Style-Based Retrieval for Ancient Syriac Manuscripts

Emma Dalton
Education Dept.
Simon Fraser University
Burnaby, BC, Canada
emma.b.dalton@gmail.com

Nicholas R. Howe
Computer Science Dept.
Smith College
Northampton, Massachusetts, USA
nhowe@smith.edu

ABSTRACT

Thousands of documents written in Syriac script by early Christian theologians are of unknown provenance and uncertain date, partly due to a shortage of human expertise. This paper addresses the problem of attribution by developing a novel algorithm for offline handwriting style identification and document retrieval, demonstrated on a set of documents in the Estrangelo variant of Syriac writing. The method employs a feature vector based upon the estimated affine transformation of actual observed characters, character parts, and voids within characters as compared to a hypothetical average or ideal form. Experiments on seventy-six pages from nineteen Syriac manuscripts written by different scribes show that the method can identify pages written in the same hand with high precision, even with documents that exhibit various challenging forms of degradation.

Categories and Subject Descriptors

I.7 [**Document & Text Processing**]: Miscellaneous; J.5 [**Arts & Humanities**]: Miscellaneous

1. INTRODUCTION

Historically, the academic study of the Christian religion has mainly focused on Latin texts. In more recent years, however, a broader view has been taken with increased interest in the study of ancient manuscripts written in the Aramaic dialect known as Syriac, a largely unexplored area due to the fact that there are relatively few scholars currently able to read and understand the Syriac language.

Syriac scribes produced over 10,000 manuscripts between 400 and 1200 AD, including some of the earliest translations of the Bible, and translations critical to the preservation of the writings of Aristotle. Syriac Christianity was also the most far-reaching branch of Christian religion in antiquity, making contact with early Islam, India, and even China. Although the study of Syriac Christianity has yet to receive the full attention of many academics, this attitude is slowly changing: with increased interest in the Syriac branch of Christianity there are now several universities offering instruction in the Syriac language, eight international Syriac conferences, and an academic journal of Syriac studies.

Scholars of Syriac have a large body of untranslated, generally unstudied manuscripts with which to work. Because attribution and the identification of related documents is often a first step in understanding the significance of a particular work, scholars would benefit greatly from an automated system capable of recommending potentially relevant comparisons. While some Syriac manuscripts are easily attributed to a particular scribe because they include a colophon recording the copyist and date, the majority of preserved documents (95 percent) do not carry this important information [12]. To address the needs of scholars for attribution, this paper describes a system that can semi-automatically identify and retrieve documents written in a similar style to a query document. In this context, *style* refers to the prototypical form taken by each character in the alphabet, and captures qualities such as elongation, curvature, and slant of individual parts of the letter shape. Although documents written by the same scribe should appear at the top of a retrieved list for any given query document, identifying other highly-ranked manuscripts with similar handwriting style is also useful. Because handwriting forms change in subtle ways over time, similarities in style may indicate a similar origin in time and place, and thus signal a document that may be of interest.

From the computer science perspective, the goal of this work is to implement and test a novel method of writer style comparison based upon the *congealing* algorithm introduced by Learned-Miller[10]. As noted above, style-based retrieval overlaps with writer identification, which has been studied in other historical contexts [2, 1, 11]. In addition, a body of work has explored the field of modern offline handwriting identification [14, 4]. The notion of writing style employed here does not necessarily require the identification of individual writers that most modern methods attempt; rather it resembles the copybook style identification work of Yoon et al.[13]. Some other recent research has also explored style-based retrieval for modern writing [3].

In addition to its algorithmic contributions, this work seeks to raise the profile of a category of historical documents that has not received much attention to date. The authors know of only one attempt to do automatic handwriting analysis in Syriac: W. F. Clocksin's studies of handwriting recognition on Estrangelo texts [6, 5].

The remainder of the paper follows a standard organization. The next section introduces the novel identification

HIP '11 September 16 - September 17 2011, Beijing, China

method that is the focus of the paper. Section 3 describes the experiments used to evaluate the method, and gives their results. The final section concludes with a short discussion of impact and future work.

2. METHOD

The method developed in this paper assumes a body of documents with known provenance, to which the style of a new document will be compared. The most similar documents found in the known library may or may not be written by the same hand as the unknown work. Yet even when produced by a different scribe, similarities in letter styling may indicate that they were written at a similar time and place, and thus they can serve as a starting point for further scholarship.

The style descriptor developed below implicitly references an idealized or *Platonic* letter form for each character. The idealized form is a sort of average over all observed samples of that letter, computed by Learned-Miller's congealing algorithm. The actual observed form of any given character sample will differ from the ideal to some degree, characterized by a mathematical (affine) transformation, both at the full-character level and more closely when comparing individual parts of a letter. The parameters of the transformations involved (translation, rotation, elongation, and shear) form the components of the style descriptor associated with that particular character instance.

An outline of the main steps in the method follows, with each step described in further detail below. Syriac has 22 letters, and samples of each are identified in every document. To begin, the congealing process [10] aligns all the samples of each letter type by applying an affine transformation to each so as to minimize the overall entropy. The aligned samples form the basis for a feature vector with components describing both the overall structure and the individual parts (i.e., significant branches and voids). When comparing a new document to the library, each character sample casts a vote for the page holding the library character it most resembles, and pages are retrieved in order of the number of votes they receive.

2.1 Character Sampling and Alignment

Currently, a human identifies character samples in each document by clicking on a point in the center of the target letter. Automatic identification should be feasible [2, 5], but is left as future work. The experiments described later use up to six samples of each letter where available, but with automatic identification this limit could be increased significantly or ultimately eliminated.

Ideally all documents would be scanned under known, identical parameters. Unfortunately, conditions in the real world rarely meet this standard because document images may come from different collections with varying practices. As a result the apparent scale cannot be relied upon, and all documents are initially rescaled to equivalent character heights. This proceeds as follows: a human identifies a representative patch of the document by drawing a bounding box. The patch is rotated by up to ±5° to align the written lines with the horizontal, by maximizing the variance of the horizontal projections. Baselines and upperlines are detected based on changes in the horizontal projection, yielding the line heights. The original documents used in the experiments exhibit median line heights ranging from 19 to 50 pixels. To achieve scale independence, all images are rescaled to match the lowest resolution present (i.e., text lines 19 pixels high). Following this they are smoothed slightly using a Gaussian filter with $\sigma = 0.5$ and then binarized using energy minimization on a Markov random field model[8].

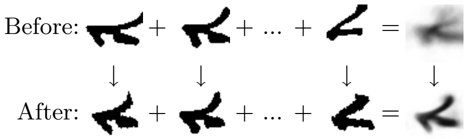

Figure 1: Congealing for samples of the character alaph. Top row shows original sample images and overall mean. Bottom row shows transformed samples after 20 rounds of congealing and new overall mean.

Character samples are taken from the binarized source images using a square radius of 24 pixels. This is large enough to capture complete letters, including most ascenders and descenders, but crops out most of the surrounding characters. (Some letters, such as *dalath* and *resh*, are smaller than others and could perhaps benefit from a smaller window, but these experiments use a uniform window size throughout.) Despite the cropping some samples inevitably contain pieces of other characters within their boundaries. The procedure described below removes a portion of this chaff, but some of it cannot be removed and remains as a source of noise in the final results.

Congealing, introduced by Learned-Miller et. al. and available as a Matlab code package[1], seeks to mutually align a set of letter samples to each other by applying gradient descent in the space of scalar transforms and seeking to minimize the pixelwise entropy of the samples. To prevent drift, the transforms are normalized after each descent step to maintain the same average position, scale, and shear over the group as a whole [10]. The results after just 20 rounds of descent are striking, as visible in Figure 1. Initially dissimilar letter forms map closely onto a common idealized shape. We exploit this fact to perform the data cleaning alluded to above: The idealized image is found by thresholding the congealed mean, and connected components in each aligned image that do not overlap with the idealized letter form are discarded in the original (unaligned) image. The original samples are then congealed again, minus their spurious components.

2.2 Letter Parts and Cavities

Bar-Yosef et. al. introduce analysis of the shape of letter cavities for style identification in ancient Hebrew[2]. This work uses cavities in a similar manner, and additionally analyzes shapes taken from parts of the letters themselves by breaking the letter forms into pieces at their main branching points. This section begins by describing how the shapes are isolated and identified. After this step, a treatment step reduces both parts and cavities in identical fashion to a descriptive feature vector.

Cavities are defined as the connected components left behind when the original letter shape is subtracted from its convex hull. Performing this operation on the idealized (i.e.,

[1] `http://www.cs.umass.edu/~elm/congealing/`

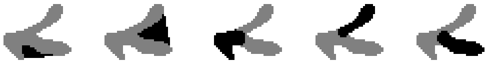

Figure 2: Cavities (left two images) and parts (right three images) for alaph. The potential cavity at the top of the letter is rejected because its thinness makes it potentially unstable.

thresholded mean congealed) image of each letter yields a canonical set of cavities, as illustrated in Figure 2. Thin cavities of only a few pixels in width tend to be unstable from one sample to another, so any that disappear under erosion by a two-pixel radius disk are thrown out and not considered in any further processing. The more robust cavities that remain can be matched across most or all of the aligned sample images.

Fortunately, the number of candidate cavities in each letter sample is usually small (on the order of zero to four) so an exhaustive search for possible matchings is feasible. In practice, it suffices to greedily find pairings using a chamfer match criterion. After all canonical cavities have been matched to their best equivalent in the observed sample, any remaining cavity components are tried in combination with the existing matches, and retained if they improve the chamfer match score. This allows for the possibility that a component may be split in two, a consideration that proves particularly important for the parts computation described below.

Letter parts are discrete segments of a branching letter form. (Some letters, such as *zayn* and the final form of *nun*, consist of a single part.) To isolate individual parts, we analyze the letter skeleton to identify junction points where three or more branches join together. With these junction pixels removed, the discrete connected components of the remaining skeleton image form the seeds of the separate letter parts. Each is reconstituted by replacing its pixels with the maximal inscribed circle in the original image. Any pixels of the original image not assigned in this manner to at least one letter part are then assigned to the part or parts they are closest to. Figure 2 shows the parts identified in the *alaph* canonical image.

As with the process for identifying cavities, the canonical parts are taken from the idealized image, and parts identified in sample images are matched to these. Because spurs in the skeleton appearing due to noise or style variation can trigger a split into extra parts, the ability to match several observed letter parts with a single canonical component becomes important here.

2.3 Feature Vector Assembly and Document Retrieval

Once the corresponding cavities and letter parts have been identified in all aligned sample images, a secondary round of congealing is applied in turn to each corresponding set of parts or cavities. The transformations found in these second-round alignments account for differences in the shape of the individual parts and cavities, independent of the rest of the letter. Since the round begins with images that are already globally aligned, any scaling or translation found will be significant.

To assemble the full feature vector describing each character sample, simply take the x and y shear components from the global affine transformation found during the first round of congealing, and concatenate these with the seven transformation components found for each cavity and letter-part set. The length of the feature vector thus varies by character, depending upon how many canonical parts and cavities are associated with its form. For example, *teth* features have 72 components, while *beth*, *lamadh*, and *kaph* each have 16. To prepare for style classification of new documents, the experiments precompute and store feature vectors for all letter samples of the library documents.

Given a previously unseen (query) document, the first step in classifying it is to align its letter samples with the previously congealed samples from the library. This employs a similar gradient-descent optimization as the original congealing process, except that the query image does not affect the steps taken by the library images because they are already fixed. The process has been referred to as *funneling* [9].

More specifically, from the standpoint of an individual image the original congealing process consists of three stages applied iteratively. First, a gradient-descent step is computed for the affine transformation that maps an individual sample onto the mean congealed image. (This is a seven-dimensional vector with components for translation, rotation, scale, and shear.) Second, the mean gradient step found over all the images is subtracted from each individual step, to ensure that the set of transformations as a whole does not drift. Finally, the transformation is updated by the remaining step and a new mean congealed image is found. To approximate the transformation that would have resulted if the query image was congealed with the rest of the library, we must record the mean congealed image and mean gradient step taken at each stage of the original congealing process for the library images. We compute the gradient-descent step in the first stage with respect to the stored mean image, and correct it in the second stage with the stored mean step. The query image does not contribute anything to these means, so they are slightly different from what they would be if the congealing were performed all at the same time, but if the number of images in the library is large then the effect of one missing image is negligible.

Once the query samples are aligned, they may be treated as all the others: their parts and cavities are identified and associated with corresponding canonical components. These also are aligned in a secondary round with the previously congealed set of library component samples, and the descriptive feature vector can be assembled.

At this point, a Euclidean distance computation gives the distance from each character sample in the query document to all the library samples of the same letter. Voting by the query samples establishes the final ranking of library documents: Each character votes for the library document containing its most similar character, and the top-ranked document is the one receiving the most votes. Ties are broken randomly. A slightly more complex system, denoted *rank-vote* in the experiments, assigns points to all documents in inverse proportion to the match rank of their letter samples as compared to the query samples. (This is a modified Borda count with Nauru point scoring.) Points are also weighted inversely to the number of samples of the query character in the library image, to avoid giving unfair weight to documents that simply have more samples to attract votes.

$$S_i = \sum_{c \in C} w_c \left(\sum_{q \in Q_c} \sum_{\ell \in L_c^i} \frac{1}{R_q(\ell)\,|L_c^i|} \right) \quad (1)$$

where C is the set of characters, Q_c is the set of samples for character c in the query document, L_c^i is the set of samples for character c in the ith library document, and $R_q(\ell)$ is the ordinal rank of ℓ over all the library samples for character c computed from the Euclidean distance of their respective feature vectors. An optional weight coefficient w_c allows different letters' votes to count more or less towards the final score. In the end, the rank order of the scores S_i determines which library documents best match the query.

3. EXPERIMENTS

The experiments employ a disparate collection of 19 documents written in the Estrangelo variant of Syriac script and believed to be written by different scribes. The test set includes four pages from each document, and for testing purposes these are treated as separate documents so that using one page as a query should retrieve the other three pages as the top-ranked hits from the library. This setup is perhaps less realistic than using entirely separate documents known to be by the same hand, because documents produced by the same scribe at different times may exhibit greater variance due to changes in writing implement, style drift over time, etc. Nevertheless the current experimental structure stands as it has thus far proved impossible to obtain a sufficent collection of different documents by the same scribe. Most of the documents used come from the Vatican's collection of Syriac manuscripts and are available on a digital CD released by Brigham Young University [7].

Of the 22 letters in the Syriac alphabet, three (namely *zain*, *yudh*, and *sadhe*) are too nondescript for non-experts to identify easily. Up to six samples of the remaining nineteen were identified by hand on each document page. An effort was made to choose clear and representative examples where possible.

The experiments all employ a four-fold cross-validation framework. In each of four repetitions, one page from each document is held out for querying, and the remaining 57 (3x19) documents form the library. By running all four folds in turn, we test each document once as a query. The results are averaged over all query attempts for all folds.

Because the system is intended as a tool for scholars looking for documents written in a related style, an information retrieval paradigm offers the most appropriate framework for evaluating the results. Every query has exactly three relevant documents in the library, so we report three numbers for all experiments: $(p_{33}, p_{67}, p_{100})$ gives the precision at 33%, 67%, and 100% recall, respectively. An ordered triple of this sort contains the same information as a precision-recall graph for these experiments, but can be expressed much more concisely.

Table 1 shows the performance of the system under various experimental conditions. The first line uses simple voting. The second uses the rank-vote scheme described above. In each of these experiments all 19 sampled characters vote equally. The third line shows results using a weighted voting scheme, with weights determined by a greedy additive scheme described below. The last section of the table shows as a comparison the results from a reimplementation of simple several text-independent measures of handwriting style: the *hinge* statistic, and the horizontal and vertical run-length encoding statistics [4]. None of the comparison schemes does appreciably better than chance for this set of documents.

Table 1: Precision at three recall levels for various experiments

Experiment	P_{33}	P_{67}	P_{100}
Simple vote	72.3%	61.8%	42.6%
Rank vote	74.2%	63.2%	48.8%
Weighted rank vote	77.7%	64.8%	48.6%
Hinge [4]	18.3%	14.1%	11.8%
RLE Horiz. [4]	16.5%	15.3%	15.4%
RLE Vert. [4]	19.5%	12.3%	11.0%

Table 2: Character weights discovered for each fold in the weighted vote-rank experiment

Letter	F1	F2	F3	F4	Mean
ܐ *alaph*	.187	.097	.048	.112	.111
ܒ *beth*	.053	.000	.024	.022	.025
ܓ *gamal*	.053	.065	.095	.045	.065
ܕ *dalath*	.000	.032	.048	.067	.037
ܗ *he*	.080	.032	.048	.000	.040
ܘ *waw*	.027	.097	.048	.067	.060
ܚ *heth*	.053	.032	.071	.045	.050
ܛ *teth*	.053	.032	.048	.067	.050
ܟ *kaph*	.053	.032	.024	.022	.033
ܠ *lamadh*	.027	.032	.048	.067	.043
ܡ *mim*	.093	.129	.095	.079	.099
ܣ *semkath*	.027	.000	.048	.067	.035
ܥ *e*	.027	.065	.048	.000	.035
ܦ *pe*	.107	.161	.024	.067	.090
ܩ *qaph*	.000	.032	.048	.067	.037
ܪ *rish*	.027	.032	.071	.045	.044
ܫ *shin*	.053	.032	.071	.045	.050
ܬ *taw*	.080	.097	.095	.112	.096

For weighted voting, the system uses a leave-one-out experiment within the library itself to set the weights. Beginning with a weight vector $W = (w_1, w_2, ..., w_{19})$ where all $w_c = 0$, the system tries adding one to each weight component in turn and computes the average precision under the new weights. The trial weight vector with the best average precision is selected as the starting point for a new round of trials. This corresponds to a greedy aggregation scheme, except that some characters may see their weights increased more than once. After 50 rounds, the weight vector W that produced the greatest average precision at any point in the trials is chosen for final use. Table 2 shows the weights discovered in each of the four folds of the experiment, normalized to sum to one. Although the numbers vary considerably from fold to fold, some letters consistently prove more useful than others.

4. CONCLUSION

This paper leaves several questions unanswered as a subject for future work. Most obviously, the selection of character samples should be automated, perhaps using Clocksin's method [5] or some other. This would make document processing much faster and open the possibility of working with thousands of documents. With more samples per character the retrieval precision may rise; the work of Bar-Yosef et. al. in Hebrew used more than twenty samples each of just three Hebrew letters and reported 100% precision on their document set despite using only analysis of cavity shape [2]. On the other hand, errors in character recognition could also degrade the results. Another open question concerns how much the resolution of a document affects the results. While the experiments herein had 19 pixel line heights at minimum, many documents are available most freely in a low-resolution format that does not meet this standard. Future work should establish how much resolution is required for effective retrieval.

Very little research has looked at handwriting style identification for historical documents, and less still at Syriac. The results described in the previous section appear promising, but should be replicated for a larger validation set. Existing writer identification techniques, while developed for modern documents and a slightly different task, should nevertheless be evaluated and compared to the techniques presented here. We hope that the methods developed in this paper will form the core of a valuable research tool for scholars who study Syriac documents and the culture behind it.

Acknowledgment

The authors thank Michael Penn of Mount Holyoke College for proposing this project, and for his help in locating suitable documents for testing.

5. REFERENCES

[1] G. R. Ball, S. N. Srihari, and R. Stittmeyer. Writer identification of historical documents among cohort writers. In *Proc. Int. Conf. on Frontiers of Handwriting Recognition*, Kolkata, India, November 2010.

[2] I. Bar-Yosef, I. Beckman, K. Kedem, and I. Dinstein. Binarization, character extraction, and writer identification of historical Hebrew calligraphy documents. *Int. J. Doc. Anal. Recognit.*, 9(2):89–99, April 2007.

[3] A. Bhardwaj, A. Thomas, Y. Fu, and V. Govindaraju. Retrieving handwriting styles: A content based approach to handwritten document retrieval. In *Proc. Int. Conf. on Frontiers in Handwriting Recog.*, pages 265–270, 2010.

[4] N. Bulacu and L. Schomaker. Text-independent writer identification and verification using textural and allographic features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(4):701–717, April 2007.

[5] W. F. Clocksin. Handwritten Syriac character recognition using order structure invariance. In *Proc. 17th International Conference on Pattern Recognition*, volume 2, pages 562 – 565, Cambridge, UK, August 2004.

[6] W. F. Clocksin and P. P. J. Fernando. Towards automatic transcription of Syriac handwriting. In *Proc. Int. Conf. on Image Analysis and Processing*, pages 664–669, Mantova, 2003.

[7] K. S. Heal and C. W. Griffin. Syriac manuscripts from the Vatican library. Bibliotheca Apostolica Vaticana and Brigham Young University, 2005. CD-ROM.

[8] N. Howe. A Laplacian energy for document binarization. In *Int. Conf. on Document Analysis and Recgnnition*, 2011.

[9] G. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *IEEE 11th. Int. Conf on Computer Vision*, pages 1–8, 2007.

[10] E. Learned-Miller. Data-driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250, 2006.

[11] M. Panagopoulos, C. Papaodysseus, P. Rousopoulos, D. Dafi, and S. Tracy. Automatic writer identification of ancient Greek inscriptions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(8):1404–1414, Aug. 2009.

[12] M. Penn, 2010. Personal communication.

[13] S. Yoon, S. Choi, S.-H. Cha, and C. Tappert. Writer profiling using handwriting copybook styles. In *Proc. Eighth Int. Conf. on Doc. Anal. and Recog.*, pages 600–604, 2005.

[14] B. Zhang and S. N. Srihari. Handwriting identification using multiscale features. *J. of Forensic Document Examination*, 16:1–20, Fall 2004.