Towards Preserving the Ephemeral: Texture-Based Background Modelling for Capturing Back-of-the-Napkin Notes

Melissa Cote and Alexandra Branzan Albu
University of Victoria, Victoria, BC, Canada
{mcote,aalbu}@uvic.ca

Abstract

A back-of-the-napkin idea is typically created on the spur of the moment and captured via a few hand-sketched notes on whatever material is available, which often happens to be an actual paper napkin. This paper explores the preservation of such back-of-the-napkin ideas. Hand-sketched notes, reflecting those flashes of inspiration, are not limited to text; they can also include drawings and graphics. Napkin backgrounds typically exhibit diverse textural and colour motifs/patterns that may have high visual saliency from a low-level vision standpoint. We thus frame the extraction of hand-sketched notes as a background modelling and removal task. We propose a novel document background model based on texture mixtures constructed from the document itself via texture synthesis, which allows us to identify background pixels and extract hand-sketched data as foreground elements. Experiments on a novel napkin image dataset yield excellent results and showcase the robustness of our method with respect to the napkin contents. A texture-based background modelling approach, such as ours, is generic enough to cope with any type of hand-sketched notes.

1. Introduction

The next generation of business productivity tools aims at supporting creative processes underlying innovation. While these productivity tools generally focus on the workplace, such as digital whiteboards in conference rooms, one cannot underestimate the power of the back-of-the-napkin concept [1]. A good idea can strike at any moment. It often takes a writing surface to capture those flashes of inspiration and further develop them into real solutions. A back-of-the-napkin idea is typically created on the spur of the moment and captured via hand-sketched notes on whatever material is available [1]. This concept refers to how an idea may be born during dinner conversations and preserved, while still fresh in mind, by sketching on the back of a paper napkin [1].

This paper focuses on the preservation of such back-of-the-napkin ideas written on perishable media via camera-captured document image analysis. Given a camera-captured image of an actual paper napkin containing hand-sketched notes, we aim to automatically extract these notes in order to facilitate further processing steps such as optical character recognition (OCR), handwriting recognition, document vectorization, or content-based information retrieval and querying. The term "hand-sketched" is more adequate than "hand-written" (which generally refers to text) since these notes can include anything from text to drawings and graphics.

Document analysis from camera-captured images typically addresses issues such as low resolution, blur, perspective distortions, sensor noise, and uneven lighting causing shading [2]. In the context of back-of-the-napkin ideas, we are also faced with potential creases, relief in the material from the pen pressure, as well as complex and colourful backgrounds that can have high visual saliency from a low-level vision standpoint. Due to the large variability of hand-sketched notes (text, drawings, graphics) and to the diversity of napkin backgrounds in terms of colours and patterns, we frame the problem of hand-sketched notes extraction as document background modelling and removal.

1.1. Related works

To the best of our knowledge, there are no other papers that address a problem similar to ours. The literature is also sparse when it comes to background modelling in the context of documents. However, works on binarization of camera-captured document images share a similar goal, which is to convert a colour or gray-level image into a bi-level one, effectively splitting the document contents into foreground (text/ink) and background. Text detection (or more specifically hand-written text detection) is a separate topic, as hand-sketched notes here are not limited to text.

While document binarization is trivial in the case of digitally-born documents with a white background, it remains an open challenge for images of physical documents with more complex backgrounds, degradation due to moisture, discoloration in the paper, ink bleed-through or show-through, or various artefacts from the digitization process. These challenges have been showcased in a series of biennial contests since 2009 for the

Document Image Binarization COmpetition (DIBCO) [3] and 2010 for the Handwritten Document Image Binarization COmpetition (H-DIBCO) [4].

Binarization methods are generally classified into global, adaptive (local), and hybrid. Global approaches such as the classic Otsu's method [5], which use a single threshold value for the entire document image, are computationally inexpensive but tend to work poorly on camera-captured document images due to the various artifacts from the acquisition process. On the other hand, adaptive approaches such as the classic Sauvola method [6], which compute threshold values for each pixel using local neighbourhood information, are computationally more expensive and tend to perform better, but can be sensitive to the selection of their free parameter values [7]. In [8], Howe presented an automatic technique for setting the free parameters in a manner that tunes them to the individual image, which is still utilized frequently as part of proposed approaches in recent DIBCO contests.

In the specific context of camera-captured document image binarization, Kiragu and Mwangi [9] used the single scale retinex algorithm (SSR) to enhance images prior to applying Otsu's global thresholding. Bukhari et al. [7] proposed an adaptive binarization approach based on the Sauvola method that utilizes different sets of free parameters values for pixels that belong to roughly estimated foreground regions and to background regions. Foreground regions are estimated from multi-oriented multi-scale anisotropic Gaussian smoothing and a ridge detection technique. Afzal et al. [10], addressing specifically the issue of blurred images, proposed an adaptive approach based on percentile filters. Zhao et al.'s adaptive approach [11] uses multi-level multi-scale local statistical information to binarize from coarse to fine and is based on variance and clarity data. Kim [12] proposed a hybrid approach via a multiple window scheme, in which local thresholds are determined from global trends and local details by applying multiple windows with their size tuned to the size and thickness of text characters. Also using a hybrid approach, Chou et al. [13] proposed to split camera-captured document images into regions and learn what simple binarization operation, based on Otsu's thresholding, is best suited for each region type. Focusing on performance assessment of camera-captured document binarization techniques, Lins et al. [14] found that the best binarization algorithm depends on the device and the camera setup.

Enlarging the scope of this literature review to include works that are not necessarily focused on camera-captured document images leads us to consider the binarization of degraded documents, whis is a very active research topic (e.g. [15-18]). Such documents are somewhat similar to napkin documents in terms of their complex and highly variable backgrounds. However, document degradation is typically studied on historic artifacts, which are quite different from the data considered in our project. Deep learning-based methods, mostly convolutional neural networks (CNNs), have recently started to permeate document binarization. Deep learning approaches proposed for document binarization typically fall in this application area of historical document restoration (e.g. [19-23]).

1.2. Contributions

Binarization methods tend to underperform when applied to documents with salient backgrounds. Also, text detection methods typically cannot cope with varied sketched notes (text, drawings, graphics). In light of those remarks, our paper makes the following contributions: 1) from a theoretical viewpoint, we utilize texture synthesis for image segmentation purposes and propose a document background model based on texture mixtures extracted from the document itself; 2) from a practical viewpoint, we apply this document background model to the challenging problem of extracting hand-sketched notes in the context of preserving back-of-the-napkin ideas; 3) for transparency purposes and as a service to the research community, we share our annotated dataset of camera-captured images of napkins with hand-sketched notes. A strength of our approach is its training-free nature: the background model is constructed "on the fly" from the napkin itself. This constitutes a sizable advantage which contributes to make our approach readily applicable to various motifs/patterns. Also, our solution of background modelling and removal applies to all documents with varied hand-sketched notes and flat or patterned backgrounds of low or high saliency. We use napkins as a case study; another example use case (not addressed in this paper) could be the improved digitization of documents with structured and salient background patterns such as documents printed/written on security paper.

The remainder of the paper is structured as follows. Sect. 2 details our approach to napkin background modelling for hand-sketched notes extraction, Sect. 3 discusses the experimental results, and Sect. 4 presents concluding remarks and future work directions.

2. Proposed method

The goal is to separate the foreground pixels (hand-sketched notes) from the background pixels (motifs on the napkins) to extract and preserve the notes. This is accomplished by framing the problem as background modelling and removal. Our method has the advantage of not requiring any training data, and of not requiring any pre- or post-processing.

We assume that the border regions of a napkin are representative of the napkin patterns, which exhibit some (unknown) spatial periodicity. The margins of the napkin are also more likely to contain little to no hand-sketched notes compared to other regions, which simplifies the construction of the background model. However, we do not

assume that the border regions should never include notes, as people usually scribble on napkins in unconstrained ways. The proposed approach is designed to handle the occurrence of some notes in the border regions (see end of Sect. 2.2). We make no assumption as to the size/type/colour/contrast of textural patterns occurring on napkins. We extract several patches at various scales and along all four borders to increase robustness to intra-napkin variations due to background changes (e.g. different patterns along some of the sides of a napkin), shadows, local distortions, and potential foreground data.

Fig. 1 illustrates the flow chart of the method. Patches are first sampled along the border of the napkin image. Each patch is then used to grow a large textured image via texture synthesis, which is then aligned with the original napkin image and cropped to its original size to create one candidate background image. This process is repeated at various scales and allows us to create a texture mixture model. All modules of the flow chart are detailed in the following subsections, along with a complexity analysis of the proposed method. Tables 1 and 2 define all notations and parameters of the method, respectively.

Table 1: Notations of the proposed method.

Symbol	Definition
I	Original napkin image
TI	Synthesized texture image
BI	Candidate background image
TMM	Texture mixture model

Table 2: Parameters of the proposed method.

Param.	Definition	Sect.
P	Number of patches at a given scale	2.1
N	Number of scales	2.1
HS_{Pi}	Patch half size at scale i	2.1
M	Size multiple w.r.t. base scale	2.1
BS	Block size for texture quilting	2.2
O	Amount of overlap at block boundary	2.2
TH	Threshold for background removal	2.5

2.1. Patch sampling

The idea is to obtain a set of patches which represents well the background that we wish to model. Thus, we extract P square patches at uniform intervals along the borders of the napkin image I. The single-scale version, where all sampled patches have the same size, is insufficient if the motifs vary in size across napkins. This is why we propose a multi-scale version in which $N*P$ patches of different sizes are sampled, where N is the number of scales. The base patch half size HS_{P1} is set as a fraction of the napkin image size, and subsequent scales (HS_{P2}, HS_{P3}, etc.) are set as a multiple M of the base. Fig. 2a shows an example of sampled patches at different scales from a given

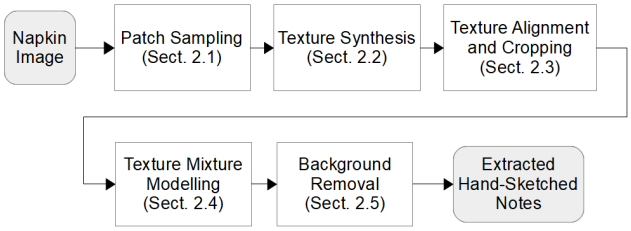

Figure 1: Flowchart of the proposed method.

napkin image, with $P = 10$, $N = 3$, $HS_{P1} = I_{size}/32$, and $M = 2^{(\text{Level-1})}$, i.e. $HS_{P2} = HS_{P1}*2^{(2-1)} = 2HS_{P1}$ and $HS_{P3} = HS_{P1}*2^{(3-1)} = 4HS_{P1}$.

2.2. Texture synthesis

From each patch obtained in Sect. 2.1, we generate a larger image (typically 1.5 times the size of I) via texture synthesis. Texture synthesis is a process in which an unlimited amount of image data is generated from a sample texture in a way that the new data will be perceived to be the same texture.

We utilize the texture synthesis method proposed by Efros and Freeman [24], called "texture quilting", which is remarkably simple and cost efficient yet works very well. Sample square blocks from the patch are quilted together to synthesize a new texture sample. In order to mitigate the "blockiness" appearance, the boundary between the "quilted" blocks is computed as a minimum cost path through the error surface at the overlap. There are two parameters: the block size (BS) and the amount of overlap (O) at the block boundary. Fig. 2b shows an example of texture image TI synthesized from a napkin patch, with $BS = 1.5*HS_{Pi}$ and $O = HS_{Pi}/6$. If any hand-sketched notes occur on a patch, they will most likely be reduced in importance in TI as they do not yield a smooth transition between blocks.

2.3. Texture alignment and cropping

Once a texture image TI is synthesized, we need to align it with the original napkin image I so that the patterns in both match each other. TI needs to be larger than I (see Sect. 2.2), as the alignment process requires some extra texture.

We use template matching with normalized cross-correlation [25] to register I with TI:

$$\gamma(u,v) = \frac{\left(\sum_{x,y}[TI(x,y)-\overline{TI}_{u,v}][I(x-u,y-v)-\bar{I}]\right)}{\left(\sum_{x,y}[TI(x,y)-\overline{TI}_{u,v}]^2 \sum_{x,y}[I(x-u,y-v)-\bar{I}]^2\right)^{1/2}}, \qquad (1)$$

where $\gamma(u,v)$ is a cross-correlation coefficient at location (u,v), $TI(x,y)$ is the pixel intensity value at coordinates (x, y) of the texture image, $I(x-u,y-v)$ is the pixel intensity value at coordinates $(x-u,y-v)$ of the napkin image, (x,y) are

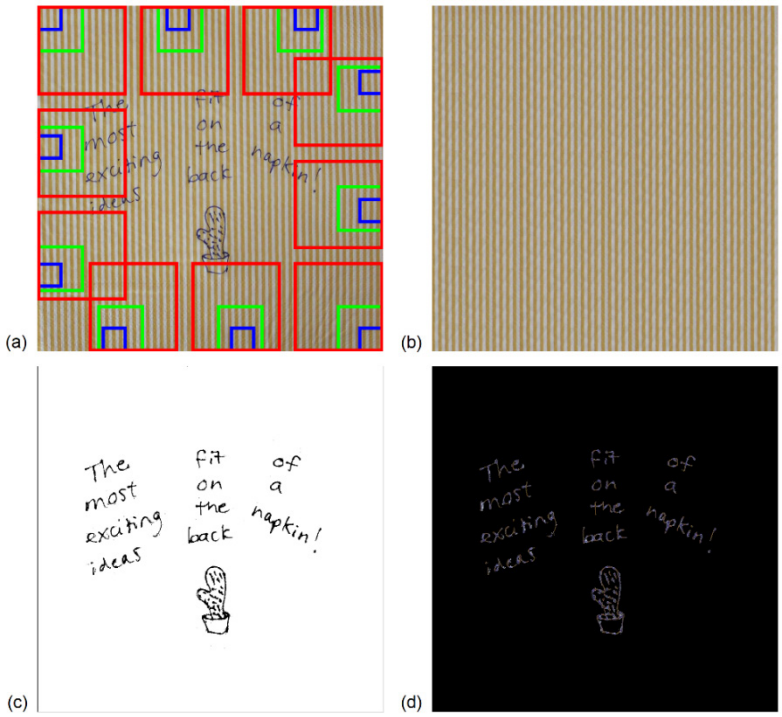

Figure 2: Example of texture-based background modelling. Sampled square patches from a napkin image with hand-sketched notes, where each colour (red, green and blue) represents a different scale (a). Synthesized texture image from one of the sampled patches (b). Background mask with pixels labelled as background in white (c) and corresponding hand-sketched note image after background removal (d).

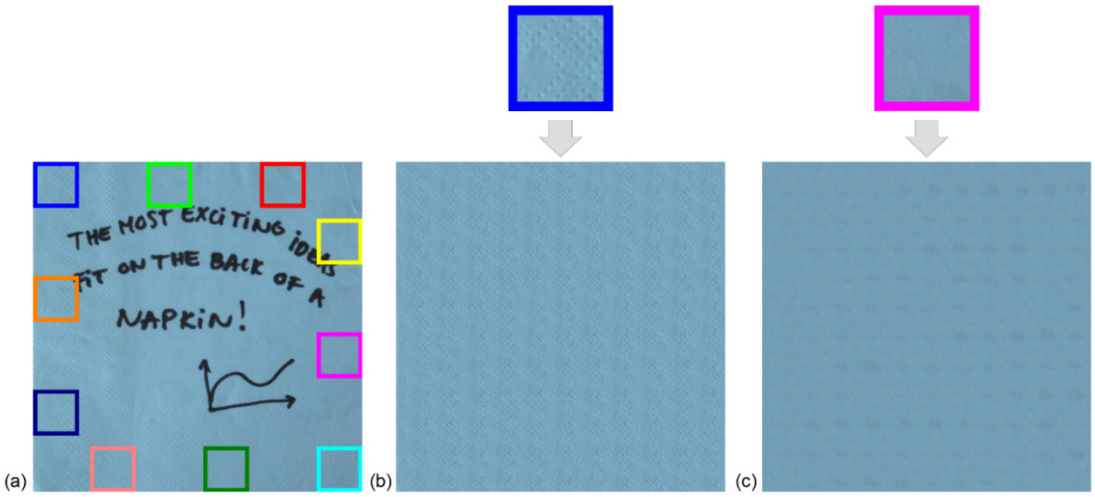

Figure 3: Examples of candidate background images within a texture mixture model obtained from two patches, displaying different etched patterns. Various sampled square patches (coloured bounding boxes) from a napkin image at a given scale (a). Two enlarged patches (with blue and magenta bounding boxes) and corresponding candidate background images (b-c).

incremented to cover all pixel coordinates of the region of the texture image where the napkin image is superimposed, $\overline{I}$ is the mean value of the napkin image, and $\overline{TI}_{u,v}$ is the mean value of the texture image in the region covered by the napkin image. We then crop TI to the size of I around the location of the best match (highest cross-correlation coefficient, i.e. (u,v) for which γ is maximal).

2.4. Texture mixture modelling

Each aligned and cropped texture image from Sect. 2.3

becomes a candidate background image *BI*. We create a texture mixture model (*TMM*) that represents the napkin background from the set of candidate background images:

$$TMM = \{BI_j \mid j = 1 \ldots N * P\} \quad (2)$$

The model gives us the set of probable background intensities at each pixel location. Fig. 3 shows several candidate background images within the same model, illustrating the importance of sampling along all borders of of the napkin to capture all possible textural variations.

2.5. Background removal

In the last phase, pixels in *I* are labelled as background if their intensity is similar to that of the *TMM*, i.e. if:

$$\exists BI_j \in TMM \mid abs\left(I(x,y) - BI_j(x,y)\right) \leq TH \quad (3)$$

for all R, G, and B channels. $I(x,y)$ is the intensity of the napkin image pixel at location (x,y) in one of the three colour channels, $BI_j(x,y)$ is the intensity of the candidate background image pixel at (x,y), and *TH* is a threshold. Equation (3) means that as long as the intensity in all three colour channels is similar to that of at least one candidate background image in *TMM*, the pixel is considered a background pixel.

The background is then removed from *I* by subtracting all pixels labelled as background. The hand-sketched notes can then be easily recovered from the remaining pixels. Figs. 2c and 2d show an example of background mask and the corresponding hand-sketched note image after background removal.

3. Experimental results

The experiments used a public implementation of the texture quilting algorithm [24] available on Matlab Central [26]. Details on the napkin dataset, compared method, quantitative and qualitative evaluations, and computational considerations are provided next.

3.1. Napkin dataset

We have created a dataset comprised of 82 napkin images containing hand-sketched notes. The images were captured with a handheld digital camera (Canon Powershot SX 600 HS) without flash, in a minimally-constrained setting (i.e. roughly perpendicular to the napkin plane and roughly centered on the napkin), then cropped around the napkin itself to remove any extraneous object. Document boundary detection and perspective rectification are separate and independent problems considered beyond the scope of this paper. Should the input images be captured at an oblique angle, a solution would be to apply a separate document rectification step informed by the napkin contour shape using a well-established method such as [27].

Due to the fragile and wrinkly nature of napkins and to the acquisition conditions, shadows, creases, and relief from the pressure of the pen are visible. There are 12 different napkin backgrounds; some have a solid colour print, while others have a checkered, lined, or dotted pattern print. All of them also include a textural pattern etched on the material over part or over the entire napkin. Six or seven pens, differing in colour and/or thickness, were used to write down notes on each napkin. We control the semantic content by using one sentence only, along with a limited set of hand drawings, which allows us to focus on several dimensions of variation, such as calligraphy styles, stroke thicknesses, ink colours, and note layouts. Each napkin image has thus one drawing (smiley, coffee cup, cactus, line chart, bar chart, or sunglasses) and the sentence "The most exciting ideas fit on the back of a napkin!" written in various styles and layouts. The images are in RGB format with 8 bits/channel; the resolution is either 868x868, 910x910, 1050x1050, or 1160x1160, depending on the original paper napkin size. Fig. 4 shows sample images from the dataset, including details of the textural patterns in the material.

Ground truth data were generated in a semi-automatic fashion with Matlab's Image Labeler app [28] using pixel labels and the flood fill and brush tools. The napkin dataset and ground truth data in the form of background masks are available online [29].

3.2. Method for comparison

As noted in Sect. 1.1, while there are no papers in the literature that address a problem similar to ours, document binarization addresses a similar goal. We thus compare our proposed method with Howe's approach to document binarization [8], a well-cited approach considered as a baseline by many who participated in the latest edition of DIBCO [3]. Howe's approach is based upon three points. 1) It defines the binarization target as the set of pixel-based labels which minimizes a global energy function inspired by a Markov random field model. 2) It is meant to be invariant to both contrast and intensity due to a data fidelity term that relies on the Laplacian of the image intensity to distinguish text/ink from background. 3) Edge discontinuities are incorporated into the smoothness term, which biases text/ink boundaries to align with edges and allows for a smoothness incentive over the rest of the image. The method introduced a stability heuristic criterion that helps to choose suitable parameter values for individual images. Its Matlab companion code is available from [8]. All comparisons reported in this paper are performed with Algorithm 3 [8].

Figure 4: Sample images from the napkin dataset. Original images (1st and 3rd rows) and zoomed-in regions showing pattern details (2nd and 4th rows)

3.3. Quantitative evaluation

Evaluation metrics found in the literature for assessing binarization methods include OCR error rate comparisons and pixel-based metrics when ground truth data are available [3]. As the handwritten notes are not limited to text and our dataset includes ground truth data, we adopt the standard pixel-based precision, recall, and F-score metrics. In all experiments, for the multi-scale approach, $P = 10$, $N = 3$, $HS_{P1} = I_{size}/32$, $M = 2^{(\text{Level-1})}$, $BS = 1.5*HS_{Pi}$, $O = HS_{Pi}/6$, and $TH = 20$ (see Table I). For the single-scale version ($N = 1$), the only difference is $HS_{P1} = I_{size}/15$.

Table 3 presents the precision and recall rates for the proposed texture-based (both multi-scale and single-scale) and for Howe's method [8], along with the F-score. The metrics are computed pixel-wise over the entire dataset, with true positives being actual background pixels predicted as such, false positives being actual foreground pixels incorrectly predicted as background pixels, and false negatives being actual background pixels incorrectly predicted as foreground pixels. The best results are shown in bold font. All methods yield a very high precision rate (> 99%); the proposed single-scale method has the highest precision and provides a small improvement over Howe's approach. While the proposed multi-scale version is slightly less precise, its recall rate is substantially superior to that of the single-scale and Howe's methods, by 4.3 p.p. and 17.4 p.p., respectively. The proposed multi-scale method thus performs significantly better in terms of not missing any of the actual background. Overall, the proposed multi-scale method outperforms both the single-scale and Howe's methods significantly, with an F-score of 0.988 vs. 0.968 and 0.892, respectively.

Table 3: Performance metric values of the proposed and compared methods over the entire dataset.

Method	Precision	Recall	F-score
Howe's [8]	0.993	0.810	0.892
Proposed (single-scale)	**0.996**	0.941	0.968
Proposed (multi-scale)	0.992	**0.984**	**0.988**

Table 4: Performance metric values of the proposed and compared methods averaged per image.

Method	Precision (μ ± σ)	Recall (μ ± σ)	F-score (μ ± σ)
Howe's [8]	0.992 ± 0.021	0.843 ± 0.260	0.884 ± 0.198
Proposed (single-scale)	**0.996 ± 0.008**	0.950 ± 0.058	0.972 ± 0.032
Proposed (multi-scale)	0.991 ± 0.011	**0.986 ± 0.013**	**0.989 ± 0.009**

μ = mean, σ = standard deviation

Table 4 presents the precision, recall, and F-score metrics averaged per image, i.e. the three performance metrics are first computed pixel-wise for each napkin image, then the values from each image are averaged over the entire dataset. This allows us to also compute a standard deviation, which is of great interest, since it informs us on how consistent the methods perform over the proposed dataset at the image level. The best results for each metric are again shown in bold font. As is the case in Table 3, the proposed multi-scale method outperforms the other methods in terms of average recall and average F-score. The standard deviation is quite high for Howe's method, especially for the recall rate and the F-score (0.260 and 0.198), while it remains quite low for both single-scale and multi-scale versions of the proposed texture-based methods (< 0.05). This means that the performance of Howe's method can change drastically from one image to the next, while the proposed method is more consistent across the entire dataset and more robust with respect to the contents of the napkin images.

3.4. Qualitative evaluation

Fig. 5 shows typical results obtained by the proposed approach and by Howe's method [8]. The 1st column illustrates a case for which Howe's method completely failed while the best results were obtained by the proposed single-scale method. Howe's method overpredicted background pixels as most of the hand-sketched notes were considered as background. One possible reason is the intensity/colour similarity between the napkin background and the notes, which did not prevent the texture-based methods from correctly identifying the background. In that particular case, the proposed multi-scale version was slightly better at identifying actual background pixels than the single-scale version. This is a general trend that can be seen in all examples of Fig. 5, confirmed by the higher recall rates in Tables 3 and 4, and can be explained by the larger and potentially more varied texture mixture model of the multi-scale version, which makes it more prone to predict pixels as background. However, the downside is that more non-background pixels were identified as background by the multi-scale version. The 2nd and 3rd columns are cases for which Howe's method locally underpredicted background pixels as many patterned regions were falsely classified as non-background pixels, while the proposed multi-scale method was significantly more successful in identifying the background. These two examples, with high contrasting patterns, are interesting cases where a background modelling approach, as proposed, is preferable to classical document binarization approaches. The last column illustrates a case for which all methods performed well, even in the presence of creases, shadows, and relief from the pen pressure.

The proposed approach does not apply any pre-processing to the napkin images nor any post-processing to the results; the pixel labelling shown in Fig. 5 is the raw output and showcases the performance of the core algorithms of the proposed approach. One could envision adding post-processing to the labelling results in order, for instance, to fill in very small gaps in the background masks or gaps that have straight shapes that are unlikely coming from hand-sketched notes, which tend to be uneven.

3.5. Computational considerations

Experiments were carried out on a PC (Intel Quad-Core i7 @ 1.8 GHz CPU with 12 GB DDR4 RAM) in Matlab R2019a. Considering the fixed parameters of Sect. 3.3, the computational load mainly depends on the size of the original napkin image as well as the number of scales (and thus the number of patches). The serial CPU runtime was found to be $6.9 \pm 0.5 \times 10^{-5}$s and $2.4 \pm 0.3 \times 10^{-4}$s per pixel on average for the single-scale and multi-scale versions, respectively. The task with the largest computational load is the texture synthesis (Sect. 2.2), which typically takes up about 60% of the execution time. Memory usage depends mainly on parameters P and N (i.e. the total number of patches), as they define the size of the *TMM*, which is $P*N$ times the size of the original napkin image. Our approach has a high potential for data parallelism on two levels: 1) each candidate texture image in the *TMM* is synthesized independently from a given napkin patch; 2) each napkin pixel is analyzed and labelled independently in the background removal phase. To take fully advantage of the spontaneous expression of back-of-the-napkin notes, a logical next step would be to implement the proposed method as a mobile app. Due to the large variability in smartphone operating systems and hardware (type of CPU, amount of storage and memory), the most feasible approach would be a cloud-based solution where the napkin image is sent for processing to a server with data parallelism capabilities. To limit data consumption by the user and to further reduce processing times, the image could easily be downsampled to at least half the current resolutions in the napkin dataset (Sect. 3.1) without impacting the end results.

4. Conclusion

This paper presents a novel approach for extracting hand-sketched notes from camera-captured images of paper napkins. We frame the problem as document background modelling and removal due to the complexity and variations of napkin motifs which may be as salient as foreground data. The background model, based on texture mixtures obtained via texture synthesis, allows us to effectively remove napkin backgrounds in order to extract hand-sketched notes such as text, drawings, and graphics. Experiments on a new napkin image dataset, made publicly available, show very promising results and showcase the robustness of our proposed method with respect to the napkin contents, compared to a baseline document binarization method. A texture-based background modelling approach, such as the proposed approach, has the main advantage of being able to cope with any type of hand-sketched notes. We believe that such an approach still has its merits in the deep learning era as it does not require any training data, the background model being constructed on the fly from the napkin itself.

Future work will focus on building a dynamic version of the proposed approach, which uses parameters that vary with respect to the local image content. This could include a dynamic patch selection based, for instance, on histogram analysis, to discard patches that are likely to contain substantial foreground data. Other future research directions also include implementing the proposed method for mobile computing, to allow for capturing back-of-the napkin notes via a phone camera in real-life situations.

Acknowledgments

This work was supported by NSERC and QuirkLogic Inc. through the Engage and CRD Grants programs.

Figure 5: Sample results on the napkin dataset for the proposed and compared methods. In the last three rows, white pixels indicate predicted background pixels, while black pixels indicate predicted non-background pixels, i.e. hand-sketched notes.

References

[1] https://www.wisegeek.com/what-is-a-back-of-the-napkin-idea.htm

[2] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: a survey," Int. J. Doc. Anal. Recog., vol. 7, no. 2-3, 2005, pp. 84–104.

[3] I. Pratikakis, K. Zagoris, G. Barlas, and B. Gatos, "ICDAR2017 competition on document image binarization (DIBCO 2017)," Proc. Int. Conf. Doc. Anal. Recog. (ICDAR), IEEE, 2017, pp. 1395–1403.

[4] I. Pratikakis, K. Zagoris, G Barlas, and B. Gatos, "ICFHR2016 handwritten document image binarization contest (H-DIBCO 2016)," Proc. Int. Conf. Front. Handwrit. Recog. (ICFHR), IEEE, 2016, pp. 619–623.

[5] N. Otsu, "A threshold selection method from gray-level histograms," IEEE T. Syst. Man Cyb., vol. 9, no. 1, 1979, pp. 62–66.

[6] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," Pattern Recog., vol. 33, no. 2, 2000, pp. 225–236.

[7] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Adaptive binarization of unconstrained hand-held camera-captured document images," J. Univers. Comput. Sci., vol. 15, no. 18, 2009, pp. 3343–3363.

[8] N. R Howe, "Document binarization with automatic parameter tuning," Int. J. Doc. Anal. Recog., vol. 16, no. 3, 2013, pp. 247–258.

[9] H. Kiragu and E. Mwangi, "An improved enhancement of degraded binary text document images using morphological and single scale retinex operations," Proc. IET Conf. Image Proc. (IPR), IET, 2012, pp. 1–6.

[10] M. Z. Afzal, M. Krämer, SS. Bukhari, M. R. Yousefi, F. Shafait, and T. M. Breuel, "Robust binarization of stereo and monocular document images using percentile filter," Proc. Int. Workshop Camera-Based Doc. Anal. Recog. (CBDAR), Springer, 2013, pp. 139–149.

[11] J. Zhao, C. Shi, F. Jia, Y. Wang, and B. Xiao, "An effective binarization method for disturbed camera-captured document images," Proc. 16th Int. Conf. Front. Handwrit. Recog. (ICFHR), IEEE, 2018, pp. 339–344.

[12] I. J. Kim, "Multi-window binarization of camera image for document recognition," Proc. Int. Workshop Front. Handwrit. Recog. (IWFHR), IEEE, 2004, pp. 323–327.

[13] C. H. Chou, W. H. Lin, and F. Chang, "A binarization method with learning-built rules for document images produced by cameras," Pattern Recog., vol. 43, no. 4, 2010, pp. 1518–1530.

[14] R. D. Lins, R. B. Bernardino, D. M. de Jesus, and J. M. Oliveira, "Binarizing document images acquired with portable cameras," Proc. Int. Conf. Doc. Anal. Recog. (ICDAR), IEEE, vol. 6, 2017, pp. 45–50.

[15] Y. Chen, and L. Wang, "Broken and degraded document images binarization," Neurocomputing, vol. 237, 2017, pp. 272–280.

[16] D. Lu, X. Huang, and L. Sui, "Binarization of degraded document images based on contrast enhancement," Int. J. Doc. Anal. Recog., vol. 21, no. 1-2, 2018, pp. 123–135.

[17] A. Sehad, Y. Chibani, R. Hedjam, and M. Cheriet, "Gabor filter-based texture for ancient degraded document image binarization," Pattern Anal. Appl., vol. 22, no. 1, 2019, pp. 1–22.

[18] A. Sulaiman, K. Omar, and M. F. Nasrudin, "Degraded historical document binarization: a review on issues, challenges, techniques, and future directions," J. Imaging, vol. 5, no. 4, 2019, 48.

[19] J. Pastor-Pellicer, S. España-Boquera, F. Zamora-Martínez, M. Z. Afzal, and M. J. Castro-Bleda, "Insights on the use of convolutional neural networks for document image binarization," Proc. Int. Work-Conf. Artif. Neural Netw (IWANN), Springer, 2015, pp. 115–126.

[20] C. Tensmeyer, and T. Martinez, "Document image binarization with fully convolutional neural networks," Proc. Int. Conf. Doc. Anal. Recog. (ICDAR), IEEE, 2017, pp. 99–104.

[21] F. Westphal, N. Lavesson, and H. Grahn, "Document image binarization using recurrent neural networks," Proc. IAPR Int. Workshop Doc. Anal. Syst. (DAS), IEEE, 2018, pp. 263–268.

[22] Q. N. Vo, S. H. Kim, H. J. Yang, and G. Lee, "Binarization of degraded document images based on hierarchical deep supervised network," Pattern Recog., vol. 74, 2018, pp. 568–586.

[23] S. He, and L. Schomaker, L. "DeepOtsu: Document enhancement and binarization using iterative deep learning," Pattern Recog., vol. 91, 2019, pp. 379–390.

[24] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," Proc. Annu. Conf. Comput. Gr. Interact. Tech. (SIGGRAPH), ACM, 2001, pp. 341–346.

[25] J. P. Lewis, "Fast template matching," Proc. Vis. Interface, CIPPRS, 1995, pp. 15–19.

[26] https://www.mathworks.com/matlabcentral/fileexchange/35828-siggraph2002-image-quilting-texture-synthesize

[27] Y. C. Tsoi and M. S. Brown, "Geometric and shading correction for images of printed materials: a unified approach using boundary," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog. (CVPR), IEEE, 2004, pp. I–I.

[28] https://www.mathworks.com/help/vision/ref/imagelabeler-app.htm

[29] http://web.uvic.ca/~mcote/Napkins/NapkinDataset.zip