A User Guide for the Spectral Total Variation Code (Version 1.0)

Joon Hee Choi (choi240@purdue.edu)

Introduction

This document provides instructions on how to use the spectral total variation denoising algorithm on MATLAB. Please read Chien-Sheng Liao, Joon Hee Choi, Delong Zhang, Stanley H. Chan and Ji-Xin Cheng, "Denoising Stimulated Raman Spectroscopic Images by Total Variation Minimization," *Journal of Physical Chemistry C*, Jul. 2015. to learn more about this algorithm. The following sections show how to download the spectral total variation (STV) code, write a new code to obtain the output image, and run the generated code.

1 Download and Install File

Download the `spectral_total_variation.zip` file to your folder and upzip the file. You should be able to see a README.txt file and two folders. The two folders are `code`, which has the .m files, and `data`, which has a few sample image files.

2 Open MATLAB

First, open MATLAB and click the `New Script` button on the upper left-hand side of the MATLAB window (Figure 1). Then, a new editor window will open (Figure 2).

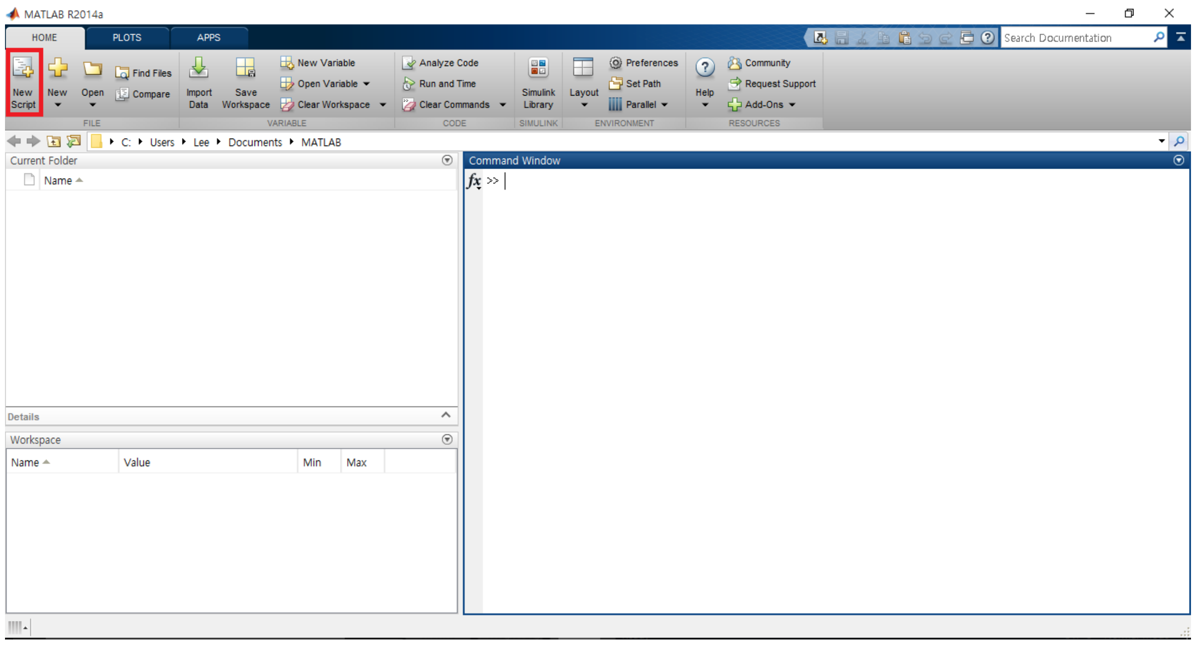


Figure 1: "New Script" button on the MATLAB Window

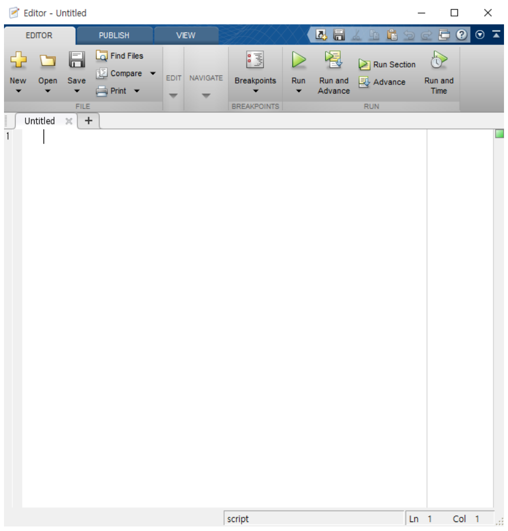


Figure 2: Editor Window on MATLAB

3 Write a New Code

To write a new code in the editor window, use the commands in Sections 3.1 - 3.4.

3.1 Add Path

For MATLAB to recognize the directories and commands, you need to enter the path of the folder you want to call. Since STV uses the files in the folder `code/spectral_tv/`, enter the path below in the editor window:

```
addpath(genpath('./spectral_tv/'));
```

3.2 Load Data

Depending on the file format, there are two different methods of loading your hyperspectral image, as shown in Sections 3.2.1 and 3.2.2.

3.2.1 .TIFF Data

If the hyperspectral image is stored as a .tif or .tiff file, which can store several images in a file, enter the following command:

```
hyper_noisy = read_hyperdata('FILENAME');
```

FILENAME should include the paths (folders) to the image file. For instance, if an image file `aaa.tif` is stored in a folder named `xx/yy/`, FILENAME should be `xx/yy/aaa.tif`.

3.2.2 .JPG, .BMP, and .PNG Data

If the hyperspectral image is stored in multiple files, such as .jpg, .bmp or .png, enter the following commands:

```
hyper_noisy(:,:,1) = rgb2gray(im2double(imread('FILENAME1.jpg')));
hyper_noisy(:,:,2) = rgb2gray(im2double(imread('FILENAME2.jpg')));
hyper_noisy(:,:,3) = rgb2gray(im2double(imread('FILENAME3.jpg')));
```

If you have 10 image files, you need to implement the command 10 times. However, if you have multiple image files named, for example, `bird1.jpg` - `bird50.jpg`, where the file names are the same except for the numbers, you can load the images using for-loop:

```
for i=1:50
    filename = ['bird', num2str(i, '%02g'), '.jpg'];
    hyper_noisy(:,:,i) = rgb2gray(im2double(imread(filename)));
end
```

3.3 STV Algorithm

To implement the denoising algorithm, you need two inputs, the image data and a parameter. The image data should be a hyperspectral (3-D) noisy image. The parameter should be a struct class, which is a set of fields (i.e., variables). Fields within a struct class can take on various formats, such as numbers, vectors, strings, etc. As shown in Table 1, a field `tv_method` within a struct class `opts` should be written as `opts.tv_method` (to learn more about classes on MATLAB, please click here). Table 1 displays a full list of the fields that we use for the input parameter (please see [1] for a detailed description of each field).

Class.Fields	Description	Default values
`opts.tv_method`	total variation method ('aniso' or 'iso')	'aniso'
`opts.rho_r`	initial penalty parameter for ||u-Df||	2
`opts.rho_o`	initial penalty parameter for ||f-g-r||	50
`opts.beta`	regularization parameter [a b c] for weighted TV norm	[1 1 0]
`opts.gamma`	update constant for rho_r	2
`opts.max_itr`	maximum iteration	20
`opts.alpha`	constant that determines constraint violation	0.7
`opts.tol`	tolerance level on relative change	1e-3
`opts.print`	print screen option	false
`opts.f`	initial f	g
`opts.y1`	initial y1	0
`opts.y2`	initial y2	0
`opts.y3`	initial y3	0
`opts.z`	initial z	0

Table 1: Struct class (opts) and its fields

STV will use the default values unless you set a specific value for each field. We recommend using the default values except for `opts.beta`. The default values for `opts.beta` are set to denoise a 2-D image. Since we are denoising a 3-D image, we need to set the field values for the rows, columns and frames. We empirically found that $[1, 1, 0.1]$ works well for most images. For example,

```
opts.beta    = [1 1 0.1];
out_stv      = spectral_tv(hyper_noisy, opts);
```

The output of the STV algorithm, `out_stv`, is also a struct class. The output image is stored in the field `out_stv.f`, and the estimated noise level of each frame is stored in the field `out_stv.sigma`.

3.4 Display Results

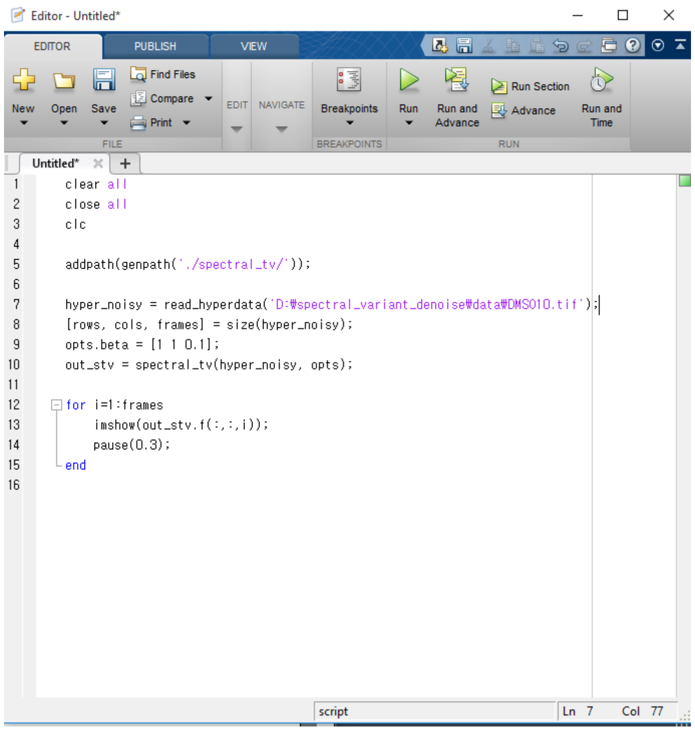
To display the denoised image of the *i*-th frame, you need the following command:

```
imshow(out_stv.f(:,:,i));
```

The output image is displayed in Figure 6.

3.5 An Example of a New Code

Your new code should look something like Figure 3.

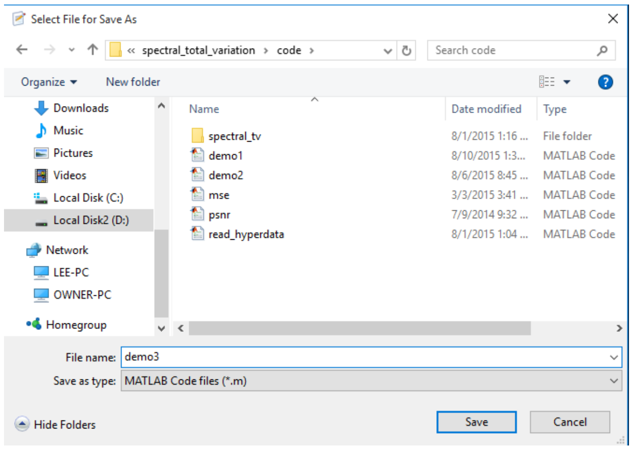


```
clear all
close all
clc

addpath(genpath('./spectral_tv/'));

hyper_noisy = read_hyperdata('D:₩spectral_variant_denoise₩data₩DMSO10.tif');
[rows, cols, frames] = size(hyper_noisy);
opts.beta = [1 1 0.1];
out_stv = spectral_tv(hyper_noisy, opts);

for i=1:frames
    imshow(out_stv.f(:,:,i));
    pause(0.3);
end
```

Figure 3: Example code

4 Save and Run

Click the Save button in the editor window and save the code in the folder `spectral_total_variation/code` (Figure 4).



Select File for Save As

« spectral_total_variation › code ›

Name	Date modified	Type
spectral_tv	8/1/2015 1:16 ...	File folder
demo1	8/10/2015 1:3...	MATLAB Code
demo2	8/6/2015 8:45 ...	MATLAB Code
mse	3/3/2015 3:41 ...	MATLAB Code
psnr	7/9/2014 9:32 ...	MATLAB Code
read_hyperdata	8/1/2015 1:04 ...	MATLAB Code

File name: demo3
Save as type: MATLAB Code files (*.m)

Figure 4: Saving the new code on editor window

Next, run the generated code by clicking on `Run` in the editor window or by pushing `F5`.

5 Noise Estimation

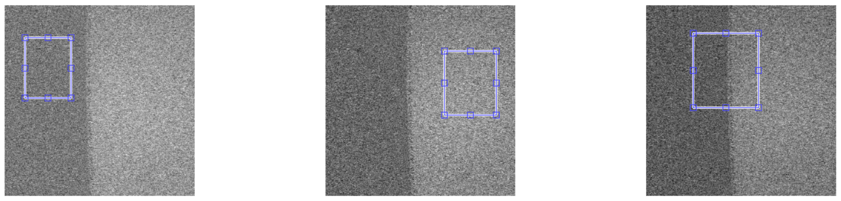
In normal situations, we do not know the noise level. Therefore, we need to estimate the noise level. Estimating the noise level can be done by tracing the variance in a uniform region. We built a simple graphics user interface to allow users to pick and choose a region where they would like to estimate the noise level. The noise level is estimated via the standard deviation of the region. Letting Ω be the region selected by the user, we compute the noise standard deviation as

$$\sigma = \left(\frac{1}{|\Omega|} \sum_{i \in \Omega} (f_i - \bar{f})^2 \right)^{1/2}, \tag{1}$$

where $\bar{f}$ is the mean of the pixels in Ω.

5.1 Graphics user interface for cropping noise region

When a new window pops up after you run the generated code, select a uniform region in the window as shown below.



Correct Correct Incorrect

The noise standard deviation within the cropped region will be estimated for all spectral wavelengths. If you do not select a uniform region, the noise standard deviation may be incorrectly estimated.

6 Results

Once a region is selected, MATLAB will produce one denoised image per frame for all frames. Therefore, if there are 50 frames, 50 denoised images will be displayed. Figure 6 shows a denoised image of a frame.

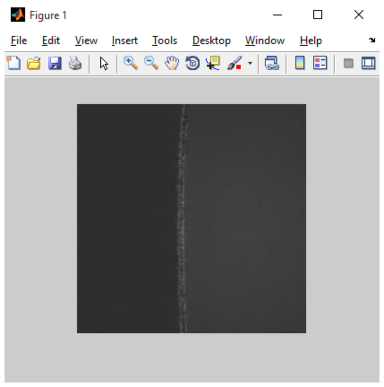


Figure 6: Denoised image output

7 Additional Commands

If you know the true image or the true noise level of each frame, you can add commands in Sections 7.1 or 7.2 to your code (Figure 3).

7.1 Performance Check

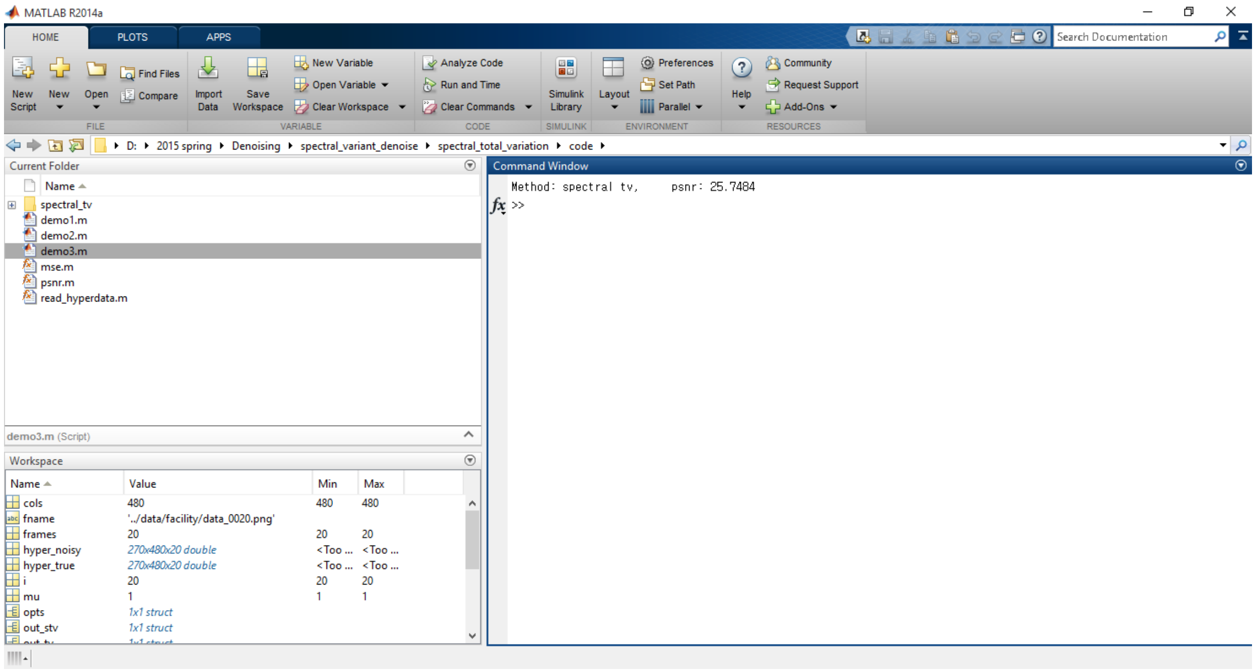
If you already know the true image, you can compare the denoised image with the true image using common metrics such as PSNR or MSE. For this comparison, load the true image in the same manner as above. Then, for PSNR, add

```
hyper_true  = read_hyperdata('FILENAME');
psnr_stv    = psnr(hyper_true, out_stv.f);
fprintf('Method: spectral tv, \t psnr: %6.4f\n', psnr_stv);
```

and for MSE,

```
hyper_true  = read_hyperdata('FILENAME');
mse_stv     = mse(hyper_true, out_stv.f);
fprintf('Method: spectral tv, \t mse: %6.4f\n', mse_stv);
```

The above commands will display the values of PSNR or MSE in the MATLAB window in addition to the denoised output images for each frame (Figure 7).



Current Folder:

```
Name
spectral_tv
demo1.m
demo2.m
demo3.m
mse.m
psnr.m
read_hyperdata.m
```

Command Window:

```
Method: spectral tv,     psnr: 25.7484
>>
```

Workspace:

Name	Value	Min	Max
cols	480	480	480
fname	'../data/facility/data_0020.png'		
frames	20	20	20
hyper_noisy	270x480x20 double	<Too ...	<Too ...
hyper_true	270x480x20 double	<Too ...	<Too ...
i	20	20	20
mu	1	1	1
opts	1x1 struct		
out_stv	1x1 struct		

Figure 7: Performance output

7.2 Checking the estimated noise level

If you already know the true noise level for each frame, you can compare the estimated and true noise levels. An example is shown below:

```
sigma_true  = [0.2, 0.1, 0.3, 0.5, 0.4];
sigma_est   = out_stv.sigma;
figure;
plot(sigma_true, 'LineWidth', 2, 'Color', 'g');
hold on;
plot(sigma_est, 'LineWidth', 2, 'Color', 'r');
hold off;
xlabel('frame');
```

```
ylabel('noise standard deviation');
legend('True', 'Estimated', 'Location', 'best');
```

where the values of sigma_true should be determined by the user. The above commands will display a graph in a new window in addition to the denoised output images for each frame (Figure 8).

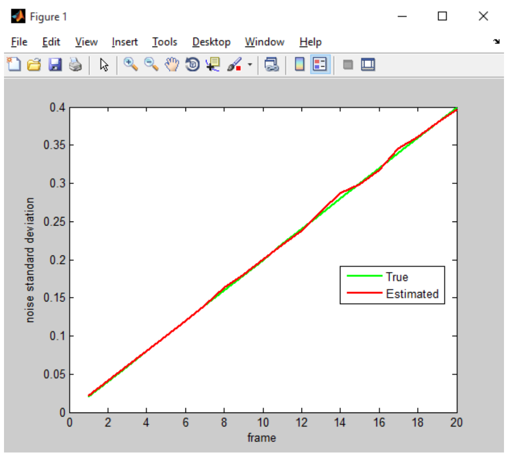


Figure 8: The output of the estimated noise level

8 New Code Generation Flowchart

Figure 9 shows the aforementioned steps in generating a new code in a flowchart.

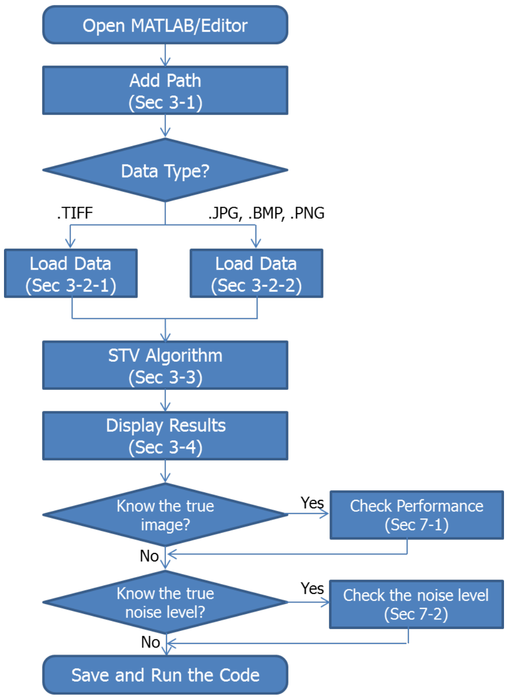


Figure 9: New code generation flowchart

Questions and Comments

See the demo1.m and demo2.m files for denoising synthetic data. For additional questions, please email Joon Hee Choi (choi240@purdue.edu).

References

[1] Stanley H. Chan, Ramsin Khoshabeh, Kristofor B. Gibson, Philip E. Gill and Truong Q. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE Trans. Image Process.*, vol. 20, issue 11, pp.3097-3111, Nov. 2011.