Information Leakage Through Document Redaction: Attacks and Countermeasures*

Daniel Lopresti[a] and A. Lawrence Spitz[b]

[a]Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015, USA.
[b]DocRec Ltd, 34 Strathaven Place, Atawhai, Nelson, New Zealand.

ABSTRACT

It has been recently demonstrated, in dramatic fashion, that sensitive information thought to be obliterated through the process of redaction can be successfully recovered via a combination of manual effort, document image analysis, and natural language processing techniques. In this paper, we examine what might be revealed through redaction, exploring how known methods might be employed to detect vestigial artifacts of the pre-redacted text. We discuss the process of redaction and circumstances under which sensitive information might leak, present an outline for experimental analyses of various approaches that could be used to recover redacted material, and describe a series of increasingly stringent countermeasures to address, and in some cases eliminate, the perceived threat.

Keywords: Security, redaction, declassification, document analysis, character shape coding.

1. INTRODUCTION

It has been recently demonstrated, in dramatic fashion, that sensitive information thought to be obliterated through the process of redaction can be successfully recovered via a combination of manual effort, document image analysis, and natural language processing (NLP) techniques. As reported in international news media, computer security researchers David Naccache and Claire Whelan demonstrated that they could recover two instances of redacted text from images of previously classified U.S. intelligence memos by measuring font design attributes and matching them against a font database.[1] An image of one of the documents is shown in Figure 1, where they were able to determine that the missing word in the second case was almost certainly "Egyptian."

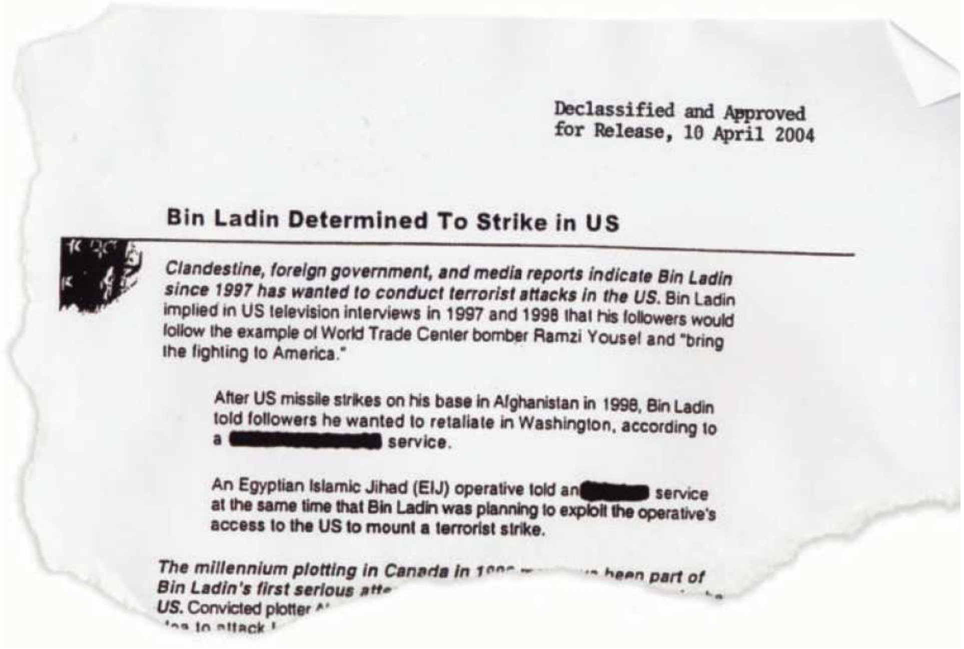
In retrospect, this achievement should come as no surprise to those versed in the field of document analysis, as efforts to recognize naturally degraded text have led to numerous research results that correspond almost exactly to some of the steps used by Naccache and Whelan. Even so, this appears to be a problem worthy of study for several reasons. Clearly, the potential exists for the unintended release of sensitive information which could be disastrous. A full and thorough understanding of what is technically possible from an adversarial standpoint can lead to better methods, and perhaps even automated or semi-automated tools, to guarantee that hidden text remains hidden. At the very least, the problem brings together aspects of document analysis research in somewhat different ways from traditional applications. Hence, it may offer the opportunity for new insights with broader implications. So far as we know, there have been no previously published papers examining this issue.†

In this paper, we examine in some detail what might be revealed through redaction, expanding on the techniques used by Naccache and Whelan to explore how known methods from document image analysis and NLP might be employed to detect vestigial artifacts of the pre-redacted text. We shall:

- Discuss the process of redaction and circumstances under which sensitive information might leak.
- Present an outline for experimental analyses of various techniques appearing in the literature that could be used to recover redacted material.

*Presented at *Document Recognition and Retrieval XII (IS&T / SPIE Electronic Imaging)*, San Jose, CA, January 2005.

†Naccache and Whelan are currently in the process of preparing a paper describing their result, which was presented informally at a European computer security conference.[2]



Declassified and Approved
for Release, 10 April 2004

Bin Ladin Determined To Strike in US

[redacted]

Clandestine, foreign government, and media reports indicate Bin Ladin since 1997 has wanted to conduct terrorist attacks in the US. Bin Ladin implied in US television interviews in 1997 and 1998 that his followers would follow the example of World Trade Center bomber Ramzi Yousel and "bring the fighting to America."

After US missile strikes on his base in Afghanistan in 1998, Bin Ladin told followers he wanted to retaliate in Washington, according to a [redacted] service.

An Egyptian Islamic Jihad (EIJ) operative told an [redacted] service at the same time that Bin Ladin was planning to exploit the operative's access to the US to mount a terrorist strike.

The millennium plotting in Canada in 1 been part of
Bin Ladin's first serious att
US. Convicted plotter
to attack

Figure 1. Part of the redacted document decoded by Naccache and Whelan.[1]

- Describe a series of increasingly stringent countermeasures to address, and in some cases eliminate, the perceived threat.

As this is very much work in progress, we are able to draw few conclusions at present and raise many more questions than we answer.

2. REDACTION

In the context of interest, redaction is defined as "the removal of exempted information from copies of a document."[3] Typically, this is accomplished by obscuring the text in question using a thick black marker pen, covering up the text with a special opaque tape, or excising a section from the document using, for example, a razor knife. In all cases, the physical redaction takes place on a copy of the original document, and a copy of the redacted document is what gets released to the public.

Beyond the above observations, which are self-evident, redaction is a "black box" process that takes place using guidelines we cannot discern directly; our access is limited to the end product, the redacted document. Hence, performing research in this area requires making certain basic (but reasonable) assumptions that are impossible for us to confirm and must be taken on faith:

1. The page in question contains a mixture of sensitive and non-sensitive material.
2. The non-sensitive material is to be released to the public.
3. The remainder of the information, when presented in the context of the original page, is confidential and must not be revealed, either directly or indirectly.
4. The goal of redaction is to prepare a facsimile of the original page for release with the sensitive material physically obliterated, while keeping as much of the non-sensitive material as possible.

Points 1 through 4 describe a set of tradeoffs that amount to what is effectively an optimization problem, the solution to which is an appropriately redacted document. Note, however, that the tools an adversary may possess must certainly be factored into the equation when deciding whether a given document is secure for release. For now, we exclude from consideration the possibility that, for political or other reasons, an entity might use poor quality redaction to release surreptitiously certain information without appearing to have done so intentionally.

3. INFORMATION LEAKAGE

If an adversary can determine with some level of confidence the identity of one or more of the redacted regions on the page, we consider it a serious breach of the process. We note that there are differing degrees of severity. In the worst case, the attacker is able to determine with high certainty precisely one interpretation for the given redaction. Under a somewhat less serious scenario, he or she might be able to narrow down the options to a small number of choices (say, two to ten). We can quantify what the attacker is able to glean in terms of orders of magnitudes: tens, hundreds, thousands of possibilities, etc. At some point, the guesses become essentially random and the document is considered to have been securely redacted.

How can information leak when text has been redacted? We can enumerate some potential ways this might happen:

1. The text has not been completely obliterated (*e.g.*, the reflective qualities of "black" may differ for laser-printer toner and a marker pen).
2. Although largely obscured, certain features of the text may remain apparent (*e.g.*, the numbers and locations of ascender and descender characters). As Spitz has shown in his work on character shape coding, this is often enough to narrow down the set of choices to a small number.[4]
3. When using a monospaced font (such as Courier), the width of the gulf between the cleartext immediately adjacent to the redacted region reveals the length, in characters, of the missing word(s). Combined with language modeling techniques, this may be sufficient information to reveal the missing text, or at least limit the possibilities.
4. For proportionally-spaced fonts, based on the locations of adjacent words and estimates of character widths (which are easy to obtain through a number of methods, including an analysis of cleartext on the page in question), the widths of missing words can be estimated and, as in the monospaced case, NLP brought to bear.

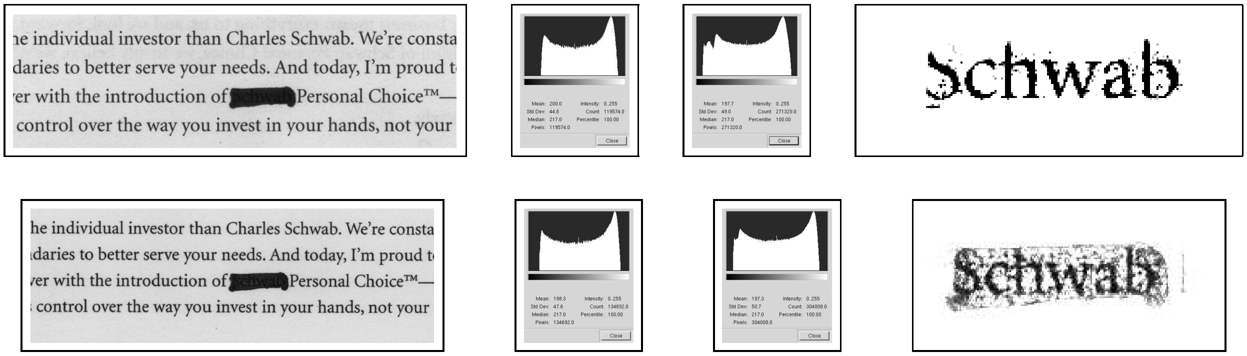
In many cases, the above attacks will not be sufficient to identify the missing text uniquely. Moreover, except perhaps for the first and second attacks listed, none will work when large regions of the page are redacted. Keep in mind, though, that the performance measure here is quite different from a traditional pattern recognition problem. In the latter case, we might demand 95% or 98% accuracy before we declare success. In a security application such as the one we are considering, however, if an adversary is able to recover redacted material even 2% of the time, that qualifies as a success for the attacker and a failure for the process.

4. TECHNIQUES USED FOR MOUNTING ATTACKS ON REDACTED TEXT

4.1. Image processing

We most often see redacted documents as photocopies of a redacted "original." Also, it is usual in the document recognition community to deal in binary (one bit per pixel) images of documents wherever possible. Processing of documents redacted using felt-tipped pens, however, seem to require that at least preliminary image processing be done in the grayscale domain.

For example, if we take the image shown in Figure 2, we can see that the recovery of the redacted word is somewhat easier in the "original" document than it is in the photocopy. Nevertheless, there is considerable useful information in the recovered redacted word even from the photocopy. From a human-readability standpoint there is almost no difference between the two resulting images.


ıe individual investor than Charles Schwab. We're consta
daries to better serve your needs. And today, I'm proud t
'er with the introduction of [redacted] Personal Choice™—
control over the way you invest in your hands, not your

he individual investor than Charles Schwab. We're consta
ıdaries to better serve your needs. And today, I'm proud t
/er with the introduction of [redacted] Personal Choice™—
; control over the way you invest in your hands, not your

Mean: 200.0 Intensity: 0..255
Std Dev: 44.8 Count: 119574.0
Median: 217.0 Percentile: 100.00
Pixels: 119574.0
Close
Mean: 197.7 Intensity: 0..255
Std Dev: 49.0 Count: 271320.0
Median: 217.0 Percentile: 100.00
Pixels: 271320.0
Close
Schwab
Mean: 198.3 Intensity: 0..255
Std Dev: 47.8 Count: 134692.0
Median: 217.0 Percentile: 100.00
Pixels: 134692.0
Close
Mean: 197.3 Intensity: 0..255
Std Dev: 50.7 Count: 304008.0
Median: 217.0 Percentile: 100.00
Pixels: 304008.0
Close
Schwab

Figure 2. A portion of a redacted page. The top row represents scanned images of the redacted original. The bottom row represents a photocopy. From left to right we have the source image, the histogram of the "background" consisting of portions of the image not including the redacted word, the histogram of the entire redacted image showing the extension of the dark portion of the histogram, and the redacted region after the histogram analysis yields threshold values.

4.2. Font metrics

The work of Naccache and Whelan, as described by Butler, employed actual font identification based on minimization of the Hamming distance between noise-free renderings of the characters found in the scanned images and the images themselves.[1] They apparently relied on high quality rendition such that the font metrics were not sufficiently distorted as to make identification impossible, and therefore selection of candidate words much more difficult.

Khoubyari and Hull have also performed predominant font identification.[5] Kopec, Zramdini and Ingold, Fang and Hull, and others have shown it is possible to perform font identification even from somewhat degraded images.[6–8] We posit here that this step may not even be necessary because we may be able to obtain character set width information from the unredacted content on the same page.

Hull, *et al.* went on to identify whole words in images artificially degraded to simulate facsimile scanning and printing.[9]

4.3. Taking advantage of artifacts

Because the obscuration of word images with a felt-tip pen is an inexact process, there may be other artifacts available to us besides word set width. We may see ascenders, descenders, or i-dots that have not been occluded in the redaction process. Noting the presence of such artifacts and, better yet, their approximate position within the word can materially reduce the candidate list. Even the absence of such artifacts, if it can be detected, is useful to an attacker.

In addition, detectability of these artifacts can improve font metric determination because of the positional synchronization they provide.

4.4. Natural language processing

If we can parse the redacted sentence (if it is, in fact, a sentence), we can reduce the number of candidate words significantly by using a part-of-speech tagged lexicon, only selecting those words that would preserve the parsability of the sentence.

Proper nouns are among the most redacted tokens. Comprehensive lexica of proper nouns are difficult to assemble and maintain, although online resources, including the growing pervasiveness of commercial search engines such as google,[10] may have the effect of making this task easier than before. Probably the best strategy for an attacker is to develop and maintain local lexica of potentially redact-prone terms.

Number strings are also often considered to represent sensitive information and may require a fundamentally different approach.

5. PRELIMINARY EXPERIMENTAL RESULTS

In this section, we describe the results of some preliminary experiments we have started to perform. We begin by noting that due to the unusual nature of the redaction problem, we are limited to using synthetic data in all of our research. We can do no better than to hypothesize about the kinds of things that might go wrong and how an attacker might try to exploit them.

Based on the work reported by Naccache and Whelan, it would appear that an important channel for information leakage is an estimate of the width of the redacted text in combination with language statistics. To study this question, we conducted a simple experiment that involved computing the widths of text strings chosen from several large lexica and determining the degree to which this one simple feature revealed information about the text itself. We conducted this test for three fonts, Times, Helvetica, and Courier, assuming a 12-point typeface and scanning at 600 dpi and using character width and typesetting information from the Adobe font metrics file.[11]

We considered four lexica:

YAWL "yet another word list" is a free, public domain list of over 264,000 English words.[12][‡]

COUNTRIES is a comprehensive list of 416 country names from around the world, including both official ("Republic of Indonesia") and informal ("Indonesia") variants.

CONGRESS is a list of the full names of the 101 Senators (including the Vice President, who presides over the Senate) and the 439 Representatives currently serving in the U.S. Congress.

NAMES is the cross-product of two lists of names provided by the U.S. Census Bureau.[13] The first is a merged list of male (1,219) and female (4,275) first names, while the second is a list of last names (88,799). By enumerating all possible $\{\text{first name}, \text{last name}\}$ pairs, a total of 487,861,706 names are generated. Many of these obviously do not correspond to real people because the current U.S. population is only 294,562,027 as of this writing.[14] Still, we can be certain that our exhaustive enumeration covers roughly 90% of the U.S. population since that is the lower bound on the coverage for each of the individual lists we used.

In each case, we computed the average number of text strings of a given width when typeset in the font in question (see Table 1). This can be taken as a measure of the overall uniqueness of information contained in redacted regions under various scenarios. If this number is low, then it would be easy, on average, for an attacker to determine the missing text based solely on the width of the redaction.

		Font	
Lexicon (Size)	*Times*	*Helvetica*	*Courier*
YAWL (264,057)	290	261	548
COUNTRIES (416)	1.33	1.28	1.97
CONGRESS (540)	1.66	1.60	2.90
NAMES (487,861,706)	481,125	427,573	969,903

Table 1. Average number of text strings of a given width.

As can be seen from this analysis, certain classes of data are simply unsafe; if it is known that the redacted text is the name of a country or a member of Congress, the adversary can, with high probability, recover it. It is also interesting to note that, from this standpoint, Courier is a safer font to use than either Times or Helvetica. This is because it conveys less information since all of its characters are the same width.

Moving beyond the average case, we believe it is also important to consider worst-case analyses. As noted previously, security applications are unlike most traditional pattern recognition problems. If it is critical to avoid

[‡] We used yawl-0.3 in our studies. The current version available online at http://www.ibiblio.org/pub/Linux/libs/ is yawl-0.3.2.

any leakage of information through redaction whatsoever, it may be desirable to develop special-purpose tools for checking redacted material to certify that the worst case has indeed not arisen.

In Table 2, we show a set of three values for each scenario. In order, these are: (1) the number of text strings which are uniquely determined by their widths, (2) the number of text strings which share their widths with only one other text string, and (3) the number of text strings which share their widths with two other text strings. These are the kinds of redactions that would be easy for an adversary to reverse-engineer; note that every lexicon-font combination possesses some number of such cases.

	Font		
Lexicon (Size)	*Times*	*Helvetica*	*Courier*
YAWL (264,057)	68, 70, 75	78, 72, 84	40, 30, 42
COUNTRIES (416)	238, 108, 48	258, 102, 33	132, 68, 60
CONGRESS (540)	186, 172, 114	201, 154, 159	63, 92, 78
NAMES (487,861,706)	9, 22, 18	16, 32, 30	3, 8, 9

Table 2. Worst-case analysis of information leakage through text string widths.

As this work is just beginning, we expect that other such suggestive experimental results would be interesting and useful to generate.

6. COUNTERMEASURES

Given an analysis of the potential weaknesses in redaction, we also wish to consider what can be done to mitigate or eliminate such problems. Below we list a series of countermeasures in increasing order of stringency:

- Insist that all of the text in question, including any signs of ascenders and descenders (or lack thereof), be completely obliterated.
- Avoid the risk of "bleed-through" by forbidding redaction using black marker pens.
- Require that several unimportant words surrounding the sensitive text be obliterated in addition (making the redaction non-minimal).
- Some fonts reveal more information via word widths than others; do not permit sensitive documents to be generated using such fonts.
- Produce special "secure-for-redaction" fonts in which the width of each character is varied randomly by a small amount every time it is used on the page.
- Through the use of document image analysis techniques, "jiggle" the placement of non-redacted text surrounding a redacted region to obscure the precise width of the missing material.
- Taking the previous idea to an extreme, reflow the text so that the widths of all redacted regions are made to be the same constant size; Breuel describes a technique that could be applied to this end.[15] A large, fixed-sized rectangle reading "REDACTED" could be inserted where the redacted text was removed.
- Apply optical character recognition to the text on the page (if the original electronic document is unavailable) and re-typeset it to hide the width of the redacted regions.

To be fair, it seems certain that many of these guidelines are already well-known and practiced by those who perform redaction; as researchers, we simply have no way of confirming this.

Note that there are some interesting tradeoffs here, as each countermeasure is more or less effective against different kinds of attacks, while its impact on the physical appearance of the document will vary as well. Most of the above techniques will hide the length of a missing word, for example, but few will obscure the fact that it is, say, a geographic place name. We have not yet begun to examine the interactions between NLP and document analysis in the context of redaction. These are questions for future research.

Beyond the sorts of studies we have proposed, it would be useful, we believe, to investigate developing an automated or semi-automated system that accepts as input a redacted document and produces one of two outputs:

1. PASS: this document is okay to release as-is; it does not appear to reveal sensitive information.
2. FAIL: this document must not be released without further redaction because it is susceptible to one or more known types of attack.

Ideally, this determination would be accompanied by a detailed analysis of how information might leak under different scenarios. While no such system could be perfect, the assistance it would provide to humans who must perform the task of redaction could be invaluable.

To this end, we have already begun to build a prototype system, which we call Plumber, to test these ideas; a screen snapshot is shown in Figure 3. Plumber is written in Tcl/Tk, a general-purpose scripting language that is popular for implementing user interfaces.[16] As demonstrated in the figure, Plumber integrates a graphical browser for scanned page images along with document analysis and language resources useful for checking for leaks. The system is semi-automated; the user interacts with the page in question, for example, by marking it up to delineate redacted regions, spacing widths, etc. and designating the suspected font, and then invokes various functions for confirming whether the information that is intended to be hidden might be revealed.

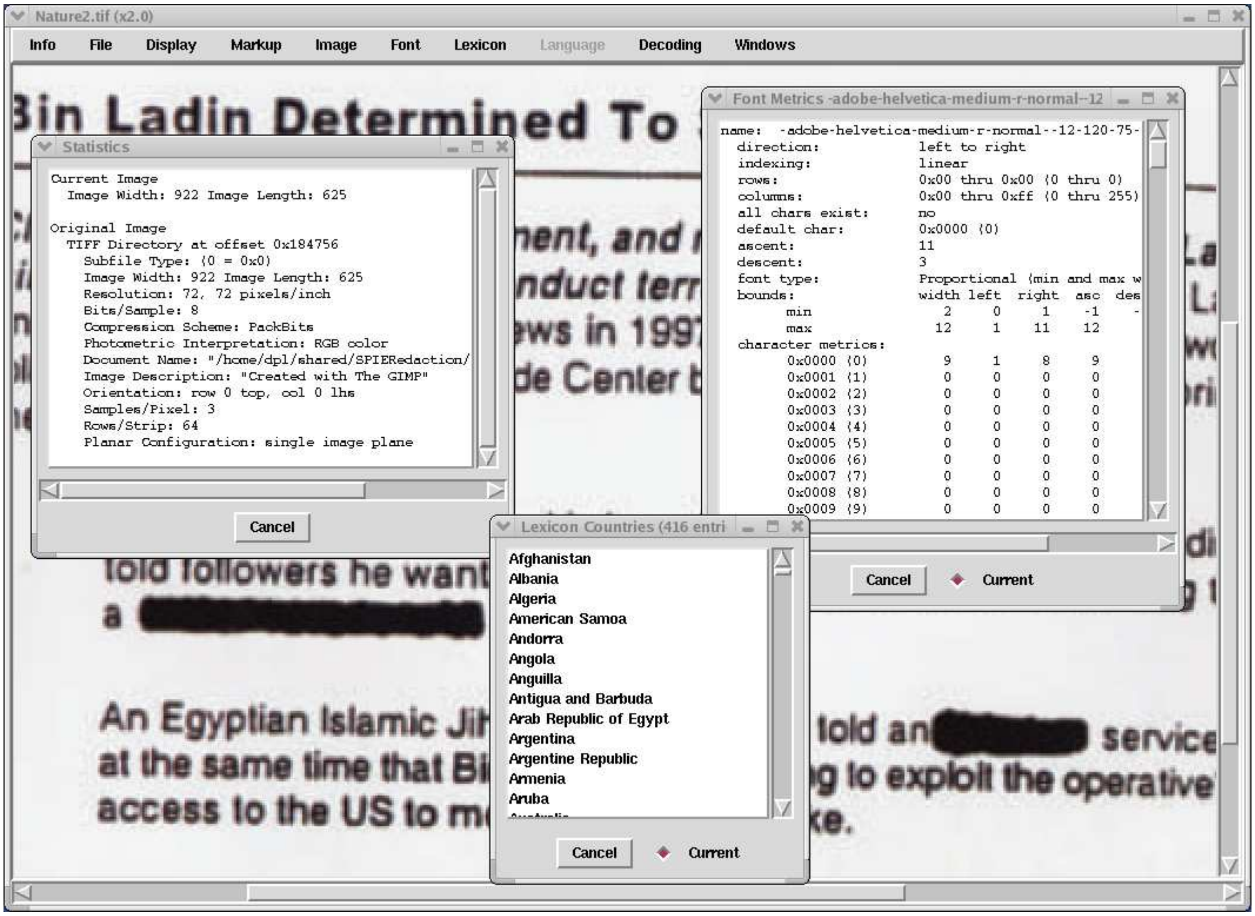
When a potential decoding is determined, Plumber allows the user to render the string in the indicated font and overlay it on the redacted region. In this way, the user can confirm whether or not the string fits based on the other image cues that are available. The current version of Plumber also includes a powerful "grep-like" search function based on Spitz's character shape code (CSC) concept.[4]

7. CONCLUSIONS

In this paper, we have examined ways in which sensitive information might leak through the process of redaction. Such attacks apply known methods from document image analysis and natural language processing. Without studies such as the ones we are proposing, our sense of security concerning redacted documents may well be misplaced. Once these weaknesses have been systematically identified and tested, effective countermeasures can be designed.

REFERENCES

1. D. Butler, "US intelligence exposed as student decodes Iraq memo," *Nature* **429**, p. 116, May 2004.
2. D. Naccache, May 2004. Private communication.
3. J. W. Leonard, "Classified National Security Information Directive No. 1," September 2003. http://www.archives.gov/isoo/policy_documents/eo_12958_implementing_directive.html.
4. A. L. Spitz, "Progress in document reconstruction," in *International Conference on Pattern Recognition*, pp. 464–467, (Quebec City, Canada), 2002.
5. S. Khoubyari and J. J. Hull, "Font and function word identification in document recognition," *Computer Vision and Image Understanding* **63**(1), pp. 66–74, 1996.
6. G. E. Kopec, "Least-squares font metric estimation from images," 1992.
7. A. Zramdini and R. Ingold, "Optical font recognition from projection profiles," *Electronic Publishing* **6**(3), pp. 249–260, 1993.
8. C. Fang and J. J. Hull, "A word-level deciphering algorithm for degraded document recognition," in *Symposium on Document Analysis and Information Retrieval*, pp. 191–202, 1995.
9. J. J. Hull, S. Khoubyari, and T. K. Ho, "Visual global context: Word image matching in a methodology for degraded text recognition," in *International Conference on Pattern Recognition*, pp. 665–668, (The Hague, Netherlands), 1992.



```
Current Image
  Image Width: 922 Image Length: 625

Original Image
  TIFF Directory at offset 0x184756
    Subfile Type: (0 = 0x0)
    Image Width: 922 Image Length: 625
    Resolution: 72, 72 pixels/inch
    Bits/Sample: 8
    Compression Scheme: PackBits
    Photometric Interpretation: RGB color
    Document Name: "/home/dpl/shared/SPIERedaction/
    Image Description: "Created with The GIMP"
    Orientation: row 0 top, col 0 lhs
    Samples/Pixel: 3
    Rows/Strip: 64
    Planar Configuration: single image plane
```

```
name:  -adobe-helvetica-medium-r-normal--12-120-75-
  direction:               left to right
  indexing:                linear
  rows:                    0x00 thru 0x00 (0 thru 0)
  columns:                 0x00 thru 0xff (0 thru 255)
  all chars exist:         no
  default char:            0x0000 (0)
  ascent:                  11
  descent:                 3
  font type:               Proportional (min and max w
  bounds:                  width left  right  asc  des
        min                   2     0     1    -1    -
        max                  12     1    11    12
  character metrics:
        0x0000 (0)            9     1     8     9
        0x0001 (1)            0     0     0     0
        0x0002 (2)            0     0     0     0
        0x0003 (3)            0     0     0     0
        0x0004 (4)            0     0     0     0
        0x0005 (5)            0     0     0     0
        0x0006 (6)            0     0     0     0
        0x0007 (7)            0     0     0     0
        0x0008 (8)            0     0     0     0
        0x0009 (9)            0     0     0     0
```

```
Afghanistan
Albania
Algeria
American Samoa
Andorra
Angola
Anguilla
Antigua and Barbuda
Arab Republic of Egypt
Argentina
Argentine Republic
Armenia
Aruba
```

Figure 3. Screen snapshot of the Plumber prototype.

10. google, October 2004. http://www.google.com.
11. Adobe Systems Incorporated, San Jose, CA, *Adobe Font Metrics File Format Specification*, October 1998.
12. YAWL (yet another word list), July 2004. http://metalab.unc.edu/pub/Linux/libs/yawl-0.3.tar.gz.
13. U.S. Census Bureau: Name Files, October 2004. http://www.census.gov/genealogy/names/.
14. U.S. Census Bureau: U.S. and World Population Clocks, October 2004. http://www.census.gov/main/www/popclock.html.
15. T. M. Breuel, "Reflowable document images for the Web," in *Second International Workshop on Web Document Analysis*, (Edinburgh, UK), August 2003. http://www.csc.liv.ac.uk/~wda2003/Papers/Section_II/Paper_5.pdf.
16. Tcl Developer Xchange, October 2004. http://www.tcl.tk/.