TV-Based Deconvolution for Medical Imaging

Pascal Getreuer, November 14, 2009

Here we discuss the extension of the semi-implicit method for TV denoising to TV deconvolution using Rician and Poisson noise models.

Notations Let $\mathcal{N}_{i,j,k}$ denote the six axial neighbors of voxel (i,j,k) in three dimensions, and define

$$g^\ell_{i,j,k} = \Big[\epsilon + \sum_{n\in\mathcal{N}_{i,j,k}} (u^\ell_n - u^\ell_{i,j,k})^2\Big]^{-1/2}$$

where superscript ℓ denotes iteration. In the following, λ is the weight on the fidelity term, σ is a parameter of the Rician distribution related to the standard deviation, and $\gamma = \lambda/\sigma^2$.

Rician We use a semi-implicit scheme. For Rician noise, it is

$$u^{\ell+1}_{i,j,k} = \frac{u^\ell_{i,j,k} + dt\big(\sum_{n\in\mathcal{N}_{i,j,k}} g^\ell_n u^\ell_n + \gamma K^* \Big[-Ku^\ell + \frac{I_1(fKu^\ell/\sigma^2)}{I_0(fKu^\ell/\sigma^2)} f\Big]_{i,j,k}\big)}{1 + dt\big(\sum_{n\in\mathcal{N}_{i,j,k}} g^\ell_n\big)}. \tag{1}$$

Unfortunately, the scheme must have dt sufficiently small for stability. It is not clear how choose dt a priori so that the scheme is gauranteed to be stable. The largest possible dt depends on the image, on K, and on the parameters. It seems that smaller dt is necessary for larger vaues of λ and for blurs K with greater support.

Rician deconvolution is implemented as a C/MEX function in `riciandeconv3mx.c`, which is called from MATLAB as

```
u = riciandeconv3mx(f,K,sigma,lambda,NumIter,dt)
```

For simplicity, the routine is limited to isotropic Gaussian blurs. The parameters are

`f`	input volumetric image (3D double array)
`K`	standard deviation of the Gaussian blur in voxels
`sigma`	parameter σ of the Rician noise
`lambda`	regularization parameter
`NumIter`	number of method iterations
`dt`	timestep parameter

Parameter `lambda` balances the accuracy of deconvolution vs. denoising strength: smaller `lambda` implies stronger denoising but at the cost of deconvolution accuracy.

A substantial amount of computation time is spent evaluating convolutions (about 50% for the example in Figure 1), which is one motivation for restricting to only Gaussian blurs. Gaussian convolution is implemented using the recursive filtering algorithm of Alvarez and Mazorra [1] in `gaussianblur.c`.

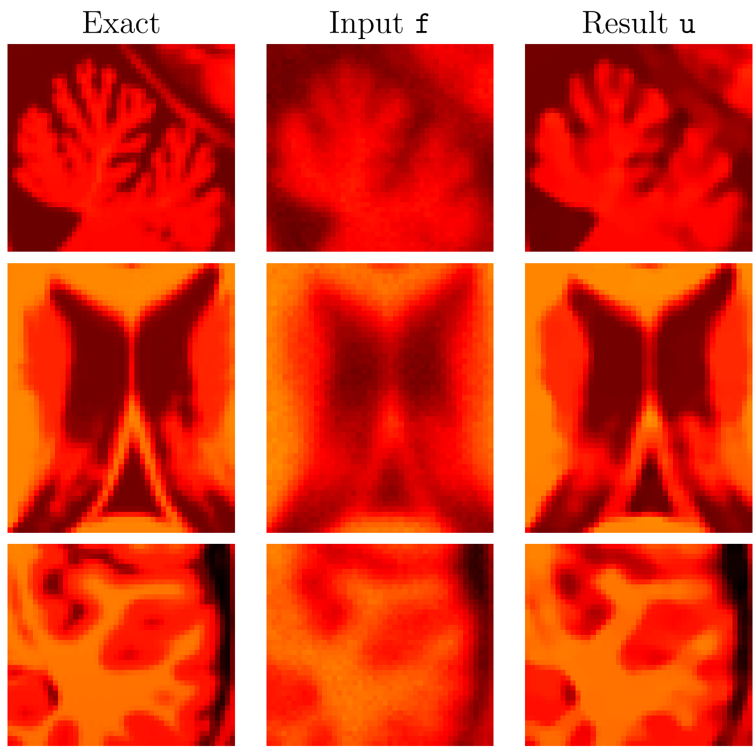

Figure 1: Experiment with TV-based Rician deconvolution. *Left column:* a simulated MRI brian volume of size $217 \times 181 \times 181$ was taken as the exact data. *Middle column:* the exact data was blurred ($K = 1.8$) and corrupted with Rician noise ($\sigma = 0.008$) to produce the input data. *Right column:* the restored result was computed using `riciandeconv3mx` (`lambda` $= 0.085$, `NumIter` $= 100$, `dt` $= 0.001$) with a computation time of 179 s.

For correct results, it is critical that `dt` is chosen small enough for stability. Unfortunately, how small `dt` must be depends on the image and the other parameters, so experimentation is required to set `dt` appropriately.

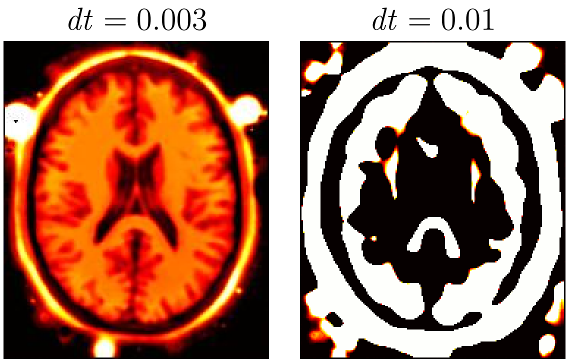

Figure 2: Examples of instability where dt is too large, computed with `NumIter` $= 5$. The maximum stable dt depends on the image and parameters.

Poisson Following [2], we can derive a similar semi-implicit scheme for Poisson noise as

$$\min_u \int |\nabla u| \; dx + \lambda \int (Ku - f \log Ku) \; dx$$

$$\partial_t u = \operatorname{div}\left(\tfrac{\nabla u}{|\nabla u|}\right) + \lambda K^* \left(\frac{f}{Ku} - 1 \right)$$

$$\frac{u^{\ell+1}_{i,j,k} - u^{\ell}_{i,j,k}}{dt} = \sum_{n \in \mathcal{N}_{i,j,k}} g^{\ell}_{i,j,k}(u^{\ell}_n - u^{\ell+1}_{i,j,k}) + \lambda K^* \left(\frac{f}{Ku^{\ell}} - 1 \right)$$

$$u^{\ell+1}_{i,j,k} = \frac{u^{\ell}_{i,j,k} + dt\left(\sum_{n \in \mathcal{N}_{i,j,k}} g^{\ell}_n u^{\ell}_n + \lambda K^* \left(\frac{f}{Ku^{\ell}} - 1\right)\right)}{1 + dt\left(\sum_{n \in \mathcal{N}_{i,j,k}} g^{\ell}_n\right)}.$$

Poisson deconvolution is implemented as a C/MEX function in `poissondeconv3mx.c`, and is called from Matlab as

```
u = poissondeconv3mx(f,K,lambda,NumIter,dt)
```

The parameters are the same as `riciandeconv3mx`, except there is no `sigma` parameter for the Poisson model. As with `riciandeconv3mx`, it is important to choose `dt` sufficiently small for stability, and the maximum stable `dt` depends on the image and the parameters.

References

[1] L. Alvarez and L. Mazorra. "Signal and image restoration using shock filters and anisotropic diffusion" *SIAM Journal on Numerical Analysis*, 1994.

[2] T. Le, R. Chartrand and T. Asaki. "A Variational Approach to Constructing Images Corrupted by Poisson Noise," *JMIV*, vol. 27(3), pp. 257–263, 2007.

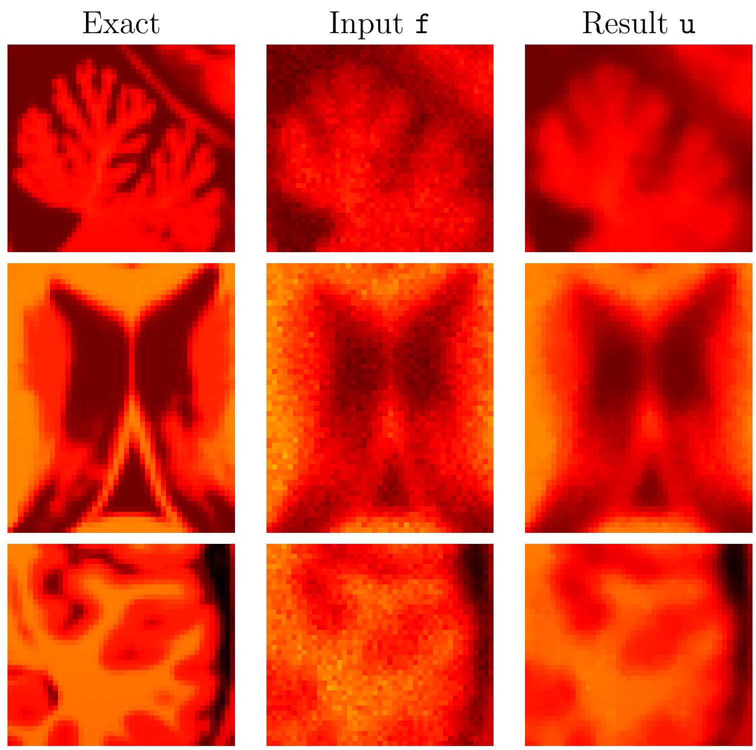

Figure 3: Experiment with TV-based Poisson deconvolution. *Left column:* a simulated MRI brian volume of size $217 \times 181 \times 181$ was taken as the exact data. *Middle column:* the exact data was blurred ($K = 1.8$) and corrupted with Poisson noise to produce the input data. *Right column:* the restored result was computed using `poissondeconv3mx` (`lambda` $= 15$, `NumIter` $= 200$, `dt` $= 8 \times 10^{-5}$) with a computation time of 340 s.