MATLAB User Guide for Adaptive Image Denoising by Targeted Databases

Enming Luo, Stanley H. Chan, and Truong Q. Nguyen,

This MATLAB user guide presents the usage of the MATLAB function "TID" (short for Targeted Image Denoising) accompanied within the papers [1, 2]:

Targeted Image Denoising:
[y_den, psnr, ssim] = TID(y, z, sigma, data)

Table 1: user interface

I. Introduction

In [1, 2], we propose an optimal denoising filter that utilizes a targeted image database to denoise a noisy image of interest. The design of the optimal denoising filter is posed as: Given a noisy patch $\boldsymbol{q} \in \mathbb{R}^d$, and assuming that the noise is i.i.d. Gaussian with zero mean and variance σ^2, we want to find a basis matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d] \in \mathbb{R}^{d\times d}$ and a diagonal spectral matrix $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_d\} \in \mathbb{R}^{d\times d}$ such that the estimate $\widehat{\boldsymbol{p}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q}$ has the minimum mean squared error (MSE) compared to the ground truth $\boldsymbol{p} \in \mathbb{R}^d$. That is, we want to solve the optimization

$$(\boldsymbol{U}, \boldsymbol{\Lambda}) = \underset{\boldsymbol{U},\boldsymbol{\Lambda}}{\arg\min}\ \mathbb{E}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q} - \boldsymbol{p}\right\|_2^2\right], \tag{1}$$

subject to the constraint that $\boldsymbol{U}$ is an orthonormal matrix.

In [1, 2], to denoise a noisy patch $\boldsymbol{q}$, the proposed algorithm first finds patches from a targeted database that is relevant to the noisy image. Then given the found clean patches $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k$, the optimal basis function $\boldsymbol{U}$ of the denoising filter is determined by solving a group sparsity minimization problem, and the optimal spectral matrix of the denoising filter is determined by considering a localized patch prior. The overall algorithm is summarized in Algorithm 1.

Algorithm 1 Proposed Algorithm

Input: Noisy patch $\boldsymbol{q}$, noise variance σ^2, and clean reference patches $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k$
Output: Estimate $\widehat{\boldsymbol{p}}$
Learn $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$

- Form data matrix $\boldsymbol{P}$ and weight matrix $\boldsymbol{W}$
- Compute eigen-decomposition $[\boldsymbol{U}, \boldsymbol{S}] = \mathrm{eig}(\boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^T)$
- Compute $\boldsymbol{\Lambda} = \left(\mathrm{diag}(\boldsymbol{S} + \sigma^2\boldsymbol{I})\right)^{-1}\mathrm{diag}(\boldsymbol{S})$

Denoise: $\widehat{\boldsymbol{p}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q}$.

We divide the entire image into over-lapped patches and denoise each patch using the above proposed algorithm. The final denoised image is an average of all the estimates for each pixel.

E. Luo and T. Nguyen are with Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093, USA. Emails: eluo@ucsd.edu and nguyent@ece.ucsd.edu

S. Chan is with School of Electrical and Computer Engineering, and Department of Statistics, Purdue University, West Lafayette, IN 47907, USA. Email: stanleychan@purdue.edu

II. Usage

After introducing the problem setup and proposed algorithm, in this section we present the usage of the MATLAB function "TID". The user interface of "TID" is:

[y_den, psnr, ssim] = TID(y, z, sigma, data)

The input and output parameters for this function are as follows

```
%Input parameters:
%                y: clean image (used for computing PSNR and SSIM)
%                z: noisy image
%            sigma: noise standard deviation
%             data: a structure containing the following external database and other parameters
%                 database: external database
%                    y_est: image used for database patch matching
%                       N1: N1xN1 is the reference block size (default: 8)
%                    Nstep: sliding step to process the next reference block (default: 6)
%                       Ns: NsxNs is the search window size for patch matching (default: 101)
%                tau_match: threshold for patch similarity (default: 2*N1^2*sigma^2)
%                       N2: maximum number of similar patches for each database image (default: 20)
%                       N3: maximum number of similar patches for the entire database (default: 40)
%Output parameters:
%            y_den: denoised_image
%             psnr: PSNR value
%             ssim: SSIM value
```

A. Input Variables

- y: Clean image. It is a gray-scaled clean image and used for computing PSNR and SSIM.
- z: Noisy image.
- sigma: Noise standard deviation.
- data: A structure of options (See below).

B. Option Fields

- data.database: External database. It consists of similar but non-identical example images for the noisy image of interest.
- data.y_est: Image used for database patch matching.
- data.N1: N1xN1 is the reference block size (default: 8).
- data.Nstep: Sliding step to process the next reference block (default: 6).
- data.Ns: NsxNs is the search window size for patch matching (default: 101)
- data.tau_match: Threshold for patch similarity (default: $2 * \text{N1}^2 * \text{sigma}^2$)
- data.N2: Maximum number of similar patches for each database image (default: 20)
- data.N3: Maximum number of similar patches for the entire database (default: 40)

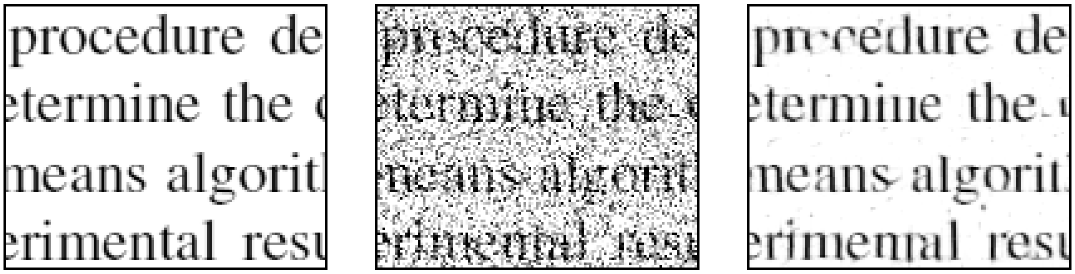
C. Out Variables

- y_den: Denoised image.
- psnr: PSNR value.
- ssim: SSIM value.

III. Example

An example of using this function is shown as follows and users can refer to "demo.m" for detailed information. In this file, for comparison purposes, we also provide the codes for some state-of-the-art denoising methods including BM3D, BM3D-PCA, LPG-PCA, and NLM. All these methods are re-implemented and modified by us such that patch search is performed over the targeted external databases. For comparison with internal BM3D denoising, one can download the original code provided by the BM3D authors[1] [3]. The text database is prepared by us from randomly selected documents. The face database is obtained from the FEI face dataset[2] [4].

Figure 1 shows the clean image, noisy image and denoised image using the proposed TID for the text image denoising.



(a) clean (b) noisy $\sigma = 100$ (c) TID: 22.20 dB (0.9069)

Fig. 1: Denoising text images: Visual comparison and objective comparison (PSNR and SSIM in the parenthesis). The test image size is of 127×104, 9 other clean images of similar sizes are used as the external database.

IV. Copyright

References

[1] E. Luo, S. H. Chan, and T. Q. Nguyen, "Image denoising by targeted external databases," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Process. (ICASSP '14)*, pp. 2469–2473, May 2014.

[2] E. Luo, S. H. Chan, and T. Q. Nguyen, "Adaptive image denoising by targeted databases," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2167–2181, Jul. 2015.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[4] C. E. Thomaz and G. A. Giraldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image and Vision Computing*, vol. 28, no. 6, pp. 902 – 913, 2010.

[1] http://www.cs.tut.fi/~foi/GCF-BM3D/

[2] http://fei.edu.br/ cet/facedatabase.html