Matlab Imaging Algorithms: Image Reconstruction, Restoration, and Alignment, with a Focus in Tomography. (Version 2.2)

Toby Sanders
email: toby.sanders@asu.edu
School of Math and Stat Sciences,
Arizona State University, Tempe, AZ.

Abstract

This document is a guide for several MATLAB imaging code, which can be used for image reconstruction, denoising, alignment, etc. There is a general focus on tomographic reconstruction and alignment, as well as ℓ_1 regularization algorithms.

1 Solvers for Signal, Image, and Volume Reconstruction

1.1 Higher order TV based ℓ_1 Regularization

- **Code name:** HOTV3D
- **Demos:** see demo_L1optimization_simple.m, demo_tomo.m, demo_MHOTV.m, demo_inpaint.m, and demo_1D_L1.m
- **References:** [6, 1, 9, 4, 7]
- **Description:** An iterative solver for higher order total variation ℓ_1 regularization minimization for inverse problems and denoising. MATLAB comments:

```
% Modifications by Toby Sanders @ASU
% School of Math & Stat Sciences
% 08/24/2016
%
%
% This code has been modified to solve l1 penalty problems with
% higher order TV operators.  Several small bugs and notation
% changes have been made as well.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         Problem Description       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [U, out] = HOTV3D(A,b,n,opts)
%
% Motivation is to find:
%
%               min_f { mu/2*||Au - b||_2^2 + ||D^k u||_1 }
%
```

```
% where D^k is kth order finite difference.
% Multiscale finite differences D^k can also be used.
% To see how to modify these settings read the file "check_HOTV_opts.m"
%
% The problem is modified using variable splitting
% and this algorithm solves:
%
%      min_{u,w} {mu/2 ||Au - b||_2^2 + beta/2 ||D^k u - w ||_2^2
%               + ||w||_1 - (delta , Au - b ) - (sigma , D^k u - w) }
%
% delta and sigma are Lagrange multipliers
% Algorithm uses alternating direction minimization over f and w.
%
%
% This algorithm was originally authored by Chengbo Li at Rice University
% as a TV solver called TVAL3.
% original code and description can be found here:
% http://www.caam.rice.edu/~optimization/L1/TVAL3/
%
% Inputs:
%   A: matrix operator as either a matrix or function handle
%   b: data values in vector form
%   n: image/ signal dimensions in vector format
%   opts: structure containing input parameters,
%       see function check_HOTV_opts.m for these
%
%
% Outputs:
%   U: reconstructed signal
%   out: output numerics
```

- **Important fields in the opts structure:**

  mu .... primary parameter balancing the data and regularization terms. This parameter is not as important if the data multiplier δ is implemented and updated a sufficient number of times (see opts.data_mlp term below).

  beta .... secondary parameter to encourage $D^k f = w$. This parameter is not as important if a sufficient number of outer iterations are used (see opts.outer_iter below).

  order .... order (k) of the finite difference operator used in the regularization term, can be set to any real number ≥ 0. Generally recommended between 1 and 3.

  levels .... Set to an integer generally between 1 and 4. For values greater than 1 a multiscale regularization approach is implemented.

  data_mlp ...... set to true or false, specifying whether or not the Lagrangian multiplier δ is used and updated. This parameter has a major effect. Set to true to approximately solve the constrained problem $Af = b$, or to simply accelerate convergence on the data term. Can also set to a positive integer value to indicate the maximum number of updates for δ.

  L1type .... set to 'isotropic' to solve the isotropic model or 'anisotropic' to solve the anisotropic model. Isotropic is generally preferred.

outer_iter ..... number of iterations or updates on the Lagrangian multiplier terms (as discussed above, the updates on δ are also limited by opts.data_mlp).

inner_iter ...... number of iterations between each update on the Lagrangian multiplier terms, thus the maximum number of total iterations is inner_iter*outer_iter

tol_out .... outer loop tolerance used to terminate the algorithm

tol_inn .... inner loop tolerance used to move to next outer loop

nonneg .... if set to true, problem is solved under the constraint $f \geq 0$

max_c .... if set to true, problem is solved under a max constraint, $f \leq$ max_v

max_v .... maximum value used if max_c is set to true

wrap_shrink ... set to to false unless signal is periodic.

** Other options are available, a full description of each is given in the "check_HOTV_opts.m" file. It is generally recommended to just use default values.*

-For automatic HOTV reconstruction of tomographic data see the code *HOTV3D_tomo.m*.

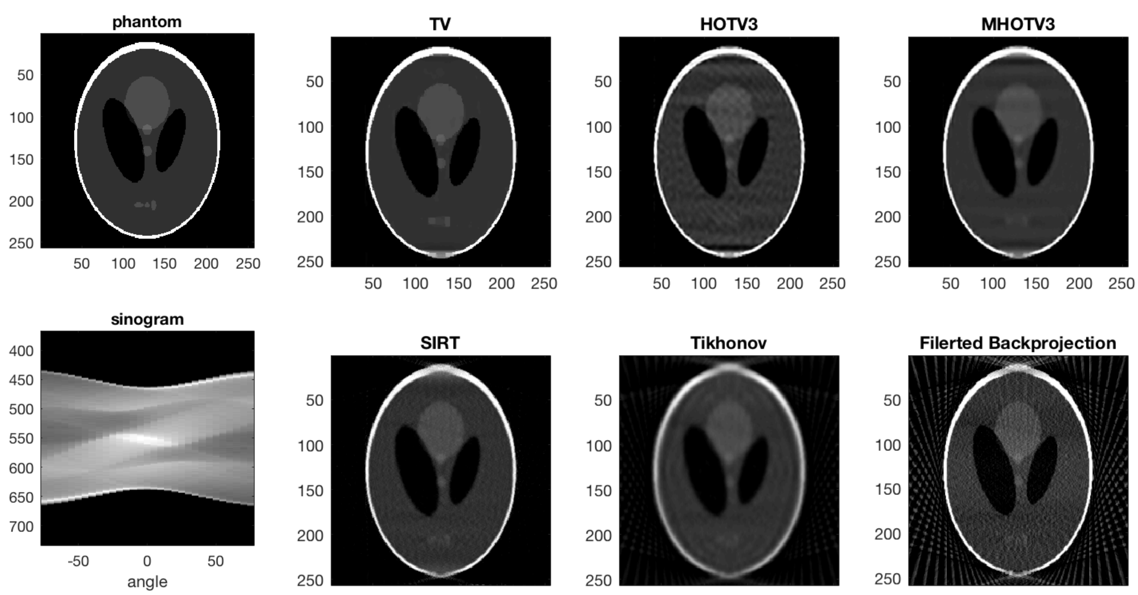

Figure 1: Tomographic reconstruction of phantom image.

1.2 General ℓ_1 regularization algorithm

- **Code name:** l1optimo
- **Demo:** see demo_MHOTV.m
- **Description:** An iterative solver for general ℓ_1 regularization minimization for inverse problems and denoising.

```
% function [U, out] = l1optimo(A,b,n,opts)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          Problem Description      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Motivation is to find:

%               min_f { mu/2*||Af - b||_2^2 + ||D f||_1 }


% This algorithm is essentially the same as HOTV3D, however,
% the user must specify the regularization operator "D".  To do this,
% the user should input into the opts structure the options opts.D and
% opts.Dt, where opts.D is the regularization transform, and opts.Dt is
% the adjoint of opts.D.
```

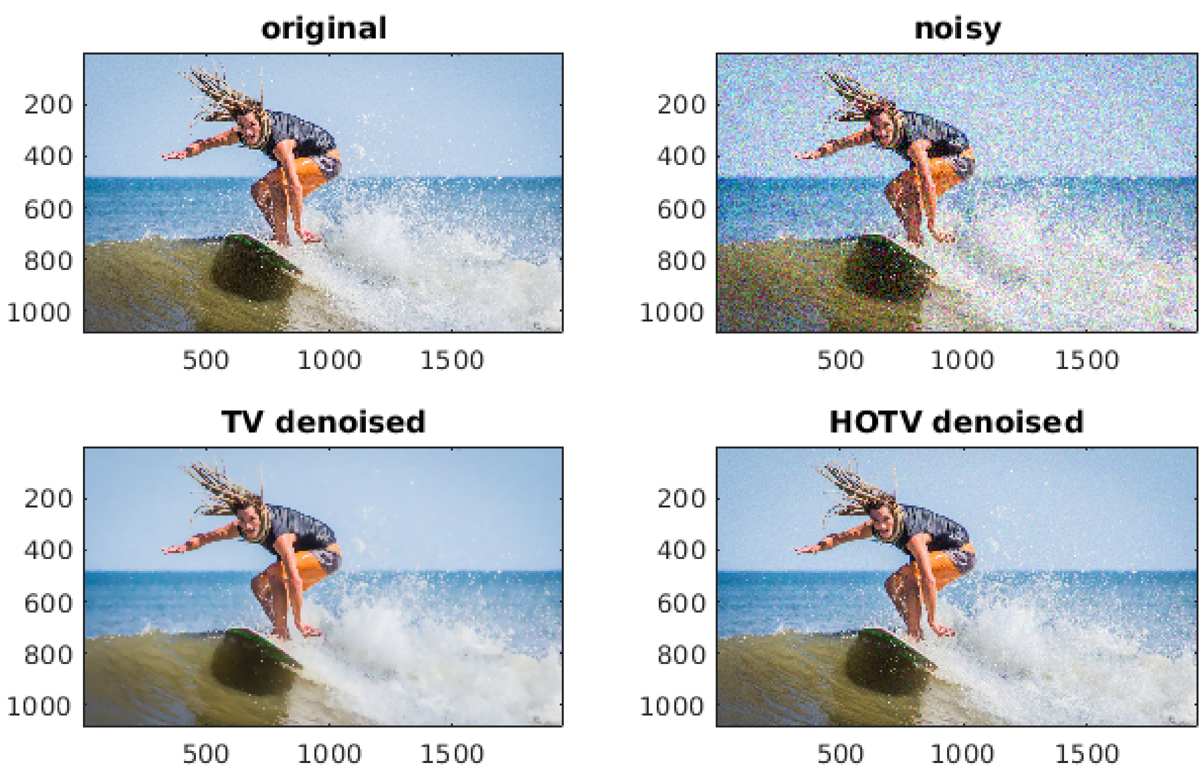

Figure 2: Denoising with ℓ_1 regularization.

1.3 Tomographic reconstruction with SIRT

- **Code name:** SIRT

- **Demo name:** see demo_tomo.m
- **References:** [11]
- **Description:** Simultaneous iterative reconstruction technique (SIRT) for tomographic reconstruction, which works by the general gradient decent method. MATLAB comments:

```
% function [x,out] = SIRT(stack,angles,recsize,iter,minc,maxc)
%
%DESCRIPTION:
    %this function performs the SIRT algorithm for tomographic reconstruction.
%
%NOTATION:
    % function [x,out] = SIRT(stack,angles,recsize,iterations,minc,maxc);
%
%INPUTS:
    % stack - the tilt series, where it is assumed the tilt axis is horizontal
    %   and located at the middle of the stack
    % angles - a vector holding the projection angles of the stack, in order,
    %   in degrees
    % recsize - the dimension of the reconstruction
    % iter - the number of SIRT iterations
    % minc - minimum density constraint, e.g. U>=0.
    % maxc - maximum density constraint, e.g. U<=1.
    %
    % default values are used for recsize, iter, and minc if they are not
    % specified, therefore one may simply input "sirtden(stack,angles)."
    % the recsize will be set to the detector count, i.e. size(stack,1)
    % 50 iterations is default, and no density constraint is used if it is not
    % specified
%
%OUTPUT:
    % x - the reconstruction from the input tilt series and other input
        %parameters
    % out - additional outputs
```

- **Additional Notes:** For automated reconstruction of a 3D volume see the algorithm *sirtauto.m.*

1.4 Discrete tomographic reconstruction with DART

- **Code name:** DIRT
- **Demo name:** demo_DART.m
- **References:** [2, 3, 8, 5]
- **Description:** Discrete algebraic reconstruction technique for tomographic reconstruction with prescribed gray levels, designed and extensively implemented by K.J. Batenburg et. al. MATLAB comments:

```
 %   function [U,out,init] = DIRT(opt,bb,init)


%INPUTS:
%Inputs that must be user defined:
    % bb - input projection data, can be left empty if user instead
        % specifies opt.data_name to reference saved data
    % init - initial solution, can be left empty and algorithm will
        % automatically compute a SIRT solution for initialization
    % fields in the opt structure:
        %opt.angles - the projection angles listed in order, in degrees
        %opt.thresh - a vector holding the thresholds used for segmenting
        %opt.grays - a vector holding the gray values used in segmentation.
        %opt.data_name - name of the file in which the aligned tilt series is
            %saved.  Note the tilt axis should be horizontal and centered.
        %opt.data_type - type of file of opt.filename, set to either 'mrc' or
            %'mat'
    % optional fields for the opt stucture that have default values if left
    % unassigned
        %opt.inner_iter - number of SIRT iterations in each DART iteration.
            %Default value is 10.
        %opt.W - if the projection matrix is precomputed, set opt.W to be
            %the projection matrix.
        %opt.disp - prints information about the reconstrution at each
            %iteration.  Set to "true" for the display and "false"
            %otherwise.  Default is "true".
        %opt.disppic - displays various images about the reconstruction at
            %each iteration. Set to "true" for this display and "false"
            %otherwise.  Default is "true".
        %opt.convergence_criteria - if set to "true", DART will iterate until
            %the reconstruction converges.  Set to "false" to simply use
            %opt.outeriter instead.  Default is "false".  Typically
            %convergence is seen after just a few iterations.
        %opt.convergence_tol - tolerance for opt.convergencecriteria.
            %Default is 10^(-3).
        %opt.max_iter - maximum number of iterations if
            %opt.convergencecriteria is used.  Default is 50.
        %opt.rx - radius of the gaussian in the slice (x,y) dimension used
            %for smoothing.  Set to 1 for no smoothing.  Default is 3.
        %opt.rz - radius of the gaussian in the slice z dimension used
            %for smoothing.  Set to 1 for no smoothing.  Default is 3.
        %opt.sigmax - sigma value used for the gaussian smoothing in the
            %(x,y) dimension.  Default is 1.  Increase for more smoothing
            %and decrease for less smoothing.
        %opt.sigmaz - sigma value used for the gaussian smoothing in the
            %z dimension.  Default is 1.5.  Increase for more smoothing
            %and decrease for less smoothing.
        %opt.chunksize - size of the chunks that DART will run on.  Default
```

```
        %is 50.
    %opt.overlap - once the chunks are computed, they are merged
        %together using a partition of unity.  The overlap is the overlap
        %of this merging.  Default is 10.


%OUTPUTS:
    %U - the 3-D DART reconstruction.  The slices change with the 3rd
        %dimension, i.e. dartrec(:,:,i) holds the ith slice of the
        %reconstruction
    %out - data about the reconstruction
    %init - the SIRT reconstruction or input reconstruction used as
        %the initial solution to compute U.




%Inputs recommended to be user defined, but default values are otherwise
%used:
    %opt.bdry_dirt_iter - number of DART iterations with boundary
        %refinements
    %opt.region_dirt_iter - number of region refinements (DIP-LS)
    %opt.t_tol - radius of interval used for the partial segmentation
        %function with DIPS.  Set a small tolerance for slower more
        %careful convergence and a high tolerance for fast convergence
    %opt.t_delta - increase in opt.t_tol whenever new pixels are not being
        %classified
    %opt.t_epsilon - this epsilon is used to determine if opt.t_tol should
        %be increased by opt.t_delta.  It is the case if the number of
        %classified pixels has only changed relatively by less that
        %epsilon.
    %opt.bdry_update_type - indicates the solver used for bdry updates.
        %Set to 'SIRT' (default) or 'CGLS'.
    %opt.region_update_type - indicates the solver used for region updates.
        %Set to 'SIRT' (default) or 'CGLS'.
    %opt.rand_pixel_probability - probability to a free pixel is set to
        %free.  Default is 0.
    %opt.resolution - the number of pixels used for reconstruction.
        %Reconstruction slices will be opt.recsize by opt.recsize. Default
        %value is the detector count
    %opt.startslice - which slice the reconstruction begins on.  Default is 1.
    %opt.endslice - which slice the reconstruction ends on.  Default is the
    %end of the stack.
    %opt.initialsolution - Set to "true" an initial solution has been
        %computed and will be used, otherwise set to "false".
    %opt.initialfile - name of the matlab file where the reconstruction is
        %saved
```

2 Image Alignment

2.1 Basic Cross Correlation

- **Code names:** cross_corr.m and cross_corr_pad.m
- **Demo name:** see demo_align.m
- **Description:** these algorithms perform 2D image alignment from an input 3D stack of images by cross correlation. “cross_corr.m” maintains the original image size by simply rotating or circular shifting the image and “cross_corr_pad.m“ pads the stack with zeros based on the minimum and maximum shifts. MATLAB comments:

```
%DESCRIPTION
    %This function performs cross correlation of an input stack of images.
    %
%NOTATION
    %[stacknew,shifts]=cross_corr(stack,startslice);
    %
%INPUTS
    %stack - 3D matrix assumed to be a stack of images to be cross
        %correlated
    %startslice - the reference image that all slices are consecutively
    %aligned to.  This image will not move
    %
%OUTPUTS
    %stack- 3D stack of images after cross correlation
    %shifts - shifts that were applied to each image
    %
%NOTE: This function automatically uses a window to improve alignment.  To
    %remove the windowing, set pwr=0.
```

2.2 Center of mass alignment for tilt series alignment

- **Code names:** COM_align.m
- **Demo name:** see demo_align.m
- **References:** [10]
- **Description:** Center of mass alignment for electron tomography data. See http://ascimaging.springeropen.com/articles/10.1186/s40679-015-0005-7 for details. MATLAB comments:

```
%  Center of mass alignment for electron tomography data
%
% function [stack,shifts,usables] = COM_align(stack,angles,ratio,s)
%
% Inputs: stack is 3-D matrix containing tomography data
% angles are the angles in degrees
% ratio is the ratio of slices to be used for each projection
```

```
% s is the number of projections used as a sequence to determine rigid
% alignment.  Setting s to be the number of angles is equivalent to the
% alignment method published here:
% http://ascimaging.springeropen.com/articles/10.1186/s40679-015-0005-7
%
%
%
% The ratio variable:
% should be input as a number between 0 and 1, where the user rates the
% consitency of the stacks through the slices.  From our experience, very
% small values of s are better.  If the user does not input a
% rating, the default value of .1 is used.  Setting ratio
% to perfect 1 would mean that through the slices, there is significant amount
% of mass in all of the slices, and very little mass moves in or out as the
% tilt angle changes.  A lower ratio would mean otherwise.
% This rating is then used to determine how many slices to use for the
% center of mass alignment.  For example, if you set the rating to .2, then
% only the best 20 percent of the slices will be used in the alignment step.
% That way, if the the stack has some slices with very little mass, yet
% mass moves into these slices at high angles, these slices may be ignored
% for the aligment.
```

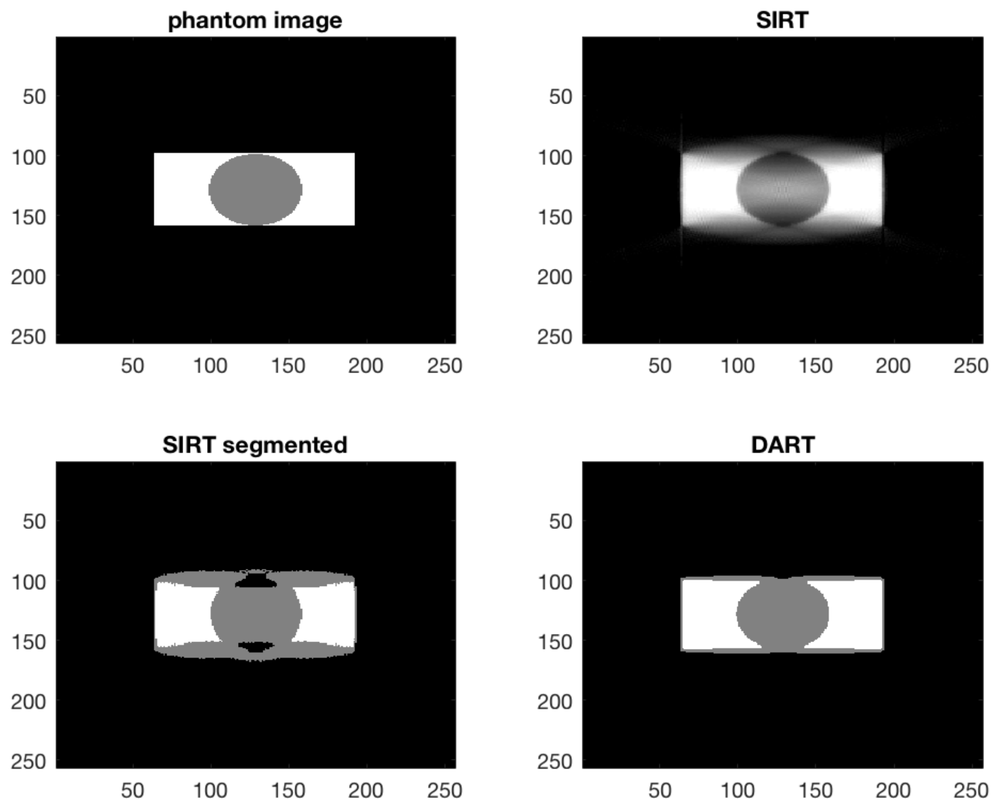

Figure 3: Tomographic reconstruction with DART.

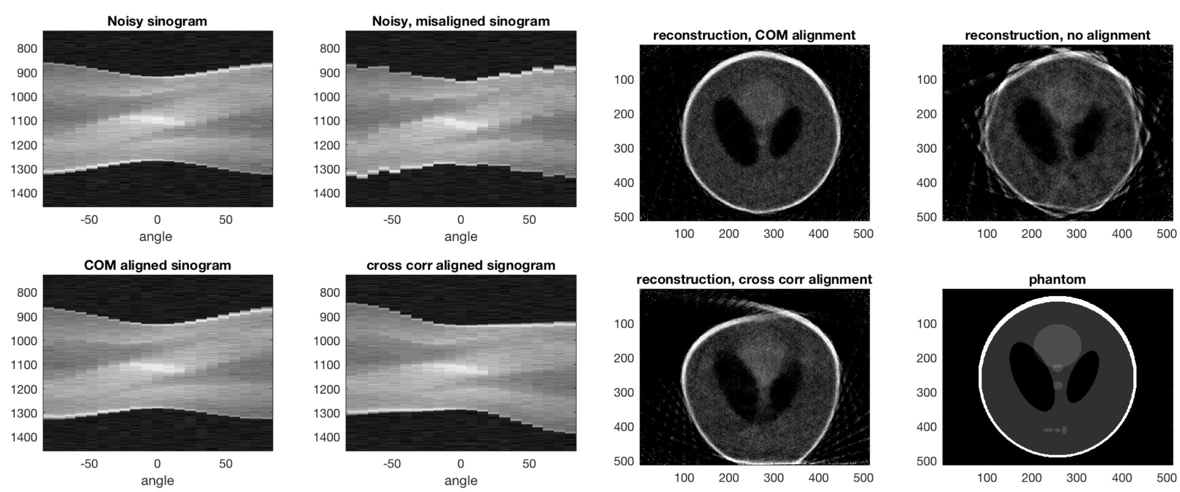

Figure 4: Tomographic alignment and reconstruction (SIRT).

3 Other utilities and codes potentially of interest

**See the comments within each code for details.*

- moviestack.m - used to visualize a 3D volume by passing through 2D cross-sections.
- radonmatrix.m - builds a sparse projection matrix for tomography based off of input geometry.
- corrx_global.m - a horizontal image alignment procedure for a tilt series based off of the conservation of mass.
- manualalign.m - a utility that can be used to manually align images.
- forward_proj.m - builds a tilt series given input projection angles and 3D volume.

References

[1] Rick Archibald, Anne Gelb, and Rodrigo B. Platte. Image reconstruction from undersampled Fourier data using the polynomial annihilation transform. *J. Sci. Comput.*, pages 1–21, 2015.

[2] K. J. Batenburg and J. Sijbers. Dart: A practical reconstruction algorithm for discrete tomography. *IEEE Transactions on Image Processing*, 20(9):2542–2553, Sept 2011.

[3] K.J. Batenburg, S. Bals, J. Sijbers, C. Kbel, P.A. Midgley, J.C. Hernandez, U. Kaiser, E.R. Encina, E.A. Coronado, and G. Van Tendeloo. 3d imaging of nanomaterials by discrete tomography. *Ultramicroscopy*, 109(6):730 – 740, 2009.

[4] Tony Chan, Antonio Marquina, and Pep Mulet. High-order total variation-based image restoration. *SIAM J. Sci. Comput.*, 22(2):503–516, 2000.

[5] B. Goris, T. Roelandts, K.J. Batenburg, H. Heidari Mezerji, and S. Bals. Advanced reconstruction algorithms for electron tomography: From comparison to combination. *Ultramicroscopy*, 127:40 – 47, 2013. Frontiers of Electron Microscopy in Materials Science.

[6] Chengbo Li. *An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing*. PhD thesis, Rice University, 2009.

[7] Chengbo Li, Wotao Yin, Hong Jiang, and Yin Zhang. An efficient augmented lagrangian method with applications to total variation minimization. *Comput. Optim. Appl.*, 56(3):507–530, 2013.

[8] T. Sanders. Discrete iterative partial segmentation technique (dips) for tomographic reconstruction. *IEEE Trans. Comput. Imag.*, 2(1):71–82, March 2016.

[9] Toby Sanders, Anne Gelb, Rodrigo B Platte, Ilke Arslan, and Kai Landskron. Recovering fine details from under-resolved electron tomography data using higher order total variation l1 regularization. *Ultramicroscopy*, 174:97–105, 2017.

[10] Toby Sanders, Micah Prange, Cem Akatay, and Peter Binev. Physically motivated global alignment method for electron tomography. *Advanced Structural and Chemical Imaging*, 1(1):1–11, 2015.

[11] Jeannot Trampert and Jean-Jacques Leveque. Simultaneous iterative reconstruction technique: Physical interpretation based on the generalized least squares solution. *J. Geophys. Res.*, 95(12):553–9, 1990.