

No Grouping Left Behind: From Edges to Curve Fragments

Amir Tamrakar
LEMS, Brown University
Providence, RI 02912
Amir_Tamrakar@brown.edu

Benjamin B. Kimia
LEMS, Brown University
Providence, RI 02912
kimia@lems.brown.edu

Abstract

We present a framework for extracting image contours based on geometric and structural consistency among edge element locations and orientations. The paper presents two contributions. First, we observe that while the traditional edge orientation operators are based on first-order derivatives, orientation as tangent of a localized curve requires third-order derivatives. We derive a numerically stable third-order edge operator and show that it outperforms current techniques. Second, we consider all discrete n -tuples of edges in a local neighborhood (7×7) and retain those that are geometrically consistent with a third-order local curve model. This results in a number of ordered discrete combinations of edges, each represented by a bundle of curves. The resulting curve bundle map is a representation of all possible local groupings from which longer contour fragments are constructed. We validate our results and show that our framework outperforms traditional approaches to contour extraction.

1. Introduction

There is perhaps a common consensus that edge detection and contour extraction has reached its peak performance and while problems with its performance are acknowledged, solutions are sought in the way these edges and contour fragments are used (edges are the output of a local filter and contour fragments are a set of grouped, ordered edges.) For example, the shape context approach and the SIFT operator rely on the histogram of edge distributions without relying on the grouping itself [4, 11]. Beyond the appearance of beating on a dead horse, a second issue with continuing research in this area is that it is very difficult to gauge the extent of any improvements. The use of ground truth databases is fraught with difficulties. For example, in the Berkeley segmentation dataset, contours have been marked selectively, *i.e.*, only those enclosing a region, and only at the user selected set of scales. Thus, those meaningful edges and contours not picked up by the subjects fall

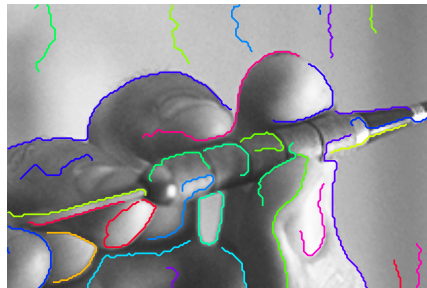


Figure 1. This figure illustrates a typical problem with edge linkers. The detected contours (randomly colored) are overlaid on the manually specified ground truth image (black), left, and over the original image, right. Observe the contour that starts at the index finger and then latches on to other spurious contours. This is the result of greedy local decision making strategies typically employed by contour tracers, which reduces the quality/utility of the detected contours for various vision tasks.

in the false positive category! Quoting a senior vision scientist, “any segmentation work is going to be five years of development and five years of convincing others it works!”

Despite these caveats, high-level vision tasks cannot avoid relying on curve fragments. Much of the recent research on patch-based methods for recognition indicates that the role of appearance needs to be augmented with the role of shape [18, 9, 15]. In our own work on perceptual organization and object recognition, we have been drafted to work on edge detection and grouping precisely because in current methods edges are often grouped across objects and many groupings are missed. We now report on two key insights which have led to a measurable and substantial increase in performance which is subjectively striking.

The first insight is that the orientation of an edge lying on a curve is computed using a derivative operation on the location of the edge. Thus, any orientation measurement should involve operators one order of derivative higher than those used in measuring its location. Edges are typically localized by finding peaks in response of some first-order derivative (gradient) operator which implies a second-order operation. By and large, however, the gradient direction, a first-order operator, is assigned as the orientation of the edge! This wrongful assignment of orientation, but not lo-

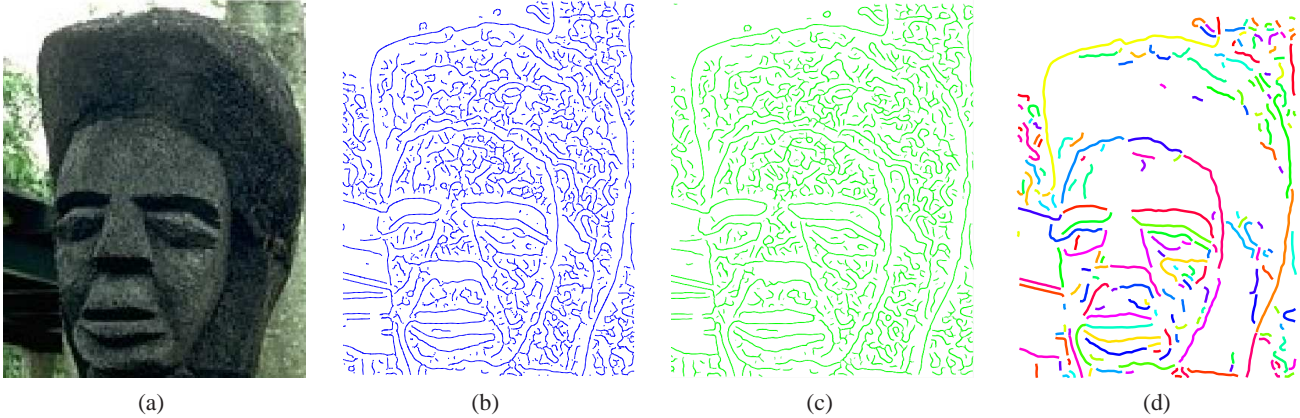


Figure 2. Our contour extraction process is illustrated on a real image. (a) The image. (b) The Third-order edge-map. (c) The curve bundle map is a collection of curves (Not visible as distinct at this resolution, see Figure 8b). (d) The extracted curve fragments.

calization, has led researchers to believe that orientation is not a reliable cue for grouping so it is either dropped altogether [16], or substantial allowance is given to its variability from the linking process and machinery is developed to fine-tune it [21]. We show that a third-order operator, with proper numerical precautions, can measure orientation reliably and that this reliability leads to a very different sort of grouping, one which counts more heavily on the orientation of the edge.

The second insight regards the issue of how globally salient, but locally weak groupings of edges can be retained as curve fragments, while avoiding the omnipresent problem of spurious curve fragments “leeching off” salient curve fragments (as noted in [1]), see Figure 1. We introduce an intermediate representation, the *curve bundle map*, which represents all geometrically viable local discrete combinations of edges. The curve-bundles ensure that all locally weak, but geometrically viable local combinations of edges are kept for a second round of grouping into longer curve fragments. Only a simple algorithm for the latter is implemented here. The term “local” refers to considering combination of edges in a small (7×7) neighborhood. The term “geometrically viable” derives from the assumption that image contours are piecewise smooth, so that only edge combinations satisfying a very weak curve smoothness model, *i.e.*, based on a third-order geometric Taylor series, are considered viable. Edges at or near a singularity will participate in discrete combinations on one or the other side, but probably not on both. The use of accurate orientation estimates, Figure 3, is key in retaining all viable groupings while discarding the spurious combinations.

These two key contributions, the third-order orientation filter and the geometric filter selecting all discrete combinations of edges in small neighborhoods, result in a rich curve bundle map of possible groupings from which curve fragments are constructed. As Figure 2 shows, the curve bundle map is a good visual representation of contour structure, but because all possible groupings are represented, there are

still many spurious possibilities. The goal of contour fragment construction is to retain the mutually consistent local discrete combinations or curve bundles.

There are at least two manually labeled datasets available to the community for evaluating edge detectors, the Berkeley dataset [13] and the Florida dataset [5]. The popular Berkeley dataset is not appropriate for evaluating our method since the tools and instructions given to the subjects restrict the results to isolated salient **closed** regions. Thus, the users could not label any open contours such as internal contours of incomplete occluding contours, even if they were salient. Evaluating even a perfect edge linker under such circumstances would lead to a relatively poor performance as properly delineated internal contours are labeled as false positive, thereby unfavorably affecting the ROC curves. We have, thus, chosen the dataset from the University of Florida, which contains 60 manually labeled ground truth images, to evaluate our method. We examine whether the use of the curve bundle map, which retains all possible local groupings of edges, can significantly improve the detection of reliable contours in images. Since most evaluation methodologies, including that of [5], are geared towards evaluating edge detectors, we have developed a new approach to evaluating linked edges – one that penalizes spurious detected contours when they link with “good” contours, see Section 5. These evaluations validate the visually striking improvements of our approach in contour extraction.

1.1. Related Work

The most commonly used strategy for contour detection is to trace edge pixels on the local 4- or 8-connected image grid, after having morphologically thinned the response of the detector [14]. The performance is thus limited by the output of the edge detection process itself, which is typically fragmented and noisy. More sophisticated versions of this kind of sequential tracing algorithms use multiresolution image pyramids to perform the sequential search [6].

Methods like Hough Transforms [3] are able to connect across large gaps since they are based on geometric consistency, but owing to their global nature they lack the sensitivity to the context in which the structure is embedded [7]. Furthermore, they are only good for simple, easily parameterized structures.

Token-based grouping algorithms [7, 17] are formulated in a local, parallel, and hierarchical framework. Edge tokens are locally grouped into larger tokens which are then further composed in a recursive manner into higher level and more global descriptors. First, local pairwise constraints are used to build a reduced search graph of edge tokens. Then local neighborhoods of this graph are analyzed for geometric consistency. Groups of tokens are replaced by a larger token employing a greedy best-first strategy.

In the perceptual grouping literature, much of the effort has been devoted to defining pairwise affinity measures between edges based on the Gestalt principles of proximity, smooth continuation, closure, *etc.*, and using them in a variety of schemes to compute the salience of edges. A good summary of these techniques can be found in [20]. Their main intent is to supplant the edge contrast measure with a measure of “figural goodness” so that spurious edges can be better detected and eliminated. Our work can be thought of as considering an n -wise affinity measure. The edge linking problem is only tackled by some of the approaches and these are usually limited to finding globally salient *closed* curves in the image [8, 12, 19]. Although figural closure is a reasonable global property to enforce for some types of curves, images also contain many open curves that are salient and need to be extracted. In our work, we draw on the insight from perceptual grouping concepts to form local groupings of edges based on continuity, to highlight salient edges and their neighborhoods prior to tracing the contours.

2. A Third-order Edge Operator

The traditional approach to edge detection using image derivatives localizes edges at the maxima of the gradient magnitude $|\nabla I|$ in the direction of the gradient $\frac{\nabla I}{|\nabla I|}$, which gives

$$\nabla|\nabla I| \cdot \frac{\nabla I}{|\nabla I|} = 0, \quad (1)$$

In Cartesian coordinates, Equation 1 can be written as

$$I_x^2 I_{xx} + 2I_x I_y I_{xy} + I_y^2 I_{yy} = 0, \quad (2)$$

which involves up to second-order derivatives. The orientation of this edge can be obtained by the normal to the edge,



Figure 3. The edge orientations computed using the third-order orientation operator, Equation 3, (shown in green) compared to those obtained from the gradient of the image (shown in red). That the results are dramatically better is evidenced by the self consistency of edge orientations as one travels along each curve, as opposed to the red edges which deviate significantly from the curve tangent.

\vec{N} , which can be obtained by differentiation

$$\begin{aligned} \vec{N} &= \begin{pmatrix} (I_x^2 I_{xx} + 2I_x I_y I_{xy} + I_y^2 I_{yy})_x \\ (I_x^2 I_{xx} + 2I_x I_y I_{xy} + I_y^2 I_{yy})_y \end{pmatrix} \\ &= \begin{pmatrix} 2I_x I_{xx}^2 + 2I_x I_{xy}^2 + 2I_y I_{xx} I_{xy} + 2I_y I_{yy} I_{xy} \\ \quad + 2I_x I_y I_{xxy} + 2I_y^2 I_{xyy} + I_x^2 I_{xxx} \\ 2I_y I_{yy}^2 + 2I_y I_{xy}^2 + 2I_x I_{xx} I_{xy} + 2I_x I_{yy} I_{xy} + \\ \quad 2I_x I_y I_{xyy} + 2I_x^2 I_{xxy} + I_y^2 I_{yyy} \end{pmatrix}, \end{aligned} \quad (3)$$

which involves up to third-order derivatives evaluated at the zero crossings indicated by Equation 2. For this reason we have chosen to call our detector a “Third-order orientation detector”.¹ Figure 3 shows the improvement in edge orientation obtained using this detector when compared to the gradient-based orientation. Furthermore, since these estimates are obtained locally and independently at each edge, they can be used for edge linking purposes without any bias.

Computation using third-order derivatives is notoriously difficult with small operator σ , so we have developed a robust numerical technique using “interpolating operators.” Specifically, as the scale of a Gaussian derivative operator, σ , becomes small and/or the order of the derivative increases, the Nyquist limit kicks in and prevents the proper representation of the derivative kernel. As a result, the derivative estimates suffer. To remedy this, we use an interpolating filter (linear is sufficient for third-order derivatives) and absorb it in the operator kernel.

We have evaluated the performance of this detector using Simon Baker’s Global Measures of Coherence (GMC) [2].

¹The source code for our third-order orientation detector has been made available at <http://www.lems.brown.edu/vision/researchAreas/EdgeDetection/Amir-Edge-detector.html>. The source code of for the edge linker described in the next sections will be made available soon.

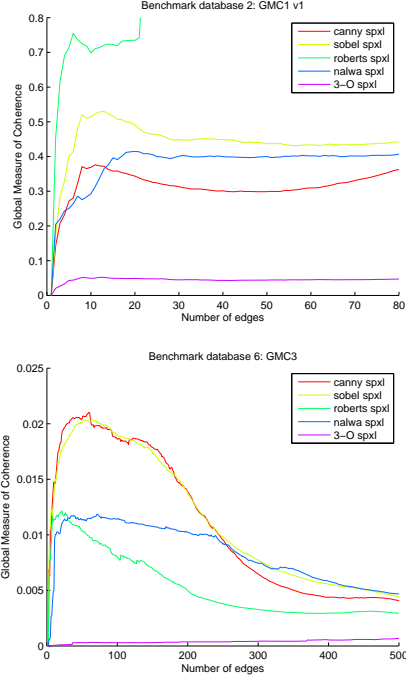


Figure 4. The GMC [2] measures on the third-order detector is compared with respect to other common edge detectors on a few benchmark datasets. Our operator outperforms all others.

This evaluation method consists of several benchmarks consisting of a large number of images of a carefully designed scenes for which a single constraint is enforced, *e.g.*, collinearity. The degree to which the constraint holds is estimated and used as a measure of performance. The performance measure is designed for detectors that produce precise sub-pixel localization and orientation estimation. A major advantage of the GMC measure is that it does not require precise ground truth, whose specification itself is prone to error. Figure 4 show the results from a few benchmark data sets. The third-order detector outperforms other detectors in a significant manner. The other benchmarks also show similar significant improvements.

3. Forming Local Hypotheses: Curve bundle Map

We make the assumption that image curves are *piecewise smooth* and that the total variation in curvature and its derivatives are bounded. Consider a geometric Taylor expansion of the curvature $k(s)$ of a smooth curve \mathcal{C} where s gives a parameterization of the curve,

$$k(s) = k_0 + k_1s + k_2s^2 + \dots \quad (4)$$

This local model for the smooth segments of image curves includes increasingly larger subclasses: (i) $k_0 = k_1 = k_2 = \dots = 0$ gives *lines*; (ii) $k_0 \neq 0$ and $k_1 = k_2 = \dots = 0$ gives *circles*; (iii) $k_0 \neq 0, k_1 \neq 0$, and $k_2 = \dots = 0$

gives *Euler Spirals* [10], *etc.* The Euler Spiral model is the lowest order curve model that allows for a geometric interpolation of two arbitrary edges. Thus, each pair of edges define a unique curve hypothesis and additional edges can be solicited to verify or disqualify this hypothesis. This is our basic notion of geometric consistency. Since each edge is computed with a some inaccuracy in position, δx , and in orientation, $\delta\theta$, (specified as parameters to the algorithm) each edge pair define a family of curve hypotheses.

To illustrate this better, consider a reference or *anchor edge*, and the set of all Euler Spirals through it, the *curve bundle* of all hypotheses, Figure 5a. Since the edge specifies a point and a tangent, the Euler Spiral requires two additional parameters to be uniquely specified, namely, its curvature k and the rate of change of curvature γ . In addition, the anchor edge is allowed to perturb up to Δx and $\Delta\theta$ in position and orientation, respectively. Thus, a four dimensional space is required to describe this curve bundle, which can be represented as a region in the $(\Delta x, \Delta\theta, k, \gamma)$ space. As a practical approach, we discretize the Δx and $\Delta\theta$ dimensions by three samples each, namely, $(-\delta x, 0, \delta x)$ and $(-\delta\theta, 0, \delta\theta)$, respectively. Having specified Δx and $\Delta\theta$, the corresponding curve bundle can be described by a region in the (k, γ) space, Figure 6d. This region is represented by a geometric polygon.

Consider next an edge pair with one edge as the anchor edge. The presence of the second edge on the putative curve hypothesis implies that only those curves of the initial curve bundle that are sufficiently close in position and orientation are viable, Figure 5b, thus pruning and significantly refining the initial bundle into a *pair-wise bundle*. This process can be continued by considering triplets of edges, two of which having just formed a pair-wise curve bundle. It is evident that the presence of a third edge on the putative curve hypothesis places an additional constraint which can signif-

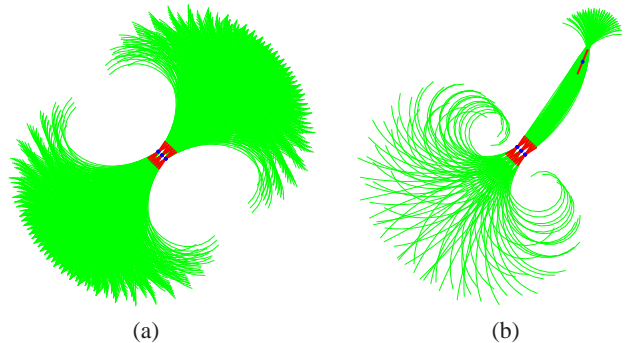


Figure 5. (a) The set of curve models through an edge is illustrated for an Euler Spiral model. It shows the bundles for the anchor edge with a perturbation in orientation, $\Delta\theta = 5$ degrees and in position, $\Delta x = 0.1$ pixels. (b) A second edge (top right) nearby the anchor edge (center) narrows the bundle only to those curves that are sufficiently close in position and orientation.

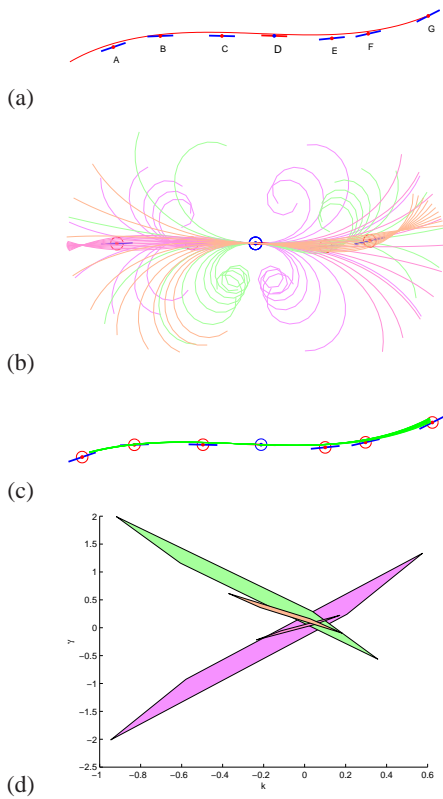


Figure 6. This figure illustrates how as the number of edges in a combination increases, the reliability of the curve bundle increases. (a) Edges are sampled from a curve and noise added. (b) Curve bundles from pairs anchored at D are shown in different colors. (c) A curve bundle is formed from the entire set of edges (a septuplet) and the corresponding curve bundle is very refined. (d) Each colored polygon represents a pairwise curve bundle in the (k, γ) space. The intersections represent larger groupings and shrinking curve bundles.

icantly remove additional curves from the bundle. In fact, given a triplet of edges, no curves may be left, in which case the triplet is discarded. For example, any edge not in the vicinity of the pairwise curve bundle in Figure 5b cannot form a triplet with this pair. On the other hand, placing an edge inside the green region, removes the majority of the curves from the bundle so that the triplet curve bundle becomes rather thin and selective. The mechanism for checking whether a curve belongs to a bundle is by checking membership in the region represented in the (k, γ) space of the anchor edge. First, all pairwise combinations of edges with the anchor edge are considered, each defining a region in the (k, γ) space. For efficiency, we next form triplets by intersecting pairs of regions; if the intersection is non-empty, quadruplets are considered, and so on until no higher order groupings can be considered. The resulting discrete combination of edges forms a *curve bundle*, Figure 6. The

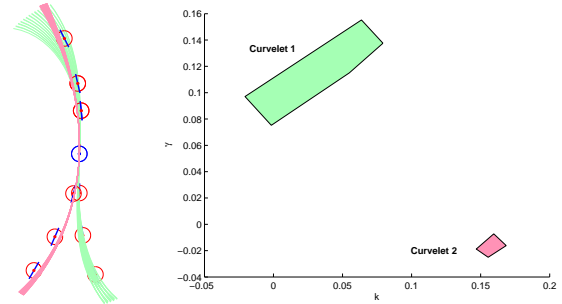


Figure 7. Alternate local curve hypotheses are explicitly maintained at edges by storing multiple curvelets each with its own curve bundle representation. In this example, the curve bundles are clearly distinct in the (k, γ) space.

intersection between regions in the (k, γ) space is computed by polygon intersection routines.

Each edge only maintains the n -tuples not subsumed in larger groupings, but **all** grouping options are retained (see Figure 7). The set of curve bundles associated with edges in an edge-map is called a *curve bundle map*, Figure 9, an intermediate representation between edges and linked curve fragments.

This process of forming curve bundles may be reminiscent of Hough transform in that groupings are represented and evaluated for consistency in the parameter space of a chosen parametric curve model, but in fact there are significant distinctions. First, our operations are very local, typically within a 7×7 window, involving on average only about 10-12 edges. Second, the curve parameter space is not discretized. Rather, curve bundles are represented by geometric polygons and intersections of these regions are computed by geometric polygon intersection routines. Third,

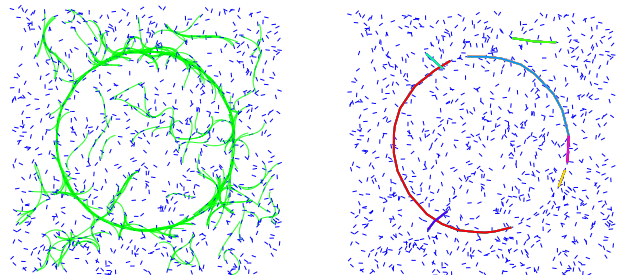


Figure 9. The utility of the intermediate curve bundle map representation is illustrated for a synthetic edge-map generated in the style of Williams and Thornber [20]. [Left] The curve bundles are shown in green. It is clear that geometric consistency of groups of edges have the capacity to eliminate most spurious edges. By keeping all the grouping hypotheses explicitly in the curve bundle map, we are avoiding making any local greedy decisions too early. [Right] The consistent curve bundles from the curve bundle map are then merged to form the curve fragment as described in Section 4. The missing section of the circle represents not a missing contour fragment, but an unresolved ambiguity in which combinations of curve bundles to retain.

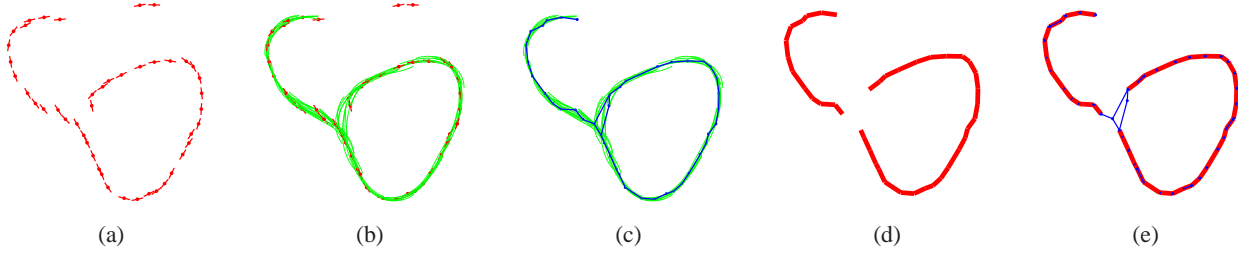


Figure 8. This figure illustrates the steps in our algorithm. (a) The edge map. (b) Curve bundles (shown in green) are formed locally around each edge. (c) Subsets of these curve bundles are used to construct the link graph (in blue). (d) Unambiguous 1-chains in the link graph (shown in red) represent contour fragments. (e) The contour fragments along with the remaining ambiguous portions of the link graph forms the contour fragment graph.

each additional edge actually rules out curves from the putative bundle that are inconsistent with it, rather than voting for some. And finally, instead of finding a single best curve from the parameter space, a whole bundle of consistent curves is reported.

4. Integrating Local Hypotheses: From Curve bundles to Curve Fragments

The curve bundle map, Figure 8b, exhaustively represents all possible local groupings of neighboring edges for each edge together with curve bundles indicating the extent of variations of the grouped edges and its geometry. Most significantly, the grouping represents an *order* for edges along the hypothesized curves. Thus, each curve bundle through an anchor edge hypothesizes a left and right neighbor. For such a neighborhood relation to be viable, at a minimum the reciprocal relationship must hold, *i.e.*, each edge should be the right neighbor’s left neighbor in some curve bundle and vice versa, Figure 10. The set of all such

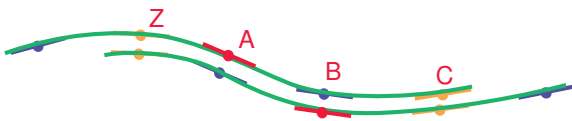


Figure 10. The overlaps between two curve bundles, one anchored at *A* and one at *B* is shown. These curve bundles are reciprocally consistent, meaning *A* links to *B* and *B* links to *A*. In fact, they are consistent to a larger degree since they agree over four edges.

edge neighborhood relations can be efficiently described by a graph, the *edge link graph*, whose nodes are edges and whose links represent viable connectivity between the edges as evidenced through consistent curve bundles, Figure 2c. Thus all potential curve fragments can be represented by paths in the link graph. In fact, most algorithms start off by first constructing this kind of graph by connecting each edge to all its neighbors. The goal of finding the curve fragments is then posed as a problem of either finding best paths in the graph or as a problem of pruning spurious links from the

graph. Our approach differs markedly in that, we first form the viable local paths as curve bundles and subsequently form the links in the graph only if there is agreement between them.

While the set of curve bundle hypotheses for any given edge is large, there is a significant degree of general agreement between the curve bundle hypotheses of neighboring edges. In practice, however, the test of reciprocity is not stringent enough to produce unambiguous paths in the link graph. We, therefore, mandate a larger degree of overlap in the participating edges of the curve bundles before allowing a link to form between edges, Figure 10. The degree of overlap required to form a link is set as a parameter but is typically set to four edges. In many cases, the neighboring relation of an edge is unanimously decided by all curve bundles through it. The resulting link graph contains many unambiguous 1-chains which represent a set of *curve fragments*, Figure 2d. However, “gaps” appearing in the curve fragments are in fact ambiguities not resolved in the above stringent method. We have yet to explore how to “iron out” the bundle of multiple ambiguous representations, but visually the grouping is compelling.

Per our philosophy of “no groupings left behind”, we retain the remaining ambiguous portions of the link graph, by constructing a *curve fragment graph*, which is a transformation of the link graph where edges belonging to unambiguous 1-chains have been replaced by curve fragments, Figure 8e. The curve bundles associated with the remaining links are also kept for future processing. The curve fragment graph is the final output of our algorithm.

5. Evaluation

We have evaluated our algorithm on a manually specified ground truth data set obtained from University of South Florida (USF) (<http://figment.csee.usf.edu/edge/>) [5]. This data set consists of 50 images containing essentially unoccluded objects of various kinds set against a natural background and 10 aerial images. Each image has a corresponding ground truth (GT) overlay image which has been man-

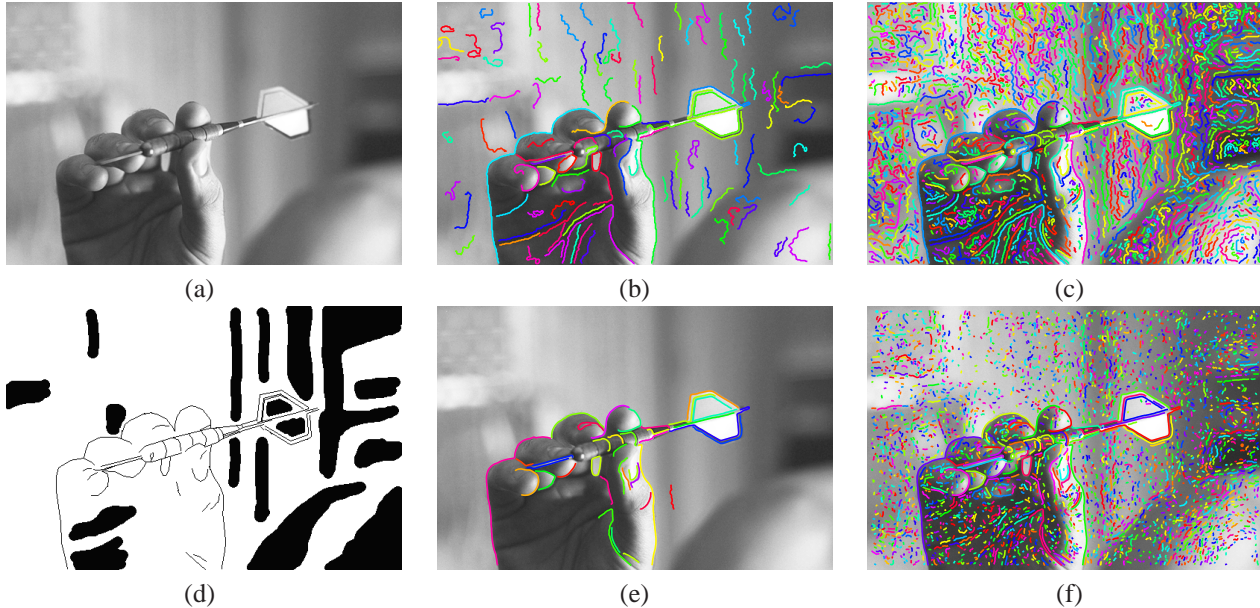


Figure 11. (a) shows a sample image from the USF database. The manually labeled GT image corresponding to it is shown in (b). The contours extracted by the generic detector and our algorithm at 90% TP is shown in (c) and (d) respectively. At this rate of detection, there are a lot of spurious contours with both algorithms. (e) and (f) shows the contours extracted by the generic linker and our algorithm respectively at 75% TP. The difference in the quality of the detectors is obvious. A point to note is that not all the contours shown are penalized. Only the ones attached to “good contours” (see text for definition) are penalized. We have included many more images from the dataset in the supplemental material document.

ually labeled with “edge” pixels, “no-edge” regions and the rest are marked as “don’t-care” regions, Figure 11b. They have used the following strategy for evaluating edge detectors. If a detector reports an edge pixel within a specified tolerance, *e.g.*, $T_{match} = 3$ pixels of an edge in the GT, it is counted as a true positive (TP). If a detector reports an edge pixel in a GT “no-edge” region, it is counted as a false positive (FP). Edges reported in the “don’t-care” regions don’t count. An ROC curve is made for the given image and detector from the (TP, FP) points from different parameter settings. For comparing performance of different detectors on the entire data set, they compute aggregate ROC curves by averaging FP values of the individual ROC curves at fixed TP values.

While this approach makes perfect sense for evaluating edge detectors, it is not sufficient for evaluating edge linkers. As mentioned earlier, one of the shortcomings of edge linking algorithms is that “good” curves tend to be linked to spurious curve fragments thereby reducing their quality for various vision tasks, Figure 1. Therefore, although the edges detected in the “don’t-care” regions are ignored as irrelevant while evaluating edge detectors, they need to be penalized in the context of edge linking if they latch on to “good” contours. We have, therefore, modified the evaluation criteria to account for such an event.

Specifically, we first classify the detected curves into three classes: (i) “good” curves, (ii) “bad” curves, and (iii) “irrelevant” FP curves. If more than 50% of the edges in a de-

tected curve overlaps with the GT edges, it is considered to be a “good” curve. All the edges in such curves, that do not overlap with GT edges are penalized and counted as FP, even if they lie in the “don’t care” regions of the GT. Curves lying mostly in the “no-edge” regions are labeled as “bad” curves and all edges in them are counted as FPs. And finally, curves detected mostly in the “don’t care” regions are deemed “irrelevant” and simply ignored. We also indirectly penalize fragmentation of contours by discarding all contours shorter than a specified length threshold prior to evaluation. This is meaningful for evaluation since short contours are routinely discarded after linking.

We have compared our contour detector against a popular generic contour detector. Since the Canny edge detector is the most widely used and is rated favorably in detection evaluations [5], we have chosen it as the front end for the generic contour detector. To keep it generic, we used the implementation provided in Matlab. The smoothing σ of the detector was fixed at 1.0 for all images while the t_{high} threshold parameter was varied for the purpose of computing the ROC curves. As for the edge linking end of the generic contour detector, we also wanted to use publicly available code (and there aren’t very many). For this paper, we have used Peter Kovese’s edge linking code downloaded from <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/edgelinek.m>. This code is basically an edge tracer using the 4- or 8-connectivity of the image grid.

As for our contour detector, we used the third-order edge detector described in Section 2 followed by the grouping and linking described in Sections 3 and 4. The operator σ was also fixed at 1.0. However, we followed a different methodology for specifying the threshold. We first used a very small threshold to produce a dense edge-map. After the contours were extracted, we computed average contrast for each fragment by averaging the contrast of the component edges. The threshold parameter was then enforced so that the entire contour was discarded if the average contrast fell below threshold.

To produce the ROC curves, we varied both the contrast threshold and the contour length threshold. But unlike in [5], we did not follow the train-test methodology due to the time constraint. All reported ROC curves are simply aggregate training ROCs. In our defense, the authors of [5] have suggested that the training ROC curves should be enough for comparison when there are only a few parameters.

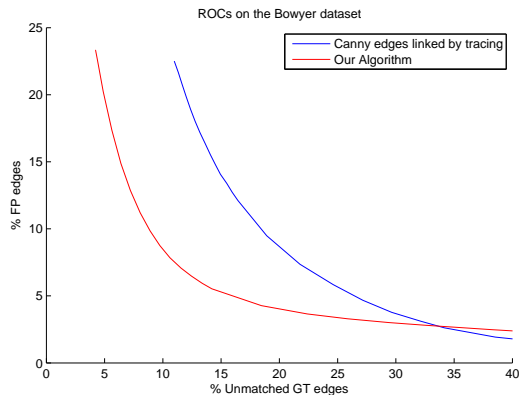


Figure 12. The aggregate ROC curves for the generic edge linker and our algorithm on the USF data set is shown. In this type of ROC, the point at (0,0) is the best operating point. Our algorithm has fewer FP edges for the same number of TP edges (% Unmatched-GT-edges is equivalent to 1-%FP edges).

Discussion

The aggregate ROC curves obtained for the USF dataset are presented in Figure 12. It is clear that our contour detector has outperformed the generic contour detector in a significant manner. To understand the meaning of the differences in the ROC curves, we present the contours extracted by the two detectors at 75% TP rate and 90% TP rate, see Figure 11. Observe that at 75% TP rate, the contours from our detector appear much cleaner than from the generic detector. At 90% TP rate, our detector also picks up a lot of spurious contours, but they are nowhere as messy as the contours produced by the generic contour detector at the same TP rate. Due to space constraints, we weren't able to add more comparisons into this document. However, a more extensive sampling of the differences can be found on our web site.

Acknowledgement: We gratefully acknowledge the support of NSF grant IIS-0413215.

References

- [1] T. Alter and R. Basri. Extracting salient contours from images: An analysis of the saliency network. *CVPR*, pages 13–20, 1996.
- [2] S. Baker and S. Nayar. Global measures of coherence for edge detector evaluation. In *CVPR*, pages II: 373–379, 1999.
- [3] D. Ballard and C. Brown. *Computer Vision*. Prentice Hall, 1982.
- [4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.
- [5] K. Bowyer, C. Kranenburg, and S. Dougherty. Edge detector evaluation using empirical ROC curves. *CVIU*, 84(1):77–103, 2001.
- [6] G. Cook and E. Delp. Multiresolution sequential edge linking. In *ICIP*, pages I: 41–44, 1995.
- [7] J. Dolan and E. Riseman. Computing curvilinear structure by token-based grouping. In *CVPR*, pages 264–270, 1992.
- [8] J. Elder and S. Zucker. Computing contour closure. In *ECCV*, pages I:399–412, 1996.
- [9] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *CVPR*, pages II: 90–96, 2004.
- [10] B. Kimia, I. Frankel, and A.-M. Popescu. Euler spiral for shape completion. *IJCV*, 54(Issue 1-3):159–182, 2003.
- [11] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [12] S. Mahamud, L. Williams, K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. *PAMI*, 25(4):433–444, 2003.
- [13] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *ICCV*, 2:416–423, 2001.
- [14] V. Nalwa and E. Pauchon. Edgel aggregation and edge description. 40(1):79–94, October 1987.
- [15] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, pages II: 575–588, 2006.
- [16] C. Rothwell, J. Mundy, W. Hoffman, and V.-D. Nguyen. Driving vision by topology. In *ISCV*, pages 395–400, 1995.
- [17] E. Saund. Labeling of curvilinear structure across scales by token grouping. In *CVPR*, pages 257–263, 1992.
- [18] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, pages I: 503–510, 2005.
- [19] S. Wang, T. Kubota, J. Siskind, and J. Wang. Salient closed boundary extraction with ratio contour. *PAMI*, 27(4):546–561, 2005.
- [20] L. Williams and K. Thornber. A comparison of measures for detecting natural shapes in cluttered backgrounds. *IJCV*, 34(2/3):81–96, 2000.
- [21] S. W. Zucker, A. Dobbins, and L. Iverson. Two stages of curve detection suggest two styles of visual computation. *Neural Comput.*, 1(1):68–81, 1990.