum gradients along the contour direction. Thus, the proposed method can handle the challenges posed by clutter more effectively. The entire method is diagrammatically explained using a flow chart in Fig. 1. For evaluation purposes, we have created and tested our system on a new dataset of hand-drawn sketches for the commonly used ETHZ dataset of images. It has also been evaluated with better computer-assisted object models ETHZ dataset, INRIA Horse and Weizmann Horse datasets.

3. Sketch decomposition and representation

Given an object category, one or a few sketches are used as its shape representative models. Some examples of such shapes can be seen in Fig. 2, Fig. 5 or in Fig. 16. Such shapes may either be manually obtained in terms of sketches from some random users, semi-automatically using segmentation algorithms [44] or fully automatically via learning from training images [14,15,24]. Given this sketch, a part-based tree-like structure is built for shape representation. Each part is further segmented into contour fragments, which are later combined to represent the given part into a collection of affine-deformable *"Connected Segment Pairs"* features, suitable for robust matching in an image.

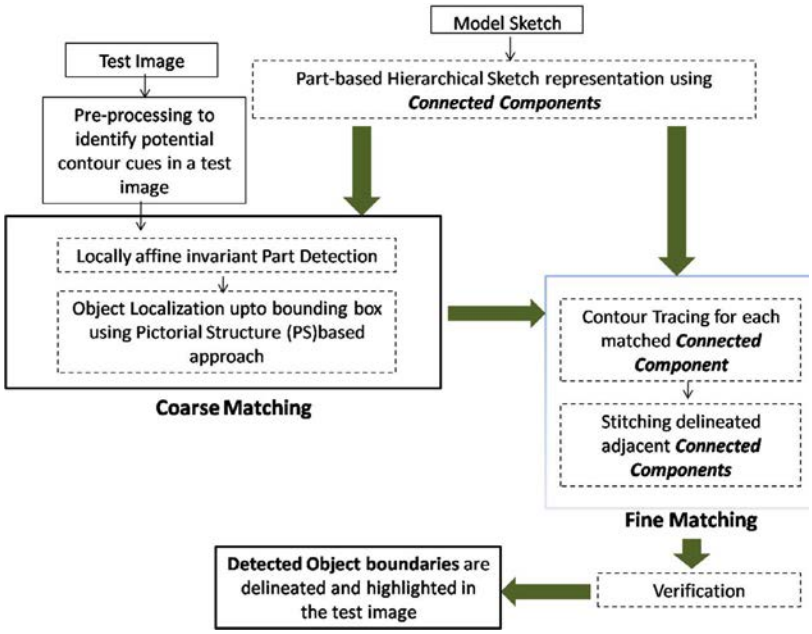

Fig. 1. An overall flowchart for our proposed method for object detection.

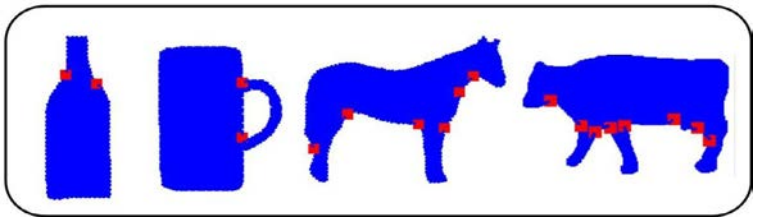

Fig. 2. Examples of some detected *concave* points, with a convexity ratio 0.8. Note that some points such as the ones near the horse's ears were not detected as they did not pass this threshold.

3.1. Decomposition of a sketch into parts

Part-based models have become increasingly popular in recent years [40,45,46], primarily due to its effectiveness to deformations which are frequent in real life images. Such models include trees [35], star graphs [32,47], k-fans [40] or fully-connected graphs [1]. Since a fully automated process for obtaining such part-based models is difficult, recently, Branson et al. [44] attempt to reduce manual intervention using a 'weak annotation' scheme, where interactive labeling on a subset of training images is used along with the other unlabeled images for learning the model. The basic assumption in most of these cases is that an object can be represented in terms of a collection of local templates that deform and articulate with respect to one another, while maintaining a good amount of independence of structural variations within each of these templates. In contrast to the established methods [32,48], which attempt to learn the optimized part model of an object from a large training set, we aim to minimize this effort by proposing an automatic part-decomposition approach inspired by a normalized cut based method by Gopalan et al. [43]. As a first step, given a closed sketch,[1] we follow an area-based approach of Ling and Jacobs [49], as also used by Gopalan et al. [43], to roughly identify some significant *concave* points (also called *non-convex* points, identified based on a convexity ratio [43] of 0.8) in the sketch. Choosing a lower convexity ratio fetches more *concave* points resulting in more number of parts, while a higher ratio detects lesser number of parts For details of how to identify these highly concave points, readers can please follow [43]. Some identified *concave* points thus obtained are shown in Fig. 2.

Gopalan et al. [43] use these *concave* points to obtain a part segmentation of the object contour by performing normalized cut, that uses the pairwise inner distance as the dissimilarity score for all the points lying within the entire silhouette. Due to this, the approach is relatively slower. Moreover, it suffers from the problem of semantically incorrect segmentation as the only criteria that has been exploited in this scenario is the convexity of parts, which does not always produce genuine parts. We would like to reiterate here that by 'genuine' (or 'best') parts, we point toward those kinds of part segmentations which are consistent with human perception. Additionally, in order to achieve the best part segmentation, this system needs to be provided with a shape-specific initial user estimate for the number of parts n. As observed in Fig. 3(a), no single estimate for n fits uniformly well for all the ETHZ shape categories. Fig. 3(b) shows some results of part-segmentation on ETHZ dataset using the automated approach described next in this paper.

We improve upon Gopalan's algorithm to not only remove this a-priori specification of the number of parts but to also amend upon the part decomposition. Fig. 3 compares our part-segmentation result with the segmentations obtained using Gopalan's approach.

Though convexity is one of the main discriminative features for shape representation and identification, it is not the only cue that humans utilize for part segmentation. As suggested by Ling and Jacob [49], several other contextual factors are involved in determining a part cut. Therefore, we do not follow only a convexity criteria for segmenting parts and allow for more flexibility by exploring a

[1] In the case of open sketches, we connect any open ends to close-by open ends with straight lines. These additional lines are used only for convexity computation in the first stage and not for any other analysis.

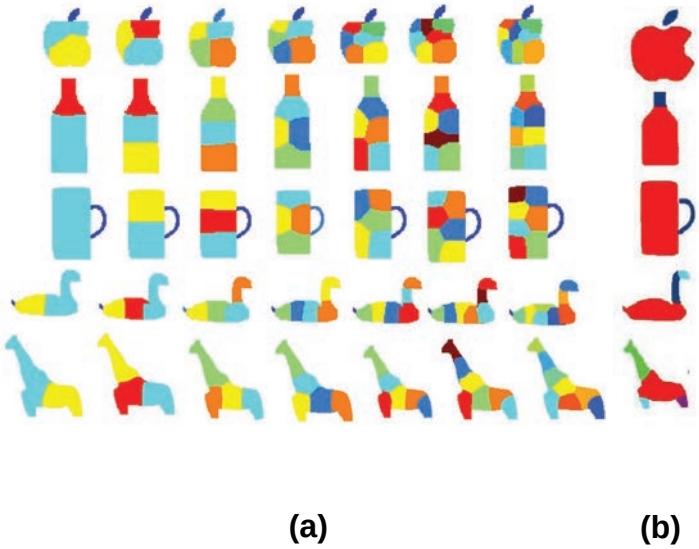

Fig. 3. (a) Results of part decomposition using the method proposed by Gopalan et al. [43]. Each row represents the results for a specific shape as the user estimate for the number of parts varies from 2 to 8. (b) Results using the proposed part-decomposition method on the sketches from the ETHZ.

close-by neighborhood around a smaller set of identified concave points and proposes to generate a part using the concave point paired with a suitably chosen neighboring silhouette point (called cut-point), which may or may not have a significant high curvature value. We have found that such extracted parts are semantically more meaningful in nature and perform better in practice than the ones obtained from a pure convexity criteria.

As illustrated in Fig. 4(a), given a *concave* point p, its paired *cut-point* was identified on the shape silhouette using a function d_p, defined in a close circular neighborhood (having a radius of 10 for a model image of size 357 $\times$ 216) around p, where for every $q(\neq p)$, $d_p(q)$ represents the Euclidean distance of q from p. A typical example of a smoothed d_p is shown in Fig. 4(b). The local-minima of d_p (q_{min}, as marked in 'red' box) having the sharpest dip observed in d_p function, yields a first guess for the paired *cut-point* p_n of a part. If another high curvature point p_h, concave or convex, is found (using a standard contour corner detector such as [50]) in a close (similarly defined as above) neighborhood N_{p_n} of p_n, then this matched *cut-point* p_n is shifted to p_h. Otherwise, this point is retained as it is, even if it is in the center of a straight line. It is important to note that the *cut-point* is characterized by a peak or a trough of d_p defined in a pre-defined neighborhood of p, and thus is conceptually different from the canonical nearest neighboring high curvature point. As can be seen in Fig. 4(a), the nearest neighboring high curvature point (H) to p does not correspond to a peak or a trough in d_p and thus failed to coincide with the chosen paired *cut-point* to p. The set of *concave* and *cut-point* pairs obtained for Giraffe shape is shown in Fig. 4(c). In another case, as shown in Fig. 4(c), given the *cut-point O* in the Giraffe's 'leg' there was no nearby high-curvature point. Hence, the initial estimate C_t for the paired *cut-point* was retained finally.

In order for sequential part decomposition, the entire set of *concave* points is sorted based on their associated curvature values and the approximately convex parts are created following that sorted order. The object is segmented at the *cut-point* pairs to obtain the segmented parts. A segmented part is treated as final if it is sufficiently long and does not carry any *concave* point within itself. Parts with one of more *concave* points, are processed again for further segmentations. Parts which are not reasonably complex according to the length criterion (less than 50 contour pixels, for a standard model size) are merged with one of its neighbors.

The entire set of parts is classified into two categories, *Root* and *peripherals*. Parts are connected to each other in a tree-like structure, where each part acts as a node connected to its neighbors based on an adjacency criterion. The part having its center of mass closest to the object centroid is defined as the *Root*. A comprehensive position estimate is finally obtained with respect to this part. All other parts directly connected to the *Root* form the first layer of *peripherals*; parts connected with parts in the first layer form the second layer of *peripherals* and so on. If a part is found to have more than one higher order *peripheral*, the cycle is broken at the connection where the inter-part distance (which is the Euclidean distance between the centroid of the two adjoining *peripherals*) between two adjoining parts is higher. The resulting unique higher order *peripheral* part (or *Root*) is known as the 'parent' to all the parts (treated as its children) at the next layer with which it is connected.

The connection between a part and its parent is identified in terms of a *joint point*, which is taken to be the mean of the *cut-point* pairs. The proposed part-representation scheme allows relative deformation of the parts about this *joint point*. The resulting part-based tree-like structure is shown pictorially in Fig. 5.

3.2. Decomposition and representation of a part

The part representation process is initiated with the decomposition of each segmented model-part into a collection of straight line-like segments, which later work as the primitives for *Connected Segment Pairs*.

Each *CSP* is essentially a pair of adjacent segments linked together at the highly discriminative corner points [50] and therefore can be defined by a tuple of three control points (p_1, p_2, p_3) (consisting of the two end points and the join of the constituting segments). By allowing flexibility at the join of two adjacent segments, one can achieve a good amount of deformation within a part. At the same time, such local features are more robust to missing segments in an image. However, retaining a very long edge segment is undesirable as it is prone to having a different deformation in its different parts. As shown in Fig. 6(a), each side of the 'Mug' model looks more or less a perfect straight line. However, the sides of a 'Mug' instance can also be curved. Hence, longer model segments are recursively segmented into n (we chose $n = 2$) equal smaller sub-segments until any of the newly generated subsegments is below this size-threshold. The structure of *CSPs*, created using pairs of adjacent segments (not sub-segments) is not affected by the internal variation within sub-segments.

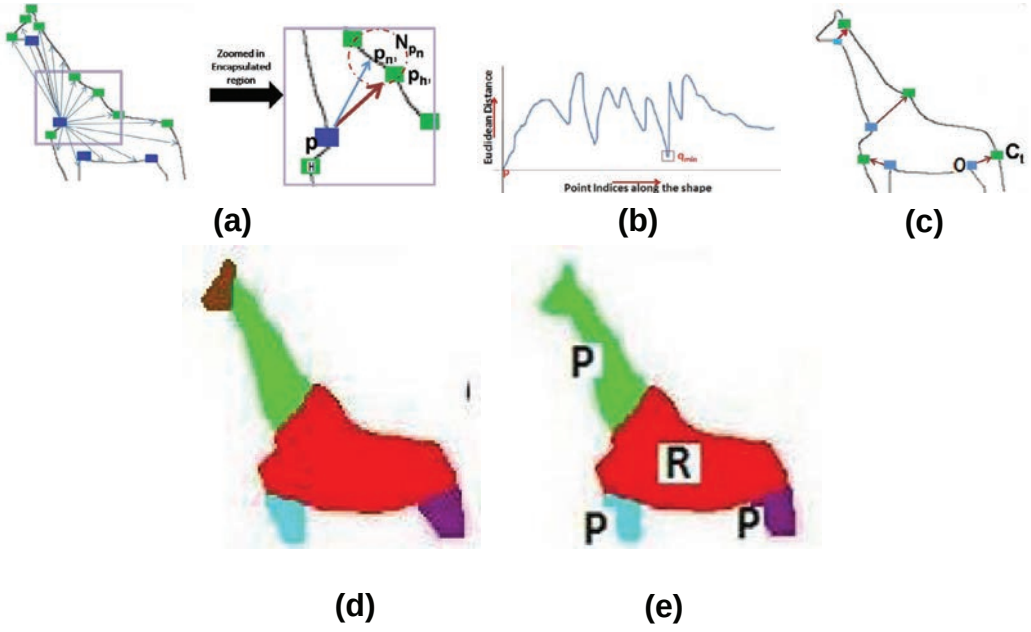

Fig. 4. The process for finding the paired *cut-point* for a *concave* point: (a) shows the identified *concave* points in 'blue'. Given every *concave* point, the zoomed-in region illustrates the method for obtaining the paired *cut-point*, (b) a typical example of function d_p used to find the paired *cut-point*, (c) shows the initial *concave* and *cut-point* pairs, each pair is shown with a pair of 'blue' and 'green' point linked with an arrow, (d) shows the resulting part decomposition obtained. The part in 'brown' was found to be very small and hence was merged with its adjacent part in 'green' and (e) shows the final part decomposition result after merging, where 'Root' and peripherals are identified as R and P respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

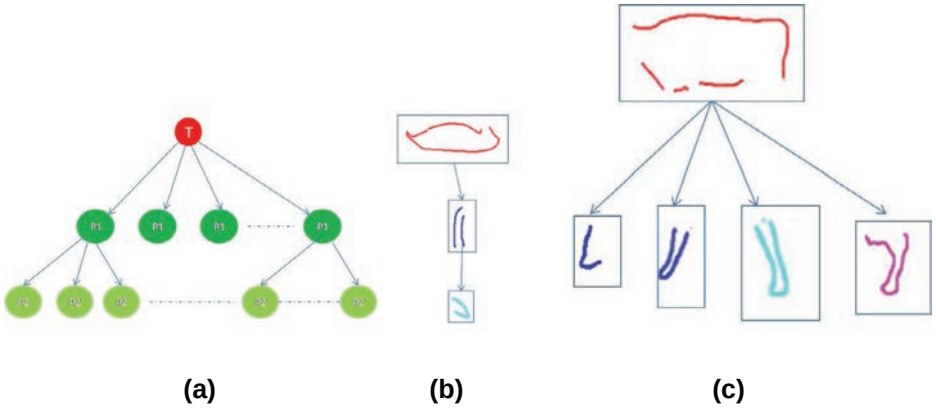

Fig. 5. (a) Tree-shaped part hierarchy. (b) and (c) The part decomposition tree derived for 'Swan' and 'Cow' sketches.

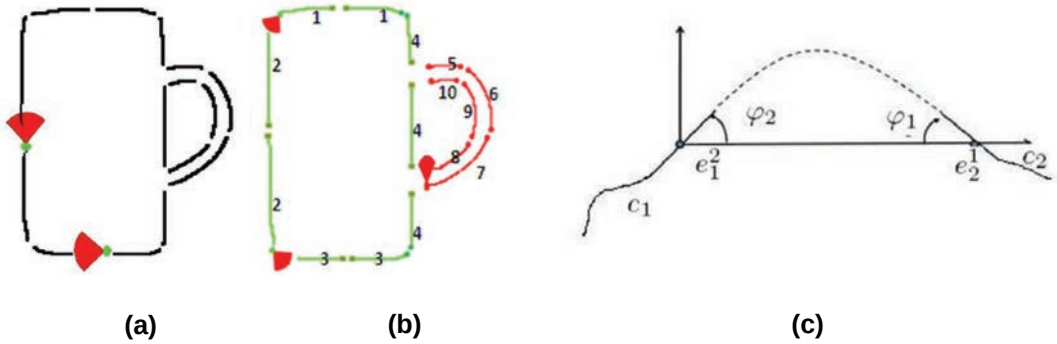

Fig. 6. Neighborhood search for (a) subsegment matches and (b) segment matches for the *CSPs* (1,2),(2,3) and (7,8), (Elastica Completion Cost measured using contour continuity at the adjoining end points of two subsegments).

Given these segments and subsegments of a part, *CSP* ensures a model representation scheme that can deal with local shape deformation and partial occlusion of a part. Simultaneously, it also helps to deal with clutter and edge noise as a certain amount of global shape perspective is retained. For example, as shown in Fig. 6(b), there are four *CSPs* in the *Root*: (1,2), (2,3), (3,4), (4,1). The structure of our *CSP* is somewhat similar to that of the *k*-adjacent segments ($k-AS$) proposed by Ferrari et al. [9,51]. However, unlike *CSP*, $k-AS$ is only similarity invariant and does not allow any deformation within its structure.

Although, the proposed part-based shape representation scheme is closely related to Pictorial Structure (PS) based models, such methods typically assume rigidity within the structure of a part and allows relative deformation only at the connections between two adjacent parts. Hence, its expressive power is limited to the relative spatial arrangement of the parts. In contrast, our proposed representation scheme can also handle the deformation within each part, represented using a collection of *CSPs*.

Based on this sketch decomposition and representation scheme, we employ a two-stage process to determine the location, if any, of the object in a given image. The object bounding box estimates are identified in the first stage with a thorough verification by tracing the exact object boundaries in the second stage. In combination, the two stages yield fairly accurate object detection and boundary delineation in real-world images.

4. Coarse matching: part-based object search

The aim of this coarse-level search process is to determine a location and the corresponding matched scale of the object in an image up to a bounding box. Toward this goal, the parts are detected from the leaves to the root in the tree-like shape representation scheme, such that each leaf can rotate about the *joint point*. The different *CSPs* comprising a part vote for its *joint point* in an affine-invariant way. A similar voting for the *Root*'s centroid determines its position. Finally, the matched parts are combined following the tree-structured deformable model representation scheme to evaluate an object location. Such a part-based bottom-up search strategy is efficiently implemented using an approach similar to Dynamic Programming-based Pictorial Structure (PS) approach by Felzenszwalb and Huttenlocher [35]. However, as opposed to prior methods, we address an affine change of a part by matching the *CSPs* in an affine-invariant way and then transferring the estimated *joint point* to the parent part using the affine transformation thus estimated.

The first stage in *Coarse Matching* is to detect each individual segmented part.

4.1. Part detection

4.1.1. Pre-processing the test images

The first step in our processing is to detect edges in the test image. We use the Berkeley edge detector [52] which combines color, brightness and texture cues to provide a probabilistic edge map, where for each pixel in the image, a probability for being an edge is computed. Hysteresis thresholding followed by efficient contour grouping as proposed by Zhu et al. [53] yields a set of salient contours derived from the image which is used as an input to our multi-stage matching process.

Search is initiated for each part independently, which asks for first identifying a set of possible locations for the basic part-primitives, called sub-segments.

4.1.2. Subsegment matching using fast directional chamfer matching

The recently proposed robust fast directional chamfer matching (FDCM) [54], due to its proven effectiveness in a clutter intensive scenario, was used for matching purpose.

Given a model subsegment, the local maxima (using FDCM score M_{dc}) of its matches in an image are determined by non-maximal suppression and identified by the location of their mid-points and the two end-points, along with other details about matched scale, rotation angle etc. This step is the most computationally expensive part of the entire algorithm since it needs to be done for each subsegment separately. However, using contour breakup into linear structures, the integral image concepts in FDCM [54] and a Dynamic Programming based approach has made the entire process efficient. In addition to it, an inverted-file indexing scheme can be incorporated for real-time matching, which we have omitted in our present proof-of-concept (PoC) implementation phase. In the next step, we try to find each segment as a combination of subsegments.

4.1.3. Search for segments

Each segment with k subsegments consists of a sequence $(\{ss_i\}_{i=1}^{k})$ of these subsegments. Each subsegment ss_i can be matched against several matches ss_i' in an image. Following a Dynamic Programming based method, potential matches are stitched together following a cost function defined in a neighborhood (see Fig. 6(a)) to attain the globally best set of matches for the entire segment. This exhaustive search process is efficiently implemented using geometric data structures such as Range Trees [55]. The cost of matching a sequence of $l(\leq k)$ subsegments from the model to an image is defined in terms of a weighted combination of four constituent costs:

$$C(l) = \sum_{i=1}^{l-1} (w_s C_s(i) + w_a C_a(i) + w_{gap} C_{gap}(i) + w_{el} C_{el}(i) + w_{CM} C_{CM}(i)) \quad (1)$$

where $C_s(i)$ and $C_a(i)$ are the Scale and Angle Dissimilarity Costs between ss_i' and ss_{i+1}', $C_{gap}(i)$ is a cost based on the Euclidean distance between the two near endpoints of these adjacent subsegments at their join and $C_{el}(i)$ is an elastica cost that depends on the continuity of the contour curvature at the join (recall again we broke the segment at a random point without any appreciable curvature and this property must be preserved in the matched contour as well). Finally, $C_{CM}(i)$ represents the corresponding FDCM cost for the ith subsegment. w_s, w_a, w_{gap}, w_{el} and w_{CM} are the corresponding weights (for our experiments, these were chosen as equal) of these five costs and could possibly be learnt automatically from training data.

While we expect a uniformity of scale estimate for the subsegments within the same segment, following [18] the Scale Dissimilarity Cost is defined as:

$$C_s(i) = 1 - e^{\left(-max\left(\frac{\sigma_{m,t}(i)}{\sigma_{m,t}(i+1)}, \frac{\sigma_{m,t}(i+1)}{\sigma_{m,t}(i)}\right)\right)} \quad (2)$$

Similarly, the Angle Dissimilarity Cost C_a is defined as:

$$C_a(i) = 1 - e^{-((\Delta\alpha_{m,t}(i))/\pi)^2} \quad (3)$$

where $\Delta\alpha_{m,t}(i) = \alpha_m(i) - \alpha_t(i)$ such that $\alpha(i)$ is the relative angle between the ith and the $(i+1)$th (matched) subsegments.

Next, the gap cost is defined as:

$$C_{gap}(i) = 1 - e^{\left(-\frac{\Delta d_{m,t}(i)}{n}\right)^2} \quad (4)$$

where $\Delta d_{m,t}(i) = (-d_t(i)/s_{avg,t}(i))$ is the scaled distance between the two adjacent endpoints of the two subsegments ss_i' and ss_{i+1}' that should ideally have zero distance between them as per the model (note that there is no gap between two subsegments in the model). The distance is normalized by $s_{avg,t}$, the estimated average scale of the subsegment set matched so far, and n, an appropriate normalization constant (we use $n = 10$ for a normalized model of size 357×216).

The Elastica term C_{el} measures the angle alone estimated from the start and end points of the subsegments while the Angle Dissimilarity

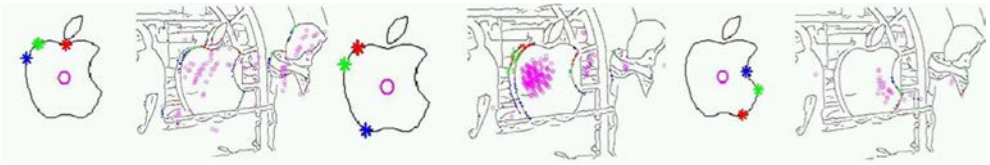

Fig. 7. Voting using *CSPs*: The three control points of a *CSP* are shown in 'red', 'green' and 'blue' in the model (left) and image (right). The centroid location and its votes in the image are shown in 'magenta' circles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Cost term looks at the continuity of the curvature within a single segment and thus is a more local phenomenon. Given two consecutive tangent directions, ϕ_1 and ϕ_2 (see Fig. 6(c)), as shown by Kokkinos and Yuille [46], the quantity $El(c_1, c_2) = 4(\varphi_1^2 \times \varphi_2^2 - \varphi_1 \times \varphi_2)$ provides a good measure for curvature inconsistency at this stage. This Elastica energy ($El(i)$) between $ss_i^{'}$ and $ss_{i+1}^{'}$ is used to define the Elastica Contour Completion cost as:

$$C_{el} = 1 - max(0.5, e^{-El(i)}) \tag{5}$$

Note that the use of the difference between the relative angles rather than the absolute angle in the cost components makes our method robust to rotations as all the terms including the component taking care of the FDCM dissimilarity are adjustable to an overall rotation of the part. Given these matched segments, we combine them to form *CSPs*.

4.1.4. Search for connected segment pairs

Given segments with their adjoining endpoints close to each other, one can find matched *CSPs* in the same way as we searched for the segment matches using subsegment matches (see Fig. 6(b)), however with lesser constraints and allowing for more flexibility.

Recall that each *CSP C* consists of a triplet of its three control points (p_1, p_2, p_3) in the sketch which are matched to a triplet of points $(p_1^{'}, p_2^{'}, p_3^{'})$ in the image. The correspondence $[p_i \leftrightarrow p_i^{'}]_{i=1}^{3}$ representing a matched *C*, defines an affine transform. This is compared with the affine transform of the matches for its adjacent *CSPs* and the match is retained only if there exists a corresponding close-by match with a similar affine transform. Therefore, given a matched *CSP* (C) and its estimated affine transformation (A_C), the cost corresponding to its neighboring matched *CSP*, $C_1([q_i \leftrightarrow q_i^{'}]_{i=1}^{3})$ is evaluated using the following dissimilarity measure:

$$C_{af} = \sum_{i=1}^{3} \left\| (A_C \cdot q_i) - q_i^{'} \right\| \tag{6}$$

The total cost of a matched *CSP* is defined as the average of the gap cost component from Eq. (4), local affine inconsistency based on Eq. (6) and an average FDCM cost of its constituent subsegments. Thus, an overall structural inconsistency is measured.

4.1.5. Estimating the candidate part locations

Given such matched *CSPs* and their associated affine transformations, one can determine the location in the image of the desired point – the *joint point* in the case of *peripherals* and the *centroid* in the case of the *Root*. A Hough-style voting is employed to collect votes for this desired point, where the votes are weighted by a score based on the average FDCM matching score of the constituent subsegments and the sum of all the costs encountered in the *CSP* generation i.e. subsegment, segment and *CSP* costs for Scale, Angle, Gap, Elastica and Affine Dissimilarities. This voting process is illustrated in Fig. 7.

Integral image concept [56] is used to smoothen the vote response using a 3×3 box filter. The local maxima in the resultant map yields matches for the desired part. At this stage, we also screen the matches to retain only the ones that got votes from a sufficient number ($> 1/2$) of the *CSPs* constituting a part.

4.2. Estimating the object centroid using Dynamic Programming

Given independent estimates for each part of an object, we combine such estimates using relative positional constraints from the lowest level of the shape representative tree to its root (i.e. the *Root*) in a manner quite similar to the 'Pictorial Structures' approach of Felzenswalb and Huttenlocher [35]. Given a parent's location and affine transform estimate, the location of its *joint point* with respect to a particular child can be predicted. Then, the total cost at different locations of the parent can be estimated as:

$$TC_p(L_p) = C_p(L_p) + \sum_{c \in childs(p)} \min_{L_c} (TC_c(L_c) + V_{pc}(L_p, L_c)) \tag{7}$$

where, where L_p and L_c denote a matched location for the parent and child part respectively and $V_{pc}(L_p, L_c)$ enforces the relative positional constraints, known as the pair-wise potential function in PS literature. This function is taken as a simple Euclidean distance between the predicted and the actual position of the (*joint point* of the) child, multiplied by a constant. $C_p(L_p)$ and $TC_p(L_p)$ are the individual and total costs respectively of a particular part. The candidate locations of the *Root* (see Fig. 8(b)) can be determined by thresholding and non-maximal suppression to extract only the best detection in a region. Given a valid object hypothesis at a location L_r of the *Root*, we can detect the best locations of all the visible *peripherals* parts.

In contrast to the original PS-based approaches, each approximately convex part is capable of handling a certain level of deformations within its structure by using a small collection of locally affine-invariant deformable *CSPs*.

This whole *Coarse Matching* process is pretty fast and yields a few bounding boxes, as shown in Fig. 8(c). The original PS approach assumes that the time taken for detecting each part is constant, which may not be the case in general. Therefore, following PS-based approach, the time complexity for detecting a part is linearly proportional to the number of its constituent *CSPs*. The entire *Coarse Matching* process can therefore be performed in $\sum_p O(h_p n_{c_p}) + O(hn)$, where h_p represents the number of possible locations for part p with n_{c_p} number of *CSPs* and h represents the total locations of all the parts.

However, there can still be some errors as only the individual subsegments and segments are matched and fragmented edges in a highly cluttered domain may yield a false match. In order to improve matching, we attempt to do a detailed contour trace in these matched bounding boxes.

5. Contour Tracing and object verification

The proposed Contour Tracing approach is primarily motivated by the method proposed by Ravishankar et al. [8] and can be treated as an improved version of their algorithm by employing a more robust Dynamic Programming-based matching strategy using a small set of more generic and comprehensive cost functions. The exhaustive Contour-Tracing stage thus greatly enhances the accuracy of our algorithm and false positives detected during the first stage are effectively screened out. At the same time, it is quite efficient since this search is done in a few small windows only.

A set of candidate locations (up to bounding boxes) detected by the previous stage, are further explored to trace the contours for each

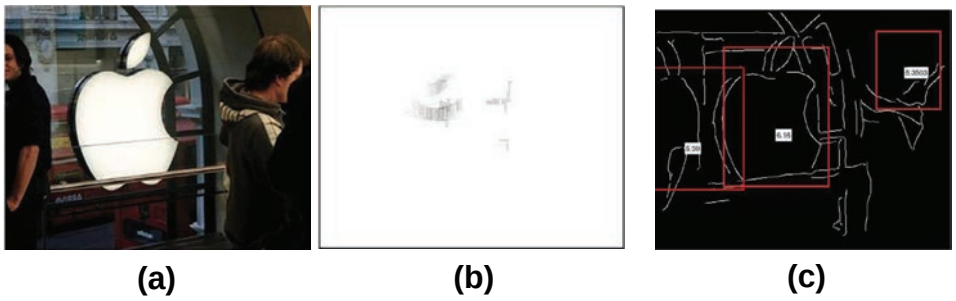

Fig. 8. (a) A test image. (b) The point map TC_{Root} corresponding to its integrated goodness value at every pixel in the image. (c) Few bounding boxes obtained from *Coarse Matching*.

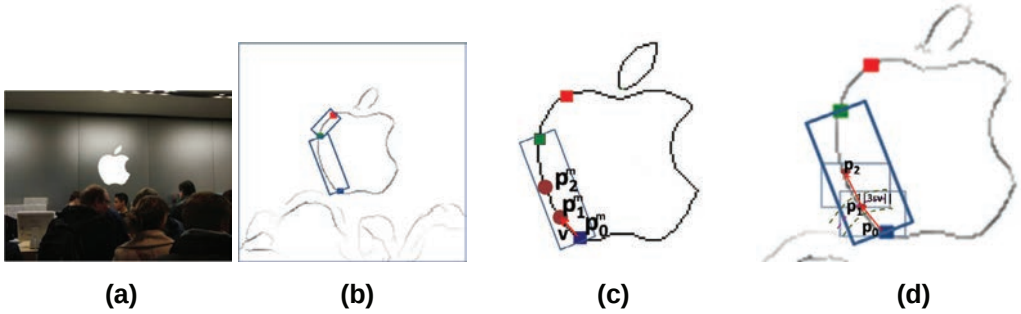

Fig. 9. (a) A test image. (b) Oriented boxes for the two segments of a *CSP* in the edge map of (a). Tracing the contour of a particular segment using search in an oriented box. The model contour is highlighted using an oriented box in (c) while the trace of the matched contour is highlighted using a similar box in the test image (which is a zoomed in version of (b)) in (d). The points p_i along the path are found in the image (in (d)) similar to such points p_i^m in the model contour (in (c)).

of the *CSPs*. The traced contour-lets are then connected with the contours of the other *CSPs* and other parts wherever possible. Search for tracing a *CSP* can be started from any *CSP* and is repeated for all *CSPs* that have not been found.

5.1. Contour Tracing of a matched connected segment pair

Given a particular bounding box for an object match, one can identify the matches for each of the *CSP* that contributed to this object match. In some cases, there might be multiple such matches and all such matches are considered. Now, in order to trace a matched *CSP*, an oriented box is placed at every pair of the control points $((e_1, e_2))$ of each of its two segments at its matched location (see Fig. 9(b)), where the width of the box depends on the locally estimated affine transform with some small leeway for deviations.

Given the sampled *CSP* in the model sketch, within each oriented box, starting from the first endpoint e_1, explore a small neighboring (appropriately scaled) patch around it to find a small set of best locally maximal next points satisfying certain cost constraints (Fig. 9(c) and (d)). This process is followed in an iterative manner using Dynamic Programming to form chains that are best in terms of a total cost of the chain so far. Local maximality in path selection ensures distinct paths and the paths are evaluated with the following cost function:

$$q = w_m C_m + w_e C_e + w_{el} C_{el} \tag{8}$$

where C_m, and C_{el} are the costs for the consistency with the corresponding model segment and the Elastica energy describing the contour curvature continuity respectively. C_e is a normalized Edge Strength Cost defined as the sum of the Berkeley gradients [52] along the Contour path normalized by the average of such gradients in the oriented bounding box of the segment. The w's are the corresponding weights determined empirically as $w_e = 0.5$, $w_m = 0.25$ and $w_{el} = 0.25$ in this work.

The first component in this equation ensures that we move along a partially explored path $(\{p_i\}_{i=1}^{i_0})$ similar to that of the model segment $(\{p_i^m\}_{i=1}^{i_0})$ (see Fig. 9(c) and (d)). In order to ensure an approximate similarity between matched path between e_1 and e_2, and the model path between the corresponding end-points e_1^m and e_2^m a matching cost is defined as:

$$C_m = 1 - max\left(0.25, min\left(\frac{\sum_{i=1}^{i_0} \|p_i - e_2\|}{\sum_{i=1}^{i_0} \|p_i^m - e_2^m\|}, \frac{\sum_{i=1}^{i_0} \|p_i^m - e_2^m\|}{\sum_{i=1}^{i_0} \|p_i - e_2\|}\right)\right) \tag{9}$$

which keeps the traced points at a distance approximately close to their expected distance from the end-point e_2. This allows for quite a bit of flexibility in the shape change of the image contour while still giving an overall shape perspective. A maximum value of 0.75 for C_m ensures some further flexibility. $\|e_1 - e_2\|$ works as a normalizing factor.

5.2. Stitching CSPs

Once the individual *CSPs* have been traced, we try to stitch them together for further confirmation of object continuity. This is done in both directions (Fig. 10).

The cost function for evaluating the extended contour at every iteration is computed as:

$$Q = w_{af} C_{af} + w_e C_e + w_{gap} C_{gap} \tag{10}$$

where C_{af} is the Affine Dissimilarity Cost (Eq. (6)), C_e is the Edge Strength Cost and C_{gap} is the Gap Cost (Eq. (4)) of the updated contour. We take a similar set of weight values as before ($w_a = 0.5, w_e = 0.25$ and $w_g = 0.25$).

The cost at the end of every iteration is compared with a threshold and the paths below this threshold are discarded.

5.3. Verification

Given such contours matched in the image, it is possible that they are still fragmented due to lack of gradients in certain regions of the

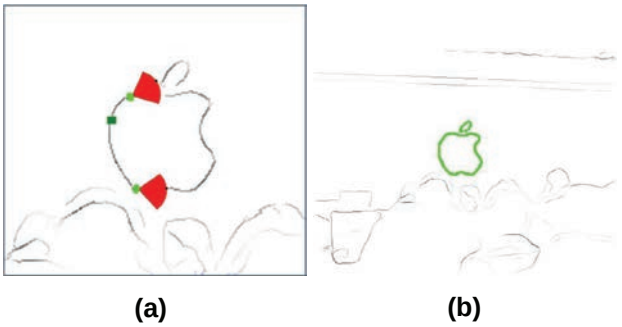

Fig. 10. (a) Stitching *CSP* using neighborhood search. (b) The final detected object contour.

object, possibly due to occlusions or a matching background. In order to deal with this situation, we retain not only the largest contour chain but also some smaller chains. Using this cleaner edge map containing all the traced contours, we again run our part-based Matching algorithm (Section 4) to verify and detect the final object along with its contour in the image. As shown in Fig. 11, two groups of matched contours obtained at the end of the Contour Tracing stage were verified to identify the true detection.

6. Experimental results and discussion

We evaluated the performance of our system on ETHZ, INRIA Horse and Weizmann Horse datasets. While INRIA horse and Weizmann horse datasets are suitable for investigating performance of deformable articulated objects with other factors being mostly constant, the ETHZ dataset has challenges due to the presence of multiple object instances at different resolutions, orientation changes, occlusions, deformation, textured and cluttered backgrounds.

We use a standard PASCAL criteria that treats the detection of a bounding box as correct when its intersection-over-union ratio (IoU) with the ground truth bounding box is more than 50%. While some of the prior works also report results at 20%-IoU, we were able to get comparable performance at even 50%-IoU.

6.1. Comparative study using the ETHZ dataset

The ETHZ is a popular dataset on which sketch-based object detection algorithms have been tested in the past. It consists of 255 total images in five categories: Applelogo, Bottle, Giraffe, Mug and Swan. The objects in the images are at various scales, illumination and clutter conditions, although orientation and rotation changes are limited. The standard ETHZ sketches were used as input models.

6.1.1. Automatic part-decomposition using shape cues

We first evaluate our proposed automatic part-decomposition scheme against a closely related approach proposed by Gopalan et al. [43]. In order to automate the choice for the number of clusters (n), we propose a heuristic formula that uses the number of *concave* points (n_c) to automatically estimate a value for n. Intuitively, in the best case scenario all the high curvature points on a shape can be *concave* in nature, n_c. In such cases, each part originate from a pair of *concave* points (which we call *cut-point* pairs), resulting in $n_c/2$ parts. With such a motivation, the best choice for n was experimentally found to be $max(2, n_c/2)$. Obtained Fig. 3(a), the part-decomposed shapes shown in Fig. 12 are intuitively some of the best part decompositions for each object sketch

As can be seen in the last two rows of Table 1 (showing the comparative category-wise detection rates at 0.4 FPPI and 0.3 FPPI on the entire ETHZ dataset), using the part structure extracted by Gopalan et al. [43], the resulting detection rates remain unaffected in 'Mug' and 'Giraffe' categories. However, our proposed part-decomposition strategy reasonably outperforms [43] for categories like 'Applelogo', 'Swan' and 'Bottle'. Some examples of failure cases for [43] are 5(e) and 6(e) of Fig. 13, where [43] fails to detect the object but the proposed method succeeded. Such improvements are attributed to the fact that the part-segmentations obtained by Gopalan et al. [43] were not very good for these three categories, while being reasonably good for others.

6.1.2. Coarse-level detection

This set of experiments attempts to evaluate our proposed *Coarse Matching* in terms of its accuracy in making an initial guess. The goal is primarily two-fold. First, a feature level comparative evaluation of our proposed *CSP* to a similar contour-based feature $k-AS$ as proposed by Ferrari et al. [9,51]. Second, comparing the coarse-level detection performance with some of the other established methods in the literature.

In order to perform the feature level comparative study, $k-AS$ is also used for describing the part-segmented (using our proposed part-decomposition scheme) sketches. Since our proposed feature *CSP* was found to be somewhat similar to $k-AS$, more specifically for the value $k=2$, we replace the *CSPs* with $2-AS$ to represent each object part and repeat the entire search process in the whole dataset. As can be observed in Table 2 (displaying the coarse-level detection rates at 1.0 FPPI), the results deteriorate in all the categories, However, for objects such as 'Bottle', 'Mug' and 'Giraffe', which have many long straight line-like segments in their models provided, the difference was more visible and the *CSPs* were found to perform considerably better for those object types. This improvement can be attributed to the fact that the *CSPs* can handle local affine transformation and a good amount of deformation within its structure by means of some flexible joints of its constituent adjacent sub-segments and segments. On the other hand, $2-AS$ was found to be more rigid in nature. It is important to note here that, in order to handle deformation, Ferrari et al. [9,51] learnt a good set of shape variants in terms of a large codebook. Therefore, in absence of an intensive learning, the performance was affected due to the lack of sufficient model information.

The coarse-level detection performance of our proposed approach was compared with some of the established methods [17,24,57,59] in the literature, which offer an object detection scheme with a similar prefixing 'hypothesis drawing' phase. Due to the lack of the results existing for the initial hypothesis generation (*Coarse Matching*) stage, we performed the comparative study of the detection rates at 1.0 FPPI,

Fig. 11. (left) A test image. (right) Two traced contour groups for the object bounding boxes detected at the first stage. The first detection was discarded while the second was taken as a true detection by the verification stage.

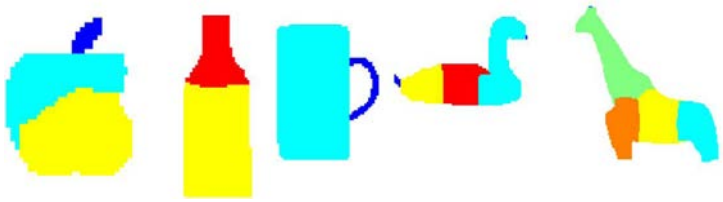

Fig. 12. Part decomposition results using a modification to the approach proposed by Gopalan et al. [43] by setting the number of parts or $n_c/2$, where n_c is the number of concave points on the shape.

Table 1
Comparison of detection rates (%) at 0.4 FPPI/0.3 FPPI for the ETHZ dataset using the original sketches at the PASCAL overlap criterion of 50%. The experimental setups are defined as; Setup-1: the proposed two-stage search approach and NO Verification, Setup-2: our full system and PD by Gopalan et al. [43], Setup-3: our full system and PD described in Section 3.1). Bold values represent the best result obtained for each category. For example, bold numbers in the first, second and sixth rows of column 1 show the best result for Applelogo category using the learning based models. On the other hand, bold values in the last row display the best result among the methods using standard sketches as model.

Ref.	Applelogo	Bottle	Giraffe	Mug	Swan	Average
Learning-based model						
Maji and Malik [57]	**95.0/95.0**	96.4/92.9	**89.6/89.6**	**96.7/93.6**	88.2/88.2	93.2/91.9
Felz et al. [47]	**95.0/95.0**	**100/100**	72.9/72.9	83.9/83.9	64.7/58.8	83.3/82.1
Ferrari et al. [9] (20%-IoU)	86.4/84.1	92.7/90.9	70.3/65.9	83.4/80.3	93.9/90.9	89.02/82.42
Ferrari et al. [24]	83.2/77.7	81.6/79.8	44.5/39.9	80.0/75.1	70.5/63.2	85.3/82.4
Gu et al. [58]	90.6/–	94.8/–	79.8/–	83.2/–	86.8/–	87.04/–
Srinivasan et al. [25]	**95.0/95.0**	**100.0/100.0**	89.6/87.2	93.6/93.6	**100.0/100.0**	95.6/95.2
Ma and Latecki [21]	92.0/92.0	97.9/97.9	85.4/85.4	87.5/87.5	100/100	92.6/92.6
Standard sketches provided with the dataset						
Ferrari et al. [18] (20%-IoU)	72.7/56.8	90.9/89.1	68.1/62.6	81.8/68.2	93.9/75.8	81.48/70.5
Ravishankar et al. [8]	97.7/95.5	92.7/90.9	**93.4/91.2**	95.3/93.7	96.9/93.9	95.2/93.0
Lu et. al. [29]	92.5/92	95.8/95.8	92.0/86.2	85.4/83.3	93.8/93.8	85.1/83.6
Zhu et al. [20]	80.0/80.0	92.9/92.9	68.1/68.1	74.2/64.5	82.4/82.4	79.5/77.6
Riemenschneider et al. [17]	93.3/93.3	97.0/97.0	81.9/79.2	86.3/84.6	92.6/92.6	90.5/89.3
Proposed system, Setup-1	97.7/95.0	92.9/91.38	93.4/91.2	96.57/94.2	96.5/93.5	95.41/93.06
Proposed system, Setup-2	93.27/92.0	93.9/92.7	93.4/91.2	97.0/97.0	93.2/93.2	94.05/93.25
Proposed system, Setup-3	**97.87/97.87**	97.0/97.0	93.4/91.2	**97.22/97.22**	**96.55/96.55**	**96.4/95.97**

Table 2
Comparison of *Coarse Matching* detection rates (%) at 1.0 FPPI for the ETHZ dataset using the original sketches at the PASCAL overlap criterion of 50%. Bold values represent the best result obtained for each category.

Ref	Applelogo	Bottle	Giraffe	Mug	Swan	Average
Hough (Ferrari et al. [24])	43.0	64.4	52.2	45.1	62.0	41.34
M^2HT (Maji and Malik [57])	85.0	67.0	55.0	55.0	42.3	60.86
w_{ac} (Ommer and Malik [59])	80.0	**92.4**	36.2	47.5	58.8	62.98
Partial Matching (Donoser et al. [17])	90.4	84.4	50.0	32.3	90.1	51.44
Our *Coarse Matching* (using $2-AS$)	90.0	66.67	54.8	55.0	87.5	70.8
Our *Coarse Matching* (using *CSPs*)	**92.0**	81.6	**79.5**	**85.5**	**90.8**	**85.88**

with a smaller number of works (shown in Table 2), for which the results were available. We follow the trend in the literature to choose the FPPI rate 1.0 at which the detection rates were available and compared usually. As seen in the table, the proposed coarse level detection technique shows a very good average detection rate 70.8% and 85.88% using $2-AS$ and *CSP* respectively as features. Other methods underperform due to a less flexible approach. For instance, Donoser et al. [17] uses a pair of three sample points along the contour, which is inherently rigid.

6.1.3. Evaluation of the full-system

This final set of experiments aims to evaluate the full multi-stage detection system (described in Sections 3, 4, 5.1 and 5.2) in terms of identifying the occurrences of similar object instances in real-life static images. The standard sketches provided with the ETHZ dataset were used as model inputs. As shown in Table 1, the final detection performance of the full system was compared with several of the state-of-the-art methods [9,17,18,20,21,24,25,29,47,57,58]. The resulting performance with and without verification stage (described in Section 5.3) is also reported in the table. While one can observe that the performance without the verification stage is pretty good, with verification our method outperforms all others, as shown in the last column in Table 1. In fact, we achieve a very good average detection rate of 95.5% at a low value of 0.2 FPPI, whereas other than a few exceptions (e.g. [8,25]), none of the rest could achieve such a high detection rate even at 0.4 FPPI. However, due to the unavailability of the results for other methods at this FPPI, an explicit comparison was not possible.

Fig. 13 shows several pictorial results of our system from the ETHZ database. As can be seen in this collage, we can reliably handle clutter (1c, 3e, 4b–e), scale variations (7b, 7e), deformations (1a–3b), pose-changes (5b–e), intra-class variations (2b–d) and multiple object instances (7b–e). 7a shows the performance of the system in the presence of a rotated object. Typical failure cases of our method are shown in Fig. 14 The failures are due to the accidental presence of similar shaped structures in the image. The problem is exacerbated in the fifth example of Fig. 14 due to clutter, while in the third example of the Fig. 14 by the search for small objects, which is necessarily more error-prone due to the availability of less data.

Fig. 15 shows the ROC curves for several of the methods for which data was available. In four of the categories (Apple, Mug, Swan and Giraffe), we achieve the best detection rates reported till date. In the

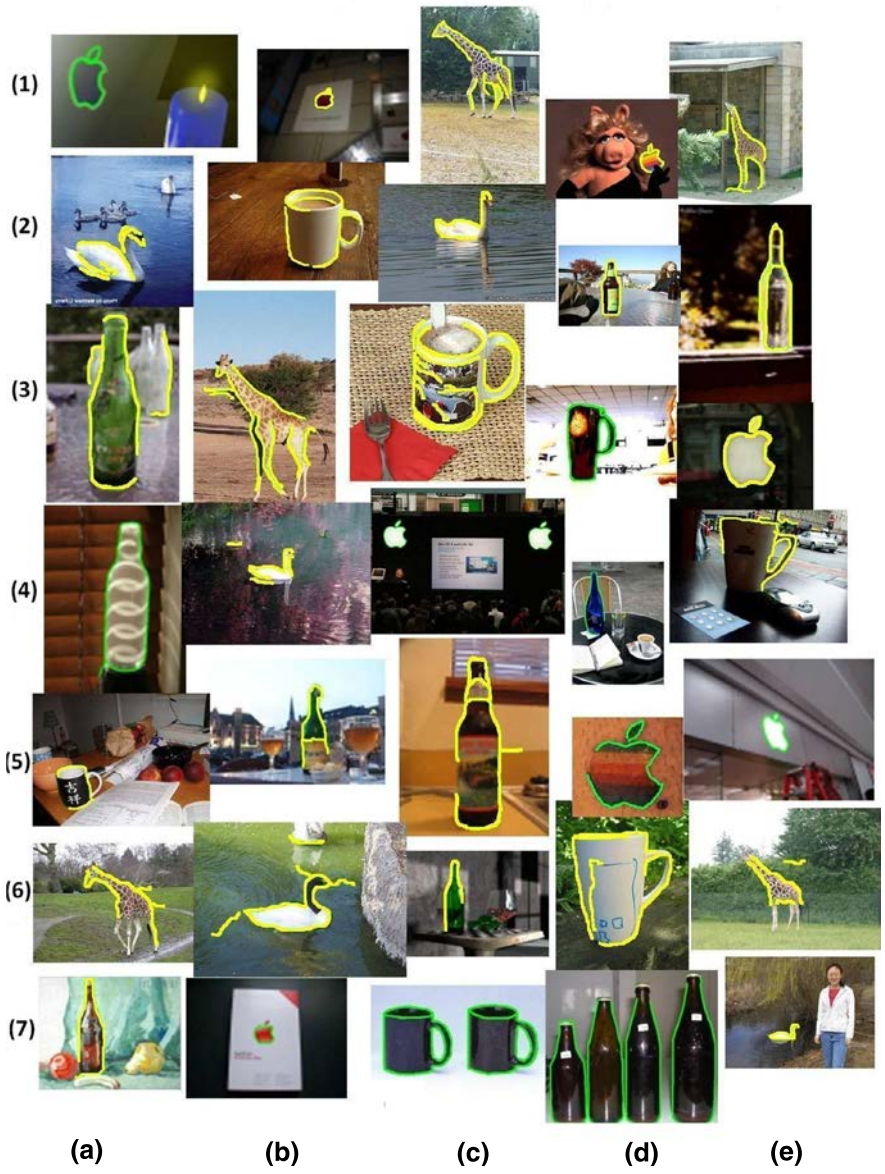

Fig. 13. Some pictorial results showing the correct results from the ETHZ dataset. The extracted object boundaries are shown in the best color possible. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 14. Some pictorial results from the ETHZ dataset show the failure cases.

'Bottle' category, the result declared by Srinivasan et al. [25] are a little better, but this is due to an extensive learning phase where they use half of the images as training data for improved performance as opposed to a single sketch in our approach. For the same problem formulation, that uses the single sketch as input [17,18,20,29], we do significantly better than most other methods.

6.1.4. Delineating object boundaries

Besides object detection, one may test the algorithms for exact delineation of the detected object's boundary. This becomes difficult since we are dealing with highly deformable objects. For evaluation, we used the *coverage* and *accuracy* measures as defined in Ma and Latecki [21]. The *coverage* measures the percentage of foreground contours that are successfully detected while the *precision* measures the percentage of detected edge pixels that are correct. As can be observed in Table 3 comparing the *coverage*/*precision* of the object boundary detection with some other methods, our proposed method was able to achieve superior results for all the five categories. Only for Giraffe, we do not do so well but this is due to the incompleteness of the model provided for this category (legs are missing).

While the above evaluation is useful, it does not tell us anything about the performance of a system on the real-world application of

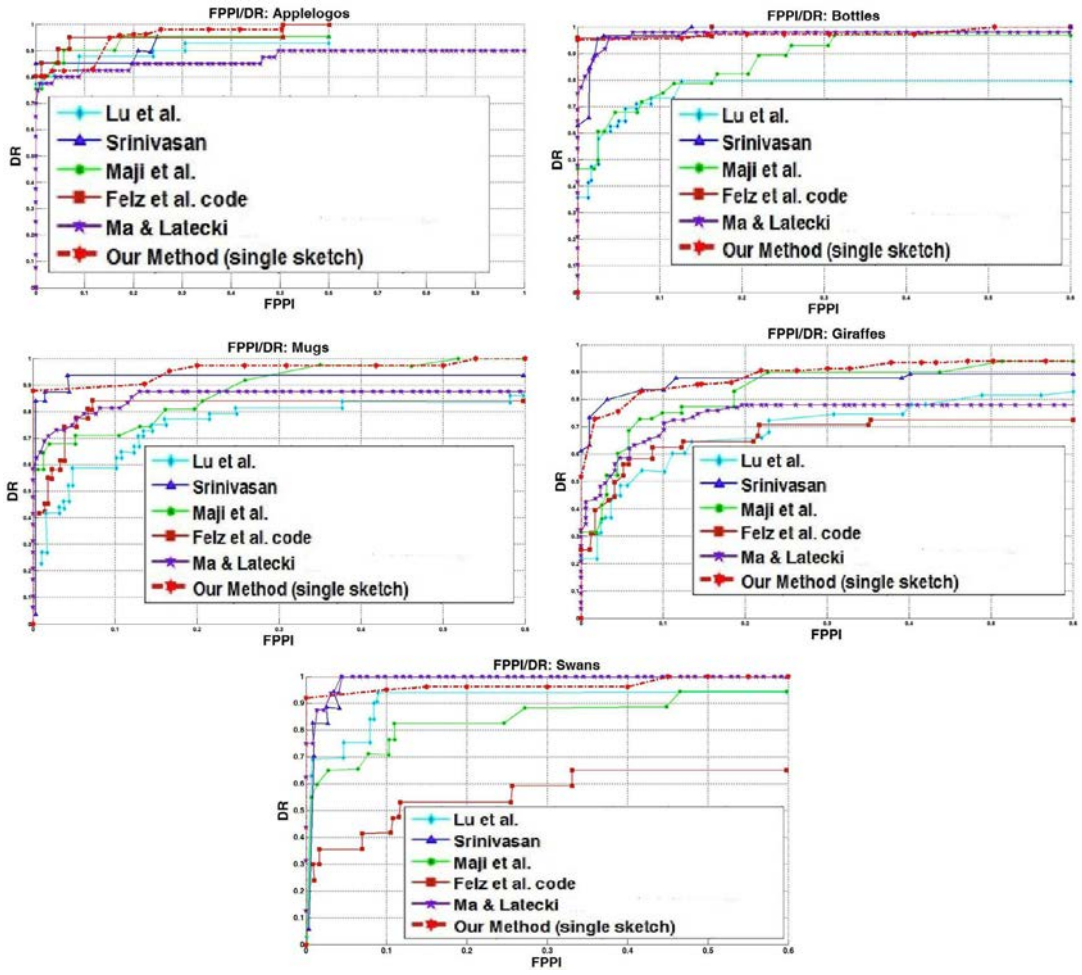

Fig. 15. ROC curves for the ETHZ dataset using the original sketches provided with the dataset. For methods other than ours, the results from Ma and Latecki [21] were used.

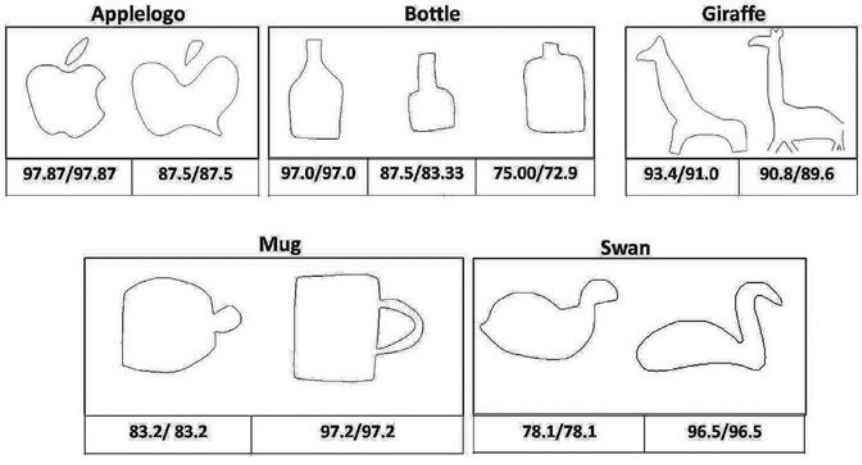

Fig. 16. First row: some real-life hand-drawn model sketches from our dataset. Detection rates (DR) are reported as a/b, where a and b represent the detection rates obtained at 0.4 and 0.3 FPPI respectively. Last row: shows the detection rates by the proposed method using the corresponding sketches shown immediately above in the first row.

object detection using hand-drawn sketches. This is becoming increasingly more important given the proliferation of touch-based smartphones and tablet devices.

6.1.5. Results using a hand-drawn sketch dataset

Most of the experiments are run using the standard sketches provided with the dataset, the models are very clean and well-drawn. In a real life scenario, user-drawn sketches may be more noisy. In order to evaluate the object detection algorithms for this problem, we developed a new dataset of hand-drawn model sketches consisting of 10 sketches per category for each of the five object categories in the ETHZ dataset. Ten different users drew these sketches by hand on a touch-based tablet and these were then cleaned up using some elementary morphological operations. This dataset is attached as supplementary material with this paper. Some sketches from this dataset are shown in the top rows of Fig. 16.

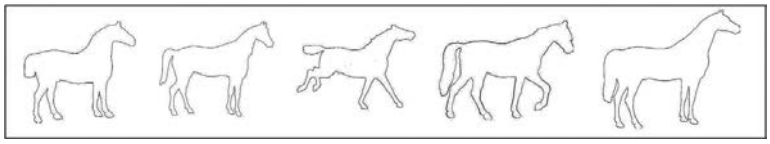

Fig. 17. Five horse sketches obtained from the MPEG-7 CE-Shape-1 dataset. These are used for the INRIA horse and Weizmann horse dataset tests.

Fig. 18. Some results on the INRIA horse dataset.

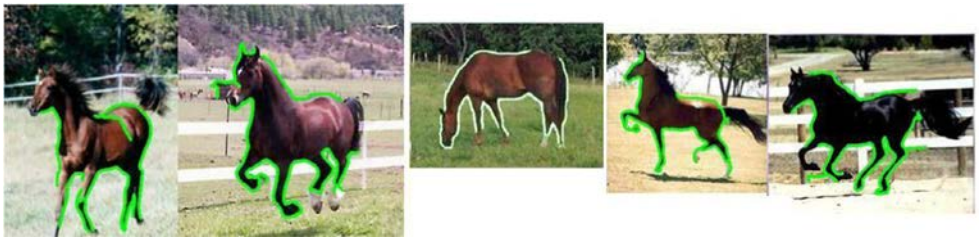

Fig. 19. Some results on the Weizmann horse dataset.

Table 3
The average *coverage*/*precision* for object boundary detection over five random trials, when measured at 0.4 FPPI.

Object type	Ma and Latecki [21]	Ferrari et al. [24]	Our system
Applelogo	0.923/**0.948**	0.916/0.939	**0.951**/0.943
Bottle	0.845/0.903	0.836/0.845	**0.914**/**0.875**
Giraffe	0.456/0.784	**0.685**/0.773	0.632/**0.817**
Mug	0.735/0.803	0.844/**0.776**	**0.929**/0.764
Swan	0.848/**0.909**	0.777/0.772	**0.915**/0.901

Table 4
Comparison of the average detection rate (ADR) of Ferrari et al. [24] at 0.4/0.3 FPPI and that achieved by the proposed method using all the 10 hand-drawn sketch models obtained for each object type.

Ref.	Applelogo	Bottle	Giraffe	Mug	Swan
Ferrari et al. [24]	64.8/54.8	71.9/71.9	42.5/38.6	74.1/74.1	57.3/51.9
Proposed method	96.25/96.25	92.3/90.3	92.6/90.4	95.5/95.5	91.0/91.0

We compute detection rates at different FPPI's for the hand-drawn sketches. There was considerable variation in the different sketches. For instance, in Fig. 16 (*Mug 2*), the inner contour of the handle is also drawn by the user while only the outer contour was drawn in *Mug 1*. Some sketches were also quite distorted (*Bottle 3*, *Swan 1*). The average detection rate (ADR) for all the 10 sketches of a category was computed and is shown in the last rows of Fig. 16. In most of the cases, when a reasonably complete sketch for an object category was provided, the performance of our system was good. However, the results deteriorated when the sketches were extremely distorted (*Applelogo 2, Bottle 3, Swan 1*) or less-detailed (*Mug 1*). Still, the results may be said to be acceptable for most of the hand-drawn sketches and gives hope for a practical sketch-based object detection module for touch-based devices. Based on the availability of the code, as can be seen in Table 4, the performance was compared with the performance obtained for the method by Ferrari et al. [24] and our method has significantly out-performed in all of the five categories, with an average improvement of about 12%.

6.2. Performance on other datasets

We also tested our algorithm on the INRIA horse and the Weizmann horse datasets which show significant intra-class deformation for a single object category. For these, no model sketches were available. In order to evaluate the robustness of our approach in the presence of an arbitrary model sketch, a random subset of 10 images from the entire set of 20 images of the Horse category in Part B, MPEG-7 CE-Shape-1 dataset[2] was used to identify a prototype shape. The binary silhouettes of horse side-views were pre-processed (also flipped wherever necessary) and represented by the outer contour only. A simplified shape context (SC) based score was used to compute pair-wise similarity between shapes. An affinity propagation-based clustering algorithm [21] was adopted to automatically partition the sketches into clusters. The largest cluster is used as the single model for our system. The experiment was repeated 5 times with 10 randomly chosen Horse shapes from the dataset. Fig. 17 shows the five models thus obtained. The variation in these five sketches may be noted here.

Some results on the INRIA Horse dataset are shown in Fig. 18. Table 5 shows the detection rates at 0.4 FPPI averaged over the five sketches on the entire dataset. The depicted detection rate of Ferrari et al. [51], as mentioned in Table 5, is taken from Ferrari et al. [24]. Apart from such results at 0.4/0.3 FPPI, we achieved a detection rate of 89.65% at 0.1 FPPI rate, which is better than the best detection rate 85.27% achieved by Maji and Malik [57] at the same 0.1 FPPI rate.

Some results on the Weizmann horse dataset are shown in Fig. 19. Table 6 shows the detection rates using the PASCAL criterion of 0.4 FPPI averaged over the five sketches, where the entire dataset was used for experiments. In this table, the results of previous papers are taken from Yang and Latecki [26], where the authors use a strict PASCAL 50%-IOU criteria as the correctness measure and the background set is taken from Caltech 101 [60]. As shown, we achieved a detection rate of 95.2% at 0.4 FPPI, which is as good as the best result achieved

[2] http://knight.temple.edu/~shape/MPEG7/datasset.html .

Table 5
Detection rate (%) on the INRIA Horse dataset at 0.4 FPPI. Bold values represent the best result obtained for each category.

Object type	Ferrari et al. [24] (learning based)	Ferrari et al. [51] (learning based)	Maji and Malik [57] (learning based)	Our system (averaged over the five learnt sketches of Fig. 17)
Horse	69.2	76.9	86.0	**96.25**

Table 6
Detection rate (%) on the Weizmann Horse dataset at 0.4 FPPI. Bold values represent the best result obtained for each category.

Object type	Yang and Latecki [26] (learning based)	Zhu et al. [10] (learning based)	Shotton et al. [23] (learning based)	Our system (averaged over the five learnt sketches of Fig. 17)
Horse	93.97	86.0	**95.20**	**95.20**

by the learning-based approach of Shotton et al. [23]. However, this detection rate was in fact first achieved at a lower FPPI of 0.33.

A good result obtained on these two datasets using sketches from another dataset demonstrate that our proposed approach is extremely flexible, robust to clutter and deformation, and can obtain a reasonably good performance without any intensive dataset-specific learning.

7. Conclusion

In this paper, we have introduced a system for deformable object detection using a single sketch, which may be hand-drawn, computer-assisted or automatically learnt from training data. The method is extremely flexible and automatically segments the sketch into parts, allowing for a different affine transformation for each part during matching. The matching strategy is also quite robust to clutter and missing edges due to its ability to find image contour paths resembling model contours even in low gradient regions. To evaluate the application of image search using sketches drawn on touch-based smart devices, we introduced a hand-drawn sketch model dataset for the standard ETHZ dataset of images. Promising results comparing favorably with state-of-the-art methods were obtained on this as well as several other commonly used datasets. Future work includes development of automatic methods for building one or more sketches from training images for a complete training-testing system using flexible sketch models.

References

[1] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 264–271.
[2] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[3] K.E.A. van de Sande, C.G.M. Snoek, A.W.M. Smeulders, Fisher and vlad with flair, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014.
[4] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: integrated recognition, localization and detection using convolutional networks, in: Proceedings of the International Conference on Learning Representations (ICLR2014), 2014.
[5] X. Wang, M. Yang, S. Zhu, Y. Lin, Regionlets for generic object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2013.
[6] A. Thayananthan, B. Stenger, P.H.S. Torr, R. Cipolla, Shape context and chamfer matching in cluttered scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 127–133.
[7] F. Jurie, C. Schmid, Scale-invariant shape features for recognition of object categories, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2004.
[8] S. Ravishankar, A. Jain, A. Mittal, Multi-stage contour based detection of deformable objects, in: Proceedings of the European Conference of Computer Vision, 2008.
[9] V. Ferrari, F. Jurie, C. Schmid, Accurate object detection with deformable shape models learnt from images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[10] L. Zhu, Y. Chen, A. Yuille, Learning a hierarchical deformable template for rapid deformable object parsing, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1029–1043.
[11] J. Shotton, A. Blake, R. Cipolla, Efficiently combining contour and texture cues for object recognition, in: Proceedings of the British Machine Vision Conference, 2008.
[12] S. Dickinson, Object representation and recognition, in: E. Lepore, Z. Pylyshyn (Eds.), What is Cognitive Science?, Basil Blackwell Publishers, 1999, pp. 172–207.
[13] M. Eitz, J. Hays, M. Alexa, How do humans sketch objects? ACM Trans. Graph. (Proc. SIGGRAPH) 31 (4) (2012) 44:1–44:10.
[14] S. Bagon, O. Brostovski, M. Galun, M. Irani, Detecting and sketching the common, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 33–40.
[15] Y.J. Lee, K. Grauman, Shape discovery from unlabeled image collections, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2254–2261.
[16] M. Donoser, H. Riemenschneider, H. Bischof, Efficient partial shape matching of outer contours, in: Proceedings of the Asian Conference of Computer Vision, 2009.
[17] H. Riemenschneider, M. Donoser, H. Bischof, Using partial edge contour matches for efficient object category localization, in: Proceedings of the European Conference of Computer Vision, 2010.
[18] V. Ferrari, T. Tuytelaars, L. Gool, Object detection by contour segment networks, in: Proceedings of the European Conference of Computer Vision, 2006, pp. 14–28.
[19] X. Bai, L.J. Latecki, Q. Li, W. Liu, Z. Tu, Shape band: a deformable object detection approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1335–1342.
[20] Q. Zhu, L. Wang, Y. Wu, J. Shi., Contour context selection for object detection: a set-to-set contour matching approach, in: Proceedings of the European Conference of Computer Vision, 2008.
[21] T. Ma, L.J. Latecki, From partial shape matching through local deformation to robust global shape similarity for object detection, in: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1441–1448.
[22] A. Toshev, B. Taskar, K. Daniilidis, Shape-based object detection via boundary structure segmentation, Int. J. Comput. Vis. 99 (2) (2012) 123–146.
[23] J. Shotton, A. Blake, R. Cipolla, Multiscale categorical object recognition using contour fragments, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 1270–1281.
[24] V. Ferrari, F. Jurie, C. Schmid, From images to shape models for object detection, Int. J. Comput. Vis. 87 (3) (2010) 284–303.
[25] P. Srinivasan, Q. Zhu, J. Shi, Many-to-one contour matching for describing and discriminating object shape, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010.
[26] X. Yang, L.J. Latecki, Weakly supervised shape based object detection with particle filter, in: Proceedings of the European Conference of Computer Vision, 2010.
[27] O. Chum, A. Zisserman, An exemplar model for learning object classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[28] S. Belongie, J. Puzhicha, J. Malik, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 509–522.
[29] C. Lu, L.J. Latecki, N. Adluru, X. Yang, H. Ling, Shape guided contour grouping with particle filters, in: Proceedings of the International Conference of Computer Vision, 2009, pp. 1–8.
[30] S.X. Yu, R. Gross, J. Shi, Concurrent object recognition and segmentation by graph partitioning, in: NIPS, 2002, pp. 1383–1390.
[31] M. Maire, S.X. Yu, P. Perona1, Object detection and segmentation from joint embedding of parts and pixels, in: Proceedings of the International Conference on Computer Vision, 2011, pp. 2142–2149.
[32] P.F. Felzenszwalb, R.B. Grishick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2009) 1627–1645.
[33] P. Felzenszwalb, J. Schwartz, Hierarchical matching of deformable shapes, in: Proceedings of the International Conference of Computer Vision, 2007, pp. 1–8.
[34] M.A. Fischler, R.A. Elschlager, The representation and matching of pictorial structures, IEEE Trans. Comput. 22 (1) (1973) 67–92.
[35] P.F. Felzenszwalb, D.P. Huttenlocher, Pictorial structures for object recognition, Int. J. Comput. Vis. 61 (1) (2005) 5579.
[36] R. Ronfard, C. Schmid, B. Triggs, Learning to parse pictures of people, in: Proceedings of the European Conference on Computer Vision, 2002, pp. 700–714.
[37] D. Ramanan, C. Sminchisescu, Training deformable models for localization, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 206–213.

[38] D. Ramanan, Learning to parse images of articulated bodies, in: Proceedings of the Neural Information Processing Systems (NIS) 2006, 2006.
[39] V. Ferrari, M. Marin-Jimenez, A. Zisserman, Progressive search space reduction for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[40] D. Crandall, P. Felzenszwalb, D. Huttenlocher, Spatial priors for part-based recognition using statistical models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 10–17.
[41] S.R. M. Andriluka, B. Schiele, Pictorial structures revisited: people detection and articulated pose estimation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2009, pp. 1014–1021.
[42] B. Sapp, A. Toshev, B. Tas, Cascaded models for articulated pose estimation, in: Proceedings of the European Conference of Computer Vision, 2010, pp. 406–420.
[43] R. Gopalan, P. Turaga, R. Chellappa, Articulation-invariant representation of non-planer shapes, in: Proceedings of the European Conference of Computer Vision, 2010.
[44] S. Branson, P. Perona, S. Belongie, Strong supervision from weak annotation: interactive training of deformable part models, in: Proceedings of the International Conference on Computer Vision, 2011.
[45] S. Agarwal, D. Roth, Learning a sparse representation for object detection, in: Proceedings of the European Conference on Computer Vision, Copenhagen, Denmark, vol. 4, 2002, pp. 113–130.
[46] I. Kokkinos, A. Yuille, Inference and learning with hierarchical shape models, Int. J. Comput. Vis. (2010) 1–25.
[47] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[48] C. Zhou, J. Yuan, Non-rectangular part discovery for object detection, in: Proceedings of the British Machine Vision Conference, 2014.
[49] H. Ling, D. Jacobs, Shape classification using inner-distance, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 286–299.
[50] X. He, N. Yung, Curvature scale space corner detector with adaptive threshold and dynamic region of support, in: Proceedings of the International Conference on Pattern Recognition, 2004.
[51] V. Ferrari, L. Fevrier, F. Jurie, C. Schmid, Groups of adjacent contour segments for object detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (2008) 3651.
[52] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural boundaries using local brightness, color and texture cues, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 530–549.
[53] Q. Zhu, G. Song, J. Shi, Untangling cycles for contour grouping, in: Proceedings of the IEEE 11th International Conference on Computer Vision, 2007, pp. 1–8.
[54] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, R. Chellappa, Fast directional chamfer matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010.
[55] H. Samet, Foundations of Multidimensional and Metric Data Structures, Elsevier, 2006.
[56] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2001.
[57] S. Maji, J. Malik, Object detection using a max-margin Hough transform, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[58] C. Gu, J. Lim, P. Arbelaez, J. Malik, Recognition using regions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[59] J.M.B. Ommer, Multi-scale object detection by clustering lines, in: Proceedings of the International Conference of Computer Vision, 2009, pp. 484–491.
[60] P.P. L. Fei-Fei, R. Fergus, One-shot learning of object categories, IEEE Trans. Pattern Anal. Mach. Intell. 28 (4) (2006) 594–611.