

User-Assisted Ink-Bleed Correction for Handwritten Documents

Yi Huang
Nanyang Technological University
School of Computer Engineering
Republic of Singapore
hu0005yi@ntu.edu.sg

Michael S. Brown
National University of Singapore
School of Computing
Republic of Singapore
brown@comp.nus.edu.sg

ABSTRACT

We describe a user-assisted framework for correcting ink-bleed in old handwritten documents housed at the National Archives of Singapore (NAS). Our approach departs from traditional correction techniques that strive for full automation. Fully-automated approaches make assumptions about ink-bleed characteristics that are not valid for all inputs. Furthermore, fully-automated approaches often have to set algorithmic parameters that have no meaning for the end-user. In our system, the user needs only to provide simple examples of ink-bleed, foreground ink, and background. These training examples are used to classify the remaining pixels in the document to produce a computer-generated result that is equal to or better than existing fully-automated approaches.

To offer a complete system, we also provide tools that allow any errors in the computer-generated results to be quickly “cleaned up” by the user. The initial training markup, together with the computer-generated results, and manual edits are all recorded with the final output, allowing subsequent viewers to see how a corrected document was created and to make changes or updates. While an ongoing project, our feedback from the NAS staff has been overwhelmingly positive that this user-assisted framework is a practical way to address the ink-bleed problem.

Categories and Subject Descriptors

J.2 [Computer Applications]: J.5 Arts and Humanities;
I.4 [Image Processing and Computer Vision]: I.4.4 Restoration

General Terms

Algorithm, Experimentation

Keywords

Ink-bleed, restoration, document processing, user-assisted systems

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

JCDL'08, June 16–20, 2008, Pittsburgh, Pennsylvania, USA.
Copyright 2008 ACM 978-1-59593-998-2/08/06 ...\$5.00.

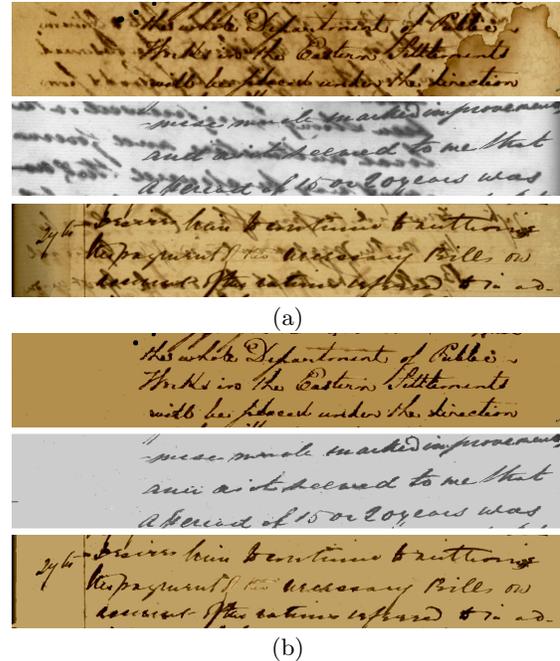


Figure 1: (a) Sections from three different documents (circa 1820-1860) from the National Archives of Singapore. (b) Our goal is to help the user significantly reduced ink-bleed interference.

1. INTRODUCTION

This paper presents work that is part of a collaborative effort with the National Archives of Singapore (NAS). The NAS houses a large collection of handwritten English language government ledgers written in the early to mid 1800s. The vast majority of these ledgers suffer from ink-bleed which occurs when ink written on the document penetrates through the paper to become visible on the opposite side (e.g Figure 1(a)). The obvious drawback of ink-bleed is the reduction in a document’s legibility.

The severity and visual characteristics of ink-bleed is related to a variety of factors, including the ink’s chemical makeup, the paper’s physical and chemical construction, the amount of ink applied (which varies spatially), the paper’s thickness, the document’s age, and the amount of humidity in the environment. This last factor is particularly troublesome for countries like Singapore where years of storage in humid conditions has exacerbated ink-bleed.

The current approach used by the NAS (and other archives) to correct ink-bleed is to place the document in a chemical wash that indiscriminately removes ink from the document. This approach takes an entire month to correct a single ledger, requiring the pages to be unbound, individual pages washed and dried, and then all pages finally rebound. Other drawbacks include the risk of damage to the primary source due to handling, potential of removing lightly written foreground-ink in addition to ink-bleed, and environmental hazards from harsh chemical solvents. Not surprisingly, a digital solution that operates on images of the documents is desired as an alternative to this chemical-based approach.

The motivation of our work is to provide a practical framework to reduce or remove ink-bleed as shown in Figure 1(b). The documents at the NAS exhibit a wide range of ink-bleed making it difficult to develop a fully automated approach that can work on all inputs. In lieu of full automation, we take a user-assisted approach that strives to reduce the amount of user involvement without sacrificing a quality output.

In particular, there are two places where the user is needed. The first is to provide simple training examples on the input image to bootstrap a classification algorithm that produces a computer-generated result that can correct the vast majority of ink-bleed. The second is to manually edit remaining ink-bleed or to correct errors made by the initial classification (e.g. foreground misclassified as ink-bleed). While manual editing might be considered more than just “user assistance”, some degree of post-processing is almost always needed. As such, our system supports this post-processing editing and provides simple visualization features to make it easy to see where manual edits are needed. In addition, our system records all processing, including the initial user markup, computer-generated results, and any manual edits. This allows subsequent viewers to verify how a restored document was created, as well as make changes to the restoration procedure. This user-assisted framework provides a practical and useful tool for reducing the effects of ink-bleed in old handwritten documents.

2. RELATED WORK

There are several existing image processing approaches targeting ink-bleed. The majority address relatively simple cases where ink-bleed intensity is significantly lighter than the foreground-ink. Such cases can be successfully handled using variations on local or global image thresholding (e.g. [1, 7, 11]).

Recent approaches have targeted more severe ink-bleed similar to that exhibited by the NAS. This includes work by Drira et al [4] that uses Principal Component Analysis (PCA) to decompose an RGB image into its PCA components and then uses iterative clustering and adaptive thresholding on these components to determine pixels that are foreground ink. Tonazzini et al [13] proposed an approach that uses blind signal separation via Independent Component Analysis to linearly decompose an RGB image into three signals assumed to be foreground, background, and ink-bleed colors. Wolf [16] extended this idea using non-linear blind separation via an Markov Random Field framework.

These previously mentioned approaches all work from a single image. A strategy to obtain more information is to use images of both the front and back side of the docu-

ment. Sharma [10] demonstrated a successful two image ‘show-through’ reduction approach for use in Xerox imaging. Show-through, however, assumes global bleeding of the front and back images. Ink-bleed exhibited in old documents is almost always spatially varying making it significantly more challenging than show-through. The most significant two image approach targeting ink-bleed are the wavelet-based approaches introduced by Tan et al [12] and Wang et al [14]. These techniques make an initial rough classification of where foreground and ink-bleed is in the image based on the magnitude of the intensity difference between coarsely aligned front and back image pixels. Iterative filtering of wavelet coefficients is then used to dampen ink-bleed while sharpening foreground pixels. While this technique produces good results, six parameters must be tuned *per example*, including two thresholds for the difference image, dampening and sharpening coefficients, the number of wavelet scale levels, and number of iterations.

Automation is the driving goal in all existing approaches; at least from an image-processing notion of automation (several algorithm parameters may need to be set). This limits these approaches to work only on examples that meet the required assumptions. Inputs that violate prior assumptions either cannot be processed or produce unsatisfactory results. Furthermore, if algorithmic parameters exist, they often have no relevance or meaning to the targeted end users (presumably archival staff or patrons). Finally, even for examples that do meet the assumptions and have had the parameters tuned correctly, achieving 100% accurate results is rare.

In our work, we take a user-assisted approach that strived to balance computer-generated results with user assistance. The idea of employing user assistance is not unique in image-based digital library applications. Two examples include Yuan and Seales [17] user-aided application to link images of manuscripts to their corresponding text transcripts and Wingenroth et al [15] related tool for transcribing imaged sheet music. In a similar vein, Dekhtyar et al [3] and Monroy et al [8] discussed approaches for building image-based editions that involve semi-automation with user assistance for establishing text-to-image linkage. In the context of restoration, work by Seales and Yun [9] used user-specified manipulation of deformable meshes derived from volumetric scans of rolled scrolls to restore the printed content. Thus, for tasks where automation is not yet achievable, semi-automation have proven a practical alternative.

3. SYSTEM OVERVIEW

This section briefly overviews our system and its overall workflow. Details to the two main components are given in Section 4 and Section 5.

A screen shot of our system’s interface is shown in Figure 2. The interface is designed as a typical image-editing GUI that allows multiple images to be opened and their views manipulated via zoom in/out, panning, etc. Additional markup features and tools pertaining to our ink-bleed correction task are provided in a control panel on the right side of the interface.

A diagram of our system’s workflow is shown in Figure 3. Our system’s input is two images of the front and back of the same document, typically $3K \times 2K$ in pixel resolution. These images are first automatically aligned using a non-linear spatial alignment approach described in Section 4.1.

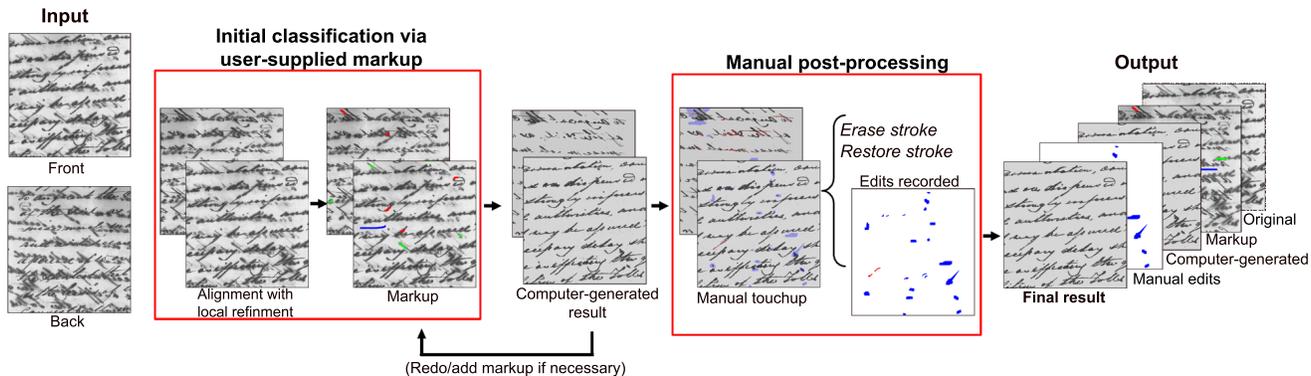


Figure 3: An overall view of our system’s components and workflow.

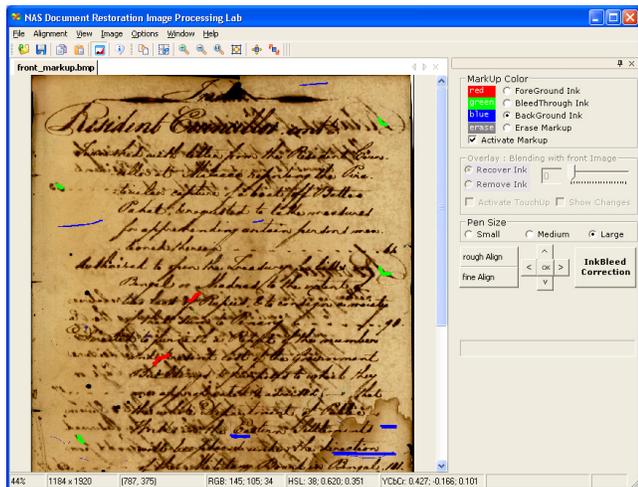


Figure 2: Our system’s interface is designed as a typical image-based GUI with additional markup tools and features to aid ink-bleed correction.

This alignment makes sure the front image pixels correspond to their appropriate locations on the back image.

After alignment, the user supplies a few examples on the image (in the form of strokes and points drawn directly on the input image) of foreground ink, ink-bleed, and background using paint-like tools. A classification algorithm then uses this user-supplied training examples to label all remaining image pixels as either foreground, ink-bleed, or background. This produces the initial computer-generated result shown in Figure 3. Pixels classified as ink-bleed and background by the computer-generated result are set to the average background color; foreground pixels are set to their original input pixel color or intensity.

While the computer-generated result reduces the majority of ink-bleed interference, two errors can occur: 1) ink-bleed or background pixels are incorrectly classified as foreground leaving noise in the image; 2) foreground pixels are misclassified resulting in broken strokes. The latter is particularly problematic when dark ink-bleed overlaps with foreground strokes. The location of these errors is not always immediately clear from the result image. An opacity overlay function is provided that blends the computer-generated results

with the original input at a specified opacity/transparency level. This can be done smoothly and in real-time helping to visually reveal errors. The user can remove errors using an *erase* brush or *restore* brush. Our system records these manual edits.

Once manual editing is completed, the user outputs the final restored document. The final result is a combination of the computer-generated results and manual edits. The software saves all information necessary to produce this final result including the original image, initial user markup, computer-generated results, and manual edits. Saving this complete output allows all steps of the restoration to be verified, replayed, and edited.

4. USER-ASSISTED CLASSIFICATION

This section describes the procedure to produce the computer-generated result. This involves three steps: 1) front and back image alignment; 2) user markup; 3) and classification.

4.1 Front/Back Image Alignment

Our software requires the front and back images to be accurately aligned such that a pixel on the front image corresponds to the exact location on the backside of the document. The images from the NAS are obtained either from scans of existing microfilm (resulting in a grayscale image) or from flatbed scans of the primary source (resulting in an RGB image). While the front and back images are carefully imaged, they are rarely accurately aligned. For gross misalignments (i.e. 50 pixels or more) the user can use “positioning widgets” on our GUI to manually make a coarse alignment.

Due to the documents’ age small 3D surface variations on the physical document make it difficult to align with global adjustments only (i.e. translation, scale, or rotation). These surface variations are further compounded by the ledgers binding which can result in noticeable warping near the page edges.

To provide an accurate pixel-level alignment, our system performs a local alignment procedure as follows. The back image is mirrored horizontally. The front image is divided into 60×60 pixels windows. Normalized cross correlation (NCC) (see [6]) is performed between the front image windows and their corresponding back image windows over a $[-10, 10]$ pixel range in the horizontal and vertical direction. NCC returns a similarity score with absolute value between

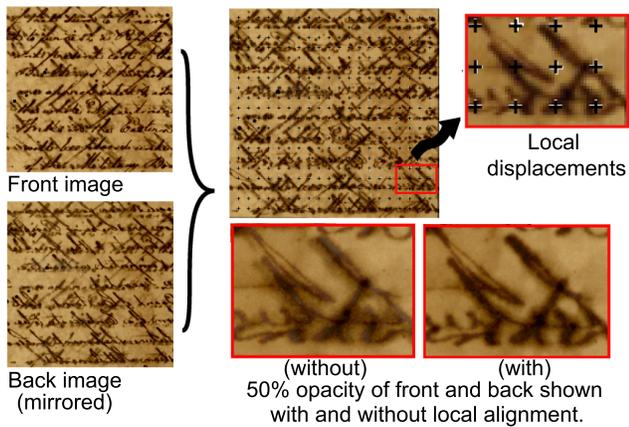


Figure 4: Local alignment procedure applied on 60×60 blocks. Non-linear warping of the image is applied to align the pixels. Zoomed aligned front and back regions are displayed simultaneously with 50% opacity, clearly showing the improvement after local refinement.

0 and 1, where 1 means most similar. This NCC score is computed for each pixel location in this search window.

The maximum NCC score is taken to be the local displacement. If the maximum score is below 0.1, it is assumed there is no displacement between the front and the back image. Once the set of window matches are found, thin-plate-spline (TPS) interpolation [2] is used to smoothly warp the back page into pixel-wise alignment with the front image. The front and back image’s width and height may need to be slightly adjusted to accommodate this alignment. Figure 4 shows an example where black crosses represent the front 60×60 window’s centers and white cross show the local displacement on the backside. Figure 4(right) shows that visible ghosting (i.e. blurring) due to misalignments between the front and back image is apparent when local refinement is not used; this ghosting is corrected via the local-refinement procedure.

4.2 User-markup and Classification

4.2.1 Markup

Due to the diversity in the characteristics of the ink-bleed exhibited by the NAS collection, training-data needs to be obtained for each input image pair. This is done by having the user draw color-coded strokes (and points) on the the front and back images, thereby labeling examples of foreground (red), background (blue), and ink-bleed (green). For each pixel, p , along these strokes, 2D features defined as $\rho_p = [C_p, C_{p'}]$ are collected, where C_p and $C_{p'}$ are the intensities of the front image pixel p and its corresponding back image pixel p' respectively. The back image feature, $\rho_{p'}$, is taken to be the reversed order of the intensities in the feature. For RGB color scans, we first convert them to intensity (grayscale) images. We have tried using the full RGB color values as feature components, but found little difference in the results.

Figure 5(left) shows an example of features being collected. These user-supplied training samples are plotted as points in a 2D grid (Figure 5(right)) where the x-axis repre-

sents the back image intensity and the y-axis represents the front image intensity.

4.2.2 Classification

Using the training samples we perform classification using a K-Nearest Neighbor (KNN) classifier which is described briefly here (for more details see [5]). The user-supplied training features can be divided into three sets: $\{\rho_i^{\mathcal{F}}\}_{i=1}^L$, $\{\rho_j^{\mathcal{I}}\}_{j=1}^M$ and $\{\rho_k^{\mathcal{B}}\}_{k=1}^N$, which represent foreground, ink-bleed and background respectively. L , M and N are the size of these sets. Each element ρ in the sets is a 2D vector of front and back pixel intensity as previously described.

To classify a pixel in the front image, we extract its feature vector ρ_p and compute the Euclidean distances (L2-norm) between ρ_p and all the $L + M + N$ training vectors and select the top-K closest vectors where K is set as $\sqrt{L + M + N}$. The top-K vectors are denoted as $\{\rho_m\}_{m=1}^K$ and are further divided into three index sets $\pi^{\mathcal{F}}, \pi^{\mathcal{I}}$ and $\pi^{\mathcal{B}}$ according to their labels. The pixel p is labeled (classified) by the majority class of $\{\pi^{\mathcal{F}}, \pi^{\mathcal{I}}, \pi^{\mathcal{B}}\}$ expressed as:

$$l_p = \arg \max_{\theta} \|\pi^{\theta}\|, \quad (1)$$

where θ is a label in $\{\mathcal{F}, \mathcal{I}, \mathcal{B}\}$, and $\|\pi^{\theta}\|$ is the number of elements in set π^{θ} .

While effective, KNN is computational costly due to its need to compute distances. Since there are only 256×256 possible feature vector combinations, we can pre-classify every possible feature vector as belonging to either \mathcal{F}, \mathcal{I} , or \mathcal{B} , resulting in a 256×256 lookup table, or decision table. This decision table is shown overlaid on top of the training-samples in Figure 5(right). Image pixels can now be quickly classified by simple lookup into the decision table based on their feature vectors ρ_p . Note that we have described how to label the pixels in the front image. The back image labeling will be performed in the same manner with the feature vectors reversed accordingly.

One advantage of using KNN is that we do not need to make assumptions about the intensity profile of the ink-bleed (e.g. ink-bleed is lighter than the foreground). All ink-bleed characteristics are embedded in the user-supplied markup. This allows our approach to work well for a variety of inputs.

Figure 6 shows a comparison of our KNN approach against two automated approaches: Drira et al [4] adaptive thresholding that operates on a single image, and Tan et al [12] two image wavelet-based approach. The adaptive thresh-

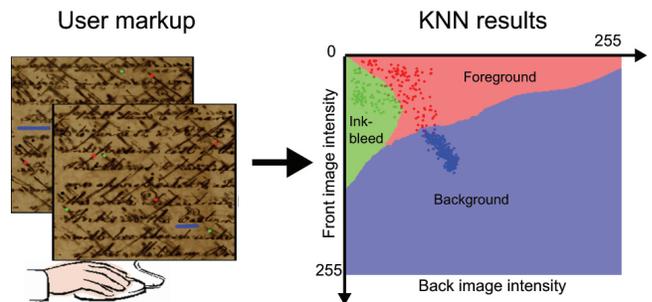


Figure 5: (Left) User markup on the input image. (Right) 2D plot of the training-samples and the resulting decision table based on these training-samples.

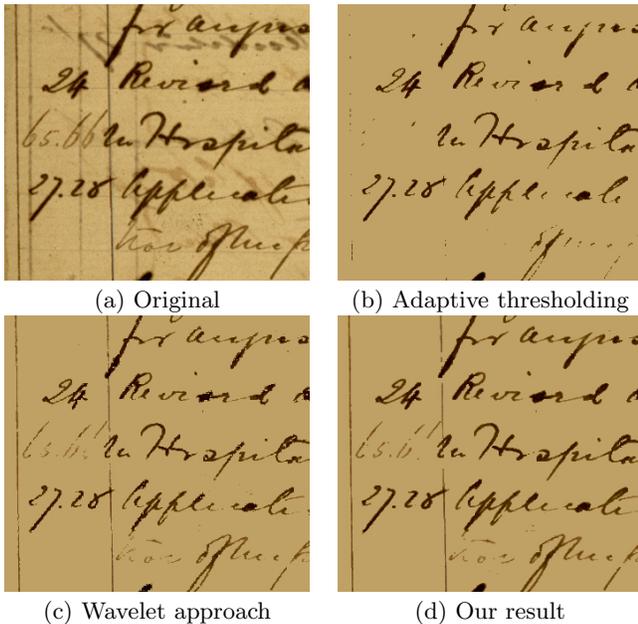


Figure 6: Comparisons: (a) portion of original image; (b) adaptive thresholding [4]; (c) wavelet-based approach [12]; (d) our result. While (c) provides similar results, it requires six parameters to be set. Our approach requires user-assistance in the form of markup, but has no parameters.

olding approach makes the assumption that the ink-bleed is lighter in intensity than foreground ink resulting in lightly written foreground ink being removed (Figure 6(b)). Tan et al [12] two image wavelet-based approach is more robust and produces results similar to ours, but has six user-tuned parameters (Figure 6(c)). Our result (Figure 6(d)), on the other hand, is generated without parameters. In additional experiments on 20 similar full page images (10 front/back pairs) our approach outperformed the adaptive thresholding and produced either better or similar results to the wavelet-based approach.

4.3 Viewing the computer-generated results

While the computer-generated result removes the majority of the ink-bleed, it is not always 100% accurate. The user therefore needs to identify where problems occur. One way to aid in this task is to blend the original input image with the computer-generated result at various opacity levels. Figure 7 gives an example with opacities levels 0%, 20%, 40%, 60%, 80% and 100%. Such blending is performed in real-time using a slider-tool on the GUI’s panel. This blending allows the user to quickly see where ink-bleed is still considered foreground, or more seriously, where a foreground stroke has been removed.

Poor results from the computer-generated step is typically because the ink-bleed is severe. Sometimes, however, it is attributed to poor markup. The user can easily modify the initial markup and try the automated procedure again.

5. MANUAL POST-PROCESSING

While the aim of the computer-generated approach is to accurately correct as much of the input as possible, errors

are unavoidable. The opacity blending feature helps the user identify erroneous regions. To correct errors, we provides simple *erase* and *restore* tools that work like a brush in a painting application. *Erase* allows the user to remove any pixels classified as foreground, resulting in these pixels being labeled as background (i.e. erased from the foreground) in the final result. Erasing is targeted at ink-bleed or background that was erroneously classified. Conversely, the *restore* brush allows any pixels that were not classified as foreground to be restored back to foreground.

Manual edits are recorded by generating an image that captures all *erase* and *restore* strokes where red represents *restore* and blue represents *erase*. This record provides an easy way for a subsequent viewer to see what portions of the final output were computer-generated, and what parts are from manual edits. Our software provides a “highlight” view, where the manual edits are shown slightly colored (like a highlight pen) to make the edits more apparent.

Figure 8 shows a complete example of an RGB scan made from an original document. Only the front page is shown. The first row shows the computer-generated result; the second row shows opacity blending; the third row shows user edits. While there appears to be a great deal of manual edits, these represent strokes made with the brush interface. Often when there is noise in the background, a single large sweeping stroke will be made to erase them. The fourth row shows a zoomed region to give a better idea of the results of the manual edits. In many cases the document is readable with the computer-generated results only, but this readability is further improved via the manual editing.

An another representative result is shown in Figure 9. This is input is from grayscale microfilm. Again we can see from the zoomed region that most of the manual cleanup is minor, however it does visually help to produce a noise-free final image void of broken strokes and specks.

6. DISCUSSION AND CONCLUSION

We have presented a user-assisted framework to help reduce ink-bleed interference that is commonly present in old documents. While full automation is undoubtedly the ultimate goal, given the difficulty of developing a one-size-fits-all automated approach and the immediacy of the problem, we opted for a design with the “user in the loop”. Our framework also provides simple visualization features to aid inevitable manual editing. Manual edits are recorded to make them distinguishable from the computer-generated results thereby allowing viewers to see when and where human-judgement was used.

The current usage scenario of our software is intended for use on an “as needed” basis. The NAS does not have an active program to digitize and process the imaged volumes solely for the purpose of ink-bleed reduction. Instead, our tool is used when a patron needs access to a few ledger pages, or when an NAS staff is looking for information inside a ledger. As such, full automation to restore the entire collection is currently not a pressing matter, however, the ability to cleanup individual pages reasonably quickly is quite desirable.

Our current work only discusses one classification technique (KNN) using simple intensity features. Different features and classifiers (e.g. Support Vector Machines) can be used instead to potentially produce better results. In addition, incorporating spatial information in the classification



Figure 7: Blending the computer-generated results with the original input to help identify errors. Percentage are how much of the original input image is shown with the result.

process, as well as higher-level semantic information such as stroke direction or character characteristic, is part of on-going efforts. The overall framework presented in this paper, however, will not change.

While a formal usability study has not been performed, feedback from our NAS collaborators has been extremely positive. The wavelet-based [12] approach discussed in Section 2 and demonstrated in Figure 6 was an earlier attempt to address ink-bleed on the same collection from the NAS. This approach generates comparable computer-generated results to our own but was not designed in a user-centric fashion. The significant draw back of this approach is that it requires six algorithmic parameters to be tuned per examples. These parameters that are specific to the algorithm itself, requiring the user to understand some notion of wavelet decomposition and image restoration. Such knowledge outside the scope of the NAS staff. In fact, the NAS staff themselves do not use this approach but relied on computer science researchers to produce results for them. In addition, for researchers who do understand the meaning of the algorithms parameters, tuning can be tedious, often requiring 10-20 minutes in a trial and error fashion until a satisfactory result is obtained. After this initial tuning, images of documents with similar ink-bleed can be tuned in 1-2 minutes. Our design on the other hand is user-centric in nature and its usage of simple markup of background, ink-bleed, and foreground examples on the input image is easily understood by the NAS staff. In addition, markup generally takes less than one minute and achieves comparable (and often better) results than the wavelet-based approach.

The amount of manual cleanup needed is also difficult to quantify at this time as it is specific to the image input. In general, small noise (e.g. Figure 9) is really not a problem, and its clean up in this paper is performed more for demonstration purposes than real need. The most serious problem is if a stroke is removed by the computer generated result. Such deleted strokes must be restored, and while the restoration of the strokes themselves is easy to perform with the *restore* brush tool, the ability for the user to quickly identify missing strokes remains challenging. Techniques to expedite error identification are part of on-going efforts. Currently, we use the opacity blending mechanism to aid in this procedure.

Interestingly, this opacity blending feature was one of the features most appreciated by the NAS staff. This was not only because it helps validate the computer-generated results, but surprisingly because the staff are accustomed to seeing some level of ink-bleed in the images. One staff member even commented she felt more comfortable reading a partially opaque result than a fully corrected result because it

looked more familiar. Another positive feedback was our ability to operate on grayscale microfilm documents. Microfilm is a preferred mechanism of access for archival materials as it avoids unnecessary handling of the primary source. Processing microfilm makes the need for the local-alignment procedure described in Section 4.1 even more relevant since imaging cannot be repeated if the documents were not completely flat during the initial imaging for microfilm. Thus, in summary, we believe our user-assisted framework provides a practical solution for archives and libraries to address the ink-bleed problem.

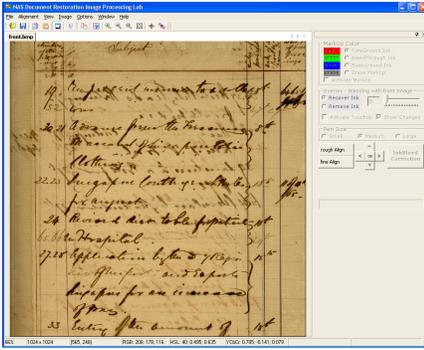
Acknowledgements

We gratefully acknowledge the support of our colleagues from the National Archives of Singapore. We thank Mr. Poon-Wei Koot and his efforts in programming our software's user-interface. This work was supported by A*STAR SERC Grant No: 0521010104.

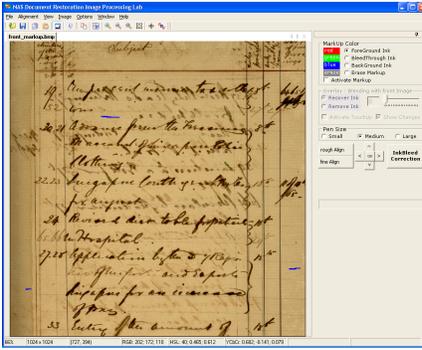
7. REFERENCES

- [1] J. Bescos. Image processing algorithms for readability enhancement of old manuscripts. *Electronic Imaging*, 1:392-397, 1989.
- [2] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 11(6):567-585, June 1989.
- [3] A. Dekhtyar, I. E. Iacob, J. Jaromczyk, K. Kiernan, N. Moore, and . C. Porter. Building image-based electronic editions using the edition production technology. In *ACM/IEEE Joint Conference on Digital Libraries*, 2005.
- [4] F. Drira, F. L. Bourgeois, and H. Emptoz. Restoring ink bleed-through degraded document images using a recursive unsupervised classification technique. In *Document Analysis Systems (DAS)*, 2006.
- [5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.
- [6] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 2nd edition, 2001.
- [7] F. C. Mintzer et al. Toward on-line, worldwide access to vatican library materials. *IBM Journal of Research and Development*, 40(2):139-162, March 1996.
- [8] C. Monroy, R. Furuta, and G. Stringer. Digital donne: workflow, editing tools, and the reader.s interface of a collection of 17th-century english poetry. In *ACM/IEEE Joint Conference on Digital Libraries*, 2007.

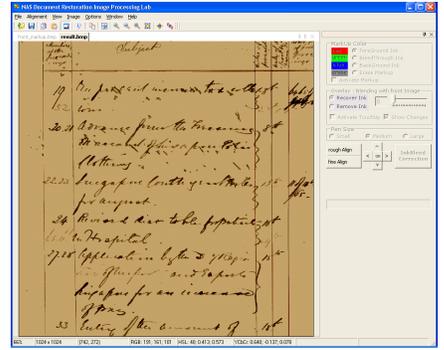
Computer-generated Results



original

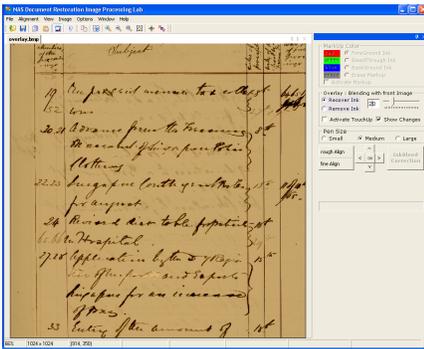


user-markup

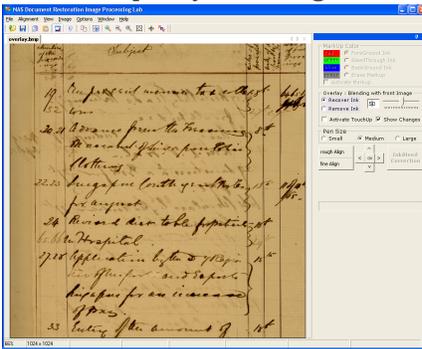


result

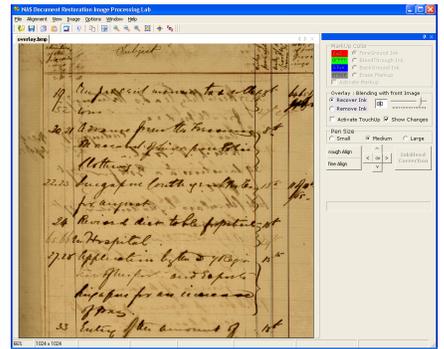
Opacity Blending



20%

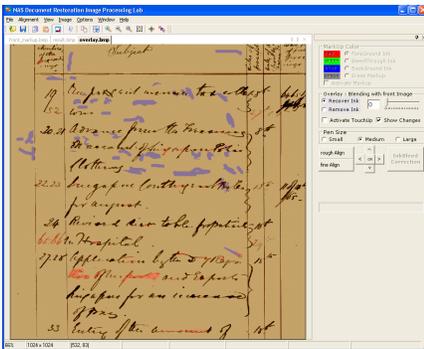


50%

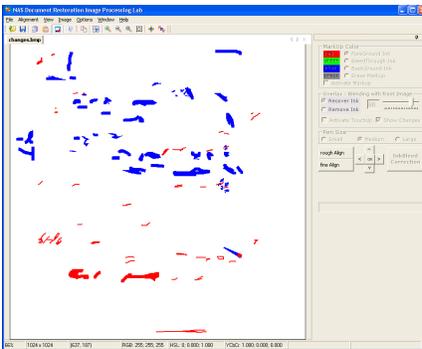


80%

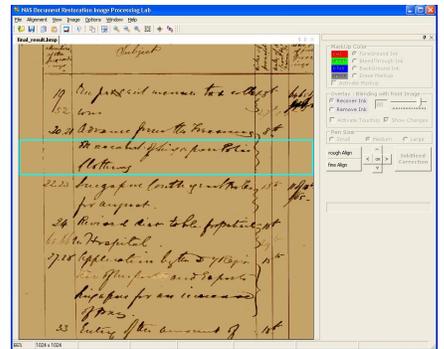
Manual Edits



edits



record



final result

Zoomed region



original



computer-generated



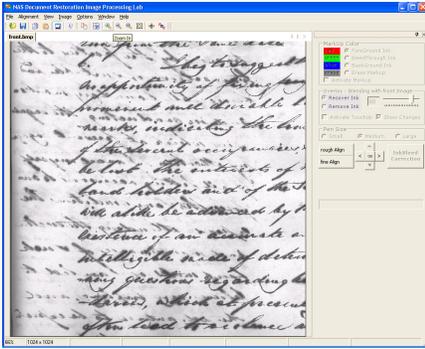
manual edits



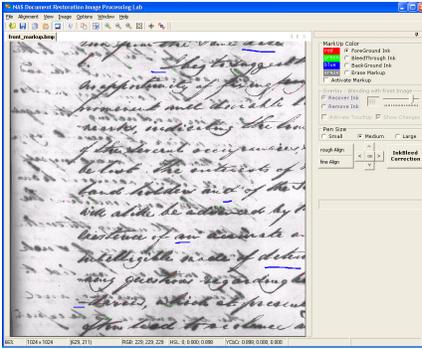
final result

Figure 8: Full Example I: the original input; user-markup; computer-generated results; opacity blending with 20%, 50%, and 80%; manual edits shown using the highlight feature; image recording manual edits; final output. A zoomed region (highlighted on the final output) is also shown.

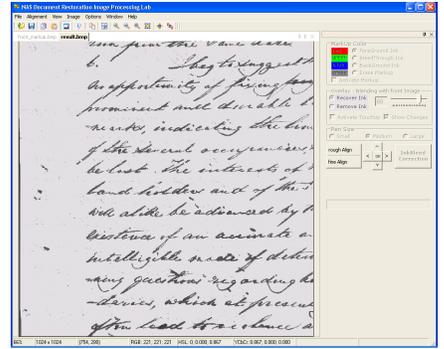
Computer-generated Results



original

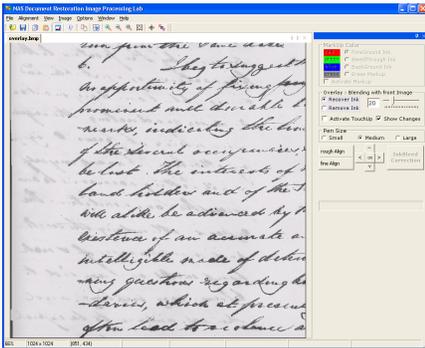


user-markup

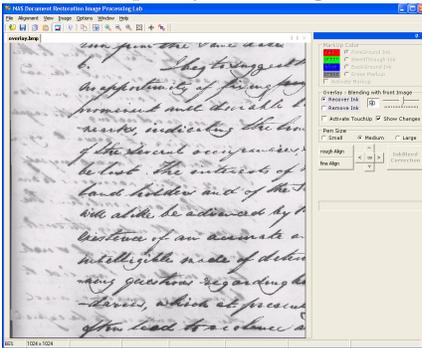


result

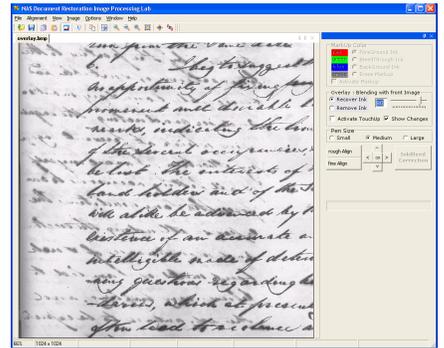
Opacity Blending



20%

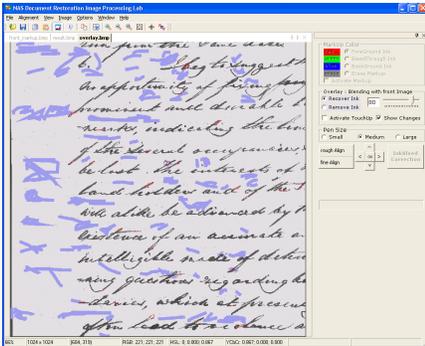


50%

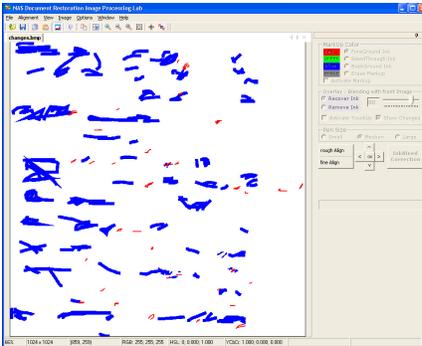


80%

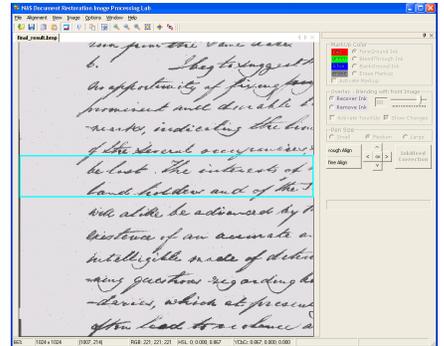
Manual Edits



edits



record

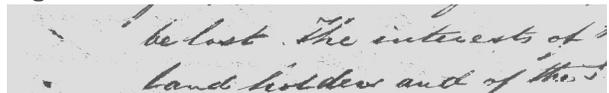


final result

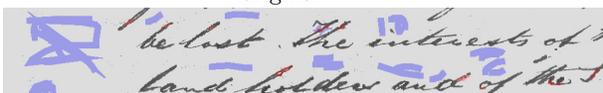
Zoomed region



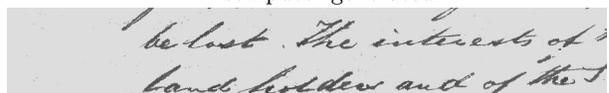
original



computer-generated



manual edits



final result

Figure 9: Full Example II: the original input; user-markup; computer-generated results; opacity blending with 20%, 50%, and 80%; manual edits shown using the highlight feature; image recording manual edits; final output. A zoomed region (highlighted on the final output) is also shown.

- [9] W. B. Seales and Y. Lin. Digital restoration using volumetric scanning. In *ACM/IEEE Joint Conference on Digital Libraries*, 2004.
- [10] G. Sharma. Show-through cancellation in scans of duplex printed documents. *IEEE Trans. on Image Processing*, 10(5):736–754, 2001.
- [11] Z. Shi and V. Govindaraju. Historical document image enhancement using background light intensity normalization. In *International Conference on Pattern Recognition*, 2004.
- [12] C. L. Tan, R. Cao, and P. Shen. Restoration of archival documents using a wavelet technique. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(10):1399–1404, Oct 2002.
- [13] A. Tonazzini, L. Bedini, and E. Salerno. Independent component analysis for document restoration. *International Journal on Document Analysis and Recognition*, 7:17–27, 2004.
- [14] Q. Wang, T. Xia, L. Li, and C. Tan. Document image enhancement using directional wavelet. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.
- [15] B. Wingenroth, M. Patton, and T. DiLauro. Enhancing access to the levy sheet music collection. In *ACM/IEEE Joint Conference on Digital Libraries*, 2002.
- [16] C. Wolf. Document ink bleed-through removal with two hidden markov random fields and a single observation field. In *Technical Report RR-LIRIS-2006-019*, 2006/2007.
- [17] C. J. Yuan and W. B. Seales. Guided linking: Efficiently making image-to-transcript correspondence. In *ACM/IEEE Joint Conference on Digital Libraries*, 2001.