Creating a high-quality pdf of a printed book

In my field of nuclear physics, there are some important books which are more than forty years old, out of print, and generally difficult to obtain physically or digitally.

For example, "Electron Radial Wave Functions and Nuclear Beta-Decay" by H. Behrens and W. Buhring, published 1982.

I got my hands on a print copy of this book from a local University, and decided to scan it so that I may maintain a personal reference for my research.

In this note, I'm going to explain the steps I took to produce a high-quality PDF document using my iPhone camera, an iOS app, and some Linux software.

ML Scans from the iPhone

I used an app called "VFlat Scan - PDF Scanner" to create the initial scan of the book. I tried 3-4 similar apps, and found that this one worked best for scanning books. Some nice features include:

- Automatic scanning - it finds the document and automatically takes the picture once it does
- 2-page scanning - two pages of an open book are scanned automatically and seperated by page
- Competent de-scewing and color correction - it does a pretty good first pass at flattening the the skewed image (due to the pages not being perfectly flat)
- Good UI with easy file management and export options, including PDF and raw images
- Free! It was free to use and free of adverstisements

My book had about 600 pages, and it took me about an hour to scan the whole thing.

Collating and post-processing

After scanning all the pages, I had one PDF with the raw scans, and a folder with all of the pre-processed images. The PDF generated by the app is pretty good! Better than many book scans I've seen.

Still, it had a number of artifacts (shadows, etc.) and the file size was quite large. I also wanted nice things that modern PDFs offer:

- OCR (searchable)
- Table of contents (bookmarks)

- Meta data
- Small size (<25 MB)

So, I decided to try my hand at further processing the files on my Linux machine. After a lot of experimentation with different tools, I used the following programs:

- imagemagick
- unpaper
- img2pdf
- ocrmypdf
- ghostscript

The script I created combined them like so:

```
rm -r source pnm unpaper png
mkdir "source" "pnm" "unpaper" "png"
cp ../Archive/*.jpg ./source/
rename -e 's/\d+/sprintf("%05d",$&)/e' -- ./source/*.jpg

for f in source/Behrens*.jpg
do
    echo $f;
    fname=${f#*/} # remove prefix ending in /

    echo "...convert to pnm"
    convert "./source/$fname" -depth 4 -threshold 85% "./pnm/$fname.pnm" ;

    echo "...unpaper filters"
    unpaper --layout single  --output-pages 1 \
        --no-deskew \
        --no-border-scan -ms 100,100 \
        --overwrite "./pnm/$fname.pnm" "./unpaper/$fname.pnm" \
        > unpaper.out 2> unpaper.log;

    echo "...compress, convert to png"
    convert "./unpaper/$fname.pnm"   \
        "./png/$fname.png" ;
done

# Cover
convert ./source/'Behrens_and_Buhring - 00000.jpg' -depth 4 \
    -white-threshold 70% -channel B -threshold 5% \
    -depth 1 ./png/'Behrens_and_Buhring - 00000.jpg.png'


echo "concatinating final pdf"
img2pdf -r none ./png/*png -o Behrens_and_Buhring_png.pdf
```

```
echo "Performing OCR analysis"
ocrmypdf --rotate-pages-threshold 1000 Behrens_and_Buhring_png.pdf Behrens_and_Buhring_ocr.p

echo "Formatting document and meta data"
gs -sDEVICE=pdfwrite -o Behrens_and_Buhring_final.pdf -sPAPERSIZE=a5 \
       -dFIXEDMEDIA -dPDFFitPage -dCompatibilityLevel=1.7 \
       -dAutoRotatePages=/None Behrens_and_Buhring_ocr.pdf bb.pdfmark
```

Results

Scanned text