Image Objects and Multi-Scale Features for Annotation Detection

Jindong (JD) Chen*, Eric Saund, Yizhou Wang †
Palo Alto Research Center
{jchen, saund, yizhou.wang}@parc.com

Abstract

This paper investigates several issues in the problem of detecting handwritten markings, or annotations, on printed documents. One issue is to define the appropriate units over which to perform feature measurements and assign type labels. We propose an alpha-shape tree that operates across multiple scales. A second issue is to devise image features that offer inferential power for machine learning algorithms. We report on a feature that measures edge turn statistics. A third issue is how to combine local and neighborhood evidence. We exploit the alpha shape tree in a direct inference architecture. Information propagation schemes such as Markov Random Fields may be readily layered on top of our output.

1 Introduction

The problem of detecting handwritten signatures, initials, and other annotations on printed documents has many applications in digital libraries and the document scanning and conversion industry.

Previous work in this area shows that annotation detection algorithms can be designed to operate at different levels of granularity. One idealized form of output can be called *layer separation*, where every pixel is labeled with the causal image marking explanation for the color value it takes. For example, a pixel's labeling might draw from the label set, { PAPER, MACHINE_PRINT, HANDWRITTEN_PENCIL, HANDWRITTEN_PEN, MULTI_LAYER, SCANNER_NOISE, UNKNOWN }. Bloomberg [2] used image morphology techniques to construct masks corresponding to labels of this sort. Recently An et al explored pixel labeling of document images [1] (although not in an annotation detection context). Both of these approaches may label not just pixels belonging to the marking themselves, but a fuzzy "region of influence" around textual elements.

Other methods for annotation detection operate at a coarser grain. A persistent question is, what constitutes an appropriate larger scale chunk, or *atomic image element* and how can these be computed? A natural starting point is connected components. These mostly correspond to characters in machine printed text, and to either characters or words in handwriting. However, it is common for handwriting to overlap machine print, so connected components are really too crude to form a foundational layer for annotation labeling. Moreover, Zheng et al found that local features measured on connected components were insufficiently distinguishing to support their Markov Random Field approach to annotation detection [8] . So they perform image processing and grouping to collect word-size blob objects to serve as the atoms over which their labeling and belief propagation algorithms operate. Shetty et al did likewise for a CRF approach [7]. Algorithms for constructing word blobs are heuristic and remain a point of art in the field, not systematic and replicable knowledge.

The issue of computable local features that might provide evidence about machine-print vs. annotation is closely related to the granularity issue. In the case of binary and even greyscale and many color scans, the immediate region around any pixel is not very informative. Deciding machine print vs. annotation requires looking at some neighborhood context. Obviously one can apply image filters of any size and shape, localized at any pixel. The larger the filter, the less spatially precise is their output. The purpose is not to gather information from arbitrary or uniform larger support regions, but to control and limit these support regions to only relevant data. This is especially critical when handwriting and machine printing are in close proximity. The sharp boundaries of a connected components or atomic word blobs are convenient in confining the support of feature measurements to data about that element alone.

*Now at Willow Garage, Inc, jdchen@willowgarage.com
†Now at Institute of Digital Media, School of Electronics Engineering and Computer Science, Peking University, Yizhou.Wang@pku.edu.cn

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

We propose to address these issues through the use of a fine-to-coarse aggregation device called the *segmentation tree*. The segmentation tree is constructed using alpha shapes, which are a generalization of the Delaunay Triangulation. Starting with connected components or fragments created by fracturing connected components into smaller parts, the construction of the segmentation tree using alpha shape triangulation offers a well-defined method for grouping fragments into successively larger chunks. Each of these chunks clearly demarks a support region in the image.

A second contribution of this paper is an exploration of a particular type of image feature, among many that could be performed on alpha shape chunks. This is the collection of edge turn statistics of foreground/background contours. While not definitive, this feature is surprisingly informative about the machine print vs. handwriting status of image regions, across many cases.

2 Alpha Shape Based Segmentation Tree

Alpha shapes are a generalization of the convex hull [4]. Given a point set, An alpha shape can be constructed by removing from the Delaunay triangulation all triangles whose radii of circumcircles are larger than α Fig. 1(a). Hull links are those forming borders between remaining triangulated points and free space. Depending on the value of alpha, hulls demark groups of points depending on their density, or spacing. alpha shapes offer a well-defined mechanism for analyzing document images at different scales(Fig. 1(b)).

A hierarchical tree of alpha-shape nodes can be built efficiently by constructing alpha triangulations with α values pre-selected on an empirical basis. Leaf nodes are formed by connected regions under the smallest alpha value. Nodes are joined to become children of non-leaf nodes in the hierarchy when they are subsumed by a connected region under larger alpha value.

For segmentation of a binary image into chunks at different scales, it is sufficient to sample points from the boundary of connected components, or else fragments of connected components.

We have observed that, by choosing a list alpha values carefully, nodes of the alpha shape segmentation tree often correspond to layout elements of a document page. For example, for a 150 DPI page image, the shapes of alpha values of 6, 12, and 24 (assuming the size of a pixel is 1 by 1), in many cases form connected regions that coincide with words, lines, and paragraphs. Fig. 1(c) shows an example for some handwritten text.

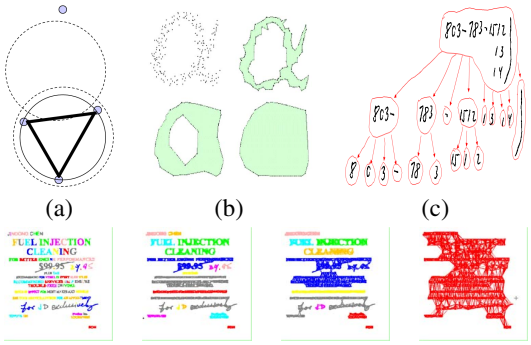


Figure 1. (a) The triangle formed by the lower 3 points belongs to the alpha shape because the its circumcircle is smaller than the circle of radius α (the dotted circle); (b) Changing α (from 0 to ∞) controls the level of details in alpha shapes. (c) A segmentation tree of a patch of handwritten annotation. The bottom row shows the layers of a segmentation tree.

3 Edge Turn Features

The segmentation tree creates well-defined chunks of image material, across scales, on which to measure features that might distinguish handwriting from machine print. We have use most of the features described in [8]. Here, we introduce a new type of feature, the *edge turn feature*.

Intuitively, the paths taken by image contours might be expected to be informative because machine printed characters often contain straight contours, and they often contain repeatable forms. For example, characters in a font either include serifs or they don't, and if they do, they use the same shapes.

Conceptually, *edge turn features* measure the frequency of certain sequences of angle changes in a sequence of contour edges. In our implementation, we have chosen to construct them based on the *mid-crack chain code representation*[3]. Considering each pixel as a square, a mid-crack chain code encodes the sequence of midpoints of black-white pixel boundaries. See Fig. 2(a)(b). At each point along a mid-crack path, at most three turn directions are possible. This makes it possible to enumerate possible paths to some depth as a trie, and to compile statistics on observed paths in image data.

We examine edge sequences of length l from each edge of a fragment, and recode the sequence of edge turns in a sequence of $l-1$ of digits from $0, 1, 2$.

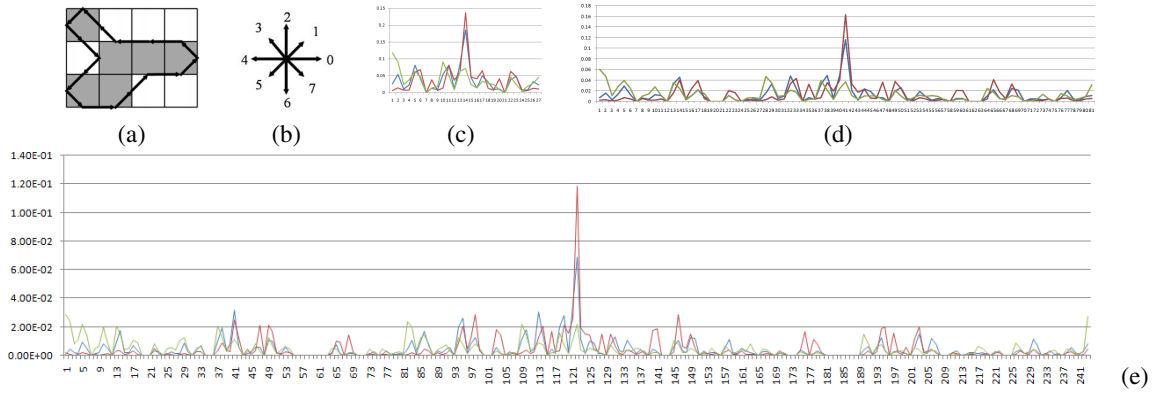


Figure 2. (a) Illustration of a mid-crack chain code of a shape. (b) Coding of direction. The chain in (a) is encoded as (from right most point) 3443357755701101. The edge turn code is 0112000220010210 (left=0, straight=1, right=2, w.r.t the edge of the pixel) (c)-(e) Turn sequence histograms, arranged in lexicographic order of the coding, of handwriting (in blue), machine printed (in red) and noise(in green), of lengths of 4, 5, and 6, respectively.

Fig. 2(c)-(e) shows the distributions (in lexicographical order) of the edge turn sequences of machine printed fragments, handwriting fragments and noise fragments. Not surprisingly, we find that the distribution of turning paths for machine printed text has peaks near the center of the plot, which corresponds to straight paths. Noise shows a more even distribution. Handwritten fragments fall in between.

We construct two kind of features from edge turn sequences. The first kind of is simply of value of a bucket in the edge turn distribution of a fragment. There are 3^{n-1} features extracted by examining edge sequences of length n.

The second kind of feature is a likelihood score measured by the similarity between the edge turn sequence distribution and a class model. For each class C, we construct an edge turn distribution model $D(C, l)$, of certain l, the length of edge sequence, by simply computing the distribution over a set of training examples. And the likelihood score of a fragment f belonging to class C is $D(f, l) \cdot D(C, l)$, the inner product of the two distributions.

We measure these two kinds of features for l equals to 2 through 6, and for two ($|R| = 2$) level resolutions, the input resolution and a downsampled resolution by a factor of 2. If we are only concerned about $|C| = 3$ classes, the total number of edge turn features is $(\sum_{l=2}^{6}(|C|^l + |C|)) \times |R| = 756$.

Notice that the above edge turn features do not take into consideration the initial orientation of the edge sequence. Ignoring the initial direction makes the features relatively independent of text orientation, and helps reduce the dimensionality of the feature space.

4 Tree-Based Inference

The segmentation tree defines neighbors relations in direction and scale. We compute output labels for each leaf node in the segmentation tree using inference techniques that combines inputs from its neighborhood.

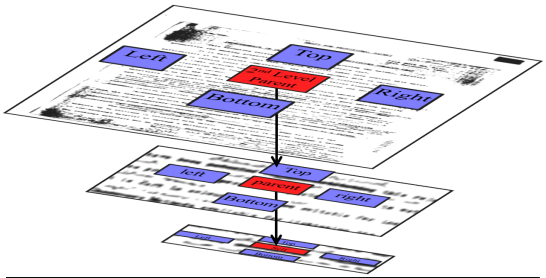
The contextual neighborhood a node in the segmentation tree, N, is the set of sibling nodes roughly to the left, right, top and bottom(denoted as N_{left}, N_{right}, N_{top}, N_{bottom}). If it exists, its parent node, N_p, and its grandparent node, N_g, and the left, right, top and bottom sibling of them (denoted as $N_{p,left}$, $N_{p,right}$, $N_{p,top}$, $N_{p,bottom}$ and $N_{g,left}$, $N_{g,right}$, $N_{g,top}$, $N_{g,bottom}$, respectively), if they exist. So the contextual neighborhood includes at most 15 nodes (See Fig.3). For notational simplicity, we use the subscripts 1 to 15.

A likelihood model measures the likelihood of certain model configuration, given the observations. Given a segmentation tree T, the likelihood of a certain label assignment A can be modeled in the following generative model,

$$P(A, T) = P(T)P(A|T) = P(T)\prod_i P(A_i|T_i), \quad (1)$$

where T_i is a leaf node (terminal node) of the T, and A_i is a label assigned to T_i.

For each leaf node in a segmentation tree, we used the classification scores of each of the 15 nodes in its



	Grandparent Layer					Parent Layer					Self Layer				
	C	L	R	T	B	C	L	R	T	B	C	L	R	T	B
M	4	0	0	0	0	4	2	2	.5	.5	4	2	2	.5	.5
H	1	0	0	0	0	5	2	2	.5	.5	4	1	1	.5	.5
N	1	0	0	0	0	5	2	2	2	2	3	1	1	1	1
U	1	0	0	0	0	2	0	0	0	0	5	0	0	0	0

Figure 3. Illustration of the contextual neighborhood of 15 nodes. The table shows the 15 λ's for each of the classes, machine-printed(M), Handwriting(H), Noise(N) and Unsure(U). Within each layer, the λ's are listed as center(c), left(l), right(r), top(t), bottom(b). Each row is subject to normalization over the sum of all 15 λ's.

Confusion Matrices

	1st cascade predictions				2nd cascade predictions			
	M	H	N	U	M	H	N	U
M	9731	654	66	151	208	15	3	17
H	18	519	8	1	11	461	7	16
N	191	192	2354	429	5	17	42	13
U	44	77	2	1864	18	34	5	352

Table 1. Confusion matrices of 2 cascades of classifier for machine printed (M), handwriting (H), noise(N) and unsure(U).

contextual neighborhood (including itself), and features based on their compatibility measurements to perform likelihood model based classification. We use the following likelihood model for each leaf node.

$$P(A_i|T_i) = c\exp\{-\sum_{k=1}^{15}\lambda_k S(N_k)\} \qquad (2)$$

where $S(N_k)$ is the local classification score from each of the 15 neighborhood nodes, the λ_k's are parameters to be estimated, and c is a normalizing factor.

5 Experiments and Discussion

We use a combination of 3 data sets. *The Maryland Set*: Professor Doermann of University of Maryland has kindly provided us a labeled set of 94 images. *The ICDAR07-HW Set:* We include the training set of the ICDAR 2007 Handwriting Segmentation Competition, which includes 20 images. *The BAT Set:* We also use 35 images downloaded from the British America Tobacco Database. In training, we use 54 images from the Maryland set (Train Set 1) and the whole ICDAR07-HW Set (to increase the number of handwriting examples). We use the rest as test data.

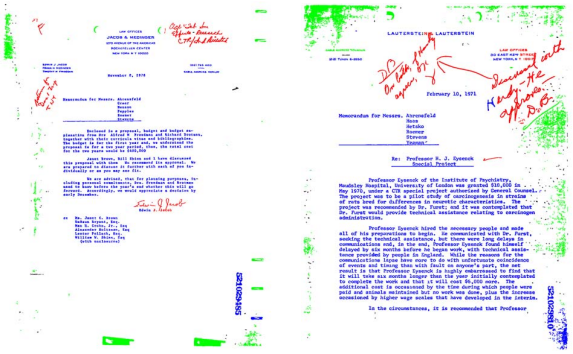


Figure 4. Testing results. Connected components are color coded as blue, red, green, and black, for machine printed, handwriting, noise and unsure, respectively.

We define 4 classes MACHINE-PRINTED, HANDWRITTEN, NOISE, and UNSURE. We introduce the class UNSURE to capture patches that are ambiguous if examined without context, e.g. small strokes and dots.

We build a 2 Level Cascade, associating handwriting false negative with high cost (see Table 1). The 1st Cascade is trained with patches generated from all training data. We select the top 300 features from a set of 888 by correlation-based method [6], and then train a LogitBoost model [5] with 108 iterations. The 2nd Cascade is trained with patches from Train Set 1 that are classified as handwritten by 1st cascade. We train with 36 iterations, without further feature selection.

Fig. 4 shows some output images from the tree-based inference, the last step. Currently, we hand-tune the λ_k's in Equation 2 for each class (see the table in Fig. 3). For machine printed fragments, as the texts tend to be in horizontal lines, neighbors to the left or rights are more informative; for noise, neighbors in all directions are equally important. The neighborhood of the grandparent layer, which measures page wide features, are not very informative, while that of the parent layer, which measures features over patches roughly in size of words,

is most informative. As appropriate to our applications, we measure the performance by the number of "isolated groups" (e.g. the left image in Fig. 4 has 4 groups; the right has 3) – currently done by manual inspection. In testing, the system detects 108 of the total 112 groups of handwriting (96.4%), with 14 groups of false positives. While the preliminary results have shown the effectiveness of the method, some of the machine-printed text (e.g. upper right region of the right image) are misclassified as handwriting, as they partially share the same contextual neighborhood as the handwriting nearby. We are working on schemes to learn the λ_k's by training, and automatic performance measurement methods that are meaningful to applications.

In conclusion, we propose an alpha shape based tree scheme that operates across multiple scales, a feature that measures edge turn statistics, and we exploit the alpha shape tree in a direct inference architecture. Preliminary results have demonstrated effective detection of handwritten annotations.

References

[1] C. An, H. Baird, and P. Xiu "Iterated Document Content Classification", *ICDAR 2007*.

[2] Dan S. Bloomberg, "Segmentation of Handwriting and Machine Printed Text." *US Patent 5,181,255*, 1993, filed Dec. 13, 1990.

[3] K. A. Dunkelberger and O. R. Mitchell, "Contour tracing for precision measurement," *Proc. IEEE Inter. Conf. Robotics and Automation*, pp. 22-27, 1985

[4] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. "On the shape of a set of points in the plane." *ACM Trans. Information Theory*, TI-29(4):551-559, 1983

[5] J. Friedman, T. Hastie, and R. Tibshirani. "Additive logistic regression: a statistical view of boosting", *Ann. Statist.*, 28(2):337-407, 2000.

[6] M. A. Hall, "Correlation-based Feature Subset Selection for Machine Learning". *Ph.D diss.*, Waikato Univ. Hamilton, New Zealand.

[7] Shravya Shetty, Harish Srinivasan, Matthew Beal, and Sargur Srihari, "Segmentation and labeling of documents using conditional random fields", *SPIE* v.6500, 2007

[8] Y. Zheng, H. Li and D. Doermann, "Machine Printed Text and Handwriting Identification in Noisy Document Images" *IEEE TPAMI* March 2004 (Vol. 26, No. 3) pp. 337-353.