

Text/Image Segmenter for DjVuLibre

CHUANHAN QIU
cq254@nyu.edu

SHIRLEY ZHAO
shirley.zhao@cims.nyu.edu

Professor Yann LeCun
G22.2565: Machine Learning
Courant Institute of Mathematical Sciences
New York University

Fall 2010

Abstract

DjVuLibre is the open source version of the DjVu system. It currently lacks a good segmenter that can separate the text and drawings from the backgrounds and continuous-tone images in a scanned document. We implemented the algorithms laid out in Bottou, et al. [1998] with some minor changes. Hierarchical K -means clustering is implemented, and the final code is written in Ruby syntax with the option for inline C code. The program runs smoothly for PPM and PGM files. Completion times vary between the two types of files, but the entire process can be sped up using inline C code. At the time of this writing, the project is almost complete. The only part missing is the output of a sepfile, which requires implementing RLE compression. Hence, we plan to open source the project and to continue working on it even after this project has been submitted.

1 Introduction

DjVuLibre is the open source version of the DjVu system. It currently lacks a good segmenter that can separate the text and drawings from the backgrounds and continuous-tone images in a scanned document. The objective of our project is to build a foreground/background segmenter by using various clustering methods, which are based on the same methods used for DjVu [Bottou et al., 1998]. Ideally, the program will produce a “sepfile” that can be fed to the command `csepdjvu` which takes the sepfile and compresses it into a composite DjVu file. A sepfile consists of a foreground image, a background image, and a bitonal mask image in a simple run-length-encoded format (RLE).

2 Methodology

We are following the same algorithms laid out for DjVu in Bottou et al. [1998] but with some minor changes. Since DjVuLibre is the open source version of DjVu, the algorithms will function the same way. We began with writing the code in Lush and then translated and continued the process in Ruby, with the option for inline C code (see **Implementation and Results** for details).

First, the foreground must be separated from the background. Greyscale images are structured differently from color images. Each pixel in a greyscale image takes on only one value between 0 and 255. Each pixel in a color image is coded in the form of a vector (R, G, B) and each element of the vector has a value between 0 and 255. We initialized k -means clustering for $k = 2$. Initial values for a greyscale image are 0 for foreground (black) and 255 for background (white). Initial values for a color image are $(0, 0, 0)$ for foreground and $(255, 255, 255)$ for background.

Distance between each pixel of the document and the initial values were calculated. For greyscale images, we used $\|I - x\|$, where I is the initial value of the foreground or background and x is a sampled pixel in the document. For color images, we used a modified distance measure: $\|(R_i, G_i, B_i) - (x, y, z)\| = |R_i + G_i + B_i - x - y - z|$, where (R_i, G_i, B_i) is

the either the foreground or background pixel value and (x, y, z) is the sampled pixel.¹ In both cases, the smaller of the two values determined the labeling of that pixel as foreground or background. Then the initial values for foreground and background were updated and the algorithm was repeated until convergence.

However, this algorithm gave only two values to segment the entire document. To get a better result, we should first divide the document into small blocks of 8 by 8 pixels and then run k -means on each block. This gave us a better end result, but as laid out in the paper [Bottou et al., 1998], there were several problems with this method.

Hence, we next used a grid of heirarchical blocks. The objective is to run k -means on the largest ones first and using the values for foreground and background obtained, we can run k -means on the smaller blocks inside the larger blocks. This is repeated on each block size, going from coarse block granularity to fine block granularity, until convergence.

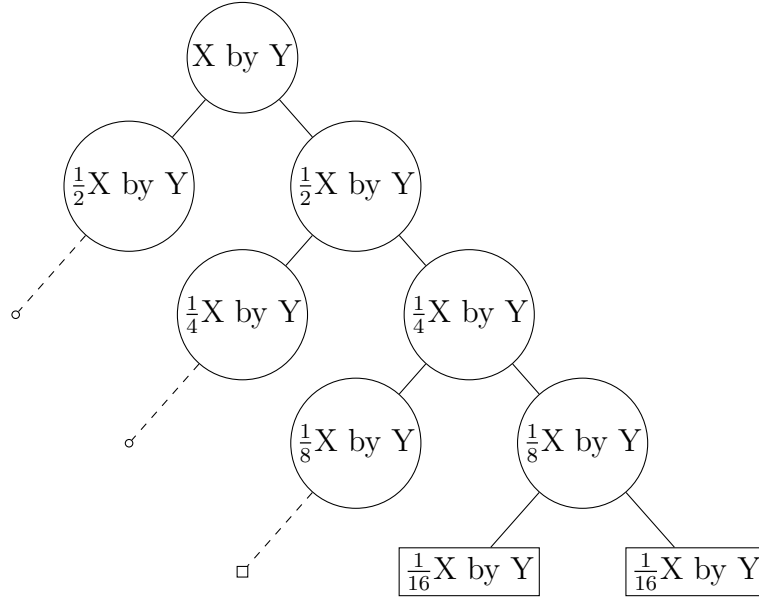


Figure 1: We begin by dividing the document vertically. This tree illustrates only division of one node on each level. The leaves are boxed.

¹The distance measure used by Bottou, et al. [1998] is computed using a line in three dimensional space, where the endpoints are the foreground and background vectors. Distance is measured between the projection of a pixel onto the line and the endpoints. We chose this particular measure because it is simpler and does the job.

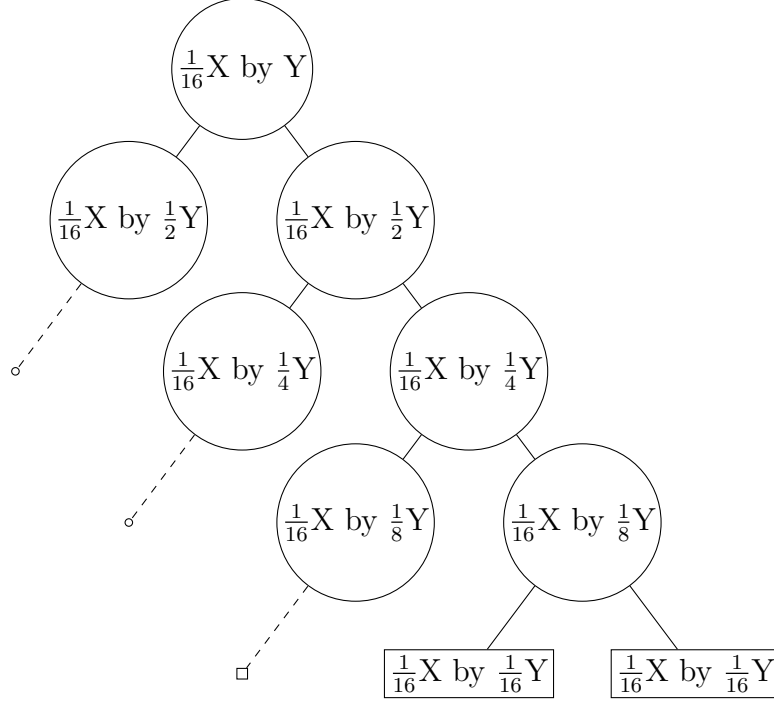


Figure 2: After dividing the document vertically, we divide horizontally each of the leaves so that we end up with a grid of 16-by-16 blocks.

To achieve this, we constructed a tree with the whole document as the root and the smaller blocks as the nodes/leaves. The goal is to have a 16-by-16 grid of blocks. **Figures 1 and 2** illustrate this idea using a document size of X -by- Y pixels. The document is first divided in half vertically and then each segment is also divided in half vertically. We get 16 nodes from this process. Then each of the segments in these nodes are divided horizontally the same way until there are 16 nodes from this process as well. In the end, we have a total of 256 blocks. K -means is run on every three levels of the tree. The parent values are passed down the the child nodes. At the end, every block has its own values for foreground and background, which are used to reconstruct the document.

3 Implementation and Results

Ruby must first be installed; if you wish to use the version with inline C code, the Ruby gem called `RubyInline` must also be installed (type into terminal `gem install RubyInline`). To run the program, simply type `ruby main.rb file_name` into a terminal window; i.e., `ruby main.rb baboon.pgm`. Please note that the pure Ruby color segmentation is not very efficient so please be patient while it runs.

All of the working files can be found on SourceForge at <https://ruby-djvu-seg.svn.sourceforge.net/svnroot/ruby-djvu-seg/>. The entire code is written in Ruby syntax. File inputs must have either the PPM or PGM extensions. **Figures 3 through 8** show

the original, the foreground image, the background image, and the reconstructed image. The masks for each figure are not shown. On the server `energon1.cims.nyu.edu`, it takes about 3 minutes to run a PGM file and 12 minutes to run a PPM file.

To increase the speed of the program, we started moving some of the heavy lifting numerical computation from Ruby to C using a Ruby gem to do inline C, which compile into `*.o` modules as needed. We have observed substantial improvements using inline C over pure Ruby implementation. Segmentation on a color PPM file that would normally take 12 minutes now only takes 3 minutes. Admittedly 3 minutes is still too long, but as we are moving more and more Ruby code into C, we should have no problem segmenting color images in the range of 30 seconds to 1 minute.



(a) original

(b) foreground



(c) background

(d) reconstruction

Figure 3: Example of reconstruction on a color picture

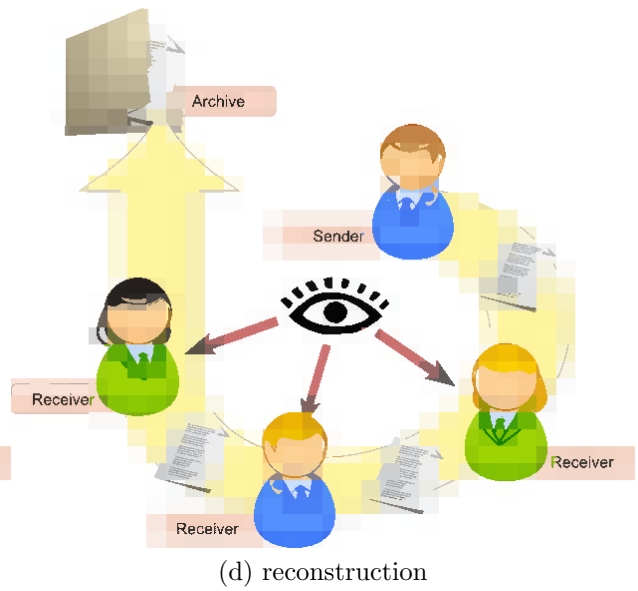
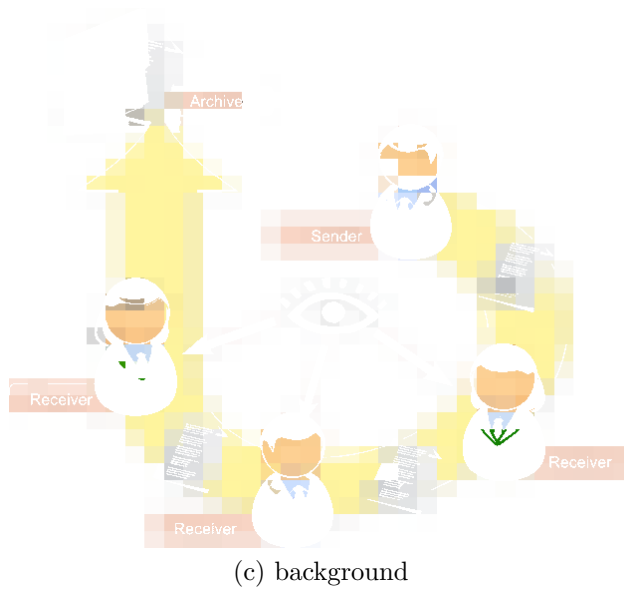
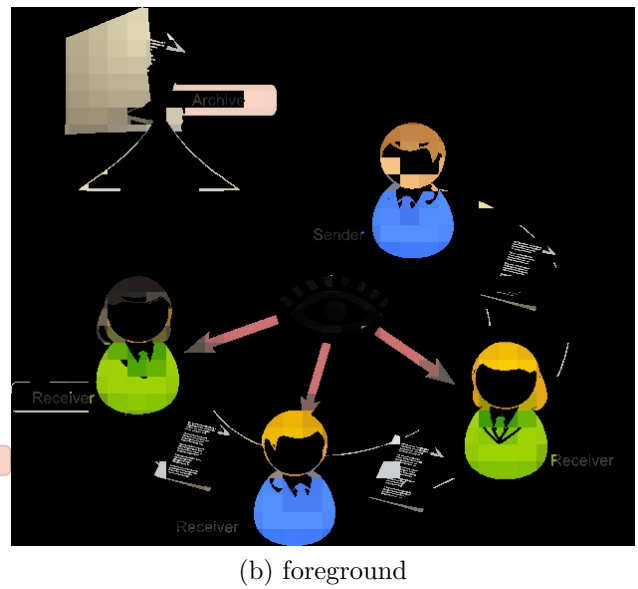
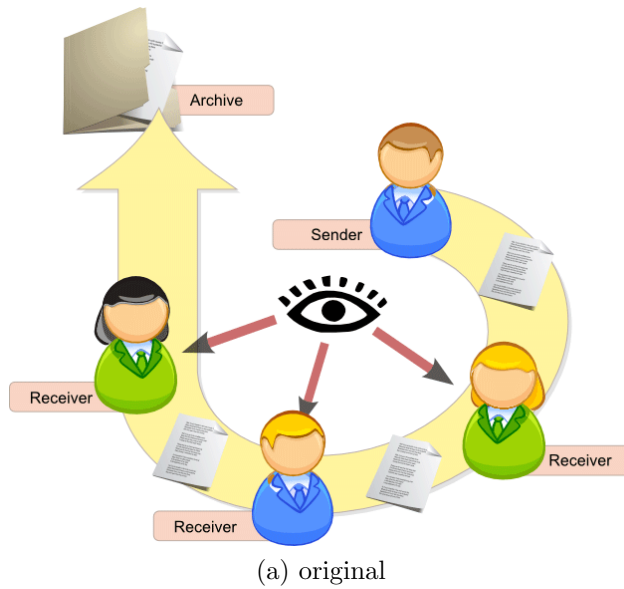


Figure 4: Example of reconstruction on a document with both text and color images



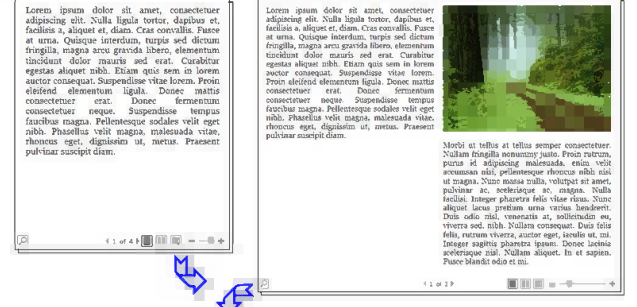
(a) original



(b) foreground



(c) background



(d) reconstruction

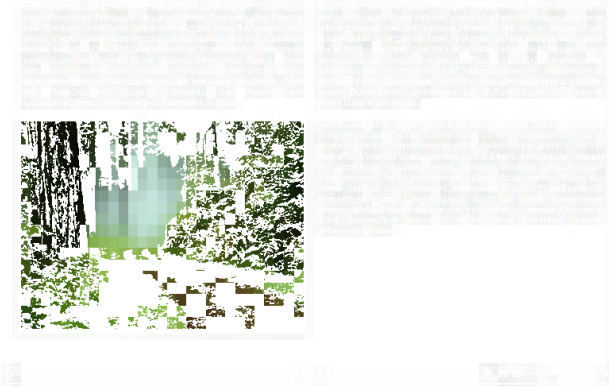


Figure 5: Another example of reconstruction on a document with both text and color images

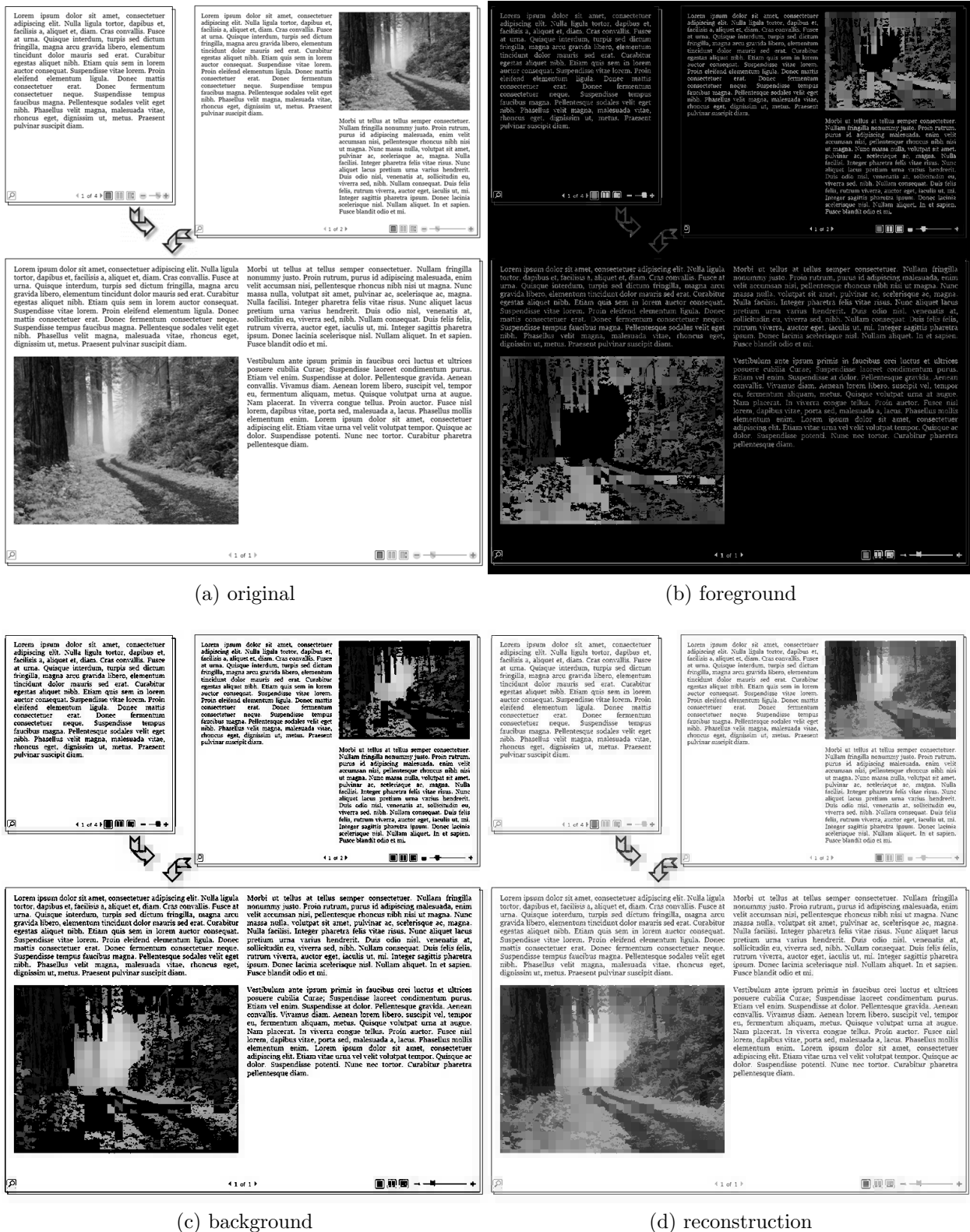
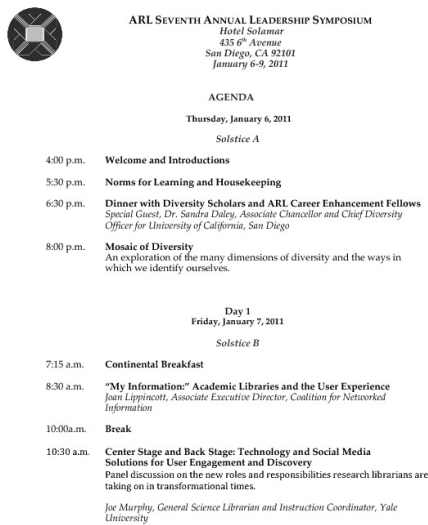


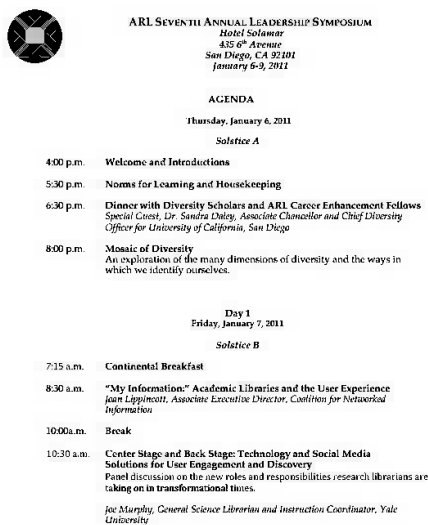
Figure 6: The same document as the previous figure but in greyscale



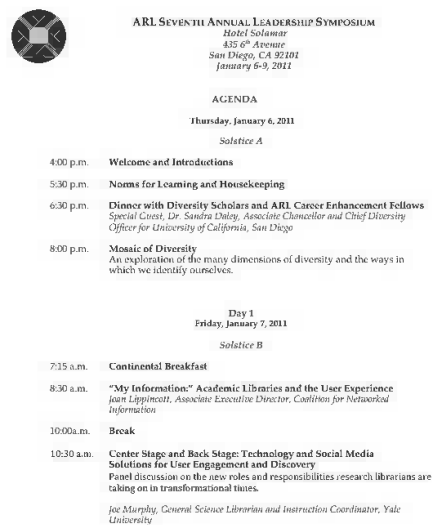
(a) original



(b) foreground




(c) background



(d) reconstruction

Figure 7: Another example of reconstruction on a greyscale document



ARL SEVENTH ANNUAL LEADERSHIP SYMPOSIUM
 Hotel Solamar
 435 6th Avenue
 San Diego, CA 92101
 January 6-9, 2011

AGENDA

Thursday, January 6, 2011
Solstice A

4:00 p.m. Welcome and Introductions
 5:30 p.m. Norms for Learning and Housekeeping
 6:30 p.m. Dinner with Diversity Scholars and ARL Career Enhancement Fellows
Special Guest, Dr. Sandra Doley, Associate Chancellor and Chief Diversity Officer for University of California, San Diego
 8:00 p.m. Mosaic of Diversity
 An exploration of the many dimensions of diversity and the ways in which we identify ourselves.

Day 1
 Friday, January 7, 2011
Solstice B

7:15 a.m. Continental Breakfast
 8:30 a.m. "My Information:" Academic Libraries and the User Experience
Juan Lippincott, Associate Executive Director, Coalition for Networked Information
 10:00a.m. Break
 10:30 a.m. Center Stage and Back Stage: Technology and Social Media Solutions for User Engagement and Discovery
 Panel discussion on the new roles and responsibilities research librarians are taking on in transformational times.
Joe Murphy, General Science Librarian and Instruction Coordinator, Yale University

(a) original



ARL SEVENTH ANNUAL LEADERSHIP SYMPOSIUM
 Hotel Solamar
 435 6th Avenue
 San Diego, CA 92101
 January 6-9, 2011

AGENDA


Thursday, January 6, 2011
Solstice A

4:00 p.m. Welcome and Introductions
 5:30 p.m. Norms for Learning and Housekeeping
 6:30 p.m. Dinner with Diversity Scholars and ARL Career Enhancement Fellows
Special Guest, Dr. Sandra Doley, Associate Chancellor and Chief Diversity Officer for University of California, San Diego
 8:00 p.m. Mosaic of Diversity
 An exploration of the many dimensions of diversity and the ways in which we identify ourselves.

Day 1
 Friday, January 7, 2011
Solstice B

7:15 a.m. Continental Breakfast
 8:30 a.m. "My Information:" Academic Libraries and the User Experience
Juan Lippincott, Associate Executive Director, Coalition for Networked Information
 10:00a.m. Break
 10:30 a.m. Center Stage and Back Stage: Technology and Social Media Solutions for User Engagement and Discovery
 Panel discussion on the new roles and responsibilities research librarians are taking on in transformational times.
Joe Murphy, General Science Librarian and Instruction Coordinator, Yale University

(b) foreground



ARL SEVENTH ANNUAL LEADERSHIP SYMPOSIUM
 Hotel Solamar
 435 6th Avenue
 San Diego, CA 92101
 January 6-9, 2011

AGENDA

Thursday, January 6, 2011
Solstice A

4:00 p.m. Welcome and Introductions
 5:30 p.m. Norms for Learning and Housekeeping
 6:30 p.m. Dinner with Diversity Scholars and ARL Career Enhancement Fellows
Special Guest, Dr. Sandra Doley, Associate Chancellor and Chief Diversity Officer for University of California, San Diego
 8:00 p.m. Mosaic of Diversity
 An exploration of the many dimensions of diversity and the ways in which we identify ourselves.

Day 1
 Friday, January 7, 2011
Solstice B

7:15 a.m. Continental Breakfast
 8:30 a.m. "My Information:" Academic Libraries and the User Experience
Juan Lippincott, Associate Executive Director, Coalition for Networked Information
 10:00a.m. Break
 10:30 a.m. Center Stage and Back Stage: Technology and Social Media Solutions for User Engagement and Discovery
 Panel discussion on the new roles and responsibilities research librarians are taking on in transformational times.
Joe Murphy, General Science Librarian and Instruction Coordinator, Yale University

(c) background



ARL SEVENTH ANNUAL LEADERSHIP SYMPOSIUM
 Hotel Solamar
 435 6th Avenue
 San Diego, CA 92101
 January 6-9, 2011

AGENDA

Thursday, January 6, 2011
Solstice A

4:00 p.m. Welcome and Introductions
 5:30 p.m. Norms for Learning and Housekeeping
 6:30 p.m. Dinner with Diversity Scholars and ARL Career Enhancement Fellows
Special Guest, Dr. Sandra Doley, Associate Chancellor and Chief Diversity Officer for University of California, San Diego
 8:00 p.m. Mosaic of Diversity
 An exploration of the many dimensions of diversity and the ways in which we identify ourselves.

Day 1
 Friday, January 7, 2011
Solstice B

7:15 a.m. Continental Breakfast
 8:30 a.m. "My Information:" Academic Libraries and the User Experience
Juan Lippincott, Associate Executive Director, Coalition for Networked Information
 10:00a.m. Break
 10:30 a.m. Center Stage and Back Stage: Technology and Social Media Solutions for User Engagement and Discovery
 Panel discussion on the new roles and responsibilities research librarians are taking on in transformational times.
Joe Murphy, General Science Librarian and Instruction Coordinator, Yale University

(d) reconstruction

Figure 8: The same example as the figure above but with color text and logo

4 Limitations

Our program is supposed to output a sepfile, which is a concatenation of an RLE-encoded foreground and a PPM background file. Unfortunately due to time constraints, we have not properly implemented RLE compression from a PGM/PPM file. However, the mechanics of concatenation for the foreground and background files are built in, and all that is left to complete is the RLE compression. The current output of our program is four separate files: a background image, a foreground image, a mask image, and a reconstructed image.

5 Conclusion

We implemented the algorithms laid out in Bottou, et al. [1998] with some minor changes. The final code is written in Ruby syntax with the option for inline C. The program runs smoothly for PPM and PGM files. Completion times vary between the two types of files, but the entire process can be sped up using inline C code. At the time of this writing, the project is almost complete. The only part missing is the output of a sepfile, which requires implementing RLE compression. We plan to open source the project and to continue working on it even after this project has been submitted.

Acknowledgements

The authors would like to thank Professor Yann LeCun for teaching this course and for his guidance in this project; and Daniel Galron (Teaching Assistant) for his knowledge and support to help us understand the algorithm better.

Reference

L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. LeCun. (1998). High Quality Document Image Compression with DjVu. *Journal of Electronic Imaging*, 7:410-425.