c't 26/2019 ; S. 152 ; Praxis

Durchsuchbare PDF-Dokumente mit OCRmyPDF - Gratishelfer fürs papierlose Büro

Als praktischer Helfer fürs papierlose Büro macht das Linux-Tool OCRmyPDF aus eingescannten Schriftstücken durchsuchbare PDF-Dokumente. Bei Bedarf kann es Scans drehen und entzerren, Bilder optimieren und auch umfangreiche Dateien einlesen.

Von Liane M. Dubowy

Briefe, Artikel und andere Schriftstücke lassen sich gut digital archivieren und sichern. Sind sie als durchsuchbare PDF-Dokumente gespeichert, braucht man auch kein Dokumenten-Management, um etwas wiederzufinden: Eine einfache Desktop-Suche stöbert ein Schreiben dann vermutlich schneller auf, als Sie einen Ordner aus dem Regal ziehen und durchblättern können.

Das kostenlose Kommandozeilentool OCRmyPDF versieht einfache PDF-Dokumente mit einer unsichtbaren, durchsuchbaren Textebene und legt sie im Format PDF/A ab, das für die Langzeitarchivierung elektronischer Dokumente gedacht ist und sich auch in vielen Jahren noch öffnen lassen soll. Dabei platziert OCRmyPDF die Textebene recht präzise im Dokument, sodass Sie Textabschnitte mit der Maus markieren und per Copy & Paste weiterverwenden können. Ganz exakt liegen die Buchstaben dabei nicht auf dem Original, unter anderem wegen fehlender Schriftarten.

PDFs könnten Sie auch mit Ghostscript oder ImageMagick in ein Bild umwandeln und dann mit Tesseract ein durchsuchbares PDF erstellen. Dabei gehen allerdings häufig Bildqualität und viele Details verloren. OCRmyPDF analysiert dagegen vor der Verarbeitung jede Seite eines PDFs, um den Farbraum und die Auflösung zu bestimmen, die nötig sind, damit keine Details verloren gehen.

Das Python-Programm greift auf bewährte Konsolenwerkzeuge wie ImageMagick, Pdftk und Ghostscript zurück. Für die Texterkennung nutzt es die bewährte OCR-Engine Tesseract; dementsprechend gut ist das Ergebnis. Rund 100 Sprachen werden unterstützt. Statt PDF/A kann auch ein anderes Format ausgegeben werden.

Ist die Qualität des Ausgangsmaterials mangelhaft, kann OCRmyPDF die PDFs vor der Verarbeitung entzerren, drehen und geraderücken.

Blätterstapel scannen

Bevor OCRmyPDF loslegen kann, müssen Sie Ihren Papierkram digitalisieren: Ein Dokumentenscanner verarbeitet schnell große Papierberge, die wichtigsten Schriftstücke lassen sich aber auch in kurzer Zeit mit einem Flachbettscanner einlesen. Zur Not tuts auch ein Foto mit dem Smartphone. Auf dem Linux-Desktop können Sie zum Scannen das mächtige grafische Tool Xsane oder übersichtlichere Werkzeuge wie Simple-Scan oder Skanlite verwenden. Achten Sie darauf, eine Scanauflösung von mindestens 300 dpi zu verwenden, bei sehr kleinen Schriften sollten es besser 600 dpi sein.

OCRmyPDF erwartet als Eingabeformat eine oder mehrere PDF-Datei(en). Xsane und Simple-Scan speichern den Scan auf Wunsch direkt als PDF-Datei, während Skanlite ihn nur als Bilddatei ablegen kann. Falls Ihnen nur Bilddateien vorliegen, sollten Sie diese vorab ins PDF-Format konvertieren, beispielsweise mit dem Konsolentool img2pdf:

```
img2pdf --output out.pdf bild.jpg
```

Handelt es sich um ein einzelnes Bild, kann OCRmyPDF es selbst in ein PDF umwandeln:

```
ocrmypdf --image-dpi 300 bild.png : out.pdf
```

Tesseract kann ebenfalls PDFs aus Bildern erstellen, doch Funktionen wie Bildverarbeitung, Metadatenkontrolle und das PDF/A-Format fehlen dem Tool.

OCRmyPDF einrichten

Am besten installieren Sie OCRmyPDF mit der Python-Paketverwaltung pip. Zwar bieten viele Linux-Distributionen das Tool in ihren Paketquellen an – so auch Linux Mint 19.2 und Ubuntu 18.04 – doch die hier vorrätige Version 6 ist stark veraltet; aktuell ist 9.03.

OCRmyPDF greift auf einige Pakete zurück, die im Python-Fundus fehlen und daher bei der Installation mit pip nicht mit eingerichtet werden. Die fehlenden Pakete müssen Sie selbst mit der Paketverwaltung nachrüsten. OCRmyPDF braucht mindestens Python 3.6, die Python-Paketverwaltung pip, Ghostscript 9.15, qpdf 8.1.0, die Bibliothek libxml2 und die OCR-Engine Tesseract 4.0 Beta samt ihrem deutschen Sprachpaket.

Einige weitere Pakete sind optional, wir raten aber, sie einzurichten, um alle Funktionen von OCRmyPDF nutzen zu können. Sind die Pakete jbig2enc und pngquant installiert, kann das Tool verlustfrei kleinere PDF-Dateien erzeugen. Das Tool unpaper ist hingegen nötig, um Scans zu verbessern. Es entfernt beispielsweise dunkle Ecken und kann Seiten geraderücken. Unter Linux Mint/Ubuntu installieren Sie die nötigen Pakete mit dem Befehl

```
sudo apt install python3 python3-pip: ghostscript qpdf libxml2 tesseract: -ocr tesseract-ocr-deu jbig2enc : pngquant unpaper
```

Der folgende Befehl installiert schließlich die aktuelle Version von OCRmyPDF:

```
sudo pip3 install ocrmypdf
```

Durchsuchbare PDFs

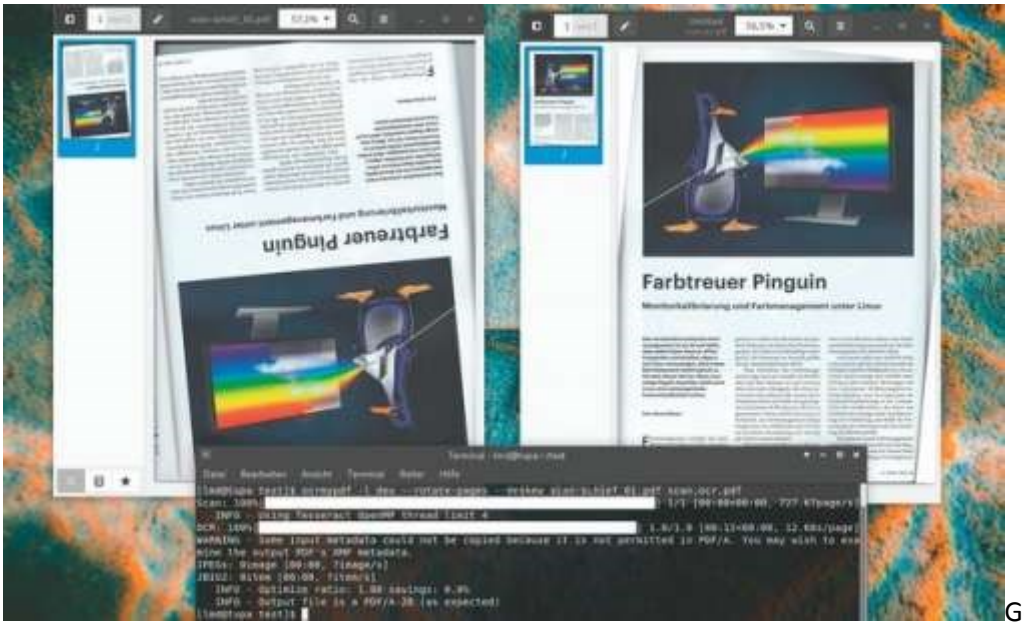
Um aus einem deutschsprachigen PDF ein durchsuchbares PDF/A-Dokument zu machen, dient der kurze Befehl:

```
ocrmypdf -l deu in.pdf out.pdf
```

Dabei ist in.pdf die einzulesende PDF-Datei, und out.pdf die durchsuchbare Ausgabe. -l deu verrät der OCR-Engine Tesseract, dass es sich um ein deutschsprachiges Dokument handelt. Wenn Sie diesen Parameter vergessen, fehlen in der Ausgabedatei die deutschen Umlaute, da Tesseract von einem englischsprachigen Dokument ausgeht. Ist ein Text zweisprachig, können Sie OCRmyPDF auch das mitteilen:

```
ocrmypdf -l deu+eng in.pdf out.pdf
```

Damit die Texterkennung in anderen Sprachen als Englisch und Deutsch klappt, müssen Sie die passenden Sprachpakete für Tesseract installieren.



Gedreht und schief: OCRmyPDF rotiert das Dokument, rückt es gerade und erkennt den Text.

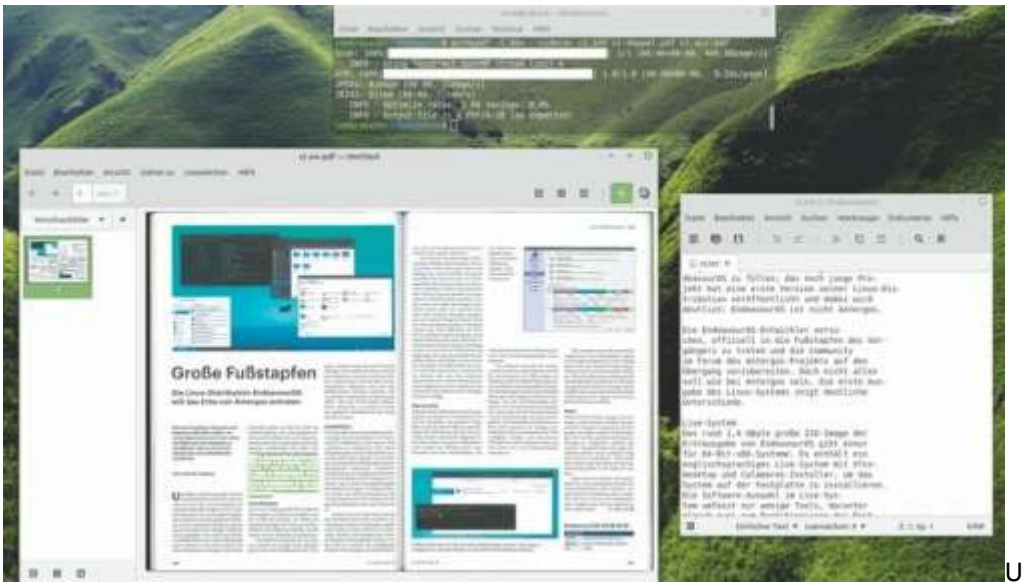
Stimmt die Seitenausrichtung des Ausgangsdokuments nicht, kann OCRmyPDF das während der Verarbeitung korrigieren, dazu geben Sie zusätzlich den Parameter --rotate-pages an. Das Tool erkennt dabei selbst, um wie viel Grad es eine Seite drehen muss. Ist das eingescannte Dokument dagegen nur leicht schief, behebt --deskew die Schräglage.

Bei Zeitungsartikeln oder Belegen kann es vorkommen, dass der Hintergrund durchscheint oder nicht ganz weiß ist, was die Texterkennung erschwert. Der Parameter --remove-background versucht, solche Störungen zu erkennen und vor der Texterkennung zu entfernen. Steht ein Dokument auf dem Kopf, ist dazu noch schief und hat einen störenden Hintergrund, können Sie diese Optionen miteinander kombinieren.

Text extrahieren

OCRmyPDF bietet viele weitere Optionen, die je nach Anwendungsbereich praktisch sein können. Geht es darum, Platz zu sparen, kann OCRmyPDF die Bilder eines PDFs verlustfrei optimieren. Mit dem Parameter --optimize 0 lässt sich die Optimierung ganz verhindern, verwendet man als Wert 1 oder 2 optimiert das Tool leicht beziehungsweise etwas stärker. Während die ersten beiden verlustfrei arbeiten, nimmt der Wert 3 eine niedrigere Bildqualität in Kauf. Eine besonders kleine PDF-Datei erzeugt daher

```
ocrmypdf --optimize 3 in.pdf out.pdf
```



Um den Text aus einem PDF zu extrahieren, geben Sie OCRmyPDF die Option **--sidecar** mit.

Geht es vor allem darum, den enthaltenen Text zu extrahieren, erzeugen Sie mit der Option --sidecar zusätzlich zum durchsuchbaren PDF auch eine Textdatei:

```
ocrmypdf -l deu --sidecar text.txt : in.pdf out.pdf
```

OCRmyPDF kann auch in eigene Python-Programme eingebunden werden, beispielsweise so:

```
import ocrmypdf

ocrmypdf.ocr('input.pdf', 'output.pdf',:  deskew=True)
```

Mehr darüber verrät die Dokumentation unter ct.de/ycb2.

Fazit

Zwar ist OCRmyPDF nicht perfekt, doch es erledigt seine Aufgabe wirklich gut. Abhängig von der Qualität des Ausgangsmaterials wird Fließtext zuverlässig in hoher Qualität erkannt und als durchsuchbares PDF/A-Dokument gespeichert. Mit einer Desktop-Suche wie Recoll finden sich so abgelegte Dokumente sehr leicht wieder. (lmd@ct.de)

Dokumentation mit vielen Optionen: ct.de/ycb2

Weiterführend: https://ocrmypdf.readthedocs.io/en/latest/cookbook.html

- Heftinhalt •