

Postal address block location by contour clustering.

Venu Govindaraju and Sergey Tulyakov
Center of Excellence for Document Analysis and Recognition (CEDAR)
Department of Computer Science & Engineering
UB Commons, Suite 202
Amherst, NY 14228-2567

Abstract

We have developed a well performing algorithm for locating address blocks in postal parcel images. Both machine printed and handwritten addresses are processed by the algorithm. The algorithm is invariant to the image orientation and scale, and it works with high noise images. It could also serve as an additional step after other address block location algorithms.

1. Introduction.

Automated machine processing and sorting of mail has become an important part of mail delivery systems. Address block location (ABL) is one of the main problems facing developers of OCR based mail processing. Scanned images of mail pieces show a big diversity of content and style. Since word and character recognizers are trained to recognize only words and characters, any extraneous information would easily confuse them. Thus successful ABL is a necessary step in machine mail processing.

In this work we are trying to create an ABL algorithm to handle diverse types of mail addresses. The main feature of our algorithm is the clustering of connected components of the image. Most ABL algorithms in essence cluster areas of predefined size of the image. To use such algorithms the proper scale, or the size of areas in which original image is split, should be specified. If the structure of documents is known, then it is possible to choose good scale.

Unfortunately, the set of images which we had did not have any good structure defined on it. These were the images scanned from postal parcels and automatically pre-segmented. Thus images could contain destination address block together with abundant extraneous graphics and writings, well separated address block, or some graphics but no address block or partial address block. Basically we needed a function to segment address block if it is present, and to leave already segmented address block intact.

In our algorithm we considered the information about connected components in the image. The address block usually consists of similar connected components, that is their position, size, width and possibly some other features are close to each other. Thus after extracting features of the connected components and clustering resulting feature vectors, it is probable that all components of the address block will belong to one cluster, and no foreign components will be there.

2. Previous work.

Earliest approaches to address block location problem were usually variations of knowledge-based systems. In such systems a complex set of rules was specified, and different functions were developed for verification of each rule. Examples of such systems could be found in [12], [9],[8].

Few algorithms try to separate text regions of the image. Transition analysis [7] technique is trying to find the regions with evenly spaced transitions between white and black areas. Jain et al. are using Gabor filters[4, 5] to find the text regions of the image. When the text regions are identified some rules are applied to find the address block location.

Lii et al.[6] present a technique of recognizing address words and subsequent labeling of recognized objects based on the address syntax.

The approaches based on the clustering of small areas of the image are the closest to our approach. Particular techniques include smearing of the image[2], multi-resolution approaches [1, 15], neural networks [10]. Algorithm of [14] involves clustering of image pixels.

In contrast to these approaches our algorithm will cluster connected components of binarized image. Considering connected components allowed our algorithm to be scale independent. By the design of our algorithm the question of whether address block would be split in different clusters does not depend on presence of other elements in the image.

3. Algorithm description.

Our algorithm has following main steps:

1. Extract contours of connected components.
2. Extract contour features. Each contour corresponds to a point in a feature space.
3. Cluster points in the feature space.
4. Using heuristic rules, find cluster corresponding to the address block.
5. Separate contours of the chosen cluster together with any close clusters.

In the first step we are doing transition from connected components to the contour representation of the image[3]. In the next step we extract contour features. Currently we consider only 5 features for each contour. Figure 1 shows how contour features are extracted.

1. $mean_x = (right + left) / 2$
2. $mean_y = (bottom + top) / 2$
3. $size = \max((right - left), (bottom - top))$
4. $avg_stroke_width = \frac{Area_inside_contour}{\frac{1}{2} Perimeter}$
5. max_stroke_length - largest interval traced in one of 8 directions.

This feature reflects the length of the stroke. There are 8 directions of pixel transitions on the contour. The interval traced in some particular direction would include a sequence of pixels having transitions with this direction as well as with two neighboring directions.

For clustering points in the feature space we have to define the way to calculate the distance between two contours. It turned out that simple Euclidean distance did not work. The problem lies in the impossibility to set a proper scale. For example if we treat a size of the image as the unity, we would get different distance for images containing only address block, and for images containing address block as a small part of the image.

What is important in address block structure is how different parts of it relate to each other, and not how they relate to the whole image. Hence we tried to derive the distance which would measure how relative two contours are. So, for example, if one contour is two times bigger than the other, we want to say that one contour stands at distance 1 from

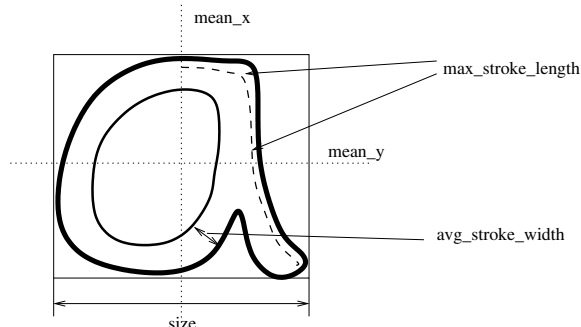


Figure 1. Extracting contour features.

the other with respect to size feature. Hence we defined logarithmic distance between two contours in the feature space as sum of following feature distances:

1. $\frac{|mean_x_1 - mean_x_2|}{min_size}$
where $min_size = \min(size_1, size_2)$.
2. $\frac{|mean_y_1 - mean_y_2|}{min_size}$
3. $\log\left(\frac{\max(size_1, size_2)}{min_size}\right)$
4. $\log\left(\frac{\max(avg_stroke_width_1, avg_stroke_width_2)}{\min(avg_stroke_width_1, avg_stroke_width_2)}\right)$
5. $\log\left(\frac{\max(avg_stroke_length_1, avg_stroke_length_2)}{\min(avg_stroke_length_1, avg_stroke_length_2)}\right)$

To cluster the feature points we at first experimented with probabilistic clustering methods, such as k-means algorithm. Soon we realized that such methods tend to incorrectly remove few contours from address block, or to add some single unrelated contours to the address block cluster.

Eventually we achieved success with agglomerative type clustering algorithm[13]. The algorithm is similar to the minimum spanning tree clustering algorithm. In our case we define a threshold θ beforehand, and resulting would satisfy these two conditions:

1. If the distance between two contours is less than θ :
 $dist(c_i, c_j) \leq \theta$ then c_i and c_j are in the same cluster.
2. The distance between any two clusters Cl_k and Cl_l is bigger than θ , that is for any $c_i \in Cl_k$ and $c_j \in Cl_l$:
 $dist(c_i, c_j) > \theta$.

We implemented a straightforward algorithm of sequential input of contours, adding contour to already existing cluster or forming new cluster, and, if necessary, merging clusters. The algorithm takes $O(n^2)$ time, where n is the number of contours. The actual number of contours in the images was usually around 100, with rare cases going over 1000. This part of the algorithm was most time consuming.

Still, when run inside Hand-Written Address Interpretation system(HWAI)[11] our algorithm took less than 10% of total processing time.

As almost all address block location algorithm, our algorithm has to decide at the end how to rank the obtained clusters, and which cluster is the most probable destination address block cluster. For this purpose we used simple heuristic rules:

1. Address block cluster should have at least 10 contours.
2. There should be contours of size bigger than some threshold in the cluster.
3. Choose the cluster satisfying 1. and 2. and of largest area as a candidate address block cluster.

Some images might contain smearing or blurring which is binarized as set of small contours. Our algorithm tends to separate the clusters of such small contours. Thus we introduced one additional step of merging found candidate address block cluster with spatially close other clusters.

The goal of our algorithm was not only to separate address block from the rest of the image, but also to preserve already separated address block intact. Hence the parameters of our algorithm (the most important is clustering threshold θ) were chosen so that most of address blocks would not be segmented.

4. Experiments

The algorithm was developed for parcel image set, which contains images with well separated address blocks as well as non-separated/incorrectly separated address blocks in different orientations. We had 1684 images. Approximately 1/3 of these images did not contain full destination address block, another 1/3 contained well separated destination address block with possible noise.

We used HWAI system[11] developed in CEDAR for our experiments. We were interested in the total performance of the system with and without address block location algorithm. Without any address block location algorithm HWAI was able to finalize 240 images. When used with our algorithm finalization number increased to 272. The increase in performance is due to successfully separated address block in some images, and removed noise in other images. There were about 10 images which were recognized before our algorithm, but not after it. Our algorithm removed some parts of these images, mostly zip code, which sometimes stands far away from other parts of the address block.

Experiments show that algorithm leaves almost all well separated address blocks intact, while doing segmentation on non-separated address block images, removing stamps, bounding boxes and different graphics. In contrast address block location algorithm previously developed at

CEDAR[9] performed purely on given image set. This algorithm was expecting the image of particular format, and thus was splitting images containing only address block.



Figure 2. Examples of original images and processed images.

Figure 2 shows 4 examples of how address block is separated from the images. Left column contains original images, and right column has corresponding processed images. First two pairs show how address block is well separated from the complex image. Second has wrong orientation, but due to algorithm design it is processed as well as normally oriented images. Third pair demonstrates that even if image contains only correct destination address block, our algorithm is still efficient in removing some noise and graphics which could Dis-orient address recognition program. Fourth image shows that there are cases where our algorithm does not separate destination address block.

5. Acknowledgments

The work was supported by a contract with USPS. Postal images are provided by USPS. Authors would like to thank Vemulapati Ramanaprasad of CEDAR for fruitful discussions and help with running the software.

References

- [1] M. Caviglione and P. Scaiola. A modular real-time vision system for address block location. In *Proceedings of the Fourth USPS Advanced Technology Conference*, pages 41–56, Washington, DC, 1990.
- [2] A. Downton and C. Leedham. Preprocessing and presorting of envelope images for automatic sorting using ocr. *Pattern Recognition*, 23(3):347–362, 1990.
- [3] H. Freeman and R. Shapira. On the encoding of arbitrary geometric configurations. *IRE Transactions on Electronic Computers*, EC-10:260–268, June 1961.
- [4] A. Jain and S. Bhattacharjee. Address block location on envelopes using gabor filters. *Pattern Recognition*, 25(12):1459–1477, 1992.
- [5] A. Jain, S. Bhattacharjee, and Y. Chen. On texture in document images. In *Proceedings of Computer Vision and Pattern Recognition*, volume 31, pages 677–680, 1992.
- [6] J. Lii, P. Palumbo, and S. Srihari. Address block location using character recognition and address syntax. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 330–335, Tsukuba Science City, Japan, 1993.
- [7] J.-C. Oriot, D. Barba, and J.-C. Salome. Address block locating method based on transition analysis approach: design and evaluation on flats objects. In *Proceedings of the First International Conference on Document Analysis and Recognition*, volume II, pages 665–673, Saint-Malo, France, 1991.
- [8] P. Palumbo and S. Srihari. Postal address reading in real time. *International Journal of Imaging Science and Technology*, 7(4):370–378, 1996.
- [9] P. Palumbo, S. Srihari, J. Soh, R. Sridhar, and V. Demjanenko. Postal address block location in real time. *IEEE Computer*, pages 34–42, July 1992.
- [10] J. Platt and R. Wolf. Convolutional neural networks for address block location. In *Proceedings of the Fifth USPS Advanced Technology Conference*, pages 1283–1293, Washington, DC, 1992.
- [11] S. Srihari and E. Keubert. Integration of hand-written address interpretation technology into the united states postal service remote computer reader system. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, Ulm, Germany, 1997.
- [12] S. Srihari, C.-H. Wang, P. Palumbo, and J. Hull. Recognizing address blocks on mail pieces: specialized tools and problem-solving architecture. *AI Magazine*, 8(4), 1987.
- [13] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, London, UK, 1999.
- [14] L. Turnbaugh and J. Weaver. The case for graph-theoretic clustering for address finding. In *Proceedings of the Second USPS Advanced Technology Conference*, pages 147–160, Washington, DC, 1986.
- [15] C. Viard-Gaudin and D. Barba. Address block location method based on a multiresolution approach. In *Proceedings of the First International Conference on Document Analysis and Recognition*, pages 954–962, Saint-Malo, France, 1991.