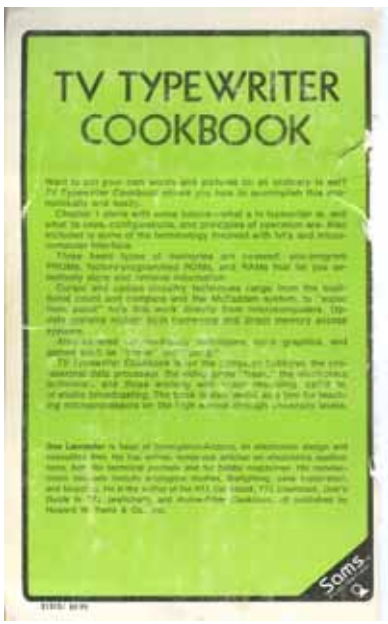
Restoring Faded or Scuffed Text for Web Distibution

Don Lancaster
Synergetics, Box 809, Thatcher, AZ 85552

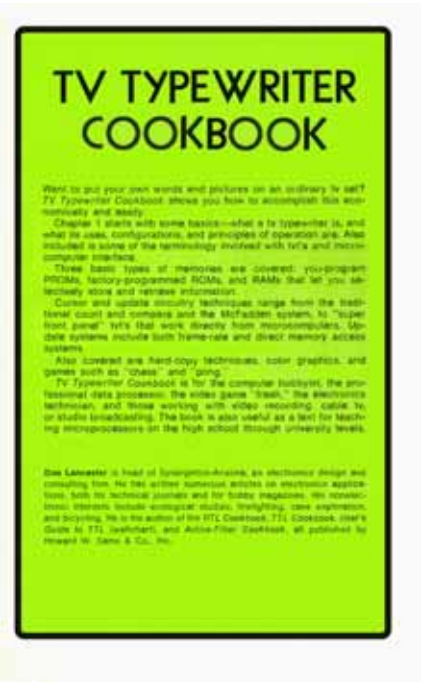
http://www.tinaja.com
don@tinaja.com
(928) 428-4073

We looked at much that is involved in remastering a classic precyber book for web eBook distribution back in GuruGram #115. One of the problems that may crop up involves restoring a text page that is badly scuffed, faded, torn, or actually has missing parts. This is most likely to crop up with a back cover, but remains a more general problem.

To fully restore these pages, we can use a technique I might call bitmap glyph remastering. The process takes little skill and less in the way of custom software, but is tedious and time intensive. Thus, finding a better original or even doing a total relayout might be competitive options. At any rate, here is an original back cover with some obvious problems...



You can click expand the image to see it full size. And here are the results of our remastering ...



Here are some details on the restoration process...

1. Do a quality high resolution scan.

Rescan the artwork to the highest RGB resolution possible. Preferably 600 DPI and saved as a .BMP bitmap. Be sure your scanner is exceptionally clean. Carefully align the page on the scanner. Do so in the upper rightmost corner. If needed, do a deskewing with IrfanView which lets you adjust the text in one tenth degree intervals. Get into Paint and crop to size by changing Image ---> Attributes. The larger your image at this point, the better. A small text letter height of 30 pixels is not unreasonable if you want the best possible results.

2. Isolate a reference background.

Pick a "typical" color background and extract a swatch of it. Make the swatch as uniform as possible. Later on, you will have the ability to HSB adjust the color, but for now, it is important that the swatch has the best "average" color over as much of the page as possible.

It is best if the swatch is slightly mottled. This will minimize any JPEG artifacts later on, compared to the "edge ghosting" you might get with a solid color. Save your swatch in a separate Paint alphabet stash.

3. Isolate the best lowercase "e".

Starting with the smallest font used, find the best lowercase "e" you can on the page and improve it as much as possible. Sometimes symmetry tricks or adopting portions of other characters can help. Allow slight to modest variations in font style if it gives you a better overall character "look". Next, replace any "speckle" pixels that are too dark, too light, or otherwise jarring. You will usually want to increase the contrast between lettering and background.

Place the character in a rectangular background swatch that is a few pixels wider and "half the ledding" higher than the character. It is important that the swatch width not exceed the minimum expected kerning between characters. Save the character to a Paint alphabet stash.

4. Cut and paste all of the lowercase "e" characters.

Be very careful to vertically and horizontally center each replacement to one pixel accuracy. It is important to replace each and every "e", unless some originals are exceptionally good. Then repeat for the other characters. Perhaps by using an ETAOIN SHRDLU sequence in honor of the famous fontographer. For efficiency, be on the lookout for reusable character pairs, triplets, or even whole words.

Repeat the isolate-->repair-->replace sequence for all needed fonts. Do so in order of increasing font size.

5. Repair the rest of the non-text artwork.

At this point, there should be no "old" text anywhere on your page. Replace the remaining color backgrounds with copies of your swatch. Borders and edges can then be dealt with by isolating a good portion, improving it, and then "chasing" it as needed both horizontally and vertically. Always aim for a one pixel accuracy in your alignments.

Should any text end up totally missing, fake it as best you can by use of your "improved" alphabet. At this point, you may want to enhance the overall HSB by way of changes in gamma, color balance, contrast, or brightness. Perhaps by using Imageviewer32 or IrfanView.

6. Send the new bitmap to Acrobat Pro.

If none of your intermediate software has changed the stored resolution in your bitmap header, your artwork should end up normal sized in your .PDF file. Should it be oversize, get into IrFanView or another bitmap manipulator and change the header resolution. It is best to let Acrobat Pro do any downsampling using Bicubic Interpolation rather than random resizing. It is also best to give Acrobat Pro the largest possible bitmap to work with.

You may want to think twice about using ClearScan text OCR recognition here. Normally ClearScan does an exceptionally good job with black and white input. But it may choke on color or pastels. It is also not all that great in recognizing exceptionally sharp cornered fonts, such as our example title. The tradeoff is a possible reduction in font quality or recognition against the ability to search text and possibly smaller file sizes.

Usually the size and searchibility benefits of ClearScan will dominate, so try it and see if it works well for you. We have included it in the final click demo.

Should your final file sizes be too large, have Acrobat do the usual downsampling and web delivery optimization.

For More Help

The eBook on which our example can be found appears here. And this related GuruGram on improving halftones here. And consulting and custom remapping services here. Or you can email me for details.