**Image registration algorithm for use in painting conservation**

# *Toolbox documentation*

Author: Damon Conover (dconover@gwmail.gwu.edu)

Last revision date: 1 June 2011

The George Washington University, Washington, DC, 20052

Contact: Dr. Murray Loew (loew@gwu.edu)

National Gallery of Art, Washington, DC, 20565

Contact: Dr. John Delaney (j-delaney@nga.gov)

The toolbox consists of a set of functions, written in MATLAB, that can be used to register related image datasets. The functions have designed with the primary purpose of registering the most common datasets captured in a museum's scientific research department. This document explains how to install and use the toolbox.

Installation

1. Extract the art\_registration folder to a location on your hard drive.

2. Open MATLAB and add the art\_registration folder to your PATH variable (File, Set Path.., Add Folder).

Description of the tools

1. ir\_extract - Extracts template images from an .img file to a local folder. The name of the local folder will be the same as the name of the .img file. The script will prompt the user for a scale factor that will be used to resize the images, unless it is provided as an argument. The script will also prompt the user to delete any unwanted images, such as images containing standards or duplicates.

Usage: ir\_extract(ir\_fn,tilt,sc)

Inputs: ir\_fn = full path and filename for the .img file

tilt = 0 or 1 depending on the relative orientation of the camera when the images were captured

sc = scale factor for resizing the template images

Outputs: template images

2. ir\_extractX - Segments a .tif file into equal-sized blocks. The blocks are then saved as separate images in a local folder with the same as the name of the .tif file. The script will prompt the user for the a scale factor that will be used to resize the blocks, block size (before scaling), and amount of overlap between the blocks (before scaling), unless they are provided as an argument.

Usage: ir\_extractX(ir\_fn,sc,sz,ov)

Input: ir\_fn = full path and filename for the .tif file

sc = scale factor for resizing the template images

sz = size of output block in pixels (before scaling)

ov = amount of overlap between blocks in pixels (before scaling)

Outputs: template images

3. rough\_mosiac - Determines the relative positions between the template images by computing the correlation between it and the next image in the set. The script will prompt the user for the minimum amount of overlap for computing the correlations (more overlap will be more accurate, but will be slower). A figure will show the progress of the script (as shown in Figure 1).

Usage: rough\_mosaic(ir\_fn)

Input: ir\_fn = full path and filename for the .img or .tif file

Output: spreadsheet called offset\_values.csv that contains the relative shift values between the template images

'

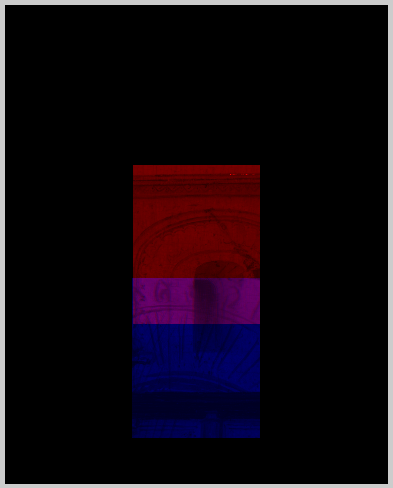


Figure : rough\_mosaic status

4. fix\_mosaic - GUI that allows a user to correct any mistakes make by rough\_mosaic (as shown in Figure 2). It is called directly from rough\_mosaic. Once the GUI is loaded, the user supplies the path and folder name containing the template images, then iterates through the image set. If an offset error is encountered, the scroll bars can be used to reposition the image. The offset\_values.csv spreadsheet will be automatically updated.

Usage: fix\_mosiac

Inputs: path to template image folder (example: C:\data\Vanorley\)

template image folder name (example: VanOrley\_ChristwithDoctors\_H)

Output: spreadsheet called offset\_values.csv that contains the relative shift values between the template images

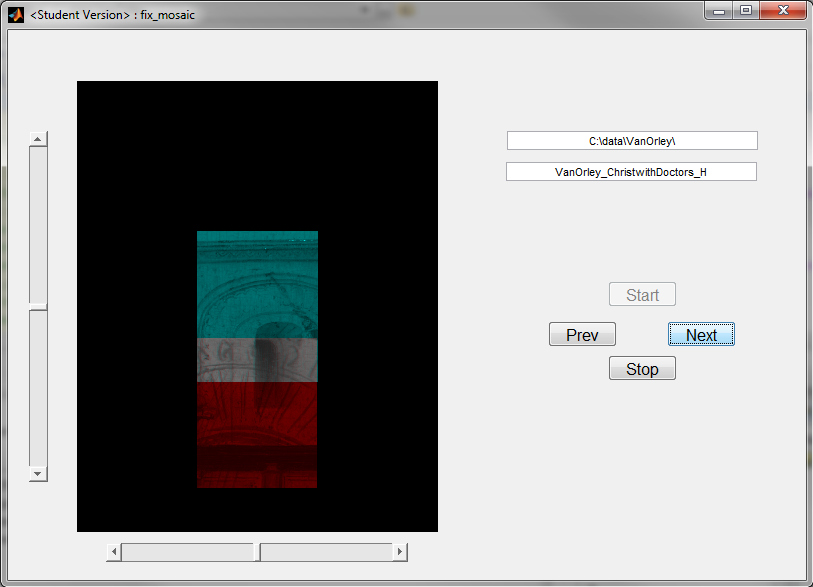


Figure : fix\_mosiac GUI

5. art\_register - This script contains all of the code for performing the registration between a reference image and a set of template images.

Usage: art\_register(rgb\_fn,sc,ir\_fn1,j01,j02,ppthresh,repcnt\_max,useresults)

Inputs: rgb\_fn = filename of the reference image

sc = scale factor or resizing the reference image

ir\_fn = full path and filename for the .img or .tif file

j01, j02 = min/max scales used in feature selection

ppthresh = modulus threshold for feature selection (note: pptthresh = 0.01 means that 1% of local maxima will be selected as initial feature points)

repcnt\_max = repeat threshold (note: repcnt\_max=2 means that ppthresh\*(2^0), ppthresh\*(2^1), and ppthresh\*(2^2) will be used before moving on to the next template image)

useresults (experimental) = 1 (includes info from the registration of neighbor blocks), or 0 (do not include other registration info)

Outputs: two registered images (The first image will be called <trial>\_RGB.tif. The second image will be called <trial>\_IR.tif, where <trial> is the name of the .img or .tif file.)

Spreadsheet, called init\_reg.csv, containing the initial horizontal and vertical shift values of each template image determined before registration

Spreadsheet, Xstar.csv, containing the transformation coefficients that were used to register each shifted template image

6. cube\_register - This script contains all of the code for performing band-to-band registration of a multispectral cube.

Usage: cube\_register(cube\_fn,lines,samples,bands,precision,offset,interleave,byteorder,

j01,j02,ppthresh,repcnt\_max)

Inputs: cube\_fn = filename of the input cube

lines = number of rows (specified in cube header)

samples = number of columns (specified in cube header)

bands = number of spectral bands (specified in cube header)

precision = precision of the data (i.e., single, double, uint8, uint16) (specified in cube header)

offset = indicates the location of the first data element the cube the data starts (specified in cube header)

interleave = indicates how the data is ordered (i.e., bsq, bil, bip) (specified in cube header)

byteorder = indicates the machine format in which the data was stored (i.e., ieee-le, ieee-be) (specified in cube header)

j01, j02 = min/max scales used in feature selection

ppthresh = modulus threshold for feature selection (note: pptthresh = 0.01 means that 1% of local maxima will be selected as initial feature points)

repcnt\_max = repeat threshold (note: repcnt\_max=2 means that ppthresh\*(2^0), ppthresh\*(2^1), and ppthresh\*(2^2) will be used before moving on to the next template image)

(Optional): If a file with the name <cube\_fn>\_mask.tif exists in the working directory, cube\_register will use it to mask out unwanted regions of the image set. The tif file should be a black and white image, where the white regions are contain the portions of the dataset from which features may be identified. Potential features in the black regions will be ignored.

Outputs: registered cube and corresponding header file (The file names will be <cube\_fn>\_registered and <cube\_fn>\_registered.hdr, respectively.

7. apply\_transform - Assume that a transformation has been applied to an image (image 1) in order to register it with another image (image 0). This function then applies that same transformation to a second image (image 2). If images 1 and 2 were originally registered, then this function will produce a transformed image2 that is also registered with image 0.

Usage: apply\_transform(fn,m1,n1,init\_reg,Xstar)

Inputs: fn = filename of image dataset to be transformed

m1, n1 = vertical and horizontal sizes in pixels of the reference image used during registration

init\_reg = full path to the init\_reg.csv spreadsheet

Xstar = full path to the Xstar.csv spreadsheet

createavi = 1 (create an AVI movie showing the transformations applied in series), or 0 (do not create the movie)

Operation

1. Open MATLAB and set the Current Folder to the location where you wish your registered images to be saved. This folder is also where all temporary files will be placed and your project-specific code will be saved.

2. Create a script containing the steps necessary for registration and save it in your current directory with a file extension of .m (example: run\_marriage\_register.m). Review the example script 1 below for guidance in creating the script. Example script 1 shows how to setup a series of two registration trials. The first registers an H-band IR image set to a color image. The second registers a K-band IR image set with the output of the first registration (VanOrley\_Marriage\_H\_IR.tif). Example 1 also show how the script can be organized into two sections: interactive and automatic. The interactive section contains all of the tools that require user input. By organizing the script in this way, a user can setup multiple trials and then leave them to run in series without needing to return to supply additional information.

**interactive**

**section**

**automatic**

**section**

ir\_fn1 = 'C:\data\VanOrley\VanOrley\_Marriage\_H.img';

ir\_extract(ir\_fn1,0)

rough\_mosaic(ir\_fn1)

ir\_fn2 = 'C:\data\VanOrley\VanOrley\_Marriage\_K.img';

ir\_extract(ir\_fn2,0)

rough\_mosaic(ir\_fn2)

rgb\_fn = 'C:\data\VanOrley\A14337\_F-OL.tif';

sc = 0.9665;

j01 = 3;

j02 = 4;

ppthresh = 0.01;

repcnt\_max = 2;

useresults = 1;

art\_register(rgb\_fn,sc,ir\_fn1,j01,j02,ppthresh,repcnt\_max,useresults)

rgb\_fn = 'VanOrley\_Marriage\_H\_IR.tif';

sc = 1;

ppthresh = 0.005;

repcnt\_max = 3;

art\_register(rgb\_fn,sc,ir\_fn2,j01,j02,ppthresh,repcnt\_max,useresults)

Example script 1: Register sets of IR template images with a color image (example filename: run\_marriage\_register.m)

3. From the MATLAB command window, run the script that you just created. The code can take several hours depending upon the size and number of images. A figure showing the status of the registration will be displayed (as shown in Figure 3).



Figure : art\_register status

4. Once the script has completed, two image files per trial will be written to the current directory. It is recommended that you rename the files to protect them from being overwritten if the script is run a second time.

Additional information

1. If the script is interrupted, it can be restarted from its current spot by running the script again. For this to work, the <trial>\_IR.tif, init\_reg.csv, and Xstar.csv files must remain from before the script was interrupted. If you wish to restart the script from the beginning, then you must delete those three file.

2. A second example script is shown below. This script shows how to register two scanned X-ray films with their correspond color image. In the interactive section, the two X-ray images are segmented to produce sets of small, overlapping images. Then the first set of images are registered and an image containing the first registered film is produced (x1\_IR.tif). The registered film is then copied with a new name (x2\_IR.tif). The second instance of art\_register will then attempt to register the second film (x2.tif) and when it sees that x2\_IR.tif already exists, if will start with that as its initial output image, rather than a blank image. By arranging the films in series this way, in the end, you will produce a registered X-ray image containing all of the films.

ir\_fn1 = 'C:\data\VanOrley\x1.tif';

ir\_extractX(ir\_fn1)

rough\_mosaic(ir\_fn1)

ir\_fn2 = 'C:\data\VanOrley\x2.tif';

ir\_extractX(ir\_fn2)

rough\_mosaic(ir\_fn2)

rgb\_fn = 'C:\data\VanOrley\A14340\_F-OL.tif';

sc = 0.9592;

j01 = 3;

j02 = 4;

ppthresh = 0.01;

repcnt\_max = 3;

useresults = 0;

art\_register(rgb\_fn,sc,ir\_fn1,j01,j02,ppthresh,repcnt\_max,useresultswith a reference image and the corresponding mosaicing.n performed).ed in series), or 0 (do not create the movie))

copyfile('x1\_IR.tif','x2\_IR.tif');

rgb\_fn = 'C:\data\VanOrley\A14340\_F-OL.tif';

sc = 0.9592;

art\_register(rgb\_fn,sc,ir\_fn2,j01,j02,ppthresh,repcnt\_max,useresults)

**interactive**

**section**

**automatic**

**section**

Example script 2: Register two scanned X-ray films with a color image (example filename: run\_vanorleyx\_register.m)

3. A third example script is shown below. This script shows how to create an AVI video showing the registration and mosaicing of each of the templates images contained in the VanOrley\_ChristwithDoctors\_H.img (registration using art\_register must have already been performed). For this example, the movie's filename will be VanOrley\_ChristwithDoctors\_H\_movie.avi.

ir\_fn = 'C:\data\VanOrley\VanOrley\_ChristwithDoctors\_H.img';

tilt = 0;

sc = 1;

ir\_extract(ir\_fn,tilt,sc)

init\_reg = 'C:\data\VanOrley\VanOrley\_ChristwithDoctors\_H\init\_reg.csv';

Xstar = 'C:\data\VanOrley\VanOrley\_ChristwithDoctors\_H\Xstar.csv';

createavi = 1;

m1 = 6202;

n1 = 4716;

apply\_transform(ir\_fn,m1,n1,init\_reg,Xstar,createavi)

Example script 3: Create a video replaying the registration of the template images contained in VanOrley\_ChristwithDoctors\_H.img with a reference image and the corresponding mosaicing.