Evaluation of Binarization Algorithms

Bruce A. Bobier
Department of Systems Design Engineering,
University of Waterloo, Waterloo, ON, N2L 3G1
Email: bbobier@engmail.uwaterloo.ca

Michael Wirth
Department of Computing and Information Science,
University of Guelph, Guelph, ON, N1G 2W1
Email: mwirth@uoguelph.ca

***Abstract*—Twenty-one algorithms from five algorithmic families were evaluated for their accuracy at binarizing line drawings. The data set contains binary line drawings (templates) that are converted to gray scale with backgrounds added from scanned elevation drawings. Using normalized cross-correlation to measure similarity between the binarized images and binary templates, clustering and histogram shape based algorithms were found to provide the most accurate results, while histogram shape based algorithms were significantly faster than all other algorithmic families. No single algorithm was observed to produce significantly more accurate results across the entire data set than all other algorithms.**

***Index Terms*—Binarization, Line Drawings, Performance Evaluation**

I. Introduction

Binarization is the process of labeling each pixel in a gray scale image as object data or background, and is a common preprocessing step in many image analysis systems [1]. Numerous binarization algorithms have been presented and currently there are no known comparative evaluations of the algorithms' accuracy at binarizing line drawings, although several survey papers have considered the problem in the context of document image analysis and optical character recognition (OCR) [1–5].

OCR problems are similar in principle to the recognition of lines, arcs and shapes in line drawings, although the unique properties and subsequent processing requirements of each document class suggests that techniques that are successful at binarizing textual documents are not necessarily optimal for preprocessing line drawings. As there are no known comparative studies of algorithms for binarizing line drawings, currently users must select an algorithm based on results from related studies, prior knowledge, trial and error, or arbitrary selection. This experiment attempts to quantify the accuracy of several prominent binarization algorithms in the context of preprocessing architectural elevation drawings for vectorization.

An architectural elevation drawing is a representation of a building's geometrical exterior as perceived from a horizontal viewpoint without dimensional perspective. The drawings are often manually created with pen and paper and digitized as 8-bit gray scale images using a scanner or overhead camera. An immense number of these drawings are contained in digital and analog archives worldwide, and in order for their contents to be fully realized for archival, retrieval and analytical tasks, a method of representing their semantic information in a non-visual format may be provided by preprocessing, vectorizing and classifying the drawings.

II. Methods

In this article, 21 algorithms [6–24] are compared for their accuracy at binarizing line drawings. The central hypothesis to be tested is that a statistically significant difference will be observed between all algorithms that will allow a full ordering to be established. The second hypothesis to be tested is that a single algorithm will perform better than all others across the entire data set. One difficulty involved in evaluating the algorithms' accuracy at binarizing line drawings is that the ground truth, or optimal binary solution, is unknown. This has been overcome through the use of a data set of synthesized line drawings that characterize elevation drawings, and for which an optimal template is available. The diverse size and complexity of the data set images used in this experiment seeks to provide a wide spread of results that can be used to verify if a significant difference between algorithms can be observed and quantified. The evaluated algorithms can be categorized into algorithmic families which allows us to test if some algorithmic families are better suited for binarizing line drawings than others.

A. Data Set

The data set used for this experiment consists of 60 gray scale images and their corresponding 60 template images, and is divided into three subsets of 20 images with images in each subset having sizes of 256×256, 512×512 or 800×600 pixels (Fig. 1). The input image I_j, template T_j and output images $\hat{I}_j$ are non-interlaced, bitmapped and non-compressed `.png` files, with file sizes of the templates ranging from 1 to 4 Kb, while the test images range from 10 to 264 Kb, depending on the complexity of the background or degradation.

Each image is first created in Adobe Photoshop as a binary `.png` image composed of lines, arcs and simple shapes, and constitutes a template T_j. Next a copy of the template is created, converted to 8-bit gray scale, and subjected to several forms of degradation or alteration to more accurately characterize an elevation drawing.

To degrade the foreground data, Gaussian white noise (15-45%) is added to some images to mimic the dirt, aging and scanning artifacts that appear in elevation drawings. Frequently, digitized elevation drawings exhibit a small amount of

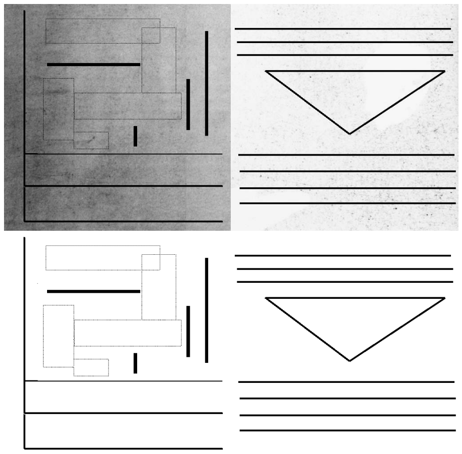

Fig. 1. Sample input images (top row) and their templates (bottom row)

blurring along the lines due to ink bleed and scanning artifacts, and this is simulated by applying a 2 pixel Gaussian blur to the lines in some test images. To create images with more realistic backgrounds, synthetic textures or image backgrounds (i.e. blank paper) are copied from elevation drawings in the Historic American Building Survey (HABS) [25] and placed beneath the template's foreground data, which remains untouched.

B. Algorithms

The 21 binarization algorithms tested in this experiment were written by Michael Wirth for the Microscopy and Imaging Toolbox [26], and can be categorized into six groups, as outlined by Sezgin and Sankur [5]:

1) Object attribute-based methods (AttribT), which search for a similarity measure between the gray level and binarized images (using for example, fuzzy shape similarity, edge coincidence, etc.);
2) Clustering-based methods (ClusterT), which cluster the gray level samples into two groups of foreground or background data, or by modeling the image as a mixture of two Gaussian distributions;
3) Entropy-based methods (EntropyT), which use the difference of entropy between the foreground and background data or the cross-entropy between the original and binarized image;
4) Local methods (LocalT), which calculate an adaptive threshold value for each pixel based on the local image characteristics; and
5) Histogram shape-based (ShapeT), which analyze the characteristics of the image's smoothed histogram, viewing it as a mixture of two Gaussian distributions attributed to foreground and background data.

The twenty-one algorithms used in this experiment are listed in Table I, and further algorithmic details can be found in [26] and the associated references. Two of the algorithms, AttribT_Ramesh and AttribT_Yager, can be executed in variant forms, the former using either the sum of square errors (AttribT_Ramesh1) or the sum of two variances(AttribT_Ramesh2), and the latter using either the Hamming (AttribT_Yager1) or Euclidean (AttribT_Yager2) distance for computing threshold values. Note that for evaluative and analytical purposes, each variant is treated as a separate algorithm. Only two other algorithms (LocalT_Niblack and LocalT_Bernsen) accept additional arguments beyond the image to be binarized. For Niblack's local algorithm, a 15×15 neighbourhood and adjustment parameter of $k = -0.2$ is selected for the experiment, following from [1, 26], while for Bernsen's local algorithm, values of 90, 16 and 1 are selected for the local contrast threshold, window size and high homogenous areas parameters respectively.

Algorithms		
AttribT_	BaradezMSA	[6]
	Huang	[10]
	moments	[19]
	Ramesh	[16]
	Yager	[20]
ClusterT_	Brink	[8]
	Otsu	[15]
	Ridler	[22]
	Yanni	[21]
EntropyT_	Brink	[9]
	Kapur	[11]
	Li	[13]
	LiL	[12]
	Shanbhag	[18]
	Yen	[24]
LocalT_	Bernsen	[7]
	mAve	[23]
	Niblack	[14]
ShapeT_	unimodal	[17]

TABLE I
BINARIZATION ALGORITHMS USED FOR EVALUATION.

C. Performance Evaluation

To evaluate each algorithm, each image $I_j\{j \in n|n = 1\ldots60\}$ is binarized by the algorithm to produce a binary image $\hat{I}_j$. Here, the accuracy of an algorithm is defined as the degree to which image $\hat{I}_j$ resembles the corresponding template T_j, which is computed as the normalized cross-correlation between $\hat{I}_j$ and T_j. Normalized cross-correlation (NCC) is a statistical approach often used in template matching and pattern recognition problems, and computes the probability that $\hat{I}_j$ is an instance of T_j [27]. Formally, the normalized cross correlation of $\hat{I}_j$ and T_j is calculated by:

$$\gamma(u,v) = \frac{\sum_{x,y}[\hat{I}(x,y) - \overline{I}_{u,v}][T(x-u,y-v) - \overline{T}]}{\{\sum_{x,y}[\hat{I}(x,y) - \overline{I}_{u,v}]^2 \sum_{x,y}[T(x-u,y-v) - \overline{T}]^2\}^{0.5}}$$

where $\overline{I_j}$ is the mean of $\hat{I}$ and $\overline{I}_{u,v}$ is the mean of $\hat{I}_j(x,y)$ in the region of the template, and is calculated using the `normxcorr2(`$\hat{I},T$`)` function in the Image Processing Toolbox for Matlab. If $\hat{I}_j$ matches T_j exactly, then γ will equal 1, whereas γ = 0 if $\hat{I}_j$ and T are entirely dissimilar.

Algorithm	Mean	Median	StdDev	Min	Max
attribT_BaradezMSA	0.569	0.531	0.203	0.203	0.997
attribT_Huang	0.687	0.932	0.380	0.003	0.997
attribT_moments	0.898	0.918	0.068	0.730	0.991
attribT_Ramesh1	0.770	0.789	0.222	0.006	0.978
attribT_Ramesh2	0.251	0.000	0.393	0.000	0.994
attribT_Yager1	0.684	0.931	0.378	0.003	0.997
attribT_Yager2	0.684	0.931	0.378	0.003	0.997
clusterT_Brink	0.936	0.946	0.053	0.762	0.997
clusterT_Otsu	0.936	0.946	0.053	0.762	0.997
clusterT_Riddler	0.936	0.946	0.053	0.762	0.997
clusterT_Yanni	0.812	0.895	0.246	0.000	0.997
entropyT_Brink	0.577	0.561	0.119	0.321	0.855
entropyT_Kapur	0.614	0.704	0.254	0.016	0.943
entropyT_Li	0.880	0.940	0.196	0.155	0.997
entropyT_LiL	0.929	0.942	0.059	0.717	0.997
entropyT_Shanbhag	0.662	0.660	0.185	0.018	0.963
entropyT_Yen	0.463	0.470	0.244	0.007	0.851
localT_Bernsen	0.881	0.907	0.094	0.546	0.997
localT_mAve	0.908	0.922	0.078	0.654	0.997
localT_Niblack	0.488	0.479	0.087	0.170	0.677
shapeT_unimodal	0.865	0.931	0.139	0.393	0.997

TABLE II
SUMMARY OF NCC DATA FOR EACH ALGORITHM'S ACCURACY ON THE 60 IMAGES.

Algorithm	Mean	Median	StdDev	Min	Max
attribT_BaradezMSA	0.387	0.399	0.167	0.022	1.532
attribT_Huang	0.127	0.124	0.048	0.045	0.436
attribT_moments	0.040	0.031	0.097	0.003	1.534
attribT_Ramesh1	0.007	0.007	0.003	0.006	0.050
attribT_Ramesh2	3.268	3.215	2.085	0.645	7.235
attribT_Yager1	0.057	0.055	0.018	0.024	0.091
attribT_Yager2	0.055	0.054	0.018	0.023	0.098
clusterT_Brink	0.032	0.020	0.031	0.018	0.217
clusterT_Otsu	0.005	0.005	0.001	0.004	0.030
clusterT_Riddler	0.006	0.006	0.004	0.002	0.041
clusterT_Yanni	0.002	0.001	0.001	0.001	0.010
entropyT_Brink	0.036	0.036	0.001	0.034	0.055
entropyT_Kapur	0.294	0.275	0.111	0.100	0.480
entropyT_Li	0.076	0.063	0.040	0.005	0.196
entropyT_LiL	0.660	0.662	0.020	0.613	0.883
entropyT_Shanbhag	0.032	0.033	0.004	0.012	0.071
entropyT_Yen	0.004	0.003	0.002	0.002	0.047
localT_Bernsen	0.092	0.092	0.027	0.009	0.148
localT_mAve	0.128	0.120	0.085	0.027	0.292
localT_Niblack	18.547	18.133	12.179	3.751	39.394
shapeT_unimodal	0.002	0.001	0.014	0.001	0.330

TABLE III
SUMMARY OF EACH ALGORITHM'S COMPUTATION TIME ON THE 60 IMAGES.

The computation time of each algorithm is also assessed and is calculated using the Matlab functions `tic` and `toc`, which records the number of seconds that have elapsed since `tic` was called.

To test the existence of a statistically significant difference between the algorithms' accuracy and computation time, the data are analyzed using an ANOVA and post-hoc individual pairwise comparison with $\alpha = 0.05$.

III. RESULTS

A total of ten trials were conducted on each of the 21 algorithms at binarizing the 60 image data set. As the algorithms are deterministic and the data set does not change, the NCC value of each algorithm-image pair is constant across each trial. Thus, the NCC values are collected from a single trial, while computation time data is collected from all ten trials.

In terms of NCC values (Table II), the highest mean values are produced by clusterT_Brink, clusterT_Ridler and clusterT_Otsu respectively, while the lowest mean values are produced by attribT_Ramesh2, entropyT_Yen and localT_Niblack. In terms of computation time (Table III), the best (lowest) means are produced by clusterT_Yanni, shapeT_unimodal and entropyT_Yen, while the three highest mean times are produced by localT_Niblack, attribT_Ramesh2 and entropyT_LiL respectively.

IV. ANALYSIS AND DISCUSSION

To determine if the data are normally distributed, regression analysis is first conducted on the recorded data. For the NCC and computation time data, r^2 values of 81.2% and 24% were observed respectively, both of which fall below the confidence level of 95%. Because of the non-normality of the data, non-parametric statistics were employed for the remaining statistical calculations.

For both NCC and computation time, an ANOVA ($\alpha = 0.05$) was calculated for all algorithms, which produced p-values of 1.42E-123 and 0 respectively, indicating that a statistically significant difference exists between groups. Post-hoc individual pairwise comparisons of the data were used to test the central hypothesis. The post-hoc comparison utilizes the difference of each pair of algorithms' means and standard error to compute a p-value, where p-values $< \alpha = 0.05$ reflect a statistically significant difference between the paired algorithms. A summary of these post-hoc comparisons based on NCC values is given in Table IV, where the algorithm name in cell (i, j) indicates a statistically significantly higher NCC value of the algorithms in the i^{th} row and j^{th} column. A blank cell indicates that there is not a statistically significant difference between the two algorithms. Additionally, a graph-based summary of the post-hoc comparisons is given in Fig. 2 as a partial ordering of algorithms, with the better performing algorithms situated atop the worse performing algorithms. Algorithms within the same node or layer are not significantly different.

Given that the graph is not a linear structure and a full ordering cannot be established, the hypothesis that a statistically

Family	Mean	Median	StdDev
AttribT	0.649	0.802	0.362
ClusterT	0.905	0.941	0.141
EntropyT	0.751	0.833	0.232
LocalT	0.759	0.854	0.211
ShapeT	0.865	0.931	0.139

TABLE IV
SUMMARY OF NCC FOR ALGORITHM FAMILIES

Family	Mean	Median	StdDev
AttribT	0.563	0.070	1.364
ClusterT	0.011	0.005	0.020
EntropyT	0.184	0.037	0.239
LocalT	6.255	0.120	11.179
ShapeT	0.002	0.001	0.014

TABLE V
SUMMARY OF TIME FOR ALGORITHM FAMILIES

	attribT_ BaradezMSA	attribT_ Huang	attribT_ moments	attribT_ Ramesh1	attribT_ Ramesh2	attribT_ Yager1	attribT_ Yager2
shapeT_unimodal	unimodal			unimodal	unimodal		
localT_Niblack		Huang	moments	Ramesh1		Yager1	Yager2
localT_mAve	mAve			mAve	mAve	mAve	mAve
localT_Bernsen	Bernsen			Bernsen	Bernsen		
entropyT_Yen		Huang	moments	Ramesh1		Yager1	Yager2
entropyT_Shanbhag		Huang	moments	Ramesh1	Shanbhag	Yager1	Yager2
entropyT_LiL	LiL	LiL		LiL	LiL	LiL	LiL
entropyT_Li	Li	Li		Li	Li	Li	Li
entropyT_Kapur		Huang	moments	Ramesh1		Yager1	Yager2
entropyT_Brink		Huang	moments	Ramesh1		Yager1	Yager2
clusterT_Yanni	Yanni				Yanni		
clusterT_Ridler	Ridler	Ridler		Ridler	Ridler	Ridler	Ridler
clusterT_Otsu	Otsu	Otsu		Otsu	Otsu	Otsu	Otsu
clusterT_Brink	Brink	Brink		Brink	Brink	Brink	Brink
attribT_Yager2	Yager2				Yager2		/
attribT_Yager1	Yager1				Yager1	/	
attribT_Ramesh2		Huang	moments	Ramesh1	/		
attribT_Ramesh1	Ramesh1		moments	/			
attribT_moments	moments		/				
attribT_Huang	moments	/					
attribT_BaradezMSA	/						

	clusterT_ Brink	clusterT_ Otsu	clusterT_ Ridler	clusterT_ Yanni	entropyT_ Brink	entropyT_ Kapur	entropyT_ Li
shapeT_unimodal					unimodal	unimodal	
localT_Niblack	Brink	Otsu	Ridler	Yanni			Li
localT_mAve					mAve	mAve	
localT_Bernsen					Bernsen	Bernsen	
entropyT_Yen	Brink	Otsu	Ridler	Yanni			Li
entropyT_Shanbhag	Brink	Otsu	Ridler	Yanni			Li
entropyT_LiL				LiL	LiL	LiL	
entropyT_Li					Li	Li	/
entropyT_Kapur	Brink	Otsu	Ridler	Yanni		/	
entropyT_Brink	Brink	Otsu	Ridler	Yanni	/		
clusterT_Yanni	Brink	Otsu	Ridler	/			
clusterT_Ridler			/				
clusterT_Otsu		/					
clusterT_Brink	/						

	entropyT_ LiL	entropyT_ Shanbhag	entropyT_ Yen	localT_ Bernsen	localT_ mAve	localT_ Niblack	shapeT_ unimodal
shapeT_unimodal		unimodal	unimodal			unimodal	/
localT_Niblack	LiL	Shanbhag		Bernsen	mAve	/	
localT_mAve		mAve	mAve		/		
localT_Bernsen		Bernsen	Bernsen	/			
entropyT_Yen	LiL	Shanbhag	/				
entropyT_Shanbhag	LiL	/					
entropyT_LiL	/						

TABLE VI

SUMMARY OF POST-HOC COMPARISONS BASED ON NCC VALUES. THE ALGORITHM NAME IN CELL (i, j) HAS A SIGNIFICANTLY HIGHER MEAN NCC $(\alpha = 0.05)$ OF THE ALGORITHMS IN THE i^{th} ROW AND j^{th} COLUMN.

significant difference exists between all algorithms based on their NCC is rejected. The hypothesis that a single algorithm performs better than all others across the entire data set is also rejected.

Post-hoc comparisons of the computation time of each pair of algorithms were also conducted, the results of which are summarized as a partial ordering graph (Fig. 3). Again, a full ordering of algorithms cannot be established based on computation time for binarizing the test images, as there is not a statistically significant difference between the 18 fastest algorithms. The hypothesis that a single algorithm will perform better across the entire image set in terms of computation time is also rejected, as the fastestcomputation times vary between clusterT_Yanni, entropyT_Yen and shapeT_unimodal, despite the latter providing the fastest time on 98.83% of the tests.

Regression analysis of the NCC and time data for each algorithmic indicates that the data is non-normal ($r^2 = 78\%$ and 24% respectively), and again non-parametric post-hoc

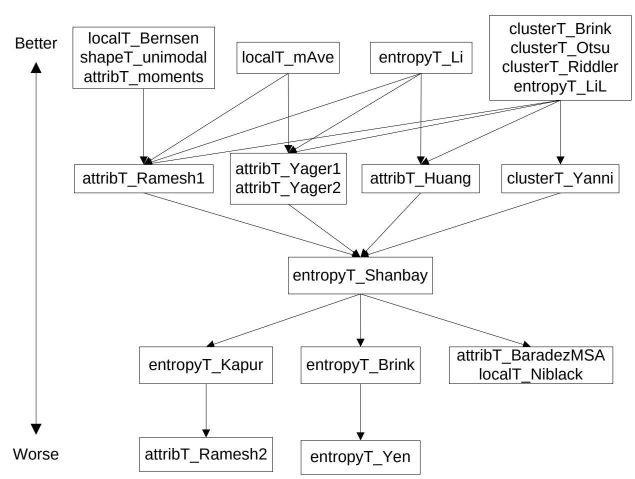

Fig. 2. Partial ordering of algorithms from post-hoc comparisons of NCC.

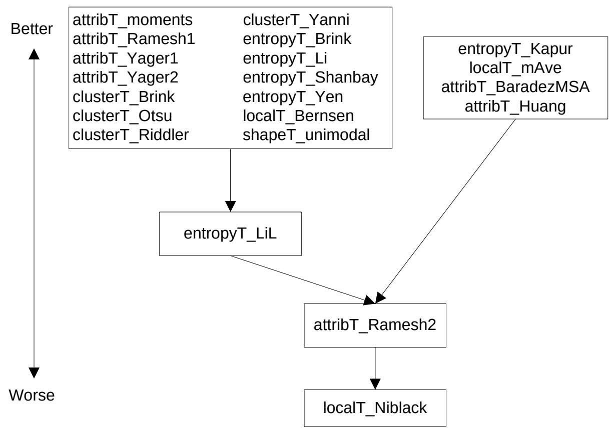

Fig. 3. Partial ordering of algorithms from post-hoc comparisons of computation time.

comparisons were performed. With regards to accuracy (Table IV), the cluster- and shape-based algorithmic families were not significantly different from each other, but were found to produce significantly higher NCC values than the other families (with AttribT, EntropyT, and LocalT not being significantly different from each other). For computation time (Table V), a statistically significant full ordering of algorithmic families was observed: ShapeT, ClusterT, EntropyT, AttribT, and Local (with ShapeT being fastest).

It is interesting to note that the accuracy of localT_Niblack is among the lowest and its computation time is statistically significantly worse than all other evaluated algorithms. This is surprising given that the algorithm has been shown to provide superior performance in comparative evaluations of algorithms as a preprocessing step for OCR [1] and its frequent use in the OCR domain [4, 5].

V. Conclusion

Twenty-one binarization algorithms were evaluated to determine which algorithm is best suited for the preprocessing of elevation drawings for vectorization, and it was found that a statistically significant full ordering of algorithms based on their NCC performance cannot be established. Sixty 8-bit gray scale images of dimensions 256×256, 512×512 or 800×600 were binarized ten times by each algorithm, with computation time (seconds) and accuracy (the normalized cross-correlation value between the binary output image and its corresponding template) being recorded for each trial. Post-hoc individual pairwise comparisons of computation time and NCC of each algorithm were used to establish a partial ordering of algorithms for each variable. Although no single algorithm can be seen as a panacea for binarizing elevation drawings, it is noted that clustering-based algorithms, specifically Brink's [8], Otsu's [15] and Ridler and Calvard's [22] may make suitable candidates for this problem, as they were shown to produce quick, accurate results for the data set.

Future work may include utilizing this experimental framework in the evaluation of text/graphics segmentation algorithms, which typically follow binarization in the preprocessing of elevation drawings for vectorization. Similarly, vectorization and direct-recognition algorithms may also be compared using this framework, with rasterized CAD-based line drawings serving as templates, and their degraded/modified versions serving as input images. Finally, the results of this experiment may also be applied in several other domains, including for example, the binarization of modern maps for GIS input, historic maps and diagrams for indexing and analysis, as well as the binarization of figures and images in primarily textual documents.

References

[1] O.D. Trier and A.K. Jain. Goal-directed evaluation of binarization methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1191–1202, 1995.

[2] C.A. Glasbey. An analysis of histogram-based thresholding algorithms. *Graphical models and image processing*, 55(6):532–537, November 1993.

[3] S.U. Lee, S.Y. Chung, and R.H. Park. A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision and Graphic Image Processing*, 52(2):171–190, November 1990.

[4] P.W. Palumbo, P. Swaminathan, and S.N. Srihari. Document image binarization: Evaluation of algorithms. *SPIE Applications of Digital Image Processing*, 697:278–285, 1986.

[5] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–165, January 2004.

[6] M.O. Baradez, C.P. McGuckin, N. Forraz, R. Pettengell, and A. Hoppe. Robust and automated unimodal histogram thresholding and potential applications. *Pattern Recognition*, 37(6):1131–1148, June 2004.

[7] J Bernsen. Dynamic thresholding of grey-level images. In *International Conference on Pattern Recognition*, pages 1251–1255, 1986.

[8] A.D. Brink. Gray-level thresholding of images using a correlation criterion. *Pattern Recognition Letters*, 9(5):335–341, June 1989.

[9] A.D. Brink and N.E. Pendock. Minimum cross-entropy threshold selection. *Pattern Recognition*, 29(1):179–188, January 1996.

[10] L.K. Huang and M.J.J. Wang. Image thresholding by minimizing the measures of fuzziness. *Pattern Recognition*, 28(1):41–51, January 1995.

[11] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Graphical Models and Image Processing*, 29(3):273–285, March 1985.

[12] C.H. Li and C. K. Lee. Minimum cross entropy thresholding. *Pattern Recognition*, 26(4):617–625, 1993.

[13] C.H. Li and P.K.S. Tam. An iterative algorithm for min-

imum cross-entropy thresholding. *Pattern Recognition Letters*, 19(8):771–776, June 1998.

[14] W. Niblack. *An Introduction to Digital Image Processing*. Prentice Hall, New Jersey, 1986.

[15] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[16] N. Ramesh, J.H. Yoo, and I.K. Sethi. Thresholding based on histogram approximation. *IEEE Proceedings Vision, Image and Signal Processing*, 142(5):271–279, October 1995.

[17] P.L. Rosin. Unimodal thresholding. *Pattern Recognition*, 34(11):2083–2096, November 2001.

[18] A.G. Shanbhag. Utilization of information measure as a means of image thresholding. *CVGIP: Graphical Model and Image Processing*, 56(5):414–419, 1994.

[19] H Tsai. Moments preserving thresholding: A new approach. *Computer Vision, Graphics and Image Processing*, 29(3):377–393, 1985.

[20] R Yager. On the measures of fuzziness and negation. *International Journal of General Systems*, 5:221, 1979.

[21] M.K. Yanni and E. Horne. A new approach to dynamic thresholding. In *EUSIPCO European Signal Processing Conference*, volume 1, pages 34–44, 1994.

[22] T.W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *System, Man and Cybernetics*, 8(8):629–632, August 1978.

[23] P. Wellner. Adaptive thresholding on the DigitalDesk. Technical report: EPC-93-110, EuroPARC, 1993.

[24] J.C. Yen, F.J. Chang, and S. Chang. A new criterion for automatic multilevel thresholding. *IEEE Transactions in Image Processing*, 4(3):370–378, March 1995.

[25] Historic American Buildings Survey/ Historic American Engineering Record. http://memory.loc.gov/ammem/ collections/habs_haer, 2007.

[26] M. Wirth. Purdue University Cytometry Laboratories. Microscopy & Imaging 3. http://www.cyto.purdue.edu, August 2005.

[27] L.G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.