2D deconvolution

Deblurring images

This page contains the computational Matlab files related to the book
Linear and Nonlinear Inverse Problems with Practical Applications
written by Jennifer Mueller and Samuli Siltanen and published by SIAM in 2012.

You can order the book at the SIAM webshop.

Go to master page

The routines below are for sharpening blurred and noisy images. We thank Sanna Tyrvainen for her help in preparing these examples.

Tikhonov regularization

Download first the file sumimage.m. Then create a blurred and noisy image using

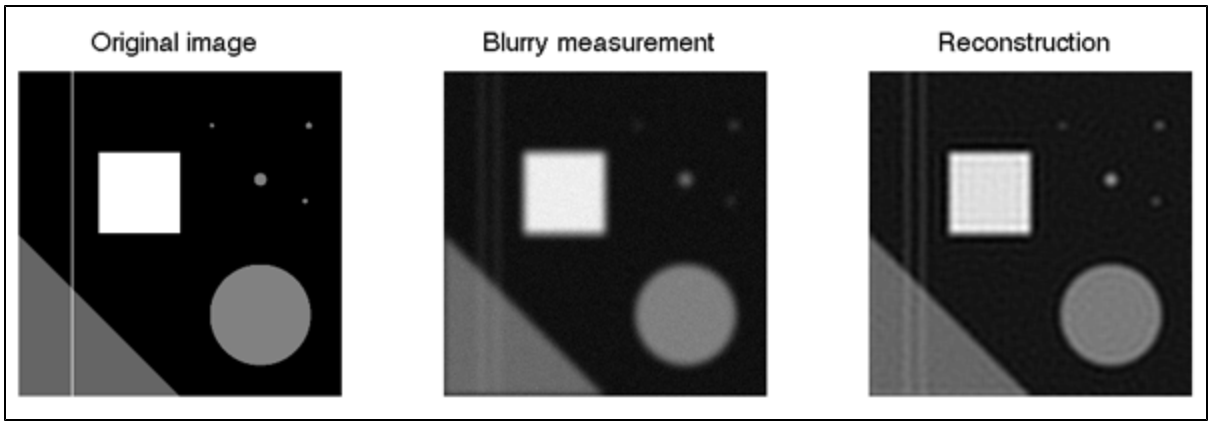
DB1_data_comp.m

A data file called img_data.mat will be created. To avoid inverse crime, the script first creates a bigger picture, blurs it using a point spread function (PSF), adds noise and then shrinks both the picture and the PSF to a fourth of their original size. You can change the parameter *noiselevel* in the above script. The default level in this example is 0.04.

The image is too big (512x512 pixels) to be deconvoluted with straightforward matrix-based methods. So will use an iterative and matrix-free method based on conjugate gradients:

DB2_Tikhonov.m

You should see something like this:



The reconstruction has some artefacts, but the method has also sharpened some of the edges and brought out smaller details.

The regularization parameter is denoted by *delta* in the script, and modifying it will change the result. The default value in this example is 0.002. The number of iteration rounds is limited by the parameter *K*. It is set to 40. If you are interested in results of different rounds you can change it to get the reconstruction of *K*'s round.

The script uses a Jacobi (diagonal) preconditioner. Different preconditioning methods can be used to speed up the iteration further.

Approximate total variation deconvolution

Now we will try a total variation technique for deblurring this image:

Please download the above image using this link: Edelweiss_noisy.png.

The Matlab script for a Barzilain-Borwein optimization approach for approximate total variation is

DB3_aTV.m

You will also need the following files:

Edelweiss.png, db_aTV_feval.m, db_aTV_fgrad.m, db_brent.m, db_f1dim.m, db_linmin.m, db_mnbrak.m, lh_db.m, lhgrad_db.m, p_aTV.m, pgrad_aTV.m

The first iteration step is taken using a line search. The line search methods above have been adapted to Matlab from the book *Numerical recipes in C*.

As a result, you should get something like this:

Above left: noisy and blurred data; middle: reconstruction; right: original sharp image.