Text Classification and Layout Analysis of Paper Fragments*

Stefan Fiel, Markus Diem, Florian Kleber, Angelika Garz, and Robert Sablatnig

Computer Vision Lab
Vienna University of Technology, Austria
fiel@caa.tuwien.ac.at

Abstract
Document image analysis such as text classification and layout analysis allow for the automated extraction of document properties. In general these methodologies are pre-processing steps for Optical Character Recognition (OCR) systems. In contrast, the proposed method aims at clustering document snippets so that an automated clustering of documents can be performed. First, localized words are classified according to printed text, manuscript, and noise. The third class permits the correction of falsely segmented background elements. Having classified the text elements, a clustering is carried out which groups words into text lines and paragraphs. A back propagation of the class weights - assigned to each word in the first step - enables correcting wrong class labels. Finally, additional features such as the detection of underlined text or the paragraph layout (e.g. left aligned, centered) are extracted. The proposed method shows promising results on a dataset consisting of document fragments with varying shapes, content writing and layout.

1 Introduction

Text localization, text classification, and layout analysis are important pre-processing steps for Optical Character Recognition (OCR) systems. Since these methods allow for an analysis of document images, they are additionally applied for indexing digitized images and the clustering of documents according to their content. The methods presented in this paper are applied to cluster document fragments.

In total, 600 million-odd snippets of Stasi documents were discovered after the fall of the Berlin Wall [9]. The documents were fragmented in 1989 when Stasi officers tried to destroy secret files. The data considered consists of manually torn documents with German, English, and Russian text. Thus, snippets have irregular shapes and their content varies from two words up to hundreds of words. Additionally machine printed and handwritten text is present. The dataset contains documents which are carbon copies, colored paper, lined or checked paper, or old fashioned copies.

To handle such amounts of document fragments an automated clustering based on the described features can be performed. Features for document clustering include amongst others, the paper color, the writing color, the background texture (e.g. lined/checked), text localization, text classification, and layout analysis. In this paper the last two methods are discussed in more detail.

For the text classification three classes are introduced, where the first two classes (*print, manuscript*) distinguish between machine printed text and handwritten, the third class (*noise*) detects falsely segmented background elements. The classification is based on so-called Gradient Shape Features (GSF) which can deal with noisy text. Multiple Support Vector Machines (SVM) are trained for the final

*This work was supported by the Fraunhofer-Institute for Production Systems and Design Technology (IPK), Berlin.

class decision. Subsequently a clustering is performed on word blobs which groups words into text lines and paragraphs. A global voting finally corrects false class decisions based on neighboring words.

This paper is organized as follows. The subsequent section discusses current state-of-the-art methods that deal with text classification. Then, Section 3 details the proposed method. Finally, an evaluation on real world data is presented in Section 4.

2 Related Work

Typical document analysis steps include skew estimation [10], document binarization [11], text line extraction [4], text classification, and layout analysis. These processing steps are on the one hand needed to perform OCR of documents. On the other hand, they allow for structuring digitized documents with respect to their content. In our case, document analysis aims at clustering document snippets that have varying supporting material, type face, and layouts.

An early work on text classification was done by Kuhnke et al. [7]. They try to distinguish between machine printed and handwritten characters in order to support OCR. Therefore, line features such as the straightness of lines and symmetry features are extracted and classified by a neural network.

Kandan et al. [6] classify text into handwritten and machine printed characters as pre-processing step of OCR. They extract invariant moments from the binary image which are classified by means of a Support Vector Machine (SVM). Subsequently, a voting scheme based on delaunay triangulation improves the classification performance.

Recently Chanda et al. [5] proposed a text classification method applied to torn documents which is capable of identifying noise, handwritten, and printed text. They implement a two tier approach where text and non text elements are identified in the first tier which is based on gabor filter features classified by a SVM. The second tier distinguishes handwritten and printed text by means of directional features again classified with a SVM.

A remarkable approach was proposed by Zheng et al. [13] which identifies handwritten and machine printed text in noisy images. They extract a total of 140 features that capture structure, stroke properties, and texture in order to identify noise, printed, and handwritten text. After feature selection, 31 features are left which are classified by means of a SVM. In order to improve the classification results, a MRF models the geometrical structure of all classes and corrects the words' class labels.

3 Methodology

To handle amounts of document fragments in the order of millions (e.g. destroyed Stasi files, see Section 1) layout analysis permits an automated clustering as a preprocessing step for further analysis. Hence, features that describe the content of document fragments, such as text classification, background texture, and the layout, can be determined to cluster documents according to their subject.

Pre-processing steps such as binarization or skew estimation, which are needed for the subsequently introduced layout analysis are shown in Figure 1. In order to localize and classify text regions, words are estimated by means of Local Projection Profiles [2]. Then, automatically detected text lines split word blobs which are falsely merged between text lines (dashed lines in Figure 1 b)). Text decorations such as underlines are additionally removed in order to improve the text localization (red/gray lines in Figure 1 b)). Finally, minimum area rectangles are found by means of Rotating Calipers [12] (see Figure 1 c)). Minimum area rectangles are the data-structure for all subsequent processing steps since they can be stored efficiently while still being a close approximation to words.

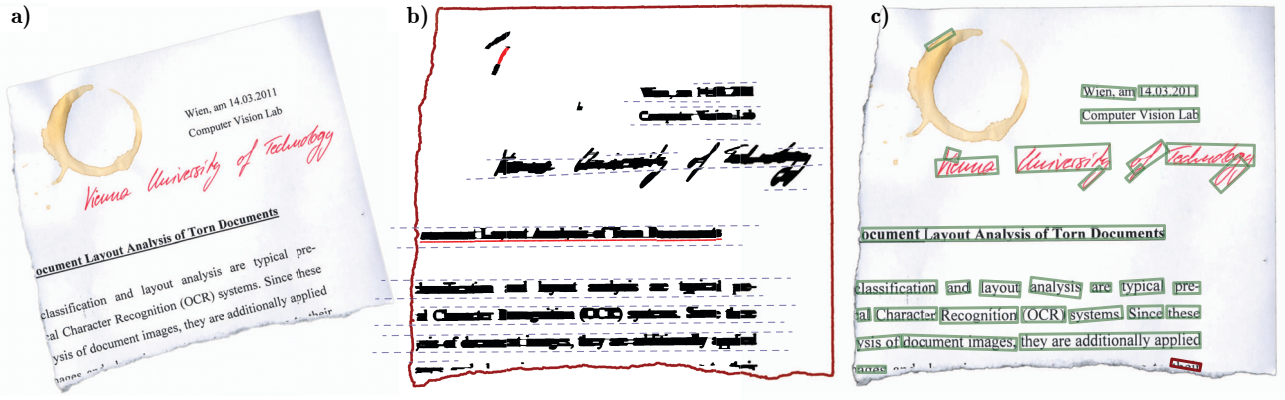

Figure 1. Input image a), word blobs obtained by eroding the LPP image b) the light blue dashed lines are text lines which separate fused blobs while red (gray) blobs denote lines which are detected in the segmented image. The final minimum area rectangles are shown in c). Note that solely one text blob is classified falsely.

Having detected possible word candidates, Gradient Shape Features (GSF) which are described in the subsequent section are computed for each character. However, an accurate character localization is not needed since it is rather desired to capture local structure than individual characters. Besides, character segmentation is still a challenging task when manuscript images are considered. Thus, characters are approximated by squared windows which fit into the minimum area rectangle. This square is shifted along the rectangle's principle axis in order to compute features of overlapping image regions (see Figure 2). The interest region detection additionally guarantees that features are extracted robust against changes of the word's scale.

The GSFs are gradient features which have a similar coordinate system to Belongie's Shape Context features [3]. However, since they are computed by means of the image's pixel values, they do not suffer by poor binarization results (e.g. caused by background clutter). The introduced features are additionally capable to distinguish between text elements and background clutter or images when trained properly.

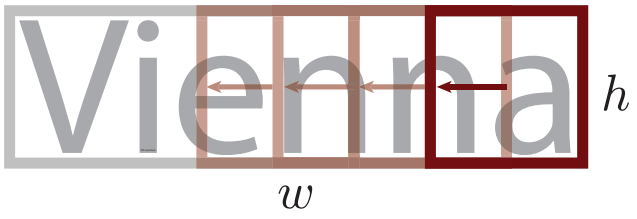

Figure 2. The gray rectangle shows the minimum area rectangle. The dark red square represents the character estimation. Having computed a feature within a square, it is shifted in order to calculate the feature of the subsequent character.

After the feature extraction, a classification is performed by means of multiple SVMs. Finally, a layout analysis is proposed which groups text lines and paragraphs while, at the same time, applying a voting which improves the classification results.

3.1 Gradient Shape Features

The proposed features for font classification are based on Shape Context features introduced by Belongie et al. [3]. In contrast to Shape Context features which are extracted in binary images, the Gradient Shape Features are computed on gradient images. Thus, they tolerate failures of previous processing steps such as binarization. As proposed by Mikolajczyk et al. [8], the features are robust against all anticipated transformations including changes of the word's scale, rotation, and illumination (contrast). They are not robust with respect to affine transformations which improves their discriminativity. In addition, the features are robust with respect to changes in the polar coordinate

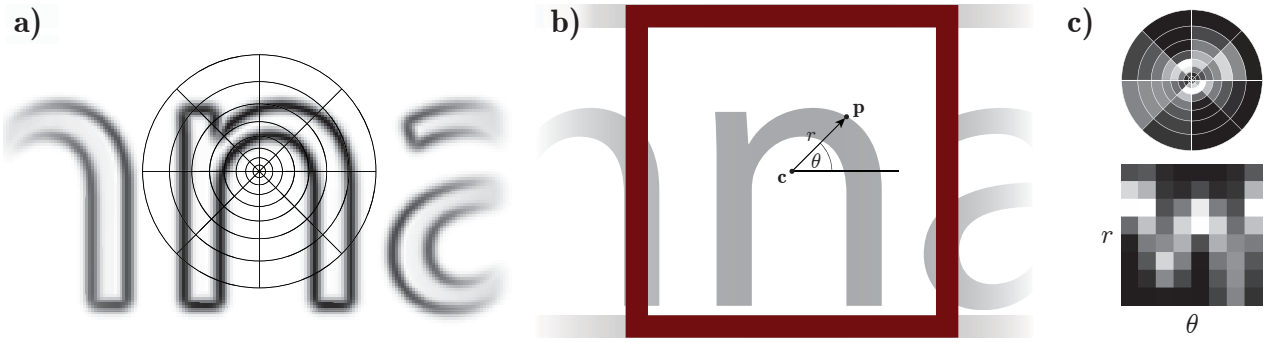

Figure 3. Log-polar coordinates of a pixel a), the log-polar grid on an inverted gradient magnitude image b), and the resulting feature vector c).

origin. Even partially visible characters can be classified correctly, since the features capture the nature of junctions and strokes rather than the topology of whole characters.

The gradient magnitude image is the basis for the feature computation. In order to compute a Gradient Shape Feature, solely pixel within the current character window are observed. First, the pixel coordinates are computed relative to the center $\mathbf{c}$ of the character window which is the point of origin in the log-polar coordinate system (see Figure 3 a)). Then, the log-polar vector $\mathbf{p} = (r, \theta)$ can be computed by:

$$r = \log \sqrt{x^2 + y^2} \tag{1}$$

$$\theta = \tan^{1} \frac{y}{x} \tag{2}$$

with x, y being the relative coordinates of the current pixel. Figure 3 b) illustrates the character window and log-polar coordinates of a relative pixel vector $\mathbf{p}$. The word's dominant orientation θ_w is subtracted from the angular coordinates in order to achieve robustness with respect to rotation, resulting in $\mathbf{p} = (r, \theta \quad \theta_w)$.

The distribution of the area of bins with increasing radius is not linear which leads to an inhomogeneous distributed gradient histogram: Bins near to the center have a lower area than those at the border. Thus, the feature's rows (see Figure 3 c) need to be normalized according to their area.

Finally, a feature is created which locally captures the gradient magnitude robust against orientation, scale, and contrast changes. The proposed feature qualifies for text classification since it captures the stroke width, the stroke's straightness, and the appearance of junctions.

3.2 Classification

For the classification process, a SVM is trained. Since one-against-all tests are performed, one Radial Basis Function (RBF) kernel is trained per class instance. For the proposed system, three classes need to be trained: *printed text*, *manuscript*, and *noise*. The former two classes differentiate machine printed text from manuscripts while the latter class targets at removing falsely segmented elements such as ruling, puncher holes, text decorations, or other artifacts.

Having trained all RBF kernels, text in document images can be classified. Therefore, the GSFs are computed for a particular word blob. Subsequently, the feature vectors are classified using one-against-all tests which results in a positive or negative weight for each class trained. This weight histogram is accumulated over all features of a word. Hence, a weight histogram is created where bins represent the probability of the current word of belonging to the respective class. The class label

of a word can be obtained by $\max w_i$ with w_i being the weight of the *i-th* class. However, the voting of text areas (see Section 3.3) is improved if these weights are taken into account rather than the final class label.

3.3 Layout Analysis

Text clustering aims at grouping the previously classified word blobs. Therefore, words are clustered according to text lines and paragraphs. The former groups words within text lines, while the latter detects paragraphs or headings.

Text Clustering In order to group the detected word blobs according to text lines and paragraphs, the minimum area rectangle of each word blob is taken into account. Its major axis is extended by a so-called *fuse factor*. Then, a fusing test is performed with all remaining minimum area rectangles, that are not extended. If a corner or a midpoint of the rectangle's sides lies within the currently observed rectangle, a potential fusing candidate is found.

If a fusing candidate is found, the minimum area rectangle of both rectangles is computed. Both word blobs are then assigned as children and the clustering is carried out with the newly created rectangle.

Global Voting As soon as the words are grouped according to text lines and paragraphs, the class labels are re-computed. In order to assign a class label to text lines and paragraphs, the weight histograms of its children are taken into account. Each histogram is weighted by the area of the corresponding word blob, since the classification decision is more reliable with increasing word size. Finally, the class label corresponding to the maximal bin in the accumulated weight histogram gets assigned to the text line or paragraph.

In order to improve the text classification, a back propagation corrects falsely classified words. Thus, the weights of the parent's histogram are voted against the weights of its children. If the maximum bin changes, a new class label is assigned. This technique especially improves the classification performance, since a global class decision is added to the local class decision and weights are propagated rather than hard class decisions. Additionally, false class labels have low weights in general and are therefore corrected by neighboring words.

Layout Characteristics After the re-computation of the class labels the formatting of the paragraphs is determined. Therefore, the text lines have to be rotated according to the dominant orientation. To distinguish between centered, justified, left, or right aligned the left and the right endings of the lines are analyzed. For each side it is calculated whether it is justified or ragged. If both sides are justified the formatting is also justified, if only one side is justified the formatting is either left or right aligned and if both sides are ragged the formatting is centered.

To distinguish between ragged and justified on one side the distances of the line endings are used. For the calculation of the distances the interquartile range is taken because it eliminates outliers like the last line of a justified text. If the difference between the two quartiles is higher than a certain threshold the side is assigned as ragged otherwise it is justified. The threshold was determined empirically by analyzing 1300 justified text lines. In these lines the distance was always smaller than $15px$ which was then taken as threshold.

With help of the lines, which were removed in the preprocessing step, it is possible to determine for each word blob whether it is underlined or not. This step does not require that the text lines have to be rotated according to the dominant orientation. For each word blob the minimum area rectangle is compared with each line. First the two angles of the edges of the rectangle are compared to the

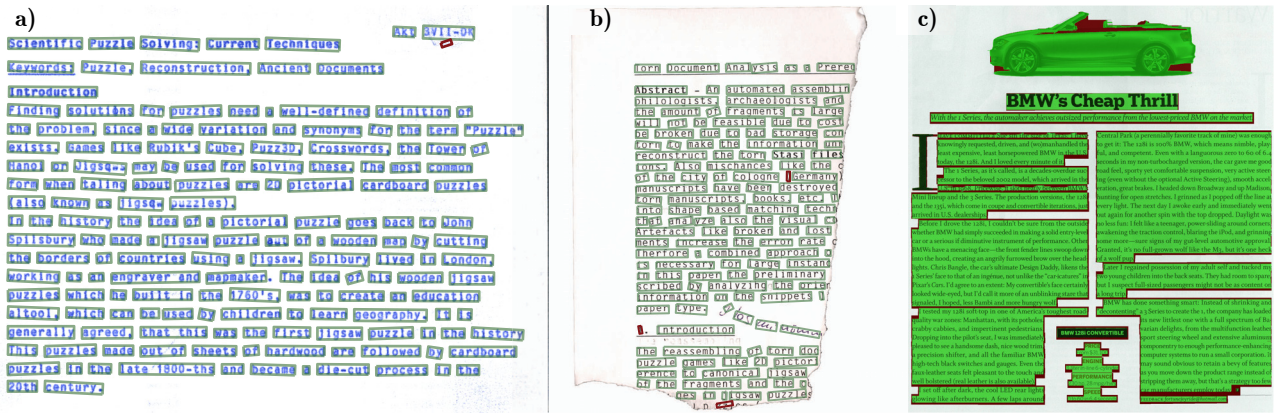

Figure 4. A carbon copy (a), a machine printed snippet with annotations (b), and page from the PRImA database (c). The light (green) rectangles in (a,b) indicate correctly classified text while dark (red) rectangles mark false classification results. In (c), light (green) areas illustrate true positives, while dark (red) areas indicate false positives and false negatives.

angle of each line. If both differences are higher than 10 degrees this line is skipped, otherwise the distances between the line and the four corners of the minimum area rectangle are calculated. If all distances are higher than two thirds of the median word height in the paragraph this line is skipped. Afterwards it is verified if the line covers the edge containing the corner with the smallest distance. Covers means that at least one of the endpoints of the line lies within the two normals of the edge at its endpoints or both endpoints of the line are lying on different sides of the normals. If the edge is covered by the line the word blob is designated as underlined.

4 Results

The proposed method was evaluated on real world data which consists of 446 Stasi file fragments. The data is particularly challenging because of its great variety. Thus it comprises snippets with varying area, background, and layout. The documents were written by varying type writers, scribes, and ink colors. Additionally, old fashioned copies with background clutters and noisy character borders are included. The original snippets must not be published due to privacy, so the examples given in Figure 4 should reasonably capture the challenges of the dataset. The snippet in Figure 4 (a) shows a carbon copy, the second snippet (b) illustrates machine printed text with annotations, and the third image (c) shows a magazin page from the ICDAR 2009 Page Segmentation Competition. The light (green) rectangles indicate correct classification results while the dark (red) rectangles mark falsely classified words.

4.1 Text Classification

In order to evaluate the proposed method, the dataset was manually tagged. In other words, each word was annotated according to its class (*print, manuscript*) while background was left blank. Table 1 shows the confusion matrix of all three classes if the method is applied to real world data. It can be seen that the *noise* has the lowest precision (63%). This can be attributed to the fact that some snippets contain bleed-through text. These text areas where annotated as noise, however their features are similar to noisy text areas. In addition, the confusion matrix shows that hardly any text (0.5% and 1.8%) is classified as noise. Machine printed text is recognized best (94.5%) by the proposed method.

In order to show the improvements of the global voting discussed in Section 3.3, the system was evaluated on the same dataset with and without global voting. The classification performance is improved by 4.8% if a voting based on the word's neighbors is performed. Considering solely the text

	predicted			
	noise	print	manuscript	#
noise	**0.625**	0.065	0.310	245
print	0.005	**0.945**	0.050	2180
manuscript	0.018	0.044	**0.938**	2034
	200	2166	2093	4459

Table 1. The rows of the confusion matrix show the groundtruth labels, while the columns represent predicted labels (e.g. 4.4% of the handwritten text is falsely classified as printed text). Additionally, the number of groundtruthed words is given for each class in the last column.

classes, global voting improves the performance by 5.1%. An overall precision of 0.924 is achieved on real world data.

Additionally, the system is compared to layout analysis systems of the ICDAR 2009 Page Segmentation Competiton [1]s in Table 2. This dataset consists of modern documents (see Figure 4). Therefore, the system was trained to differentiate between printed text, images and noise. The results in Table 2 show that the proposed method is comparable to state-of-the-art page segmentation methods.

	Non-text	Text	Overall
Vienna UT	**94.58**	94.35	**94.47**
Fraunhofer	75.15	**95.04**	93.14
FineReader	71.75	93.09	91.90
Tesseract	74.23	92.50	91.04
DICE	66.22	92.21	90.09
REGIM-ENIS	67.13	91.73	87.82
OCRopus	51.08	84.18	78.35

Table 2. F-scores of the Page Segmentation Competition 2009 [1] compared to our method (Vienna UT).

4.2 Formatting

The method for the determination of the formatting of a paragraph has been evaluated on a test dataset containing 185 paragraphs. These paragraphs have been generated with a lorem ipsum generator with a random width and also a random formatting type. Thus, each formatting type occurs 44 to 50 times. The font size has been fixed to 12 pixels. 184 (99.5%) paragraphs were identified correctly, only one paragraph was assigned to a wrong class. This paragraph has centered as formatting type but 10 of 13 lines nearly fill the width of the paragraph. Since the interquartile range eliminates the outliers the distance of the line endings on both sides is small and the paragraph is identified as justified.

5 Conclusion

Text classification and layout analysis of paper fragments was presented in this paper. The challenge of document analysis on fragments, is their varying content and the fact that methods must be capable of dealing with sparse and noisy data.

Compared to the current state-of-the-art in text classification, we employ local grayscale features which can handle noisy text, since they do not suffer from poor binarization results. The features are computed at the estimated location of characters. This methodology was adopted from the document analysis community and is different to the concept of interest points. In doing so, we save computation time and improve the class decision as a character has a more distinct shape than corners or junctions of characters. Robustness against scale changes (i.e. headlines) is achieved by the interdependence

of the features' scales and the minimum area rectangle. It was additionally shown, that the proposed system outperforms state-of-the-art layout analysis systems.

References

[1] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos. ICDAR 2009 Page Segmentation Competition. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 1370 –1374, jul. 2009.

[2] Itay Bar-Yosef, Nate Hagbi, Klara Kedem, and Itshak Dinstein. Line Segmentation for degraded handwritten historical documents. In *Int. Conference on Document Analysis and Recognition, ICDAR*, pages 1161–1165. IEEE Computer Society, 2009.

[3] Abdel Belaïd. Recognition of table of contents for electronic library consulting. *IJDAR*, 4(1):35–45, 2001.

[4] S.S. Bukhari, F. Shafait, and T.M. Breuel. Script-Independent Handwritten Textlines Segmentation Using Active Contours. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 446 –450, jul. 2009.

[5] Sukalpa Chanda, Katrin Franke, and Umapada Pal. Document-Zone Classification in Torn Documents. In *ICFHR*, pages 25–30, 2010.

[6] R. Kandan, Nirup Kumar Reddy, K. R. Arvind, and A. G. Ramakrishnan. A Robust Two Level Classification Algorithm for Text Localization in Documents. In *ISVC (2)*, pages 96–105, 2007.

[7] K. Kuhnke, L. Simoncini, and Zsolt Miklós Kovács-Vajna. A system for machine-written and hand-written character distinction. In *ICDAR*, pages 811–, 1995.

[8] Krystian Mikolajczyk and Cordelia Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[9] Bertram Nickolay and Jan Schneider. *Virtuelle Rekonstruktion "vorvernichteter" Stasi-Unterlagen. Technologische Machbarkeit und Finanzierbarkeit - Folgerungen für Wissenschaft, Kriminaltechnik und Publizistik*, volume 21, pages 11–28. Berlin, 2007.

[10] J. Sadri and M. Cheriet. A New Approach for Skew Correction of Documents Based on Particle Swarm Optimization. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 1066 –1070, jul. 2009.

[11] Bolan Su, Shijian Lu, and Chew Lim Tan. Binarization of historical document images using the local maximum and minimum. In *DAS '10: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 159–166, New York, NY, USA, 2010. ACM.

[12] Godfried Toussaint. Solving Geometric Problems with the Rotating Calipers. In *In Proceedings IEEE MELECON*, pages 10–17, 1983.

[13] Yefeng Zheng, Huiping Li, and David S. Doermann. Machine Printed Text and Handwriting Identification in Noisy Document Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(3):337–353, 2003.