# A* Path Planning for Line Segmentation of Handwritten Documents

Olarik Surinta, Michiel Holtkamp, Faik Karabaa, Jean-Paul van Oosten, Lambert Schomaker and Marco Wiering
Institute of Artificial Intelligence and Cognitive Engineering, University of Groningen
Nijenborgh 9, Groningen, The Netherlands
Email: {o.surinta, m.j.holtkamp, m.f.karabaa, j.p.van.oosten, l.r.b.schomaker, m.a.wiering}@rug.nl

***Abstract*—This paper describes the use of a novel $A^*$ path-planning algorithm for performing line segmentation of handwritten documents. The novelty of the proposed approach lies in the use of a smart combination of simple soft cost functions that allow an artificial agent to compute paths separating the upper and lower text fields. The use of soft cost functions allow the agent to compute near-optimal separating paths even if the upper and lower text parts are overlapping in particular places. Experimental results on several medieval, historical and contemporary handwritten manuscripts show that the proposed method needs little tuning and performs very well.**

***Keywords*—Document analysis, $A^*$ path-planning algorithm, Handwritten historical manuscripts, Line segmentation.**

## I. Introduction

Current search engines are very useful for people to search for information on the Internet. However, there is still a lot of information available on the internet that cannot be used efficiently, because this information is contained in documents that cannot be read or understood in an effective way by current search engine technology. We are especially interested in making handwritten documents accessible to people by recognizing the contents of these documents and making them searchable with a new generation of search engines tailored to handwritten documents.

The world digital library consists of a collection of millions of handwritten historical manuscripts, maps, photographs and other cultural documents. Several projects focus on disclosing this information to the public. The digital dead sea scrolls (*DDSS*) project[1] allows people to explore the ancient dead sea scroll manuscripts, and in this project several advanced imaging and web technologies have been developed. The technology behind the *DDSS* project allows the user to retrieve the manuscript images and translations so that people can have access to this important historical information.

The *MONK* system[2] system is a historical manuscript recognition system, consisting of many techniques for searching for words in historical manuscript collections. The system consists of different handwriting recognition algorithms, which are trained by crowd sourcing techniques where people all over the world can create ground-truth labels for words and lines that occur in the historical documents [1], [2].

In this paper we describe a novel line segmentation algorithm for handwritten documents. Line segmentation is one of the first techniques that needs to be applied to a document, before individual words or characters can be found and (parts of) the handwritten text can be automatically recognized. Our method is based on A* path-planning [3] and combines this well-known method with different cost functions to find near-optimal paths separating upper and lower handwritten text areas. By using soft cost functions, our method can find very good separating lines, even if upper and lower text parts are overlapping in some areas.

**Related work.** The first step that is performed by a line segmentation algorithm is to find potential candidates for starting points of lines separating upper and lower text fields. Most often this step uses horizontal projection profiles, in which the amount of black ink is summed over the x-axis to obtain a profile indicating text areas having a lot or little to no black ink. Bulacu *et al.* [2] proposed the smoothed horizontal projection profile to more robustly detect peaks and valleys of the binarized document image. In [4], the handwritten document is divided into chunks, and then the smoothed projection profile in each chunk is calculated. Then, the valleys in the projection profile are considered as the starting state of the text lines. Also in [5], the baselines of the valleys are used to define the starting states for the separating lines.

After defining the starting states, various methods for finding the line separating upper and lower text ares have been proposed. In [2], a droplet method that preserves the ink connectivity is developed. Beginning in the starting state, first an initial straight path is generated. Then, the document image is turned 90 degrees and an artificial water droplet is moved from the top to the bottom of the page. This droplet tries to move around the ink along the straight path with the aim to preserve the ascenders and descenders in the final segmentation. The experimental results on a part of a dataset named "the cabinet of the Dutch Queen" containing 32,816 lines (31.6 per page) showed that 99.8% of the lines were correctly segmented.

Garz *et al.* [6] proposed a binarization-free text line segmentation algorithm. First, parts-of-character interest points are located by means of the Difference of Gaussians (DoG) filter after which locations of local minima and maxima are found in the gray-scale image. These detected interest points then represent the most significant locations of portions of the text. The energy map is computed around the located text points, and the so-called seam carving technique is used to

[1] The *DDSS* project is available at `http://dss.collections.imj.org.il`

[2] The *MONK* is available at `http://www.ai.rug.nl/∼lambert/Monk-collections-english.html`

find a connected path with minimum cost that goes through low energy parts. This technique provides a hit rate of 0.9865 on 1,431 text lines of the *Saint Gall*[3] dataset.

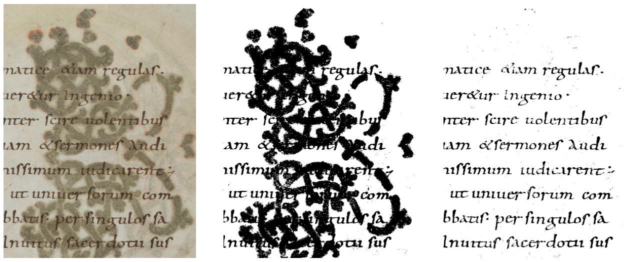
Louloudis *et al.* [7] proposed the use of the Hough transform to perform line segmentation. In this method, the average width and height of connected components in the whole document are computed and used for partitioning the text into sub-areas. The sub-areas are again partitioned into equally sized blocks, after that the ink gravity enter is computed in each block. Finally, the set of all gravity center points is processed by the linear Hough transform to find a straight separating line. This technique provided a detection rate of 90.4% on the ICDAR 2007 handwriting segmentation contest dataset.

Although these previous methods perform well when the text is quite well structured, they still suffer from inaccurate line segmentations in case text blocks above and below the separating line are overlapping. Therefore the aim of this research is to develop a robust method to deal with this overlapping text problem and to obtain accurate line segmentations for different kinds of manuscripts.

**Contributions.** This paper proposes a novel line segmentation method based on the $A^*$ path-planning algorithm. This well-known path-planning algorithm is combined with a number of cost functions to determine the optimal path separating upper and lower text areas. The cost functions have been designed in order to allow the separating path to go through text areas, which is very useful in case upper and lower text fields are overlapping or connected. The $A^*$ path-planning algorithm has been widely used in the field of artificial intelligence, however, this paper shows that this technique can also be very beneficial for line segmentation purposes. We have performed experiments on the *Saint Gall* dataset, as well as on several other historical and contemporary handwritten manuscripts.

## II. Text Line Localization

An important aspect of line segmentation is to specify the exact location of the text line. Text line localization is performed in two steps: binarization and projection profile analysis. In the first step, the handwritten document images are binarized using a binarization technique such a technique takes into account the diversity of document images, texts, images, mixtures of texts and images, line drawings, noisy and even degraded document images. Bulacu *et al.* [2] and Surinta *et al.* [8] use Otsu's algorithm, a global binarization technique, in their work. Otsu's algorithm uses one threshold value to process an entire document image. This algorithm is not performing well when the background of the document image is complicated as shown in Fig. 1(b). On the other hand, Sauvola's algorithm [9] for local binarization copes effectively with complex background documents as shown in Fig. 1(c). Because the contrast between handwriting and background is low (see Fig. 1(a)), the threshold value is calculated by the mean and standard deviation of the local neighbourhood of the gray pixel values. This threshold value has to be calculated for each pixel [10], [11].

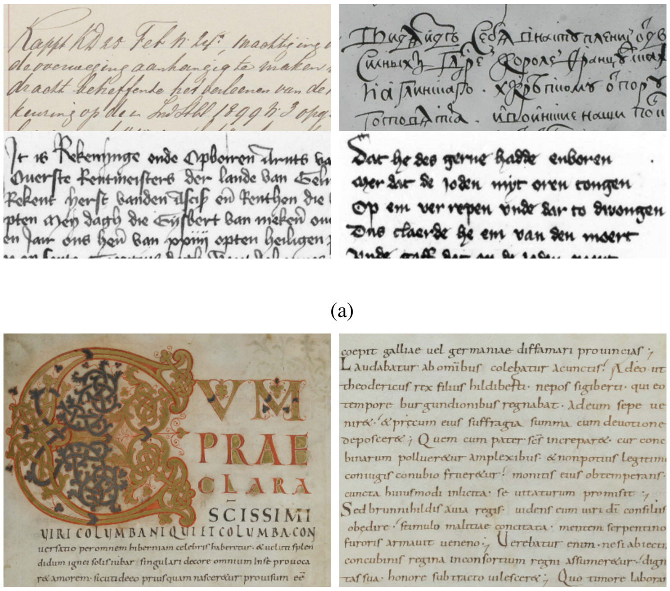


(a) (b) (c)

Fig. 1. Results of the document image after using binarization technique. (a) The original handwritten document image, (b) background noise is removed by Otsu's algorithm and (c) Sauvola's algorithm.

The experimental evaluation of the proposed method is realised on the handwritten historical manuscript datasets including the *MONK* and the *Saint Gall* dataset (Fig. 2). The *Saint Gall* dataset is written in $9^{th}$ century, Latin script and contains 60 pages. Each page is written in one column and contains a graphic [6], [12]. In this research, we have used Sauvola's algorithm with a window size of 20 × 20 pixels [12] to convert the gray document image to a binary document image, the result of which is shown in Fig. 1(c). The structure of the characters is legible [13] when zoomed into pixel level, see Fig. 3.

The second step uses the concept of projection profile analysis [14] for finding the location of the text lines in the handwritten document image. The horizontal ink density histogram $h$ of the document image is computed by taking the sum of the black pixel values in the corresponding row, then storing the value into a vector. Subsequently, the extrema are extracted from the vector. In this case, we are considering



(a)

(b)

Fig. 2. A variety handwriting styles of handwritten historical manuscript samples. (a) Samples from the *MONK* dataset. (b) The *Saint Gall* dataset, which is one dataset of the *IAM* Historical document database (*IAM-HistDB*).

[3] The *Saint Gall* dataset is available at http://www.iam.unibe.ch/fki/databases

Fig. 3. Result of the Sauvola's algorithm when zooming image to pixel level.

local maxima and local minima [2] according to persistence[4]. The persistence threshold is defined as $(\mu_h - \sigma_h)$. The local maxima represents the estimated text lines. Furthermore, the local minima $lm$ is set as the starting state of the $A^*$ path planning $(s_i = (0, y_{lm}))$ and the goal state is set at the same y-value of $s_i$ $g_i = (w, y_{lm})$, where $w$ is the width of the handwritten image and $y$ is the y-value.

## III. The novel $A^*$ Path-Planning Algorithm for Text Line Segmentation

Path planning can be applied extensively to robotic systems, driving directions, and even games. The aim of path planning is discovering and generating a full path, which allows the agent to reach to the destination. Path planning can be completely recognized and computed depending on the environment map representation [15]. The environment map contains many obstacles. Hence, the $A^*$ algorithm acquires some knowledge to compute the cost function from the environment map.

The $A^*$ path-planning algorithm uses the heuristic function to discover and appropriate solution to reach the goal state. It combines the cost of the actual path and the heuristic function to estimate cost from the given state to the goal state. The equation is defined as follows:

$$F\left(n\right) = G\left(n\right) + H\left(n\right) \tag{1}$$

Where $G\left(n\right)$ is the actual traveling cost from the starting state to reach state $n$. Therefore, the $G\left(n\right)$ of each state indicates the total path cost from starting state until state $n$. $H\left(n\right)$ is the heuristic cost of the distance from state $n$ to the goal state [16].

The heuristic function, which provides the information to estimate the possible path to the goal state, uses the Euclidean distance measurement. The agent that applies the Euclidean distance measurement moves more realistically than the Manhattan distance. Consequently, the Euclidean distance produces the shortest possible path. The heuristic function is computed as follows:

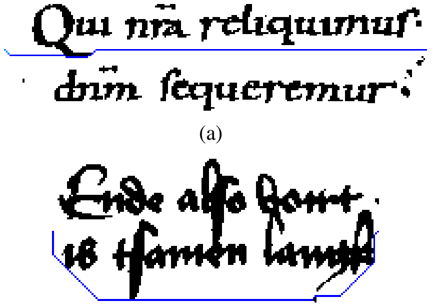
$$H\left(n\right) = \left\|X_n - X_g\right\| \tag{2}$$

Where $X_n$ is current state $n$. $X_g$ is the goal state.

### A. The Problem with The Unreachable Goal State

$A^*$ path-planning algorithm cannot address the problem of the unreachable goal state. For example, a goal state within a circle. The agent is not allowed to move through obstacles and then never figures out the optimal path. However, the new approach of the $A^*$ path-planning algorithm allows the agent to move through the obstacles.

We are interested in the line segmentation of the handwritten historical manuscripts. In this study, the A* path-planning algorithm is highly effective when the components of two lines do not overlap. On the other hand, the output of the algorithm is incorrect, when the handwriting between two lines are overlapped. Results of the $A^*$ path-planning algorithm are shown in Fig. 4(a) (correct) and Fig. 4(b) (incorrect).


(a)

(b)

Fig. 4. Illustrations for $A^*$ path-planning algorithm. (a) The agent is completely divided betweeen two character lines. (b) The agent cannot divide the two overlapping lines, because it cannot move through the obstacles.

The $A^*$ path-planning algorithm operates differently in our approach. In the $A^*$ algorithm, obstacle nodes are not considered as valid movements. This knowledge makes the agent to move away from the obstacle. In the proposed $A^*$ path-planning algorithm, obstacles are excluded. The proposed $A^*$ algorithm allows the agent to move over the obstacle. Consequently, the problem of the overlapping text lines is solved at the algorithm level.

### B. New Cost Functions of The $A^*$ Path-Planning Algorithm

The approach $A^*$ path-planning algorithm consists of two cost functions: $G\left(n\right)$ and $H\left(n\right)$. The cost functions estimate the traveling cost from the node $n$ until the goal state is achieved. In addition, the lowest $F\left(n\right)$ cost is computed from cost functions. The new cost function concentrates on two functions: *1*) the minimum distance cost function and *2*) the neighbor cost function.

*1) The Minimum Distance Cost Function $D\left(n\right)$:* This cost function controls the agent to move upward $(y_u)$ and downward $(y_d)$. It consists of the distance $d$ of the agent from obstacles in the vertical direction. The minimum distance cost function is computed as follows:

$$D(n) = \frac{C}{1 + min\left(d\left(n_{y_u}\right), d\left(n_{y_d}\right)\right)} \tag{3}$$

Where $C$ is a constant value. $d\left(n_y\right)$ is the distance between $n_{yu}$ and the closet object in the $y_u$, the distance is set to a

[4] Extracting and filtering minima and maxima of 1D functions package is available at http://www.mpi-inf.mpg.de/~weinkauf/notes/persistence1d.html

maximum value when no object is found in that direction. The closet distance value is used.

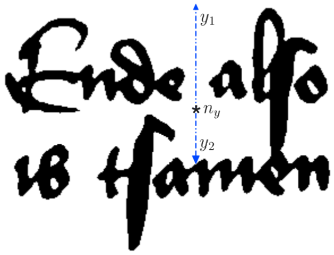
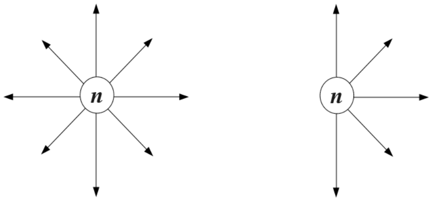


Fig. 5. Illustration for the minimum distance cost function. The minimum cost is finding the closest distance value between $n_y$ and objects in upward direction $y_u$ and downward direction $y_d$. The minimum distance value between $y_1$ and $y_2$ is selected

*2) The Neighbor Cost Function $N(n)$:* The neighbor cost $N(n)$ is explored through all possible neighbor nodes around the current node $n$. The neighbor cost is defined as 10 and 14. The cost of 10 for vertical and horizontal directions and the cost of 14 for diagonal directions. The reason is that, the diagonal direction distance is approximately 1.4 times longer than distance of the horizontal and the vertical direction.[5] Generally, the algorithm explores in 8 directions: vertical, horizontal and diagonal, as shown in Fig. 6(a).



(a) (b)

Fig. 6. Direction of the neighbor cost from the current node $n$ to its neighbor nodes. (a) The 8-directional movement as represent in the $A^*$ path-planning algorithm. (b) The proposed 5-directional movement.

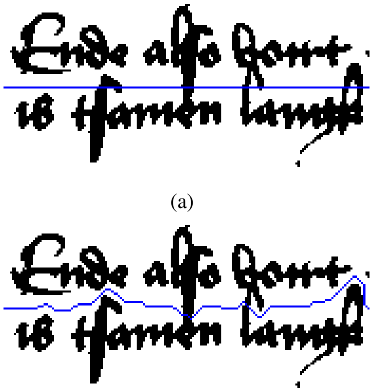
This approach for the $A^*$ path-planning algorithm to the handwritten historical manuscripts is applied. We have used 5-directional movement, as shown in Fig. 6(b), preventing the agent from moving backwards.

In order to apply the cost function, in Fig. 7(a), by the low $C$ value of $D(n)$ the line is made more straight. The $D(n)$ controls the agent not to move far away from $s_y$. Whereas, in Fig. 7(b) the agent moves more flexible from $s_y$ with the high $C$ value. Furthermore, Fig. 7 is applying $D(n)$. The distance cost is set to 0.9.

The proposed $A^*$ path-planning algorithm can now be described by the following equation:

$$F(n) = G'(n) + H(n) \tag{4}$$

Where $H(n)$ is the heuristic cost as before and $G'(n)$ equation is defined as follows:



(a)

(b)

Fig. 7. Results of the new $A^*$ path-planning algorithm. The $C$ values of $D(n)$ that are used in these examples are 10 and 250, respectively.

$$G'(n) = D(n) + N(n) \tag{5}$$

Where $M(n)$ is the minimum distance cost function and $N(n)$ is the neighbor cost function.

## IV. Experimental Evaluation

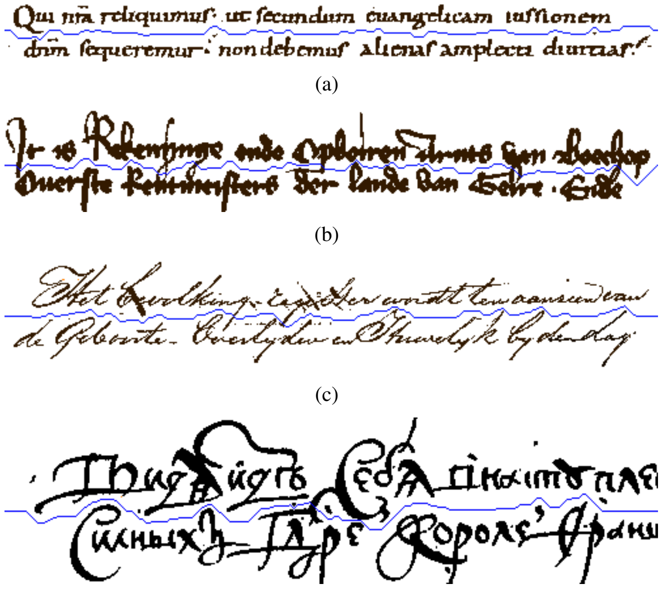
The line segmentation system is applied on two handwritten historical manuscripts from the *MONK* and the *Saint Gall* dataset. Results of the novel $A^*$ path-planning algorithm are shown in Fig. 8. The text lines are separated by the optimal path (i.e. the path with the lowest cost). Firstly, the handwritten historical manuscripts are binarized using Sauvola's algorithm. Secondly, the smoothed horizontal ink density histogram of the binary image is calculated. Then, peaks of the horizontal ink density histogram are detected by the local maxima method. The starting state of each line is set between each peak. Finally, the novel $A^*$ path-planning algorithm is applied. Most importantly, the agent of the algorithm allow to move through the handwritten texts. Some output samples of our method are shown in Fig. 9.

The ground truth is acquired manually with the help of a tool developed for this task. This tool presents a scanned document in a web page, and allows the human user to mark handwritten text with the mouse pointer by selecting a vertical area (denoted by two y-values in the image) using JavaScript (See Fig. 10). After annotating these text lines, the page image is split as follows: for each line annotation the area above and below the line area are whitened in a copy of the original image. This preserves the original image size. Descenders and ascenders from the lines above and below respectively are not removed by this process; they are removed manually by whitening them with a simple paint program (e.g., xpaint).

Both the input images and output images are losslessly compressed to prevent any influence from lossy compression artefacts. Each line annotation image is named as the original image, appended with a line number. This allows the matching of the ground truth images with the output images of the approach $A^*$ path-planning algorithm.

For evaluating the performance of the line segmentation algorithm, Li *et al.* [18] proposed the pixel-level hit rate and

[5] the neighbor cost are followed from $A^*$ Pathfinding for Beginners of `Patrick Lester`.



(a)

(b)

(c)

(d)

Fig. 8. The novel $A^*$ path-planning algorithm results. The $C$ value is experimentally established of 250.

the line accuracy measure on the binary image. Suppose a matrix $P$ of size $M \times N$ is computed, where $M$ is ground-truth lines and $N$ is detected lines. $(P_{ij})_{M \times N}$ where $P_{ij}$ is the shared black pixels between the $i^{th}$ ground-truth line and $j^{th}$ detect line for $i = 1, 2, ..., M$ and $j = 1, 2, ..., N$ [6], [19]. The pixel-level hit rate equation is defined as follows:

$$Hr = \frac{G\left(S_{max}\right)}{|GT \cup R|} \tag{6}$$

Where $G$ is the maximum value of shared black pixels, $GT$ is the set of black pixels in the ground-truth line, $R$ is all black pixels found by our algorithm including pixels which are not in the ground-truth. $S_{max}$ is defined as follows:

$$S_{max} = arg \max_{s} G\left(S\right) \tag{7}$$

Where $G\left(S\right)$ is the total number of shared black pixels between ground-truth line and detected line.

The performance method at the text-line level [18], [19] is evaluated. The line $i$ is correctly detected if $\frac{G_{ij}(S_{max})}{|GT_i|} \geq 0.9$ and $\frac{G_{ij}(S_{max})}{|R_j|} \geq 0.9$. Where $GT_i$ is the set of the black pixel in the ground-truth line $i$. $R_j$ is the set of black pixels that are found by our novel $A^*$ path-planning algorithm for the line $j$ and $G_{ij}$ is the number of shared pixels between the line $i$ in the ground-truth line and the detected line $j$ in the result.

The results of the *Saint Gall* dataset have a total of 1,410 lines, in which we obtain a pixel-level hit rate 0.985 and a line accuracy of 0.976.

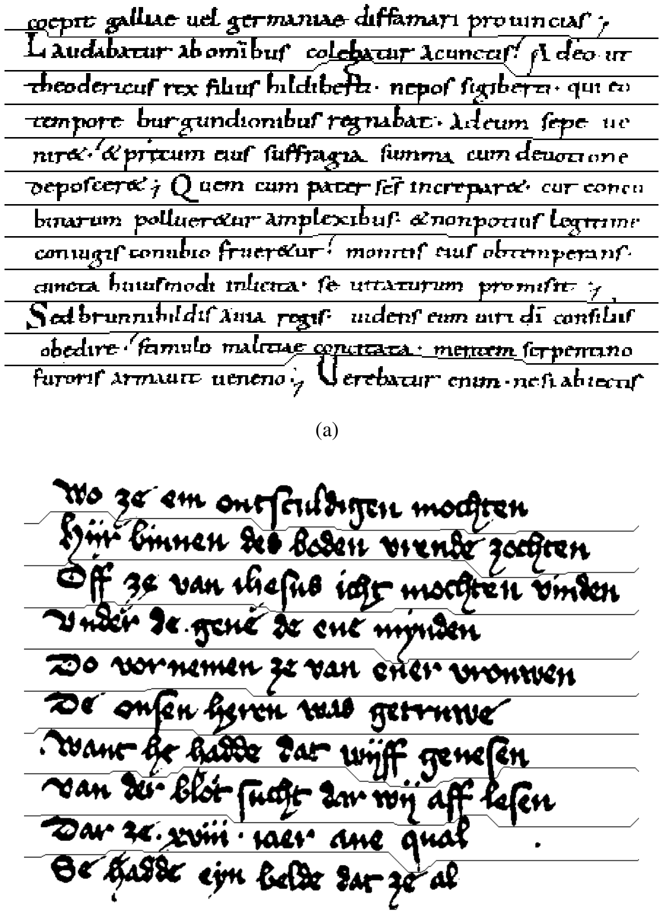
## V. Conclusion

In this paper a novel $A^*$ path-planning algorithm is proposed which uses the minimum and distance cost function to control the line segmentation agent. The $A^*$ algorithm is computed on the binarized image by using the smoothed horizontal ink density histogram. The starting state is determined. The optimal path of the line segmentation is generated using the $A^*$ path-planning algorithm. The novel cost function consists of the minimum distance and the neighbor cost function by using 5-directional movement, the agent cannot move backwards. The novel $A^*$ path-planning algorithm is successful in dividing the overlapping text lines. The cost function prevents the agent from moving far away from the initial y-value and allows the agent to move over the overlapping text lines.

In future work, layout analysis [2], [19] will be used to handle the document image before applying the line segmentation technique. Object detection will be used to detect a graphic in the handwritten image. We plan to use an energy function [6] from as the extra energy map to improve our $A^*$ algorithm. Then, the $A^*$ path-planning algorithm will be applied a handwritten Thai dataset.

## References

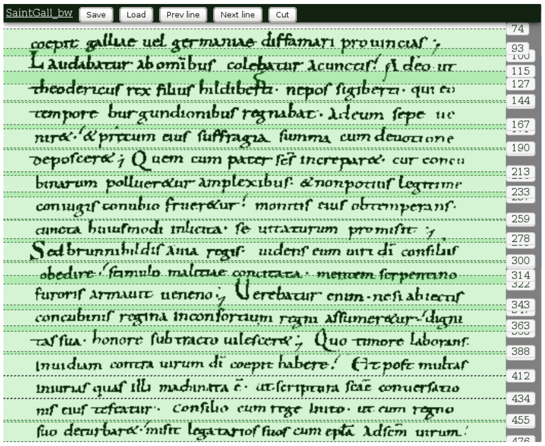
[1] M. Bulacu, A. Brink, T. Zant, and L. Schomaker, “Recognition of handwritten numerical fields in a large single-writer historical collection,” in *Document Analysis and Recognition, 2009. ICDAR ’09. 10th International Conference on*, 2009, pp. 808–812.



(a)

(b)

Fig. 9. Line segmentation results on the handwritten document from (a) the *Saint Gall* dataset and (b) the *MONK* dataset.



coepit galliae uel germaniae diffamari prouinciaſ ;
Laudabatur ab omībuſ colebatur acunctiſ! Adeo ut
theodericuſ rex filiuſ hildiberti · nepoſ ſigiberti · qui eo
tempore burgundionibuſ regnabat · adeum ſepe ue
niret · & pr[e]cum eiuſ ſuffragia ſumma cum deuotione
depoſceret ; Quem cum pater ſcſ increparet · cur concu
binarum pollueretur amplexibuſ · & nonpotiuſ legitime
coniugiſ conubio frueretur ! monitiſ eiuſ obtemperanſ ·
cuncta huiuſmodi inlicita · ſe uitaturum promiſit ;
Sedbrunnihildiſ auia regiſ · uidenſ eum uiri dī conſiliiſ
obedire · famulo malitiae concitata · mentem ſerpentino
furoriſ armauit ueneno ; Uerebatur enim · neſi abiectiſ
concubiniſ regina inconſortium regni aſſumeretur · digni
taſ ſua · honore ſubtracto uileſceret ; Quo timore laboranſ ·
inuidiam contra uirum dī coepit habere ! Etpoſt multaſ
iniuriaſ quaſ illi machinata ē · ut ſcriptura ſcāe conuerſatio
niſ eiuſ teſtatur · conſilio cum rege inito · ut eum regno
ſuo deturbaret · miſit legatarioſ ſuoſ cum eplā adſcm̄ uirum !

Fig. 10. Illustration for the ground-truth tool. The text line area is marked by the human user.

[2] M. Bulacu, R. van Koert, L. Schomaker, and T. van Der Zant, “Layout analysis of handwritten historical documents for searching the archive of the cabinet of the dutch queen,” in *Document Analysis and Recognition, 2007. ICDAR ’07. 9th International Conference on*, vol. 1, 2007, pp. 357–361.

[3] N. Nilsson, *Principles of Artificial Intelligence*, ser. Symbolic Computation / Artificial Intelligence. Springer, 1982.

[4] M. Arivazhagan, H. Srinivasan, and S. Srihari, “A statistical approach to line segmentation in handwritten documents. center of excellence for document analysis and recognition,” in proceedings of Document Recognition and Retrieval XIV, SPIE, Tech. Rep., 2007.

[5] R. Chamchong and C. C. Fung, “Text line extraction using adaptive partial projection for palm leaf manuscripts from thailand,” in *Frontiers in Handwriting Recognition, 2012. ICFHR ’12, 14th International Conference on*, 2012, pp. 588–593.

[6] A. Garz, A. Fischer, R. Sablatnig, and H. Bunke, “Binarization-free text line segmentation for historical documents based on interest point clustering,” in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, 2012, pp. 95–99.

[7] 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.

[8] O. Surinta, L. Schomaker, and M. Wiering, “A comparison of feature extraction and pixel-based methods for recognizing handwritten bangla digits,” in *Document Analysis and Recognition, 2013. ICDAR ’13. 12th International Conference on*, 2013, pp. 165–169.

[9] J. Sauvola and M. Pietikäinen, “Adaptive document image binarization,” *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000.

[10] T. R. Singh, S. Roy, O. I. Singh, T. Sinam, and K. M. Singh, “A new local adaptive thresholding technique in binarization,” *International Journal of Computer Science Issues*, vol. 8, 2011.

[11] F. Shafait, D. Keysers, and T. M. Breuel, “Efficient implementation of local adaptive thresholding techniques using integral images,” in *Document Recognition and Retrieval. DRR ’08.*, 2008, p. 681510.

[12] A. Fischer, V. Frinken, A. Fornés, and H. Bunke, “Transcription alignment of latin manuscripts using hidden markov models,” in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing. HIP ’11.*, 2011, pp. 29–36.

[13] E. Badekas and N. Papamarkos, “Optimal combination of document binarization techniques using a self-organizing map neural network,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 11–24, 2007.

[14] R. Ghosh, D. Bhattacharyya, T.-h. Kim, and G.-s. Lee, “New algorithm for skewing detection of handwritten bangla words,” in *Signal Processing, Image Processing and Pattern Recognition*, ser. Communications in Computer and Information Science, T.-h. Kim, H. Adeli, C. Ramos, and B.-H. Kang, Eds. Springer Berlin Heidelberg, 2011, vol. 260, pp. 153–159.

[15] A. Stentz, “Optimal and efficient path planning for partially-known environments,” in *Robotics and Automation, 1994. ICRA 1994. IEEE International Conference on*, 1994, pp. 3310–3317 vol.4.

[16] I. Rekleitis, J.-L. Bedwani, E. Dupuis, and P. Allard, “Path planning for planetary exploration,” in *Computer and Robot Vision, 2008. CRV ’08. Canadian Conference on*, 2008, pp. 61–68.

[17] D. Ferguson, M. Likhachev, and A. Stentz, “A guide to heuristic-based path planning,” in *in Proceedings of The Workshop on Planning under Uncertainty for Autonomous Systems at The International Conference on Automated Planning and Scheduling (ICAPS)*, 2005.

[18] Y. Li, Y. Zheng, D. Doermann, S. Jaeger, and Y. Li, “Script-independent text line segmentation in freestyle handwritten documents,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 8, pp. 1313–1329, 2008.

[19] M. Baechler, M. Liwicki, and R. Ingold, “Text line extraction using DMLP classifiers for historical manuscripts,” in *Document Analysis and Recognition, 2013. ICDAR ’13. 12th International Conference on*, 2013, pp. 1029–1033.