A Multi-stage Approach to Curve Extraction

Yuliang Guo, Naman Kumar, Maruthi Narayanan, and Benjamin Kimia

Brown University, School of Engineering,
Providence, RI 02912, USA
{yuliang_guo,maruthi_narayanan,benjamin_kimia}@brown.edu,
namank@andrew.cmu.edu
http://vision.lems.brown.edu

Abstract. We propose a multi-stage approach to curve extraction where the curve fragment search space is iteratively reduced by removing unlikely candidates using geometric constrains, but without affecting recall, to a point where the application of an objective functional becomes appropriate. The motivation in using multiple stages is to avoid the drawback of using a global functional directly on edges, which can result in non-salient but high scoring curve fragments, which arise from non-uniformly distributed edge evidence. The process progresses in stages from local to global: (*i*) edges, (*ii*) curvelets, (*iii*) unambiguous curve fragments, (*iv*) resolving ambiguities to generate a full set of curve fragment candidates, (*v*) merging curve fragments based on a learned photometric and geometric cues as well a novel *lateral edge sparsity* cue, and (*vi*) the application of a learned objective functional to get a final selection of curve fragments. The resulting curve fragments are typically visually salient and have been evaluated in two ways. First, we measure the stability of curve fragments when images undergo visual transformations such as change in viewpoints, illumination, and noise, a critical factor for curve fragments to be useful to later visual processes but one often ignored in evaluation. Second, we use a more traditional comparison against human annotation, but using the CFGD dataset and CFGD evaluation strategy rather than the standard BSDS counterpart, which is shown to be not appropriate for evaluating curve fragments. Under both evaluation schemes our results are significantly better than those state of the art algorithms whose implementations are publicly available.

1 Introduction

Interest in *contour extraction*, an age-old fundamental problem in computer vision, which diminished beginning in the late 1990's with a paradigm shift towards the use of keypoint-based approaches, has recently seen a revival, especially in application to object recognition and multi-view applications. Contours, which along with regions have been a classic intermediate-level representation of images, are attractive since they are robust to illumination variation, view variation, and other visual transformations. An increasing number of approaches now rely on extracted contours for recognition [6,16,20,31,11], or for 3D object

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 663–678, 2014.
© Springer International Publishing Switzerland 2014

recognition and pose estimation [21], and others, leading to a number of recent approaches to contour extraction [22,28]. Multiple terms such as boundary, contour, silhouette, curve, edges, *etc.*, have been used with some ambiguity to denote at least three distinct notions: (*i*) *edge maps*, line tokens, and other local and unorganized representations of a contour where a pixel/point is denoted as a contour point; (*ii*) *saliency maps* representing at each pixel the probability of a contour, or a subset of contours such as silhouette (boundaries), *e.g.*, Pb [18,2] and others; (*iii*) *curve fragments* which are essentially chains of long ordered edge points [27,13,14,15,5,22,11,17,4,24,9]. Relatively fewer papers address the latter, but it is a arguably the most useful. The focus of this paper is to extract contour fragments, Fig 1.

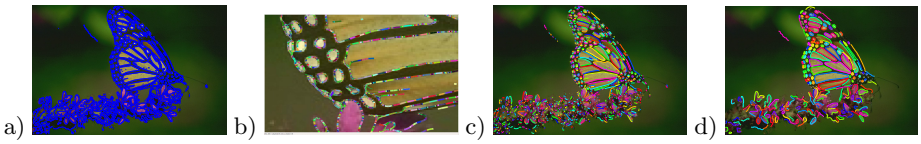

Fig. 1. A brief overview of some of the stages in our curve extraction approach: a) Initial Edge Map, b) Curvelet Map, zoomed in to show details, c) Contour fragments from [27], d) Contour Fragments resulting from this work. Readers are invited to zoom in to compare the two contour fragments shown in randomized color

The early work of Fischler *et al.* [8] is canonical of work which followed the next three decades in this area: a local road detector generates a spatial map which is then searched by the A^* algorithm for the minimum cost path between any two given points. We show in Section 2 that previous work essentially follows this pattern by searching the space of curves using a linear objective functional. The drawback of this type of a model is that for any given path depicting a low cost (high edge evidence), there is a *key underlying assumption that cost is uniformly distributed*. This assumption, however, is not always true and this caveat allows for paths passing through very high strength evidence but otherwise sparse, leading to either numerous false positives or missed detections.

Fig 2 shows examples of curve fragments achieving good response from the global optimization process which are not salient because they are in fact spatially sparse in edge evidence. The existence of such curves can overwhelm those arrangements of edges which have consistency but possibly low edge evidence. We posit that the jump from edges to curve fragments needs to be mediated by multiple stages of increasingly more global considerations, each maintaining recall while reducing the space of undesirable curves, thus safely limiting that the curve fragment search space. This is in analogy to the 20 question approach but done in stages.

Specifically, the curve extraction process proceeds in six stages: (*i*) edge maps are first computed at thresholds generating the highest recall, therefore also generating the highest level of false positives; (*ii*) the formation of curvelets from neighboring edges at each edge, identifies edges with geometrically consistent support and potential neighbors. This allows false positives to be reduced with

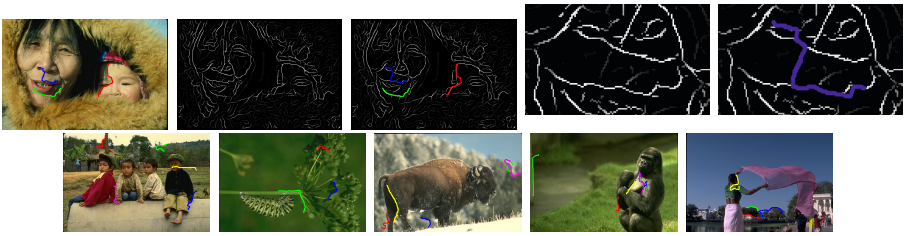

Fig. 2. An example demonstrating the drawback of directly going from edges to curve fragments: top row, left to right: original image and its edge map, some selected contours which score well based on a standard objective functional but which are not salient, the edge map zoomed in the area of the blue curve without and with the blue curve superimposed. Second row: additional examples.

a very insignificant drop in recall; (*iii*) edges whose local curvelet structure indicates an unambiguous groupings with their neighbors are grouped and all these edges are replaced with the resulting curve fragment; (*iv*) the ambiguity in grouping the remaining edges is explicitly represented in a hypothesis graph rooted at end-points of contour fragments formed in the previous stage. The ambiguity is then resolved by applying an objective functional operating locally which takes into account smoothness and minimizes gaps; (*v*) The resulting curve fragments are often over-fragmented, so geometric and photometric cues and a novel edge sparsity cue introduced here are learnt in combination to merge pairs of curves sharing an end-point; (*vi*) The resulting curve fragments are rated by a learned objective functional using cues almost identical with the above, to prune false positives. The end result of our process is a *contour fragment graph*, representing curve fragments and their relative arrangements at junctions.

The results are evaluated in two ways. First, the stability of curve fragments when images undergo visual transformations is examined. Second, results are compared against human annotations using the boundary evaluation framework introduced by [18] on the Berkeley Segmentation DataSet (BSDS) [18] as well as the curve evaluation framework introduced in [10] on the Curve Fragment Ground-Truth Dataset (CFGD). In all cases it is shown that our approach improves upon the state of the art.

2 Related Works

The need for non-linearity at the level of edge detection was identified by Iverson and Zucker [12] who proposed that instead of a linear sum over the support of an edge detector, a non-linear function in the form of a logical-linear operation be used. Our approach is similar to this idea at the level of curve fragments. A **Curve Fragment (CF)** is formally defined as a piecewise smooth parameterizable curve segment, $C(s) = (x(s), y(s)), s \in [0, 1]$. The literature on boundary detection is broad, diverse, and spans a few decades. Much of this work, however, focuses on edge or saliency map detection and relatively few focus on detecting curve fragments. We present a large portion of the existing literature in the

following abstract framework. Let $V = \{v_i | i = 1, ..., N\}$ represent a set of possible samples along the curve fragments: These can be image pixels [5], image edges [30], or small grouping of edges (super-nodes) in the form of line tokens [14] or local subgraph [13]. Let $E = \{e_{ij} | d(v_i, v_j) < d_0, \forall v_i, \forall v_j\}$ represent all possible links among these candidate curve samples. The graph $G(V, E)$ is therefore a superset of all possible curve fragments. The curve extraction problem can then be cast as selecting a set of links $E^* \subset E$ where

$$E^* = \arg\min_{\bar{E} \subseteq E} [f_{fg}(\bar{E}) + f_{bg}(E \setminus \bar{E})], \tag{1}$$

and where f_{fg} and f_{bg} are objective functionals that capture characteristics of the foreground curve and the background clutter, respectively. For example, in the min-cover approach [5], v_i are image pixels and the algorithm seeks curves to minimizing an objective function in the form of Equation 1. The two approaches KGS [30,13] and FPG [14] follow a two-stage strategy: in KGS [30,13], normalized cut [26] divides the graph into clusters which then represent v_i as a super node. In FPG [14] short straight lines are first formed and represent v_i. Both methods then proceed to solve an optimization problem of the form described in Equation 1 using a linear objective function (see supplementary material for further detail). This two-stage optimization increases efficiency, but has serious drawbacks: The initial clustering in normalized cut can introduce spurious links in the k-way normalized cut [29], which includes non-convex optimization, random initialization results in random clusters which in turn lead to poor stability, *e.g.*, over a video sequence. Similarly, the initial stage in FPG [14] needs to be conservative due to the geometric limitations of the line model, leaving much of the work for the second stage.

In our approach, we have avoided both problems by employing a range of simple to complex models covering local to global information arranged in stages. Our approach bears a strong relationship with compositional systems such as [11,7], which begin with Gabor-like filters and progressively construct higher level constructs by composing simples structures. Our work, however has only bottom-up processing and may benefit from a full compositional structure. The work of Todorovic [22] is also hierarchical and generates a vast collection of contour fragment candidates which though extensive are selected to form a final output. The latter stage of this work can probably work with the results of this paper.

3 Managing Grouping Ambiguity

The goal of the multi-stage approach is to entertain all possible curve hypothesis and then progressively remove unlikely ones, *i.e.*, maintaining recall while increasing precision, all based on geometric constraints, as explained below.

Stage 1) Edge Detection: The set of candidate curves is significantly reduced by only initiating them through detected edges. The caveat is that curves with significant gaps would not have a representation in the gap area, so these curves would have to rely on representation in the non-gap areas to be initiated; see more below. The idea of initiating curve candidates from edges implies that the results

of edge detection should not be pruned at this early stage. Rather, this step should maintain the highest possible recall at the expense of also maintaining numerous false positives, Fig 1. The edges are connected by a neighborhood graph:

Definition 1. The Edge Topology Graph ***(ETG)*** *is a graph whose nodes are image edges and where links exist between any two edges which are linked by some* curvelet, Fig 3 (d).

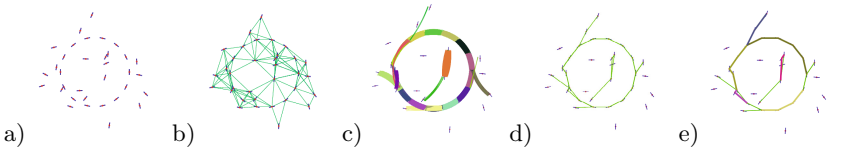

Fig. 3. (a) And edge map, (b) An Edge Neighborhood Graph connects each edge to every edge in a local neighborhood, (c) The set of discrete curvelets, (d) The Edge Topology Graph **(ETG)**, (e) The Curve Fragment Map **(CFG)** is formed by identifying unambiguous and geometrically viable 1-chains and replacing the image edges and connecting links by a single curve fragment

Stage 2) Curvelet Formation: The next stage in removing unlikely candidates is to require a curve segment to have more than one edge support. We follow the approach of [27] who considered all geometrically reasonable combinations of edges, referred to as *curvelets, in some neighborhood typically (7x7). The purpose of using* curvelets is twofold: (*i*) establish candidate pairing among edges from which candidate curve fragments can be constructed, and (*ii*) discarding all potential pairings which do not participate in a geometrically consistent relationship say in a set of say five edges. This stage removes only those curve fragments whose edge support is below five edges, which implies a very minimal reduction in recall, but a significant reduction in false positives. Note that this is quite different than a greedy approach where the most likely combination is considered. Rather, this approach retains almost all of the likely combinations, relegating the job of deciding which may be veridical to a later, global stage.

Stage 3) Forming unambiguous curve fragments: Since all pairings of edges which have passed the above stage are considered viable, ambiguities arise when an edge forms pairings with more than one edge, occurring often in junctions, corners, but also in other portions of a curve. However, in numerous other cases an edge forms pairing with only one other edge (each end of an edge is considered separately). In this case an unambiguous curve fragment forms, which need not be challenged [27]. The formation of these unambiguous curve fragments to replace all edges and their groupings which led to it helps resolves ambiguity in their neighborhood, as described next. Observe that thus far no objective function has been used, but rather the geometric configuration of edges with barely minimal contrast has been used to reduce the search space.

Definition 2. The Curve Fragment Graph ***(CFG)*** *is a transformation of the* ***ETG*** *where these nodes and links which form unambiguous and geometrically viable 1-chains are replaced by a single contour fragment, Fig 3 (e).*

Stage 4) Resolving Grouping Ambiguity: The approach to resolve the ambiguity in grouping the remaining edges is to first construct a graph of all possible groupings. The first observation is that grouping ambiguity occurs in clusters of edges, thanks to the formation of unambiguous curve fragments. The canonical curves can enumerated by considering possibilities starting from the end point of an existing unambiguous curve fragments: (i) The end point is truly an end point, with on continuities, *e.g.*, endpoint of A in Fig 4a. (ii) The curve continues to another curve with ambiguity in the specific paths among a cluster of options, *e.g.*, continuations of B onto C in Fig 4a. (iii) the continuations of a curve onto one or two curves or both, as in the continuations of C onto D and E as in Fig 4a, representing a Y-junction or T junction. (iv) The curve continues but does not terminate onto another curve fragment, *e.g.*, as in G in Fig 4a. Observe that in many cases clusters of possible curves agree on the underlying topology but disagree slightly about the geometry of connectivity. The set of possible continuations can be represented by a graph, Fig 4c:

Definition 3. The Contour Hypothesis Graph *(CHG)* *is a rooted graph with the root node representing a contour fragment end-point (the anchor contour end-point) with links representing the next edge selection and with with nodes representing image edges. The leaf nodes, each representing a unique path from the root, give the set of all contour fragments continuing the anchor contour at the root.*

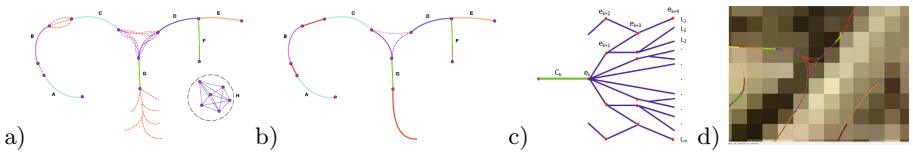

Fig. 4. a) Several distinct types of ambiguities with candidate paths from end-points. The region labeled "H" represents a noisy area not connected to any viable contour fragments. b) Newly formed **CFG** is shown after the resolution of the ambiguities using quality metric as described in the text. c) A Contour Hypotheses Graph **(CHG)** rooted at the end point e_K of contour fragment C_K. In this case, six contributions are possible which include e_{K+1}. d) A realistic example of how the CHG represents and manages ambiguity in grouping among three end-points.

The construction of CHG follows a best-first strategy guided by the quality of the path constructed thus far. Quality is represented by an energy functional **E** capturing the saliency of a contour is described below. Three limitations are imposed on the construction: *(i)* The total number of nodes cannot exceed $K = 1000$. This is experimentally never reached; *(ii)* The energy of a path can not exceed a Threshold E_0; and, *(iii)* the individual link energy cannot exceed a threshold E, a large threshold devised to prune away the very unlikely continuations. It should be noted that the contour hypothesis graph is typically very constrained since it represents local grouping ambiguity between contour fragment end-points, thus performance is not affected significantly by changes in the above system parameters.

Contour Saliency Measure: The contour hypothesis graph localizes ambiguities to clusters of topologically equivalent but geometrically different groupings. The latter can be resolved by employing a standard energy function, representing structural saliency [25]. Specifically, given an ordered sequence of edges, $C = (e_0, e_1, ...e_N)$, the energy $E(\mathcal{C})$ depends on two factors (i) maximizing smoothness by minimizing orientation difference $|\theta_{i+1} - \theta_i|$, where θ_i is the orientation of edge e_i, and (ii) minimizing the gap length or the Euclidean Distance between pairs of subsequent edges $|e_{i+1} - e_i|$. Finally, the energy is normalized by the number of edges to represent a measure that can fairly capture short and long sequences of edges:

$$E(\mathcal{C}) = \frac{1}{N+1} \sum_{i=0}^{N} (w * |e_{i+1} - e_i| + (1-w) * |\theta_{i+1} - \theta_i|), \quad 0 < w < 1 \qquad (2)$$

where we set $w = 0.2$ which is optimized by grid search in $(0, 1)$ for highest F-measure.

It should be noted that the mutli-stage approach above does not attempt to bridge gaps beyond what is possible within a small local neighborhood. Rather, gaps beyond a couple of pixels need to be completed in a post processing stage such as those presented in [23,19].

4 Learning-Based Merging of Curve Fragments

The result of the previous set of stages is a set of curve fragments which are represented as a graph, where nodes represent curve fragment endpoints or junctions where two or more curve fragments come together Fig 5a. These curve fragments are typically small and cascaded, so that over-fragmentation of expected contour is perceptually evident. Thus, additional grouping is required. Consistent with the progression from local information (edges) to increasing more global information (curvelets, hypothesis graph) in previous stages, which chiefly involve geometric information, the next stage uses a slightly more global basis of reasoning. While the chief organizational principle has thus far been geometric continuity, the resulting curve fragments are typically sufficiently long to allow the use of photometric and geometric cues in deciding whether additional grouping is required and in disambiguating conflicts when such grouping is warranted. Specifically, the main question is whether two cascaded curve fragments, i.e., those that share an end point in exclusion of other curve fragments (no T or Y junctions) should be merged or not, essentially a classification problem, Fig 5b.

We consider the following set of photometric and geometric cues in relation to whether they contribute to the merge/not-merge decision and to what extent. First, the **photometric cues** are length-normalized integrals of those photometric cues traditionally used in edge detection: *(i) Brightness* gradient: $\int \nabla B(s)\, ds$; *(ii) Saturation* gradient: $\int \nabla S(s)\, ds$; *(iii)* Hue gradient: $\int \nabla H(s)\, ds$; These three cues are computed for each of the two curve fragments and then compared.*(iv)* Texture differences between the respective sides of the two connected curve fragments, i.e., differences between the texture attributes of regions 1 and 2

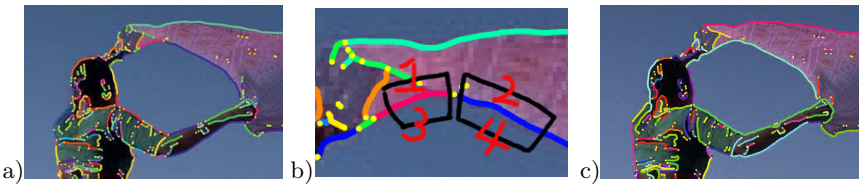

Fig. 5. (a) The curve fragment graph structure provides an opportunity to reduce the over-fragmentation by using various cues to label each node sharing two curve fragments as either "merge" or "do not merge." The decision uses only local portions of curve fragments as shown in b). (c) shows curve fragments after this stage.

in Fig 5(b), and regions 3 and 4, respectively. These differences are captured by the Chi-Square distance between the histograms of textons in each region, and then summed for the two pairs of regions.

Second, the **geometric cues** are also traditional measures: *(v)* Length-normalized integral of absolute curvature: $\int |\kappa(s)|\, ds$; *(vi) Wiggliness*: the number of inflection points along the curve fragment; These two cues are computed for each of the two curve fragments and then compared. Finally, *(vii) Geometric Continuity:* the angle between the tangents of the two curve fragments at their connecting node is a measure of how likely the two curve fragments are a single curve fragment.

Third, we introduce a **novel cue** which to the best of our knowledge has not been reported before, namely, *(viii) Lateral Edge Sparsity*: this is defined as the number of edges in a narrow neighborhood of the curve fragment, see the regions in Fig 5b. We observe that veridical contours lack edges in a small neighborhood and therefore conjecture that the cue is inversely correlated with the likelihood that a candidate curve is veridical (see next section). In addition to being a cue for veridicality, this cue can also be used as an attribute to the two potentially connecting curve fragments, expecting the number to be small for both curves.

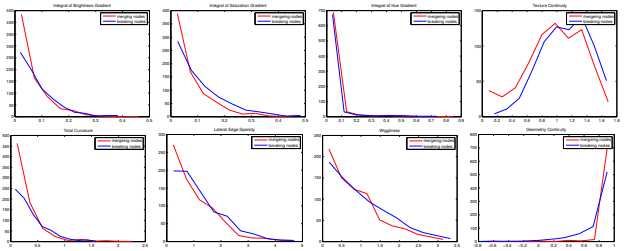

Fig. 6. The histogram of each of the eight cues, shown separately for "merge" in red and "do not merge" in blue. Observe that each cue in isolation offers a modest measure of distinction between the two cases, while the distinction is significantly compounded when all cues are used in conjunction with each other.

We explore the utility of these eight cues by using a logistic regression classifier on a training set where positive (merge) and negative (do not merge) examples are specified. Specifically, we train on the curve fragments annotated in the *Contour Fragment Ground-Truth Dataset* (CFGD) which is introduced in [10].

We use the first 20 images of the CFGD dataset for training and leave the remaining 30 for the testing stage . By matching the candidate pair of curve fragments to ground-truth curve fragments, the connection node is selected as "merge" or "do not merge" depending on whether the corresponding ground-truth curve is connected. The histograms of positive (merge) and negative (do not merge) samples for each cue reveal the discriminative power of each cue in isolation, Fig 6. The cues act in concert giving the probability of the "merge" vs "do not merge," which when exceeding a selected threshold (0.5 by default) leads to a concrete decision.

The above discussion has focused on augmenting the geometric continuity measures of previous stages with more global photometric and geometric measure with the goal of merging the typically over-fragmented curve fragments arising from local geometric grouping in previous stages. While this will be shown to be effective, the opposite problem must also be addressed: in certain instances, the geometric cues happen to be so powerful that they result in long curve fragments. In such cases, caution is required to verify that indeed the geometric factors are not inconsistent with photometric cues.[1] We therefore use the same machinery in reverse: we virtually break a long curve at the points along its length and consider whether the two pieces of the curve should be merged: the probability of (re)merging at each point of the curve is calculated, local minima are identified, and if the probability is below the merge threshold, breaks are introduced.

5 Learning-Based Sifting of Curve Fragments

The curve fragments obtained in our approach are the end result of a very cautious process: extremely low edge thresholds, maintaining all possible local groupings in the form of curvelets, stringing together unambiguous curvelets while sorting out ambiguous ones in a hypothesis graph, and resolving ambiguity by grouping cues all results in making sure that any contour hypotheses with the slightest degree of evidence is retained. As a result, it is unlikely that a veridical contour is missed in the process (very high recall), while at the same time it is highly likely that the process generates (hallucinates) false positives from scant data (low precision). Since the high degree of false positives can overwhelm processes relying on contour fragments, *e.g.,* for recognition, a final stage is needed to reduce the false positive while necessarily also reducing the recall.

This is done by learning a veridicality measure for curve fragments based on training data. Specifically, curve fragments are attributed with several image-based measures, essentially the same cues presented in the previous section, so that each curve fragment is represented as a point in a high-dimensional space. The ground-truth curve fragments of a training set from CFGD separate curve fragments into positive and negative exemplars: Veridical (or false positive) curve fragments are randomly sampled from those computed curve fragments which

[1] Since producing erroneously grouped curves can cause irreversible damage, *e.g.,* confusing object and background (as many camouflaged objects hope to achieve), it is wiser to er on the side of caution.

match (do not match) the ground truth curve fragments under the matching method of [10]. Finally, a query curve fragment interrogates this space to decide which class it belongs to. This approach is analogous to [18] who use training data in cue combination on edge data.

The cues used to indicate veridicality are very similar to those used to indicate whether two connected curve fragments should be merged. Specifically, these cues are the photometric cues of brightness, saturation, and hue gradients, and the geometric cues of total curvature, wiggliness, and finally lateral edge sparsity. We also use curve fragment length as a cue in addition[2]. Fig 7 shows the histograms of each cue on veridical and false positive contours, showing meaningful distinctions.

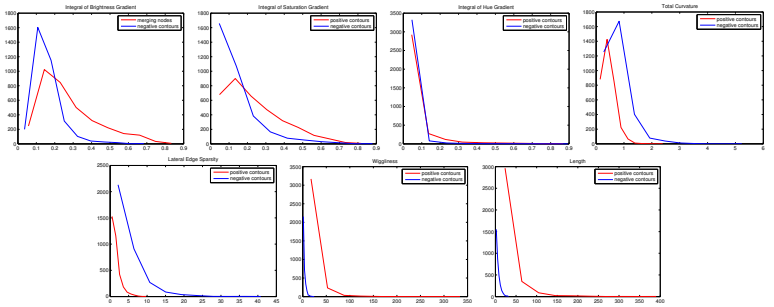

Fig. 7. The comparison of histograms of various cues for veridical (red) and false positive contours (blue), on the **CFGD** data depicting various degrees of separation

To combine the various cues we use a logistic regression classifier. The classifier is trained on the first 20 images of the CFGD and the rest are reserved for testing. Utilizing the resulting classifier output, each contour is assigned a probability of being veridical, and the curve fragment map is thresholded to yield a final set of curve fragments. This approach is quite flexible and the same framework can be applied to any other curve extraction system [13,14,15]. To quantitatively evaluate the extracted curve fragments the classifier probability is thresholded at varying levels to generate the PR curve, see Fig 13. The visual examples shown in Fig 1 and Fig 14 correspond to the highest F-measure of the PR curve.

6 Experiments

The algorithm is evaluated in two ways, one using stability analysis and one using human annotation. Both evaluations are compared to those curve extraction algorithms for which *(i)* code is available and *(ii)* an output is in the form of an independent set of ordered edges is available. Among a series of candidate algorithms we considered [13,14,15,5,22,23,11,28], only [13,14,15] satisfy these criteria.

[2] We have not included texture gradients due to technical difficulties arising just before the paper deadline.

6.1 Contour Stability/Self Consistency

A key requirement of extracted contour fragments is that they should remain stable as the viewing pose and viewing distance change, as illumination changes, as noise is introduced into the system, *etc.* Baker and Nayar [3] introduced global measures of coherence that reported self consistency of edge detectors, *e.g.* colinearity. We now expand on this idea to stability of extracted curves under general visual transformations as a form of evaluating contour fragments.

Illumination Stability Test: A requirement of contour fragment algorithms is stability with moderate illumination variations. Figure 8a) shows several frames of a sequence from the Robot Dataset [1] where the lighting is varied over a wide range. Despite the large range of variations, our method produces contour fragments, Figure 8b), that exhibit a high degree of invariance to illumination when compared to competing methods.

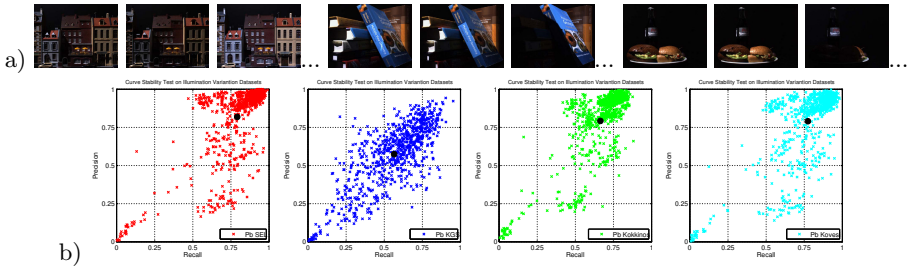

Fig. 8. Illumination Curve Stability Test: a) Several Images from the Robot Dataset [1] where images undergo illumination changes. b) Scatter plots where each dot represents an evaluation using [10] of curve fragments between two subsequent images.

Multi-View Stability Test: Fig 9 compares the stability of various curve extraction methods on multiple views of the same scene taking from slightly different poses.

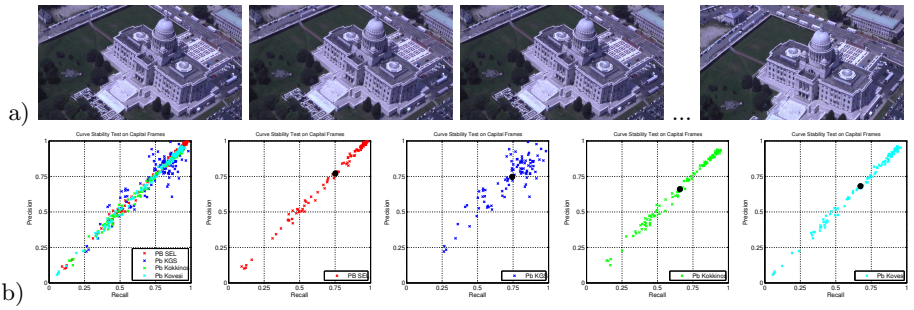

Fig. 9. Stability of curve fragments with slight changes in viewpoint is demonstrated on the Capitol Dataset images (a). Scatter plot where each dot represents an evaluation using [10] of curve fragments between two subsequent images is shown in (b).

Stability Test under Noise: A more sensitive evaluation is to test each contour extraction algorithm's stability under random noise. Fig 10 During this test, images are generated through adding five different level of Gaussian noise, and each algorithm's stability is measured by matching contour fragments generated from noise image to the curve fragments extracted from original image.

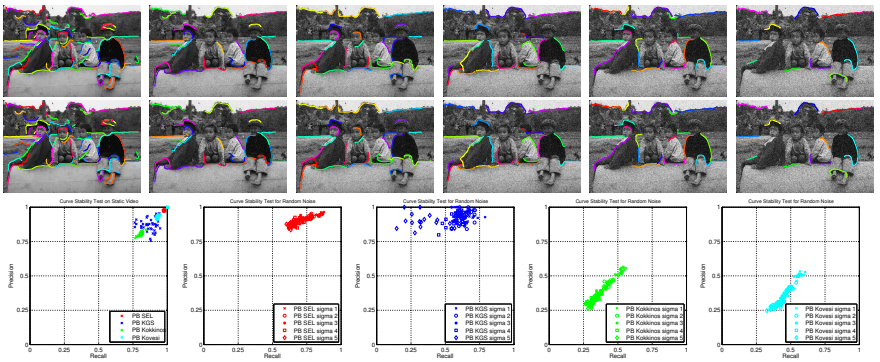

Fig. 10. Stability under noise: First row shows an image with five different levels of noise. Second rows shows extracted contours for each. Third row measure the stability of extracted contours. It is clear that our method produces the most stable results.

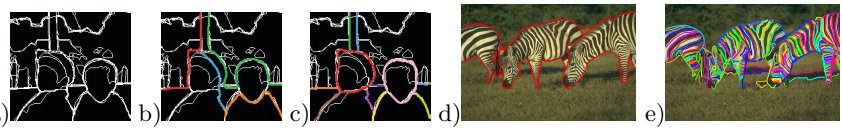

Fig. 11. The contour fragments shown in (b) and (c) represent two distinct groupings, yet at the edge map level (a), they are completely equivalent. BSDS evaluation takes into account edge position (not even edge orientation which by itself is very significant) but not which edges go together. This motivates a ground truth which explicitly represents ordering among edge points. BSDS only annotates silhouettes (d) as compared to CFGD [10] (e) which annotates all contours, including internal contours.

6.2 Evaluation Based on Human Annotation

The standard approach for boundary evaluation is the BSDS framework. This dataset and evaluation strategy is inappropriate for evaluating curve fragments for a number of reasons: *(i)* the human annotators are asked to delineate boundaries by painting regions. As such non-closed internal contours have no possibility to be annotated. (*ii*) The annotators are asked to delineate meaningful object silhouettes through semantic segments. Thus, they were discouraged from marking clearly perceivable contours, *e.g.,* zebra patterns in Fig 11. *(iii)* The BSDS evaluation evaluates edges not contours which are instead treated as a set of unorganized edges; *(iv)* even for edge evaluation only the match in edge location is checked but edge orientation, which is critical, is not included in the evaluation. Thus, random contours on a busy image have a good chance of matching ground truth. In fact, including edge orientation error in the evaluation changes the results and the rankings of algorithms.

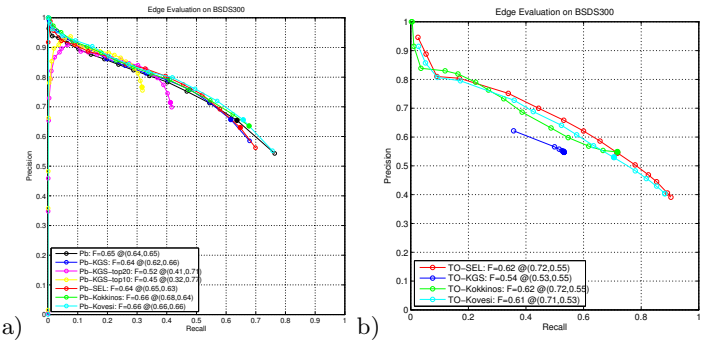

Fig. 12. Berkeley boundary evaluation on BSDS given the same edge input Pb [18] (a) and TO [27] (b), respectively, for a number of linkers. Applying our method on Pb does not improve the performance when compared to the raw Pb input. However, our full pipeline using TO with SEL results in higher recall compared to other methods. We note that KGS is limited on the total number of edges due to the limited range of the eigensolver system.

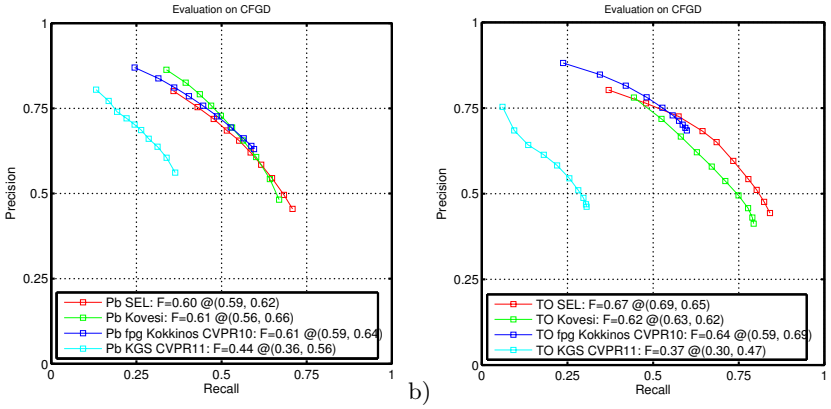

Fig. 13. Using the CFGD evaulation to compare the performance of several curve extraction algorithms using the Pb [18] (a) and TO [27] (b) edge input. Observe that SEL produces the highest recall for both edge maps.

6.3 Edge Evaluation on BSDS

Despite these shortcomings it is informative to evaluate the extracted contours based on the evaluation of edge detection. Using Pb [18] as the baseline and the input, we compare our algorithm's performance to KGS [13], FPG Kokkinos [14], Kovesi [15] in Fig 12a). The results show that the PR curves between Pb and Pb+Sel have nearly identical performance. This is due to the fact that the curve linking process maintains a high edge recall, throws out few edges. Identical performance between BSDS evaluation of edge detector output and edge linking output is consistent with other methods in Fig 12a), and can be seen from Fig 12 b) when comparing algorithms using the third order edge detector

(TO) [27]. KGS [13] gets much lower performance than the others because of its computational difficulty in solving eigensystem.

6.4 Curve Evaluation on CFGD

We rely on the curve evaluation framework introduce by CFGD [10] to evaluate edge grouping accuracy. Figures 13a and 13b compare contour extraction algorithms using retrained Pb and TO as a common edge input, respectively. While for the Pb input the algorithms perform rather similarly, for the TO input our algorithm (SEL) performs distinctly better, producing a higher recall rate. Visual comparisons qualitatively confirm this in Fig 14. It can be seen that SEL generates long smooth curve fragments covering most veridical boundaries while rarely groups boundaries from different objects or object parts together.

Fig. 14. A visual comparison of four curve extraction algorithms: from left to right columns: KGS [13] using Pb, FPG Kokkinos [14], Kovesi [15] and our method (SEL), all using TO. The selection of edge map type is based on which gives the best performance. Each is individually sifted by the veridicality probability threshold that gives the highest F-measure score. The color images are show in grey for better visualization.

7 Conclusion

There are three major contributions in this paper. First, a multi-stage approach is presented which relies on geometry in forming curve fragment pieces from edges and utilizes multiple curve-level cues in grouping over-segmented curve fragments as well as measuring the veridicality of curve fragments. Second, a hypothesis graph is proposed for the representation and management of grouping ambiguities in an approach based on "least early commitment" which demonstrably gives better contour fragments. Third, a learning-based approach and a novel and effective lateral edge sparsity cue is proposed that effectively reduces the false positive groupings and sifts out veridical curve fragments. The resulting

system is shown to lead to improved contour fragments when validated against human-annotated data, and which is stable with noise, viewing and viewing angle variations, and to moderate illumination variation.

References

1. Aanæs, H., Dahl, A.L., Pedersen, K.S.: Interesting interest points - a comparative study of interest point performance on a unique data set. International Journal of Computer Vision 97(1), 18–35 (2012)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 33(5), 898–916 (2011)
3. Baker, S., Nayar, S.: Global measures of coherence for edge detector evaluation. In: CVPR, pp. II:373–II:379 (1999)
4. Coughlan, J.M., Yuille, A.L.: Bayesian a* tree search with expected o(n) node expansions: Applications to road tracking. Neural Computation 14(8), 1929–1958 (2002)
5. Felzenszwalb, P., McAllester, D.: A min-cover approach for finding salient curves. In: Conference on Computer Vision and Pattern Recognition Workshop, CVPRW 2006, p. 185 (June 2006)
6. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. IEEE Trans. Pattern Analysis and Machine Intelligence 30(1), 36–51 (2008)
7. Fidler, S., Berginc, G., Leonardis, A.: Hierarchical statistical learning of generic parts of object structure. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 182–189. IEEE Computer Society (2006)
8. Fischler, M., Tenenbaum, J., Wolf, H.: Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. Computer Graphics and Image Processing 15(3), 201–223
9. Geman, D., Jedynak, B.: An active testing model for tracking roads in satellite images. IEEE Trans. Pattern Anal. Mach. Intell. 18(1), 1–14 (1996)
10. Guo, Y., Kimia, B.B.: On evaluating methods for recovering image curve fragments. In: Proceedings of IEEE Workshop on Perceptual Organization in Computer Vision, POCV, pp. 9–16 (June 2012)
11. Hu, W., Wu, Y.N., Zhu, S.C.: Image representation by active curves. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1808–1815 (2011)
12. Iverson, L., Zucker, S.: Logical/linear operators for image curves. PAMI 17(10), 982–996 (1995)
13. Kennedy, R., Gallier, J., Shi, J.: Contour cut: Identifying salient contours in images by solving a hermitian eigenvalue problem. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, pp. 2065–2072. IEEE Computer Society (2011)
14. Kokkinos, I.: Highly accurate boundary detection and grouping. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2520–2527. IEEE Computer Society Press, San Francisco (2010)
15. Kovesi, P.D.: MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia (2009), `http://www.csse.uwa.edu.au/~pk/research/matlabfns/`

16. Lin, L., Zeng, K., Liu, X., Zhu, S.C.: Layered graph matching by composite cluster sampling with collaborative and competitive interactions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1351–1358. IEEE Computer Society Press, Miami (2009)
17. Mahamud, S., Williams, L., Thornber, K., Xu, K.: Segmentation of multiple salient closed contours from real images. PAMI 25(4), 433–444 (2003)
18. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5), 530–549 (2004)
19. Narayanan, M., Kimia, B.: To complete or not to complete: Gap completion in real images. In: Proceedings of IEEE Workshop on Perceptual Organization in Computer Vision, POCV, pp. 47–54 (June 2012)
20. Payet, N., Todorovic, S.: From a set of shapes to object discovery. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 57–70. Springer, Heidelberg (2010)
21. Payet, N., Todorovic, S.: From contours to 3D object detection and pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 983–990 (2011)
22. Payet, N., Todorovic, S.: Sledge: Sequential labeling of image edges for boundary detection. International Journal of Computer Vision 104(1), 15–37 (2013)
23. Ren, X., Fowlkes, C., Malik, J.: Learning probabilistic models for contour completion in natural images. International Journal of Computer Vision 77(1-3), 47–63 (2008)
24. Sharon, E., Brandt, A., Basri, R.: Segmentation and boundary detection using multiscale intensity measurements. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, December 9-14, pp. 469–476. IEEE Computer Society Press, Kauai (2001)
25. Shashua, A., Ullman, S.: Structural saliency: The detection of globally salient structures using a locally connected network. In: ICCV, pp. 321–327 (1988)
26. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 731–737. IEEE Computer Society (1997)
27. Tamrakar, A., Kimia, B.B.: No grouping left behind: From edges to curve fragments. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE Computer Society, Rio de Janeiro (2007)
28. Widynski, N., Mignotte, M.: A particle filter framework for contour detection. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 780–793. Springer, Heidelberg (2012)
29. Yu, S.X., Shi, J.: Multiclass spectral clustering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 313–319. IEEE Computer Society, Nice (2003)
30. Zhu, Q., Song, G., Shi, J.: Untangling cycles for contour grouping. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1–8. IEEE Computer Society (2007)
31. Zhu, Q., Wang, L., Wu, Y., Shi, J.: Contour context selection for object detection: A set-to-set contour matching approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 774–787. Springer, Heidelberg (2008)