An iterative graph cut optimization algorithm for a double MRF prior
Christian Wolf

To cite this version:
Christian Wolf. An iterative graph cut optimization algorithm for a double MRF prior. 2008. hal-00422548

HAL Id: hal-00422548
https://hal.archives-ouvertes.fr/hal-00422548
Preprint submitted on 7 Oct 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
An iterative graph cut optimization algorithm for a double MRF prior

Christian Wolf

Technical Report LIRIS
Laboratoire d’informatique en images et systèmes d’information - UMR 5205
Université de Lyon
INSA-Lyon, Bât. J.Verne; 20, Av. Albert Einstein
69621 Villeurbanne cedex, France
Tel.: 0033 4 72 43 63 08 Fax.: 0033 4 72 43 71 17
Email: christian.wolf@liris.cnrs.fr
Web: http://liris.cnrs.fr/christian.wolf

July 19, 2008

Abstract
In a previous publication we presented a double MRF model capable of separately regularizing the recto and verso side of a document suffering from ink bleed through. In this paper we show that this model naturally leads to an efficient optimization method based on the minimum cut/maximum flow in a graph. The proposed method is evaluated on scanned document images from the 18th century, showing an improvement of character recognition results compared to other restoration methods.

Keywords
Markov Random Fields, Bayesian estimation, Graph cuts, Document Image Restoration.

1 Introduction
In a previous paper [15] we presented a double Markov random field (MRF) model designed for document restoration, more specifically document ink bleedthrough removal, i.e. the extraction and replacement of the verso ink traversing the paper and showing on the recto side. The original method used simulated annealing for optimization, which is theoretically known to converge to global solution but which is painfully slow in practice. In this paper we propose an efficient optimization technique based on graph cuts.

The paper is organized as follows: section 2 very briefly describes the graphical model, details on its motivation and its derivation can be found in [15]. Section 3 formulates the iterative optimization algorithm based on graph cuts. Section 4 illustrates the experiments performed to evaluate the results on a dataset of scanned documents, and section 5 finally concludes.

2 The graphical model
The model introduced in [15] consists of two hidden label fields, one for the recto side and one for the verso side, as well as a single observation field (the scanned image). We therefore consider a segmentation problem where each pixel corresponds to two different hidden labels, one for each field, and where each label is chosen from a space of two labels: text and background. Formally, we have a graph $\mathcal{G} = \{V, E\}$ with a set of nodes $V$ and a set of edges $E$. $V$ is partitioned into three subsets: the two fields of hidden variables $F^1$ and $F^2$ and the field of observed variables $D$. The three fields are indexed by the same indices corresponding to the pixels of the image, i.e. $F^1_s$, $F^2_s$ and $D_s$, respectively, the hidden recto label, the hidden verso label and the observation for the same pixel $s$. The hidden variables $F^1_s$ and $F^2_s$ may take values from the set $\Lambda = \{0, 1\}$, where 0 corresponds to background and 1 corresponds to text. The set of edges $E$ defines the neighborhood on the graph, i.e. there is an edge between to nodes $r$ and $s$ if and only if $r \in N_s$ and $s \in N_r$.

The dependency graph (see figure 1) contains the following cliques types: first order and second order “intra-field” cliques in the subgraph $F^1$, first order...
and second order “intra-field” cliques in the subgraph \( F^2 \) (we will assume the 3-node clique potentials to be zero) and finally the “inter-field” cliques between \( F^1, F^2 \) and \( D \).

The joint probability distribution of the whole graph can therefore be given as follows:

\[
P(f^1, f^2, d) = \frac{1}{Z} \exp \{-U(f^1) + U(f^2) + U(f^1, f^2, d)\} / T
\]

The terms \( U(f^1) \) and \( U(f^2) \) correspond to two Potts models, one prior for each field:

\[
U(f^1) + U(f^2) = U(f^1, f^2) = \sum_{\{s\} \in C_1} \alpha^1 f^1_s + \sum_{\{s, s\} \in C_2} \beta^1_{s, s'} \delta f^1_s f^1_{s'} + \sum_{\{s\} \in C_1} \alpha^2 f^2_s + \sum_{\{s, s'\} \in C_2} \beta^2_{s, s'} \delta f^2_s f^2_{s'}
\]

where \( C_1 \) is the set of single site cliques, \( C_2 \) is the set of pair site cliques and \( \delta \) is the Kronecker delta defined as \( \delta_{i,j} = 1 \) if \( i = j \) and 0 else.

The term \( U(f^1, f^2, d) \) corresponds to the data likelihood, which factorizes as follows:

\[
P(d|f^1, f^2) = \prod_s N(d_s; \mu_s, \Sigma_s)
\]

where \( \mu_s \) is the mean for class \( f_s \) (in the degraded image) and \( \Sigma_s \) is the covariance matrix for class \( f_s \) given as follows (note that \( \mu_r \neq \mu^r \) etc.):

\[
\mu_s = \begin{cases} 
\mu_r & \text{if } f^1_s = \text{text} \\
\mu_v & \text{if } f^1_s = \text{background} \text{ and } f^2_s = \text{text} \\
\mu_{bg} & \text{else}
\end{cases}
\]

\[
\Sigma_s = \begin{cases} 
\Sigma_r & \text{if } f^1_s = \text{text} \\
\Sigma_v & \text{if } f^1_s = \text{background} \text{ and } f^2_s = \text{text} \\
\Sigma_{bg} & \text{else}
\end{cases}
\]

where \( \mu_r, \mu_v, \) and \( \mu_{bg} \) are, respectively, and in the degraded image, the means for the recto class, the verso class and the background class, and the covariances are denoted equivalently.

3 The posterior probability and its maximization with graph cuts

Applying the Bayes rule, the posterior probability of the two label fields can be given as follows:

\[
P(f^1, f^2|d) = \frac{1}{P(d)} P(f^1, f^2) P(d|f^1, f^2)
\]

\[
= \frac{1}{P(d)} \prod_{\{s\} \in C_1} [\alpha^1 U_1(f^1_s) + \alpha^2 U_1(f^2_s)] + \sum_{\{s, s'\} \in C_2} [\beta^1_{s, s'} U_2(f^1_s, f^1_{s'}) + \beta^2_{s, s'} U_2(f^2_s, f^2_{s'})] + \sum_{\{s\} \in C_1} U'_2(f^1_s, f^2_s, d_s)
\]

As usual, we can ignore the factor \( \frac{1}{P(d)} \) not depending on the hidden variables and maximize the joint probability, or minimize its energy. Combining it with the two Potts models and the data likelihood, we get the following energy potential function:

\[
U(f^1, f^2, d) = \sum_{\{s\} \in C_1} \alpha^1 f^1_s + \sum_{\{s, s\} \in C_2} \beta^1_{s, s'} \delta f^1_s f^1_{s'} + \sum_{\{s\} \in C_1} \alpha^2 f^2_s + \sum_{\{s, s\} \in C_2} \beta^2_{s, s'} \delta f^2_s f^2_{s'}
\]

where and \( \mu_s \) and \( \Sigma_s \) are the sufficient statistics for the observation model given by the labels \( f_s \) and \( f_s' \). To estimate the binary images, equation (5) must be maximized. Unfortunately, the function is not convex and standard gradient descent methods will most likely return a non global solution. Simulated Annealing has been proven to return the global optimum under certain conditions [4], but is painfully slow in practice. Loopy belief propagation is another option, giving an approximative solution by iteratively applying Pearl’s belief propagation algorithm originally designed for belief networks [7]. In this work we will take advantage of the nature of the dependency graph (binary labels and cliques with not more than 2 hidden labels) in order to derive an optimization algorithm based on the calculation of the minimum cut/maximum flow in a graph [1][2][3][5].

For convenience we will rewrite the energy function for the whole graph in terms of unary functions \( U_1 \) and two types of binary functions \( U_2 \) and \( U'_2 \) as follows:

\[
U(f^1, f^2, d) = \sum_{\{s\} \in C_1} [\alpha^1 U_1(f^1_s) + \alpha^2 U_1(f^2_s)]
\]

\[
+ \sum_{\{s, s\} \in C_2} [\beta^1_{s, s'} U_2(f^1_s, f^1_{s'}) + \beta^2_{s, s'} U_2(f^2_s, f^2_{s'})] + \sum_{\{s\} \in C_1} U'_2(f^1_s, f^2_s, d_s)
\]
where $U_1(f_s) = f_s$, $U_2(f_s, f_s') = \delta_{f_s, f_s'}$ and $U_3(f_1^s, f_2^s; d_s) = \frac{1}{2}(d_s - \mu_s)^T \Sigma_s^{-1}(d_s - \mu_s)$. We consider $U_3(\ldots)$ a binary function since we do not maximize over the third argument, which is an observed variable.

Although the problem involves two possible labels for each hidden variable ($|\Lambda| = 2$), the exact solution for equation (7) cannot be found using algorithms based on graph cuts. As shown by Kolmogorov et al. [5], a function of binary variables composed of unary terms and binary terms is graph-representable, i.e. it can be minimized with algorithms based on the calculation of the maximum flow in a graph, and if and only if each binary term $E(\ldots)$ is regular, i.e. it satisfies the following equation:

$$E(0, 0) + E(1, 1) \leq E(0, 1) + E(1, 0) \quad (8)$$

It can be easily seen that this is the case of the terms $U_3(\ldots)$ in equation (7), but not necessarily for all terms $U_3(\ldots)$. According to the value of the observation $d_s$ at site $s$, $U_3(f_1^s, f_2^s; d_s)$ may be regular or not. In other words, if the observation likelihood for equal labels $f_1^s$ and $f_2^s$ is higher than the observation likelihood for different labels, then the term is regular for site $s$.

We therefore propose an adaptation and extension of the iterative $\alpha$-expansion move algorithm proposed by Boykov et al. [2] for labeling problems with multiple labels ($|\Lambda| > 2$) and improved by Kolmogorov et al. [5]. In the original formulation for multi label problems, each subproblem is a binary problem where each hidden variable may take two virtual labels: $x_s$ and $\alpha$, where $x_s$ is the original (current) label, and $\alpha$ is a new label, whose value is changed at each iteration.

In our case, the iteratively solved binary labeled and regular subproblems arise fixing the hidden label of one of the two fields $F^1$ and $F^2$ and estimating the labels of the other one. Completely fixing a whole set of variables corresponds to running an $\alpha$-expansion move algorithm on a single field dependency graph where each single hidden variable $f_s$ may take 4 values (background, recto, verso, recto-verso) and the pairwise clique potentials are adapted accordingly. This optimization schedule may be improved by fixing only the variables whose site $s$ is not regular, and jointly estimate the variables $f_1^s$ and $f_2^s$ for the regular sites $s$.

For convenience we introduce a binary matrix $H$ indicating for each site $s$ whether it is regular or not, i.e. whether the associated function $U_3(f_1^s, f_2^s; d_s)$ is regular or not:

$$H_s = \begin{cases} 1 & \text{if } U_3(0, 0, d_s) + U_3(1, 1, d_s) \leq U_3(0, 1, d_s) + U_3(1, 0, d_s) \\ 0 & \text{else} \end{cases} \quad (9)$$

Figure 2 outlines the inference algorithm, which iteratively calculates the exact solution of two different binary subproblems, maximizing, respectively, $U^{-1}(f_1^s, f_2^s, d, H)$ and $U^{-1}(f_1^s, f_2^s, d, H)$. These two energy functions are actually equivalent, however, the set of fixed variables and the set of estimated variables being different, they lead to two different cut graphs. In order to show the derivation of the cut graph, we will rewrite the two functions by reordering some terms. W.l.o.g., in the rest of this section we describe $U^{-1}(f_1^s, f_2^s, d, H)$, i.e. the subproblem where a subset of the variables in $F^1$ is fixed, whereas the variables of $F^2$ and the complementary subset of $F^1$ are estimated.

After separating terms according to the contents of $H$, the corresponding energy function can be
given as follows:

\[
U^{\rightarrow 2}(f^1, f^2, d, H) = \sum_{\{s\} \in C_1; H_s = 0} \alpha^1 U_1(f^1_s) + \sum_{\{s\} \in C_1; H_s = 1} \alpha^1 U_1(f^1_s) + \sum_{\{s\} \in C_1} \alpha^2 U_1(f^2_s) + \sum_{\{s, s'\} \in C_2; H_s = 0 \land H_{s'} = 0} \beta^1_{s, s'} U_2(f^1_s, f^1_{s'}) + \sum_{\{s, s'\} \in C_2; H_s = 1 \land H_{s'} = 1} \beta^1_{s, s'} U_2(f^1_s, f^1_{s'}) + \sum_{\{s, s'\} \in C_2; H_s \neq H_{s'}} \beta^1_{s, s'} U_2(f^1_s, f^1_{s'}) + \sum_{\{s, s'\} \in C_2} \beta^2_{s, s'} U_2(f^2_s, f^2_{s'}) + \sum_{\{s\} \in C_1; H_s = 0} U_2^1(f^1_s, f^2_s; d_s) + \sum_{\{s\} \in C_1; H_s = 1} U_2^1(f^1_s, f^2_s; d_s)
\]

(10)

Written in this notation, the energy functions can be directly translated into a cut graph using the method introduced by Kolmogorov et al. [5]. The cut graph then contains, besides the terminal nodes source and sink, one node for each variable $F^2_s$ as well as one node for each variable $F^1_s$ satisfying $H_s = 1$. Each unary term is translated into an n-edge as well as two t-edges.

The terms in lines 1 and 4 of equation (10) do not depend on estimated variables and therefore can be omitted during the minimization. The terms in lines 2 and 3 contain standard unary functions and will be represented by t-edges. The terms in lines 5 and 7 contain standard binary functions (pairwise cliques of the Potts model) and will be represented by n-edges. The terms in line 6 are binary functions (also pairwise cliques of the Potts model) in the full original expression (equation (7)), but one of the two arguments is fixed in equation (10) describing the sub problem. They can therefore be represented as t-edges in the cut graph. Similarly, the terms in line 8 are non-regular pairwise functions of the observation model, which can be represented as t-edges. The terms in line 9, finally, correspond to the regular pairwise function of the observation model, which can be represented as n-edges.

Table 1 gives a full description of the different edges of the cut graph and their weights. Figure 3 shows an example of a dependency graph for a toy problem, a $3 \times 1$ image, and two different cut graphs. Figure 3b shows the cut graph for the α-expansion move like algorithm, i.e. all sites $s$ are considered as non-regular. The cut graph is shown for the case where the complete set of variables $F^1_s$ is fixed whereas the complete set of variables $F^2_s$ is estimated.

Figure 3c shows the extended algorithm, where the middle and the right site are considered regular, whereas the left site is considered non-regular. For the middle and the right site, the variables $F^1_s$ and $F^2_s$ are jointly estimated, whereas for the left site only $F^2_s$ is estimated whereas $F^1_s$ is fixed.

### 4 Experimental results

Evaluating document restoration algorithms is a non trivial task since ground truth is very hard to come by. Short of manually classifying each pixel in a scanned image, the only way to get reliable ground truth data on pixel level is to test the algorithm on synthetic data. These tests, on the other hand, may not be realistic enough to capture all the subtleties of a real environment. To evaluate our algorithm we therefore decided to test its ability to improve the performance of an OCR algorithm when applied to real scanned documents.

We chose a dataset consisting of 104 pages of low quality printed text from the 18th century, the *Gazettes de Leyde*. This journal in French language was printed from 1679 to 1798 in the Netherlands in order to escape the censorship in France at the 18th century and relates news of the world. The Gazettes are currently used by several research projects in social and political sciences, some of which are currently collaborating with our team in the framework of digitization projects.

From an image processing point of view, the data situates itself between the difficulty of manuscripts and the regularity of printed documents. The images of sizes around 1030×1550 pixels are of very low quality compared to modern printed text. Recognition is possible, although the performance on the non-restored images is not very high. We chose the open source OCR software Tesseract published by Google\(^1\) for our experiences, mainly because it is easily scriptable. We performed some selected experiments with the product of the market leader, Abby Finereader 8\(^2\), which performs slightly better without changing the ranking of the restoration methods.

We compared the proposed method with several competing methods. One group of algorithms purely exploits the fact that, according to the hypothesis that set recto pixels completely cover verso pixels, without taking into account interactions between neighboring pixels. Examples are the k-

---

2. [http://finereader.abbyy.com](http://finereader.abbyy.com)
In a personal communication sent for the experiments, the source images was the same. The source planes between different images, tests showed that for all 104 images of the Gazettes de Leyde, the order of the source images was the same. The source planes resembling the most to the assumed recto plane where, for both methods, source #1 and source #2, which we both included into the experiments. This was not the case for other images, as for instance the manuscripts shown in figures 7 and 8.

The last method compared to the proposed algorithm is a standard single MRF with a Potts model and three labels (recto, verso and background), optimized using Kolmogorov et al.’s version of the α-expansion move algorithm [5] and combined with the same parameter estimation and pre- and post-processing as our proposed method.

Figures 4 to 6 illustrate the OCR results on a small image taken from the Gazettes dataset. As we can see, being based on segmentation, the results for k-means and the two MRF methods are similar. The k-means result (Figure 4b) is noisy as opposed to the MRF results, the double MRF (Figure 4d) improves the regularity of the single MRF (Figure 4c). The OCR output is a little bit cleaner for the double MRF case.

The results of Niblack’s algorithm and Sauvola et al.’s algorithm show the typical weaknesses of these approaches: Niblack (Figure 5a) produces spurious components, especially in areas with few text, and Sauvola (Figure 5b) tends to cutting characters into

<table>
<thead>
<tr>
<th>n-edges for node pairs:</th>
<th>Weight</th>
<th>Line in eq. (10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_x^1, F_y^2 : (s, s') \in \mathcal{C}_2 )</td>
<td>(-\beta_{s,s'}^1)</td>
<td>7</td>
</tr>
<tr>
<td>( F_x^1, F_y^2 : (s, s') \in \mathcal{C}<em>2 \land H_s = 1 \land H</em>{s'} = 1 )</td>
<td>(-\beta_{s,s'}^2)</td>
<td>5</td>
</tr>
<tr>
<td>( F_x^3 : H_s = 1 )</td>
<td>( U_s^2(0, 1, d_s) + U_s^2(1, 0, d_s) - U_s^2(0, 0, d_s) - U_s^2(1, 1, d_s) )</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 1: The edges added to the cut graph for the proposed inference algorithm; each edge corresponds to a term in eq. (10). Each t-edge is connected to the source if the weight is positive, or connected to the sink if the weight is negative, in which case the absolute value of the weight is used. Multiple edges between same nodes (taking into account the orientation) are replaced by a single edge, its weight being the sum of the individual weights.
Figure 3: (a) The dependency graph of a simple model containing three pixels in a single row; (b) the cut graph for an α-expansion move like inference algorithm: inference of the verso pixels; (c) the cut graph for the proposed inference algorithm: joint inference of the verso pixels and of a subset of the recto pixels. In this example, the potential functions related to the observation model are regular for the middle and for the right pixel ($H_s = 1$), but not for the left one ($H_s = 0$).

<table>
<thead>
<tr>
<th>Method-Type</th>
<th>Method</th>
<th>Recall (in %)</th>
<th>Prec. (in %)</th>
<th>Cost (abs.)</th>
<th>Size of dataset (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>No restoration</td>
<td>65.65</td>
<td>49.91</td>
<td>76,752</td>
<td>100</td>
</tr>
<tr>
<td>Context-free</td>
<td>Niblack [6] (segm. only)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Sauvola et al. [8] (segm. only)</td>
<td>78.75</td>
<td>66.78</td>
<td>45,363</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>K-Means (k=3)</td>
<td>79.82</td>
<td>68.43</td>
<td>42,675</td>
<td>100</td>
</tr>
<tr>
<td>Source-sep.</td>
<td>Tonazzini et al. [12] - src #1</td>
<td>†</td>
<td>41.00</td>
<td>30.05</td>
<td>74,819</td>
</tr>
<tr>
<td></td>
<td>Tonazzini et al. [12] - src #2</td>
<td>†</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Tonazzini et al. [9] - src #1</td>
<td>†</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Tonazzini et al. [9] - src #2</td>
<td>†</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Tonazzini et al. [9] - 3 sources</td>
<td>†</td>
<td>50.52</td>
<td>33.90</td>
<td>101,280</td>
</tr>
<tr>
<td>MRF</td>
<td>Single MRF &amp; α-exp. move [9]</td>
<td>†</td>
<td>81.99</td>
<td>72.12</td>
<td>36,744</td>
</tr>
<tr>
<td></td>
<td>Double MRF (proposed method)</td>
<td><strong>83.23</strong></td>
<td><strong>74.85</strong></td>
<td><strong>32,537</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

†Not available: lack of OCR performance makes a correct evaluation impossible
‡results obtained with a subset of the images only (absolute cost is not comparable).

Table 2: OCR results on a database of 104 scanned document images: non-restored input images and different restoration methods.

The result seems to be better than the ones consisting of a single source component only. Surprisingly, this result is the only one which produces at least limited OCR output, whereas the other images do not produce anything meaningful.

In order to evaluate the amount of recognition improvement of the restoration method, we manually created groundtruth for the 104 images, and calculated the Levenstein edit distance between two strings [14], which finds the optimal transformation from on string into another with elementary operations (insertion, deletion, substitution) minimizing the global cost of these operations. Additionally, we calculated character recall and character precision derived from the transformation operation of this distance. Table 2 compares the measures for the different methods described above, as well as the recognition performance on not restored images. Note, that precision and recall are indepen-
Figure 4: Small extracts of the OCR results obtained on scanned document images: (a) input image (no restoration); (b) k-means segmentation + restoration; (c) single MRF segmentation [5] + restoration; (d) double MRF (proposed method).
bras de la Compagnie : Il etait accompagné de son Oncle, Frère de sa Mère. Le jeune Prince avait pour motif de sa fuite le désir de trouver les moyens de rejoindre son Père du trié état, où il étoit réduit.

Figure 5: Small extracts of the OCR results obtained on scanned document images: (a) segmentation with Niblack [6]; (b) segmentation with Sauvola et al. [8]; (c) Tonazzini et al. [12] source #1, Tonazzini et al. [12] source #2
Figure 6: Small extracts of the OCR results obtained on scanned document images: (a) Tonazzini et al. [9] source #1 (b) Tonazzini et al. [9] source #2 (c) Tonazzini et al. [9] source #3 (d) Tonazzini et al. [9] all three sources combined.
Figure 7: Restoration results on real data. From left to right, top to bottom: input image, k-means, single MRF & α-exp. move [5], double MRF (proposed method), Tonazzini et al. [12] source #1, Tonazzini et al. [12] source #2, Tonazzini et al. [9] source #1, Tonazzini et al. [9] source #2 (source #3 not displayed).
Figure 8: Restoration results on real data. From left to right, top to bottom: input image, k-means, single MRF & α-exp. move [5], double MRF (proposed method), Tonazzini et al. [12] source #2, Tonazzini et al. [12] source #1, Tonazzini et al. [9] source #3, Tonazzini et al. [9] source #1 (source #2 not displayed).
nant of the dataset size, whereas the total transformation cost is not.

We can see that all methods based on identifying the verso component (k-means and the two MRF methods, including the proposed one) are capable of significantly improving the recognition results compared to no restoration at all. Not surprisingly, regularizing the segmentation with a priori knowledge boosts the performance. Separating the regularization of the recto and verso side further improves recognition, gaining 1.2 percent points in recall compared to the single MRF and 2.7 percentage points in precision. Totally, compared to no restoration at all, the proposed method improves recognition at about 17 percentage points in terms of recall and around 25 percentage points in terms of precision.

Recognition on the results of Niblack’s method produces only gibberish, probably because of the small ghost objects it creates. Sauvola et al.’s method overcomes this problem and the recognition performance almost attains the quality of the three class segmentation of performed by the k-means algorithm.

Surprisingly, the recognition performance on the results of the two source separation results was very disappointing. We performed recognition experiments for both planes of the first method [12] and all three planes of the second method [9], respecting the author’s recommendations to darken the images after applying the inverted mixture matrix. In a personal communication for the experiments in this paper, Prof. Tonazzini recommended subtracting the K component of the CMYK color decomposition. However, we obtained better results with a histogram stretch instead of the proposed method.

Unfortunately, the recognition performance on these results was not good enough to include it in the table. Most of the output was blank or gibberish, making an evaluation impossible. We managed to get some statistics on the first source plane of the first method, as well as on output images combining all three source planes of the second method. However, this was only possible when a subset of the dataset was removed. Even then, the results were not competitive.

Figures 7 and 8 show restoration results on two different manuscript images. The source separation methods remove more of the verso text in Figure 7, but unfortunately the contrast is very low and they are significantly disturbed by the JPEG artifacts in the input image. The performance shown in figure 8 reveals similar strengths and weaknesses, typical to the two types of approaches: the regularized segmentation approaches create crisp images but show localized artifacts, whereas the artifacts created by the source separation methods are more spread out across the image and seem to touch more of the low frequency components.

### 4.1 Computational complexity

The computational complexity of the proposed method is dominated by the inference part with minimum cut/maximum flow whose complexity is bounded by $O(|E| + f)$, where $|E|$ is the number of edges in the graph and $f$ is the maximum flow. We use the graph cut implementation by Boykov and Kolmogorov [1] which has been optimized for typical graph structures encountered in computer vision and whose running time is nearly linear in running time in practice [2]. Table 3 gives effective run times measured on a computer equipped with an Intel Core 2 processor running at 2.5Ghz and 4GB of RAM (only one core was used). The running time of the proposed method is comparable to the running time of a single MRF with graph cut optimization and quite competitive given its restoration performance.

### 5 Conclusion and Outlook

In this paper we presented a method to separate the verso side from the recto side of a single scan of document images. The novelty of the method is the separation of the MRF prior into two different label fields, each of which regularizes one of the two sides of the document. This separation allows to estimate the verso pixels of the document which are covered by the recto pixels, which, again through the MRF prior, improves the estimation of the verso pixels not covered by recto pixels, thus increasing the performance of the regularization. We showed that this formulation leads to an efficient

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>1026 \times 1587</th>
<th>2436 \times 3320</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-free</td>
<td>Niblack [6]</td>
<td>0.6</td>
<td>2.9</td>
</tr>
<tr>
<td>Source-separation</td>
<td>Sauvola et al. [8]</td>
<td>0.6</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>K-Means (k=3)</td>
<td>1.9</td>
<td>10.5</td>
</tr>
<tr>
<td>Source-separation</td>
<td>Tonazzini et al. [9]</td>
<td>36.9</td>
<td>134.6</td>
</tr>
<tr>
<td></td>
<td>Tonazzini et al. [12]</td>
<td>17.0</td>
<td>74.5</td>
</tr>
<tr>
<td>MRF</td>
<td>Single MRF [5]</td>
<td>7.2</td>
<td>34.9</td>
</tr>
<tr>
<td></td>
<td>Double MRF</td>
<td>7.4</td>
<td>36.4</td>
</tr>
</tbody>
</table>

† Code in C++
‡ Code in Matlab/GNU Octave

Table 3: Execution times in seconds for various methods. The replacement of the background pixels is included if the method is based on segmentation.
algorithm based on graph cuts.
The performance of the method has been evaluated on scanned document images from the 18th century, showing that the restoration is able to improve the recognition performance of an OCR significantly, compared to non restored images but also compared to competing methods.

6 Acknowledgements

We thank Anna Tonazzini for providing us with the source code of the two source separation methods and her kind help in setting up the corresponding experiments as well as for the interesting discussions.

References


