

Automated Color Clustering for Medieval Manuscript Analysis

Ying Yang*, Ruggero Pintus[†], Enrico Gobbetti[†] and Holly Rushmeier*

*Department of Computer Science, Yale University, USA

[†]CRS4, Italy

Abstract—Given a color image of a medieval manuscript page, we propose a simple, yet efficient algorithm for automatically estimating the number of its color-based pixel groups, K . We formulate this estimation as a minimization problem, where the objective function assesses the quality of a candidate clustering. Rather than using all the features of the given image, we carefully select a subset of features to perform clustering. The proposed algorithm was extensively evaluated on a dataset of 2198 images (1099 original images and their 1099 variants produced by modifying both spatial and spectral resolutions of the originals) from the Yale’s Institute for the Preservation of Cultural Heritage (IPCH). The experimental results show that it is able to yield satisfactory estimates of K for these test images.

Keywords—Medieval manuscript, color clustering, figure and capital letter extraction

I. INTRODUCTION

Recently, massive numbers of historic manuscripts have been digitized and made public. For massive collections, automatic computer-assisted techniques are increasingly important [1], since they can offer efficient access to various types of information in the documents. Most existing methods treat classic issues in content analysis such as word spotting [2], [3], [4] and text line segmentation [5], [6]. However, only a limited amount of effort has been dedicated to automatically investigating manuscript material properties, which are particularly important for medieval manuscript analysis.

An important issue in studying material properties is to obtain K , the number of colors used in a manuscript page. Automatically obtaining expected K for medieval manuscripts is by no means an easy task, in that variations in ink density and aging result in a large number of different RGB or spectral values on the page, while the actual number of colors is quite small. Particularly, it is commonly agreed that K is of great importance in pigment or ink-related applications such as (i) identifying possible pigment changes that indicate changes in scribal hand, (ii) identifying initials and differentiating between types of initials and (iii) identifying pages, from massive numbers of historical documents, that have unique features and thus require special attention, by analyzing their K (those pages are probably associated with particular values of K).

Estimating K is closely related to clustering. Existing methods using clustering techniques for analyzing historical handwritten document images include [1], [6], [7]. By using the SVM (Support Vector Machine), Grana et al. [1] propose a system for automatically extracting graphical elements from historical manuscripts, as well as identifying significant pictures from them. Recently, Pintus et al. [6] train an SVM classifier using the features of images with salient features and

then obtain text blocks and lines. Again with the SVM, Chen et al. [7] develop a method to classify pixels into *periphery*, *background*, *text block* and *decoration* categories. All these methods provide a high-level clustering, meaning that they can only group pixels into few broad classes, but are unable to fully satisfy scholar’s expectations such as distinguishing normal and historiated initials. Note that image clustering and segmentation are closely related, but not identical. The former emphasizes pixel similarity or dissimilarity from a global perspective, while the latter generally involves integrating and clustering features over local image patches [8], [9].

While K can indeed be obtained *manually* by inspecting each manuscript page of interest, this is very inefficient and often infeasible, when a lot of pages are involved. This motivates us to propose, to best of our knowledge, the *first completely automatic* K estimation method, assisting scholars in analyzing manuscript material properties. Moreover, another advantage of using a pre-computed K is the possibility of it making some semi-automatic, K -dependent manuscript analysis systems fully automatic. We demonstrate in Section IV-B the need for and use of K in an application, where an automated technique is particularly important. We are here interested in color number, but the idea can be extended to obtain the number of other characteristic-based pixel groups.

Contribution. The main contributions are: (i) to the best of our knowledge, the first, domain-specific algorithm for estimating K ; (ii) an extensive evaluation on more than 2000 images of pages of 7 different manuscripts and (iii) a demonstration of the importance of using K in a practical application.

Limitation. Our method is tailored to medieval manuscripts written in western languages (bar-like text). Nevertheless, a re-design of the template image and/or template matching strategy could adapt our pipeline to a broader range of documents.

II. PROBLEM FORMULATION

Our algorithm is built upon the observation that the text in a manuscript page image \mathbf{I} generally have the same color assignment (perceptually), but are colored distinctively from non-text pixels. Note that we only consider the *foreground* pixels within the main text blocks, which are computed using [6].

Let \mathbf{F} denote the features extracted from \mathbf{I} and $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{\tilde{K}}$ denote the clusters obtained from applying one *unsupervised* clustering algorithm to \mathbf{F} . We measure the quality of the clustering result using the widely used Davies-Bouldin criterion [10]:

$$\varepsilon(\tilde{K}, \mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{\tilde{K}}) = \frac{1}{\tilde{K}} \sum_{k=1}^{\tilde{K}} \max_{k' \neq k} \left\{ \frac{\bar{d}_k + \bar{d}_{k'}}{d_{k,k'}} \right\}, \quad (1)$$

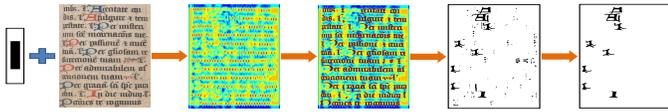


Fig. 1. Pixel candidates selection. Left to right: matching the template \mathbf{T} and image \mathbf{I} , we obtain matching scores \mathbf{S} and \mathbf{S}' . Then the mask \mathbf{P} is produced and pixels are selected for computing the features \mathbf{F} . High and low matching scores are shown in red and blue, respectively (zoom in to observe details).

where \bar{d}_k is the average Euclidean distance between each point in the k -th cluster \mathcal{C}_k and the centroid of \mathcal{C}_k and $d_{k,k'}$ is the Euclidean distance between the centroids of \mathcal{C}_k and $\mathcal{C}_{k'}$. The smaller $\varepsilon(\cdot)$ is, the better the clustering quality. Thus, we estimate K as

$$K = \arg \min_{\tilde{K} \in [K_{\min}, K_{\max}]} \{\varepsilon(\tilde{K}, \mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{\tilde{K}})\}. \quad (2)$$

K_{\min} and K_{\max} define the range of \tilde{K} , i.e., the number of colors that manuscripts generally use. We fix $K_{\min} = 1$ and $K_{\max} = 7$.

Rather than using all the foreground pixels when computing \mathbf{F} , we consider a subset. The reason is that an unbalanced data distribution (there are generally significantly more text than other foreground pixels) could make both the clustering algorithm used and quality measure in Eq. 2 fail. Thus, we propose a text-identification strategy (see Section III-B) to address the data dominance issue, ensuring the pixels used for \mathbf{F} computation are selected from *non-text* pixels.

III. APPROPRIATE K ESTIMATION

A. Pre-processing: Image Binarization

We start from converting the R , G , and B components of \mathbf{I} into the grayscale values according to $0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B$. For a multi-spectral image \mathbf{I} , we can obtain its R , G , and B values by downsampling the spectral data.

Next, a 256-bin histogram of the grayscale values is constructed and subsequently averaged using the nearest n data. We deem *foreground* pixels as those whose corresponding grayscale values are within $[0, \tilde{i} - \delta]$, with

$$\tilde{i} = \arg \max_i \{h_i : i \in \{\eta, \eta + 1, \eta + 2, \dots, 255\}\}, \quad (3)$$

and consequently assign to them 0 in the binary image \mathbf{B} ; otherwise *background* pixels with 1 in \mathbf{B} . Here, h_i denotes the number of elements in the i -th bin of the averaged histogram.

\mathbf{B} is produced after performing median filtering of \mathbf{B} using $S \times S$ neighborhood. The intention is to remove noise or outlier pixels. We experimentally found that the proposed binarization with $n = 20$, $\delta = 40$, $\eta = 64$ and $S = 8$ provides satisfactory results and outperforms the well-known method [11].

B. Pixel Candidates Selection

The selection of pixels used for computing the features \mathbf{F} has three steps as shown in Fig. 1.

Template image \mathbf{T} : Let H and W denote the text height/leading and text stroke width, respectively. We define the text-stroke-like \mathbf{T} as an $H \times 2W$ -sized matrix: $\mathbf{T}(i, j) = 0$, if $\alpha_H \cdot H \leq i \leq \beta_H \cdot H$ and $\alpha_W \cdot W \leq j \leq \beta_W \cdot W$; otherwise, $\mathbf{T}(i, j) = 1$. The parameters α_H , β_H , α_W and β_W are used to control the size of the black rectangle in \mathbf{T} . See Fig. 1 (left).

The text height H is computed by Pintus et al.'s method [12]. To obtain the text width W , we consider the lines that are produced by applying the LSD [13] to \mathbf{B} and whose angles with respect to the y -axis are smaller than 8 degrees. We define the line-to-line distance $d(i, i', j) = 0.5 \cdot \text{abs}\{x_1(i', j) + x_2(i', j) - x_1(i, j) - x_2(i, j)\}$, where $x_1(i, j)$ and $x_2(i, j)$ are the x coordinates of the two end points of the i -th line from the j -th text line segment by [6]. By taking the single stroke width as $W(i, j) = \min_{i', i' \neq i} d(i, i', j)$, we obtain W by averaging all $W(i, j)$ corresponding to the peak of the histogram of all $W(i, j)$ with $2 \cdot \lceil \max_{i,j} W(i, j) - \min_{i,j} W(i, j) \rceil$ bins. Here, $\lceil \cdot \rceil$ denotes the mathematical ceiling function.

Obtaining α_H and β_H exploits the fact that the text strokes in \mathbf{I} can be considered as the connected components in \mathbf{B} (corresponding to black pixels). For the i -th text line segment, we compute α_H^i and β_H^i as the averages of all the minimum and maximum row indices of the pixels within the *valid* components. A component is deemed *valid* if it is not too small, i.e., its height and width must be larger than $\lceil H/3 \rceil$ and $\lceil 0.8 \cdot W \rceil$, respectively. Finally, all α_H^i and β_H^i are separately averaged over all the text line segments, yielding α_H and β_H . As W already represents the stroke width, we set $\alpha_W = 0.5$ and $\beta_W = 1.5$ based on empirical observation to include background information in \mathbf{T} .

Template matching: Given \mathbf{B} and \mathbf{T} , we match them in the frequency domain using the FFT-based correlation, resulting in a matrix \mathbf{S} of matching scores normalized to $[0, 1]$.

We update \mathbf{S} , yielding \mathbf{S}' with large scores being assigned to *text*. Let κ be the threshold differentiating high and low scores. For each $\mathbf{S}(x', y') \geq \kappa$, an $H \times 2W$ rectangular window, centered at (x', y') , is computed. $\mathbf{S}(x, y) = \mathbf{S}(x', y')$ is repeatedly applied to every within-window *foreground* pixel if $\mathbf{S}(x, y) < \mathbf{S}(x', y')$, i.e., $\mathbf{S}(x, y) = \max_{(x', y') \in \Psi} \{\mathbf{S}(x', y') : \mathbf{S}(x', y') \geq \kappa\}$, where Ψ is the set of positions whose corresponding windows contain the pixel at (x, y) .

Pixel selection: Preliminarily, the pixel candidates are selected as those with $\mathbf{S}'(x, y) < \kappa$ ($\kappa = 0.75$) and their positions are indicated by the zero-valued elements in \mathbf{P} : $\mathbf{P}(x, y) = 0$, if $\mathbf{S}'(x, y) < \kappa$ and $\mathbf{B}(x, y) = 0$; otherwise, $\mathbf{P}(x, y) = 1$.

Considering that there could still exist some text in the preliminary selection, we apply the k -means ($k = 2$) to the numbers or densities of the connected pixels and then prune the pixels (likely text) whose corresponding group has a lower density, producing the final pixel candidates for \mathbf{F} computation as described blow.

C. \mathbf{F} Computation

For each pixel candidate, we compute an N -dimensional feature \mathbf{F}_i . Here, N varies, depending on image type, e.g., $N = 60$ for RGB images, while, for 8-channel multi-spectral images, $N = 68$ to include data from distinct bands.

The first 9 components of \mathbf{F}_i are the color information, including H , S , V (hue-saturation-value), R , G , B , R^* , G^* and B^* . Here, R^* is the normalized R , i.e., $R^* = R/(R + G + B)$. Analogously, G^* and B^* can be computed. Note that we experimentally found that combining information from multiple color spaces can make \mathbf{F}_i more discriminative and hence K more accurate.

We also take into account the relationship between neighboring pixels when computing \mathbf{F}_i . For a varying size $\{H \times n \cdot W : n = 2, 3, 4\}$ window centered at each pixel candidate, we obtain the H , S , V components of all the foreground pixels within it and separately compute the *mean*, *standard deviation*, *skewness*, *energy* and *entropy*, thus creating another 45 components of \mathbf{F}_i . The final 6 components are from intensity perspective, i.e., they are the numbers of the within-window zero-valued pixels of \mathbf{P} with $\kappa = 0.7$.

To improve classification, each type of feature in \mathbf{F} is scaled into $[0, 1]$ so that the features with large numerical ranges do not dominate those with small numerical ones [14].

D. K Computation

Given the features \mathbf{F} , we can compute K . Rather than employing Eq. 2 directly, we use a variant defined as follows:

$$K = 1 + \arg \min_{\tilde{K} \in [K_{\min}, K_{\max}]} \{\varepsilon(\tilde{K}, \mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{\tilde{K}})\}, \quad (4)$$

due to the fact that \mathbf{F} used for clustering is text-independent. By adding 1, we create an extra group for the “ignored” texts.

However, if \mathbf{F} is text-dependent (due to the existence of text in the pixel candidates), Eq. 4 might yield an over-estimated K . To deal with this issue, we apply an unsupervised learning method to classify all the foreground pixels into K clusters given their features computed by the method in Section III-C. Considering the significant number of the text pixels, we easily identify the *text-cluster* that contains text. For each non-text cluster, we compare its averaged 9 color components against those of the *text-cluster* and keep K intact if the color information is dissimilar, i.e., there are four or more difference values larger than 45, which we found reasonable in our tests; otherwise, $K = K - 1$. Note that these 9 color components are scaled into $[0, 255]$ for comparison. To avoid confusion, we shall use K^{bf} (before update) and K^{af} (after update) to denote K computed directly from Eq. 4 and after addressing the potential over-estimation issue, respectively.

IV. EXPERIMENTAL RESULTS

The proposed method was evaluated on the images from the Yale’s Institute for the Preservation of Cultural Heritage (IPCH) including 1027 RGB and 72 8-band multi-spectral images of pages of 7 different manuscripts¹, as well as their 1099 variants produced by modifying their spatial and spectral resolutions. Since we observed that the two unsupervised learning methods (k -means and Expectation-Maximization (EM) algorithm for the Gaussian mixture model) have similar performance in K estimation, we only show the results from the EM method. Note that although numerous methods have been proposed to analyze medieval manuscripts, they have completely different focuses as ours and thus we will not provide any comparison results.

A. Accuracy of Estimated K

To compute the estimation accuracy, we obtain the difference $\Delta K_{\text{EM}} = K_{\text{EM}} - K_e$ between the automatically generated K_{EM} by the EM method against the ground truth data K_e ,



Fig. 2. The triple $(K_{\text{EM}}^{\text{bf}}, K_{\text{EM}}^{\text{af}}, K_e)$ is $(4, 3, 3)$, $(6, 6, 6)$, $(4, 4, 4)$, $(4, 4, 4)$, $(4, 3, 3)$, $(6, 5, 5)$, $(6, 3, 3)$ and $(5, 5, 3)$ for few images (zoom in to observe details). The two right-most images are multi-spectral data.

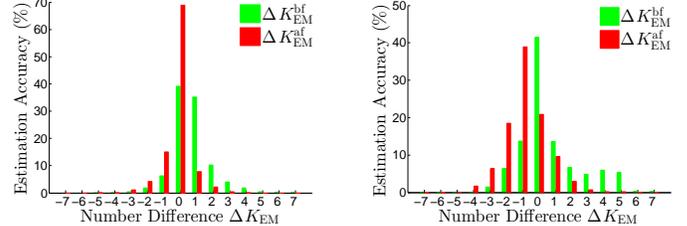


Fig. 3. Accuracy of K for the original 1027 RGB images (left) and for the same 1027 RGB images at lower spatial resolution with a scaling factor of 0.25 (right).

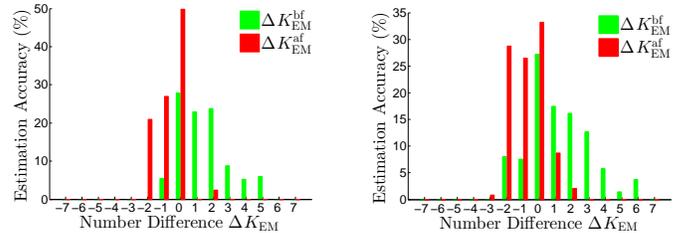


Fig. 4. Accuracy of K for the original 72 8-channel multi-spectral images (left) and for the same 72 images converted to RGB only (right).

which were established through manually inspecting all the test images. Fig. 2 shows the estimated $K_{\text{EM}}^{\text{bf}}$, $K_{\text{EM}}^{\text{af}}$ and K_e for few challenging pages.

Accuracy w.r.t spatial resolution. Fig. 3 (left) plots the histograms of $\Delta K_{\text{EM}}^{\text{bf}}$ and $\Delta K_{\text{EM}}^{\text{af}}$ for the original 1027 RGB images. We obtain perfect K ($\Delta K = 0$) for up to 70% of the test images and $|\Delta K_{\text{EM}}^{\text{af}}| \leq 1$ for 89%, demonstrating high-level reliability of our method. Also, the figure demonstrates that updating K yields improved estimation performance, since there are more $\Delta K_{\text{EM}}^{\text{af}} = 0$ than $\Delta K_{\text{EM}}^{\text{bf}} = 0$.

Fig. 3 (right) illustrates the results for the same 1027 RGB images at lower spatial resolution. Still, we are successful in obtaining satisfactory K , as we have both $|\Delta K_{\text{EM}}^{\text{bf}}| \leq 1$ and $|\Delta K_{\text{EM}}^{\text{af}}| \leq 1$ for about 70% of the test images. Indeed, the performance degrades slightly, but this is expected as higher resolution means more useful information available.

Accuracy w.r.t spectral resolution. Fig. 4 shows the results for images with different spectral sampling. As expected, we have greater chance of success for the multi-spectral data than their spectral resolution-reduced RGB due to their larger volume of useful information.

For multi-spectral data, we can have similar conclusions as made for the original RGB images, i.e., successful K estimation and improved performance due to K update. However, note that it is not reasonable to compare the results in Figs. 3 and 4, since the RGB and multi-spectral data were produced from digitizing different manuscripts. As Fig. 2 shows, the multi-spectral images are far more challenging than RGB, in terms of both layout structure and text shape.

¹Publicly available at: <http://hdl.handle.net/10079/cz8w9v8>



Fig. 5. The figures, capital letters and line fillers extracted from challenging pages are highlighted using red rectangular boxes (zoom in to observe details).

TABLE I. FIGURES, CAPITAL LETTERS AND LINE FILLERS EXTRACTED FROM ALL THE ORIGINAL 1027 RGB IMAGES. TP, FP AND FN INDICATE TRUE POSITIVE, FALSE POSITIVE AND FALSE NEGATIVE, RESPECTIVELY.

Manuscript Name	TP	FP	FN	Precision	Recall
BeineckeMS10	642	89	5	87.82%	99.23%
BeineckeMS109	1253	162	55	88.55%	95.80%
BeineckeMS310	2363	262	90	90.02%	96.33%
BeineckeMS360	3125	52	350	98.36%	89.93%
# Manuscripts	— 4	7383	565	92.89%	93.66%

B. Application in Segmentation

We demonstrate the need for and use of K in automatic extraction of special symbols (here, figures, capital letters and line fillers), which are generally colored distinctively from texts. For scholars, figures and capital letters are important in differentiating penwork and painting and line fillers identify litanies.

Analogous to the K update process, the extraction starts from obtaining $K = K_{EM}^{af}$ groups and then considers all the $K - 1$ clusters rather than the *text-cluster*, in that they likely correspond to the figures, capital letters and line fillers. After creating the binary image \mathbf{M} , where 0 is assigned to all the pixels within the $K - 1$ groups of interest and 1 otherwise, we extract from it a set of candidate connected components $\{\mathcal{T}_i\}$ that meet $H_{\mathcal{T}_i} \geq \lfloor 0.2 \cdot H \rfloor$ and $W_{\mathcal{T}_i} \geq 2 \cdot W$. $\lfloor \cdot \rfloor$ stands for the mathematical floor function; $H_{\mathcal{T}_i}$ and $W_{\mathcal{T}_i}$, respectively, denote the height and width of \mathcal{T}_i . Then, we update \mathbf{M} by assigning 1 to all the pixels that do not belong to $\{\mathcal{T}_i\}$, yielding $\mathbf{M}^{(1)}$. Next, we compute the horizontal-direction distance maps for $\mathbf{M}^{(1)}$ by:

$$\mathbf{D}_H(x, y) = \min_{y'} \{|y - y'| : \mathbf{M}^{(1)}(x, y') = 0\}. \quad (5)$$

We set $\mathbf{D}_H(x, y)$ to be the image width if $\mathbf{M}^{(1)}(x, y') = 0$, $\forall y'$ does not exist. The binary image $\mathbf{M}_H^{(2)}$ is generated with $\mathbf{M}_H^{(2)}(x, y) = 0$ if $\mathbf{D}_H(x, y) \leq 2 \cdot W$; $\mathbf{M}_H^{(2)}(x, y) = 1$ otherwise. Similarly, we obtain the vertical-direction distance map $\mathbf{D}_V(x, y)$ and its corresponding binary image $\mathbf{M}_V^{(2)}$ with the threshold $\lfloor H/4 \rfloor$. Given $\mathbf{M}_H^{(2)}$ and $\mathbf{M}_V^{(2)}$, we generate $\tilde{\mathbf{M}}$ after performing element-by-element bitwise AND (&) operation between $\mathbf{M}_H^{(2)}$ and $\mathbf{M}_V^{(2)}$, i.e., $\tilde{\mathbf{M}} = \mathbf{M}_H^{(2)} \& \mathbf{M}_V^{(2)}$. Finally, we extract the figures, capital letters and line fillers as the connected components \mathcal{T}_i in $\tilde{\mathbf{M}}$ that satisfy (i) the zero-valued (non-text) pixels occupy more than 1/10 of the total pixels embraced by a \mathcal{T}_i ; (ii) the zero-valued pixels is at least 2 times the texts embraced by \mathcal{T}_i in terms of pixel number; as well as (iii) either of $H_{\mathcal{T}_i} \geq \lfloor 0.6 \cdot H \rfloor$ and $W_{\mathcal{T}_i} \geq 4 \cdot W$, or $H_{\mathcal{T}_i} \geq \lfloor 0.3 \cdot H \rfloor$ and $W_{\mathcal{T}_i} \geq 6 \cdot W$. Note that producing $\tilde{\mathbf{M}}$ is to fill the “holes” between the isolated parts from a figure, capital letter or line filler and also that the numerical values used are set based on empirical observation to the general physical layout and structure of various manuscripts.

Table I shows that we can achieve a precision of up

to 98.36% and a recall of up to 99.23%, indicating that reasonable K is critical for the clustering algorithms to succeed in manuscript analysis. Fig. 5 visualizes the figures, capital letters and line fillers extracted from some challenging pages.

V. CONCLUSION

We have presented an *automatic* method for obtaining K for medieval manuscripts. The proposed template matching-based strategy that distinguishes *text* and *non-text* pixels contributes greatly to the success of obtaining K . Our method has been evaluated on 2198 images of pages of 7 different manuscripts and the results show that it can find expected K with a high likelihood, i.e., we achieve $K = K_e$ and $|K - K_e| \leq 1$ for up to 70% and 92% of the test images. K can be used in various scenarios such as detecting ink changes, differentiating between painted and penwork initials and identifying litanies. Our method has been tested mainly on the manuscripts with straight stroke-like text; however, the general idea can be extended to work for the manuscripts with other style-shaped texts via re-designing the template image or matching strategy. This extension will be one direction of our future work.

Acknowledgments. This work was partially supported by the Digitally Enabled Scholarship with Medieval Manuscripts (DESM) project funded by the Mellon Foundation. We also acknowledge the contribution of Sardinian Regional authorities for Data Intensive Computing activities.

REFERENCES

- [1] C. Grana, D. Borghesani, and R. Cucchiara, “Picture extraction from digitized historical manuscripts,” in *Proc. of the ACM International Conference on Image and Video Retrieval*, 2009, pp. 22:1–22:8.
- [2] T. M. Rath and R. Manmatha, “Word image matching using dynamic time warping,” in *CVPR*, vol. 2, 2003, pp. II–521 – II–527.
- [3] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, “Segmentation-free word spotting with exemplar svms,” *Pattern Recognition*, vol. 47, no. 12, pp. 3967–3978, 2014.
- [4] A. Fischer, A. Keller, V. Frinken, and H. Bunke, “Hmm-based word spotting in handwritten documents using subword models,” in *ICPR*, Aug 2010, pp. 3416–3419.
- [5] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, “Text line and word segmentation of handwritten documents,” *Pattern Recognition*, vol. 42, no. 12, pp. 3169–3183, 2009.
- [6] R. Pintus, Y. Yang, E. Gobbetti, and H. Rushmeier, “A TaLISMAN: Automatic text and Lline segmentation of historical MANuscripts,” in *12th EUROGRAPHICS Workshop on GCH*, 2014.
- [7] K. Chen, H. Wei, J. Hennebert, R. Ingold, and M. Liwicki, “Page segmentation for historical handwritten document images using color and texture features,” in *ICFHR*, 2014.
- [8] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Trans. on PAMI*, vol. 33, no. 5, pp. 898–916, 2011.
- [9] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *IJCV*, vol. 59, no. 2, pp. 167–181, 2004.
- [10] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE Trans. On PAMI*, vol. PAMI-1, no. 2, pp. 224–227, 1979.
- [11] J. Sauvola and M. Pietikäinen, “Adaptive document image binarization,” *Pattern recognition*, vol. 33, no. 2, pp. 225–236, 2000.
- [12] R. Pintus, Y. Yang, and H. Rushmeier, “ATHENA: Automatic text height extraction for the analysis of text lines in old handwritten manuscripts,” *ACM JOCCH*, vol. 8, no. 1, pp. 1–25, 2014.
- [13] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, “LSD: A fast line segment detector with a false detection control,” *IEEE Trans. On PAMI*, vol. 32, no. 4, pp. 722–732, 2010.
- [14] Y. Yang and I. Ivrişimtzis, “Mesh discriminative features for 3d steganalysis,” *ACM TOMM*, vol. 10, no. 3, pp. 27:1–27:13, Apr. 2014.