FKgrain User's Manual

(Version 2021/6/22)

Prepared by: Hong Ping Lo, Chi-Kuei Wang, and Fu-Chun Wu

Table of Contents

1 Software Installation 2

1.1 Dependencies 2

1.1.1 Matlab Runtime 2

1.1.2 R 2

1.1.3 GDAL 3

1.2 Main Program 3

2 Generate Zero-Contour Image 4

3 Morphological Grain Segmentation 6

3.1 Stage 1 – Processing of Zero-Contour Image 6

3.2 Stage 2 – Multi-Level Grain Segmentation 7

3.2.1 Level #1 7

3.2.2 Subsequent Levels 9

3.2.3 Export Results 11

1 Software Installation

1.1 Dependencies

1.1.1 Matlab Runtime

(1) Download Matlab Runtime Installer from the link below:
https://drive.google.com/file/d/1AM-gQMwe66ClW6BzuNx_CHVNxmIi1Gte/view?usp=sharing

(2) Unzip MatlabRuntime.zip and start the installer:
./MatlabRuntime/for_redistribution/MyAppInstaller_web.exe

(3) Follow the steps from the installer and accept terms of agreement

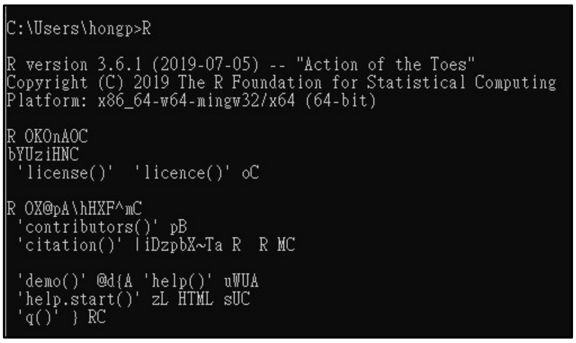
1.1.2 R

(1) Download R 3.6.3 installer from the link below:
https://cran.r-project.org/bin/windows/base/old/3.6.3/R-3.6.3-win.exe
(Different version may not have the required library. Only R 3.6.1 to 3.6.3 have been tested.)

(2) Follow the steps to install R

(3) Add C:\Program Files\R\R-3.6.3\bin to PATH system environment

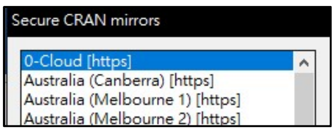
(4) Open a terminal and run R Console by typing R and press enter



(5) Install gstat and sp packages by running the following commands in R console:
install.packages("gstat")
install.packages("sp")

(6) Type in "yes" when asked if personal library is required

(7) Select the mirror server to download the packages. The default server is 0-Cloud.



(8) Check if the library is correctly installed by running the following commands in R console. If no error is shown, then the library has been successfully installed.
library("gstat")
library("sp")

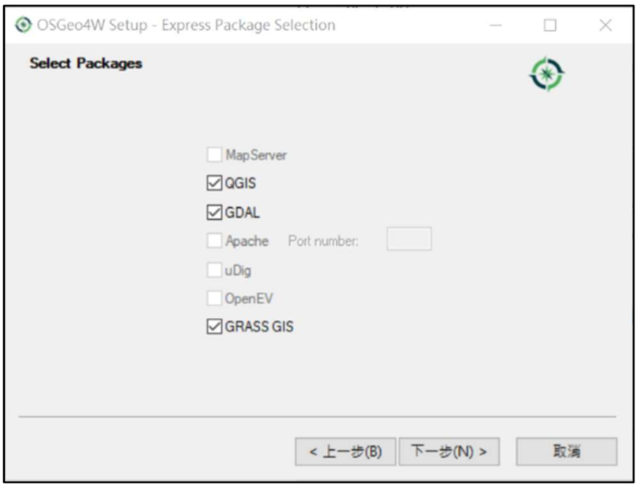
(9) If the library installation has been unsuccessful, try to run it by using a Windows PowerShell or run the terminal using Administrator.

1.1.3 GDAL

(1) Download OSGeo4W from the link below:
http://download.osgeo.org/osgeo4w/osgeo4w-setup-x86_64.exe

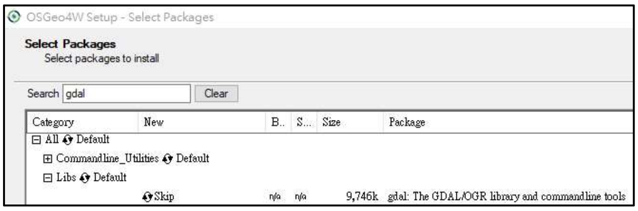
(2) Choose Express Desktop Install and press Next

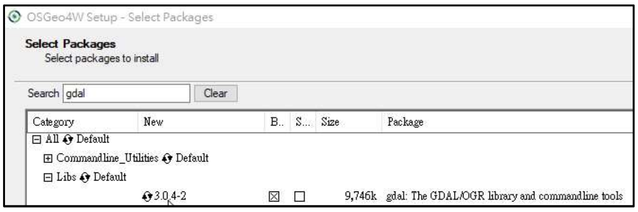
(3) Make sure GDAL is selected when prompt.



(4) Choose a download site and press Next. Default site: http://osgeo4w-oslandia.com

(5) Search for “gdal: The GDAL/ORG library and command line tools” and click on “Skip” if it is not selected.





(6) Press Next and agree on all license terms to start the installation process.

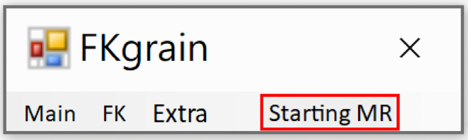
1.2 Main Program

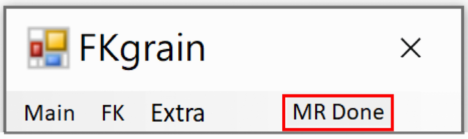
(1) Download the *FKgrain* program package from the link below:
https://github.com/ncku-arsem/FKgrain

(2) Unzip the downloaded program package

(3) Run the main program by launching FKgrain.exe.

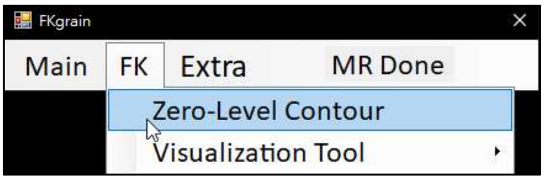
(4) “Starting MR” will appear during the Matlab Runtime is being launched, “MR Done” will appear when it is ready.





2 Generate Zero-Contour Image

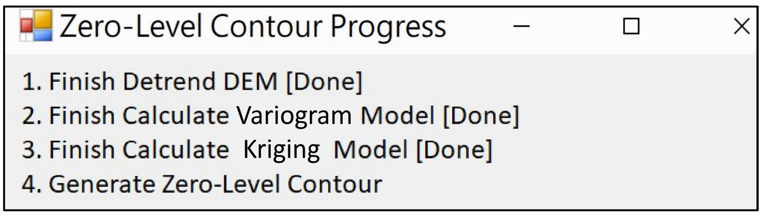
(1) In the Menu bar, click on FK tab and select “Zero-Level Contour”.

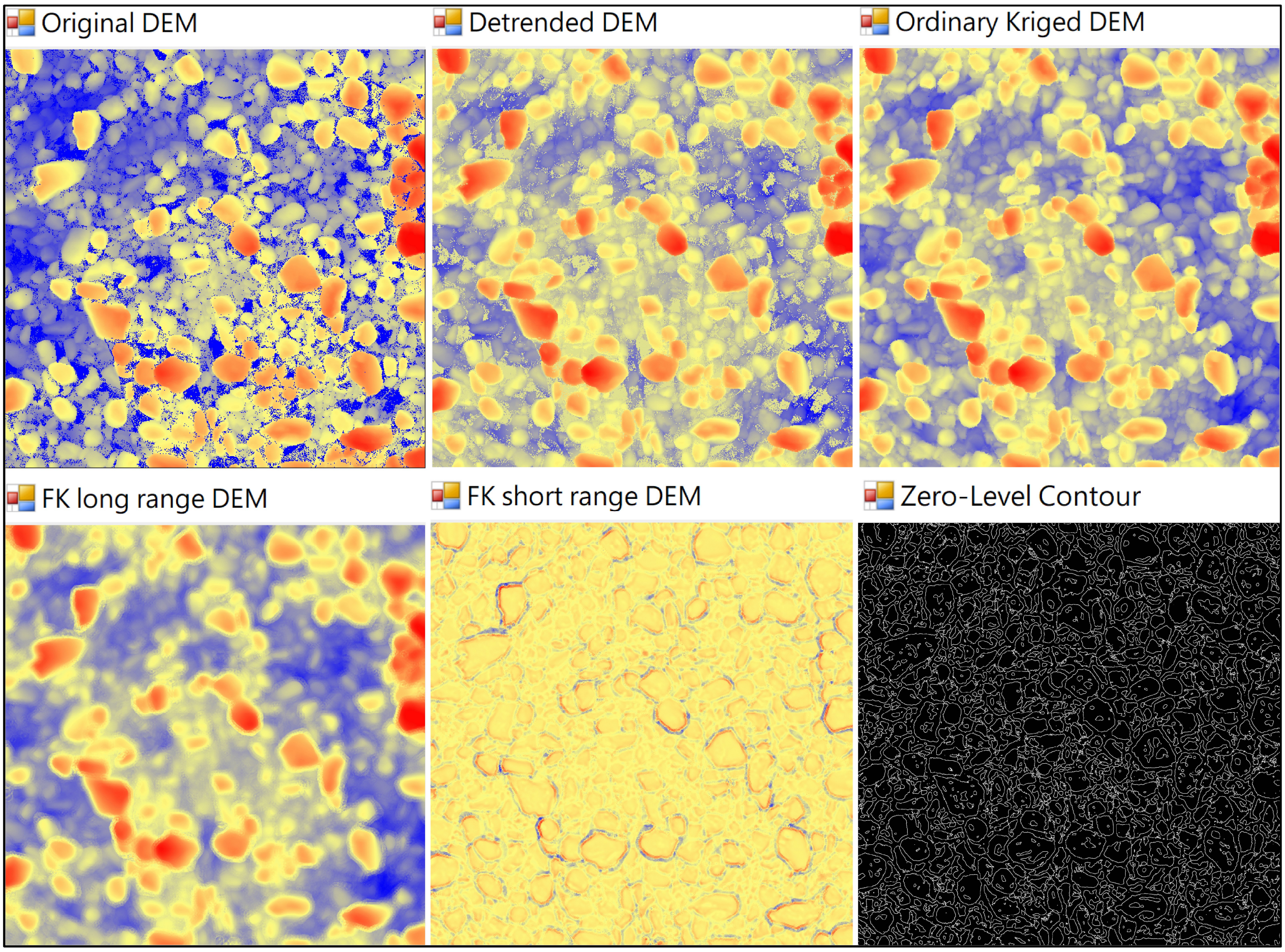


(2) Select an input DEM file (in .txt format). A 6 m × 6 m DEM (with 1 cm × 1 cm resolution) is provided in SampleDEM folder. (Note: space in the path name will result in error)

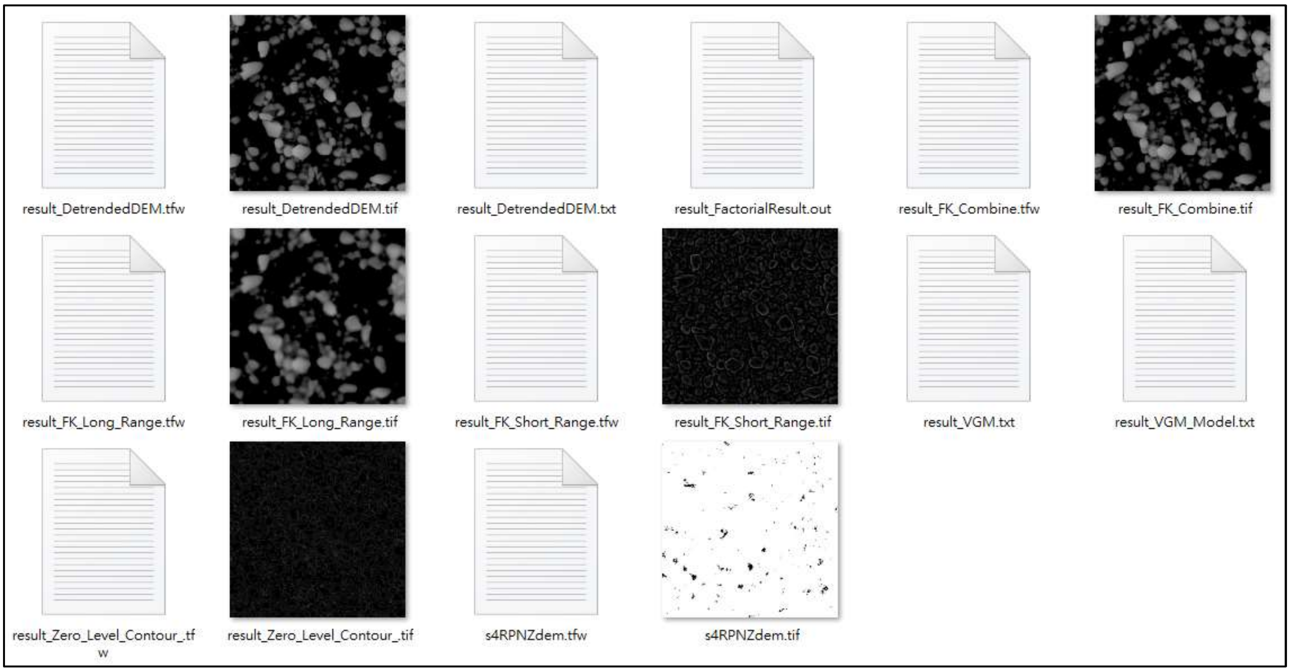
(3) Select a folder and type in a filename where the output image files will be saved.

A box will appear (see below) showing the progress of zero-level contour calculations. Wait until all calculations are done and the results show up. The whole process can take up to 30 minutes, depending on the size of data and speed of CPU. When DEM detrending is finished, an original DEM and a detrended DEM will show up; when kriging calculations are done, the ordinary kriged DEM, long-range and short-range factorial kriged (FK) DEM will show up. When all calculations are done, a binary image of zero-level contours will show up. These images may be also viewed by clicking FK in Menu bar and selecting “Visualization Tool”. They are shown in the next page.



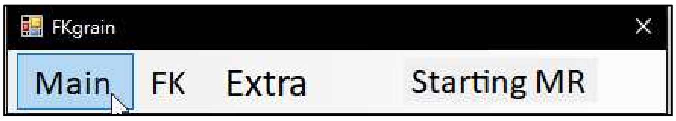


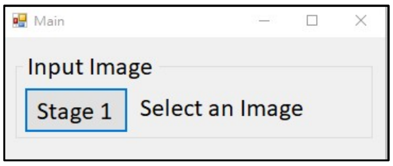
The figure below illustrates all output files saved in the selected folder. DetrendedDEM is the planar-detrended DEM, FK_Combine is the ordinary kriged (OK) DEM, FK_Long_Range is the long-range FK DEM, FK_Short_Range is the short-range FK DEM, and Zero_Level_Contour is the zero-level contour image of the short-range FK DEM. These image files can be retrieved by the GIS software.



3 Morphological Grain Segmentation

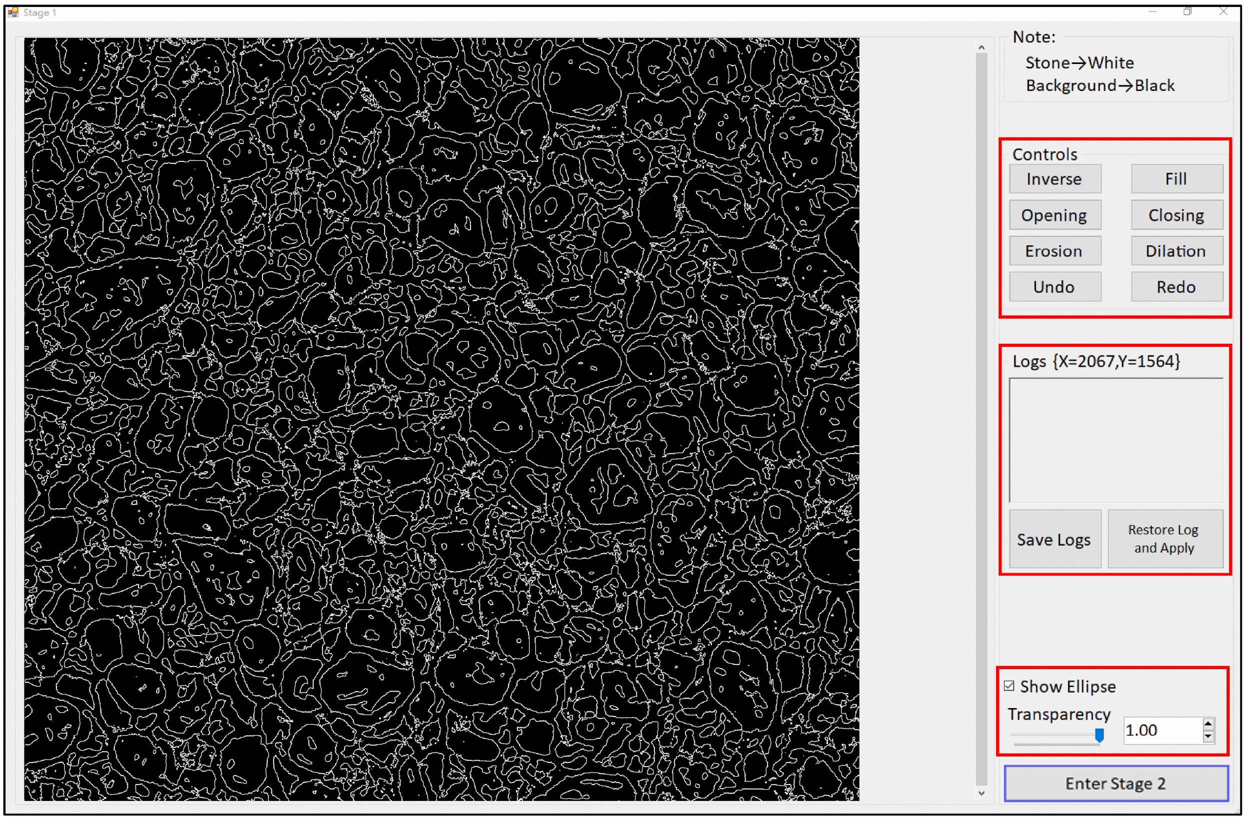
In the Menu bar, click on Main tab → Stage1, and select an input zero-contour image file. If no tfw file is found, users will be asked for the scale of the image.





3.1 Stage 1 – Processing of Zero-Contour Image

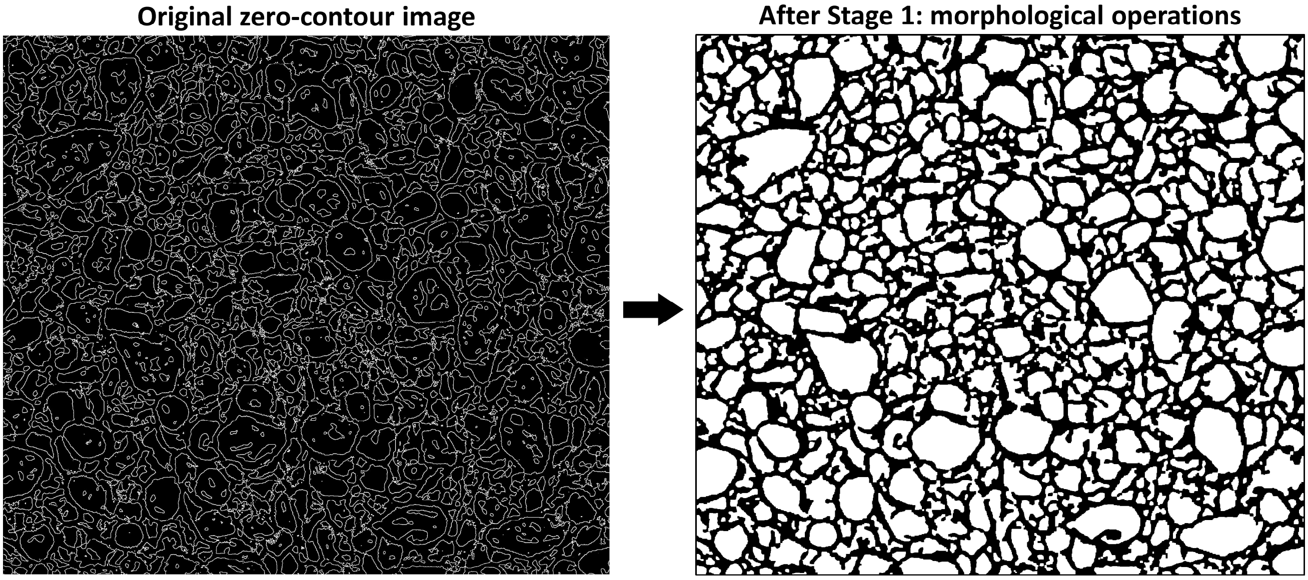
A morphological operation window will show up (see below). On the right side of the window, the Controls panel contains a set of morphological operations: Inverse, Fill, Opening, Closing, Erosion, and Dilation. Morphological operations work with binary images.



The example given below shows the original zero-contour image and the result of morphological operations (i.e., delineated grain segments) obtained by going through the following 5 steps:

1. Fill → Coordinates (x, y) = (1, 1). (x, y) should be in the non-object background to be filled.
2. Inverse
3. Fill → Holes

   This step is to eliminate the fragmentations that arise from the grain surface texture. Such textural features on the grain surface should be removed as they represent subgrain-scale noises, providing no useful information for grain segmentation.
4. Closing, with a square structuring element of size equal to 7.
5. Opening, with a square structuring element of size equal to 7.



Each operation implemented is recorded in the Logs panel, which can be saved as a log file and retrieved for later use. After implementing the morphological operations, the elliptical fits to the delineated grain segments will show up automatically, which can be turned off/on. For the sake of better visualization, left clicking to flash between the processed and original images or adjusting the transparency would be useful. Press “Enter Stage 2” when all operations in Stage 1 are done.

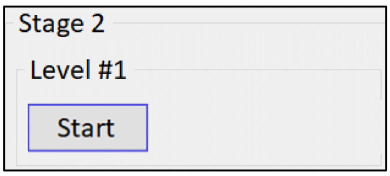
Note that the parameters used in steps 4 – 5 are suited for the present example. When processing other images, users should select suitable parameters to optimize the result. The image processing used in *FKgrain* is based on the Matlab morphological operations. For details, users are referred to https://www.mathworks.com/help/images/morphological-filtering.html?s_tid=CRUX_lftnav.

3.2 Stage 2 – Multi-Level Grain Segmentation

Stage 2 proceeds with multiple levels of morphological grain segmentation. The goal is to process the whole population of grain segments, while the operations divide the population into multiple levels of segment size and process one level at a time. By doing so, grain segmentation becomes tractable. Care must be taken, however, to balance the processing performance and efficiency.

3.2.1 Level #1

Press “Start” in the dialogue box to launch Level #1. Each level proceeds with two operational steps: (1) **Step 1** – grain thresholding; (2) **Step 2** – grain segmentation.

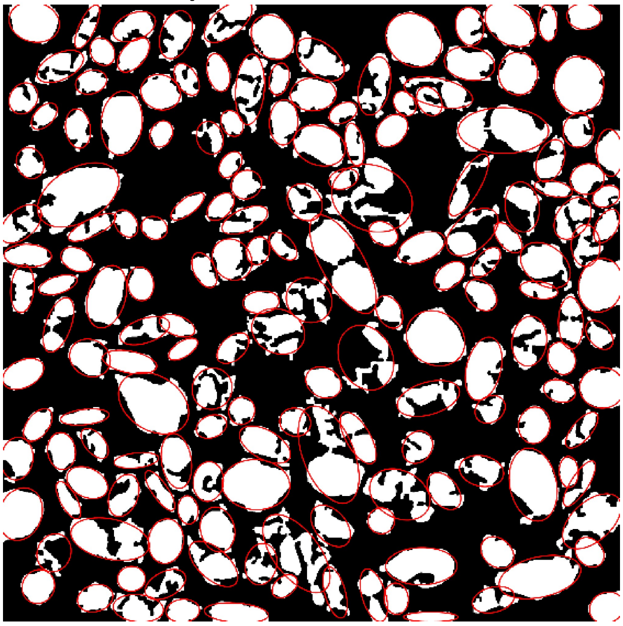


Step 1 thresholds the smaller grain segments. Only those grain segments with sizes greater than the specified threshold (number of pixels) will remain. The thresholded smaller segments will be processed in subsequent levels. The threshold can be specified by moving the scroll bar or typing in the input box. Level #1 starts with the largest threshold. When finishing **Step 1**, press “Done” to enter **Step 2**.

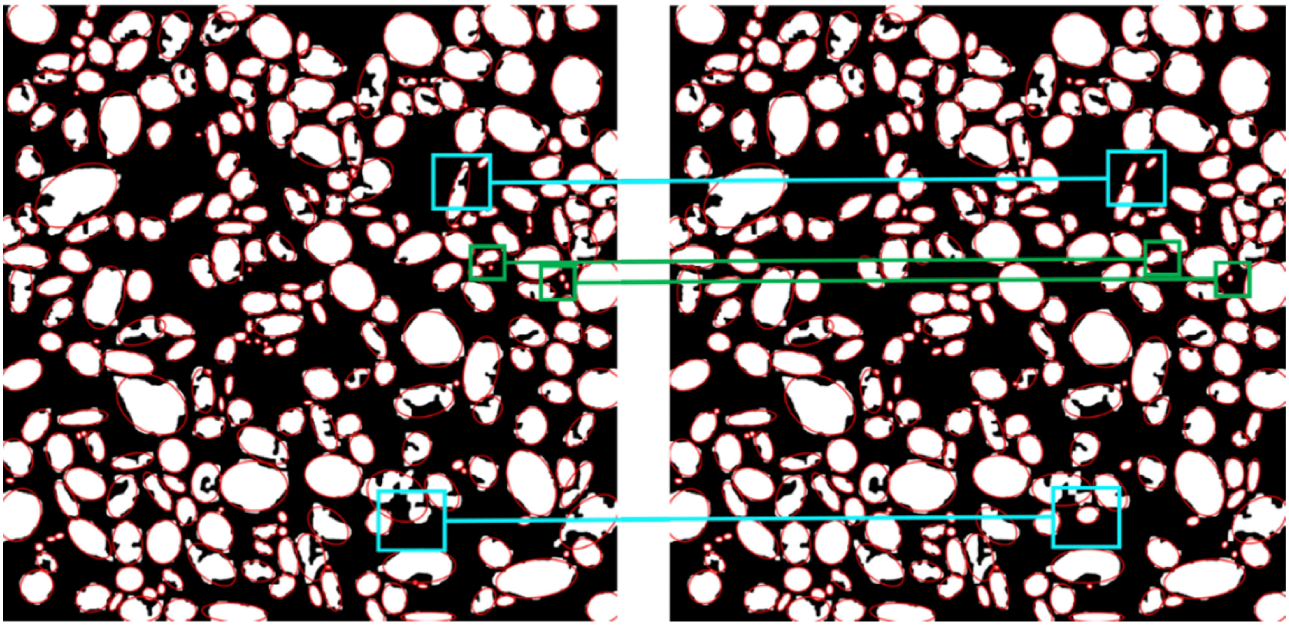
Step 2 is to separate connected grains that are delineated as a single segment. A morphological operation window will show up where segmentation operations can be selected and performed.

As an illustration, the result shown below is obtained using a threshold of 6000 in **Step 1**, with 172 grain segments remaining. In **Step 2**: as a larger opening (square size = 16) is used, some smaller grains are lost (marked in green); as a smaller opening (square size = 15) is used, some grains tend to be under-segmented (marked in cyan).

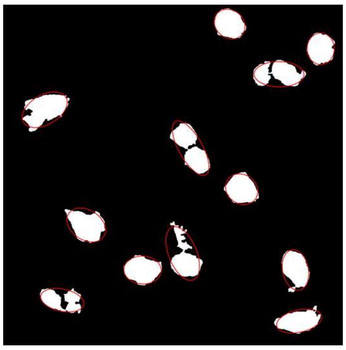
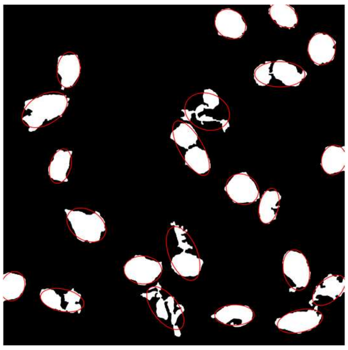
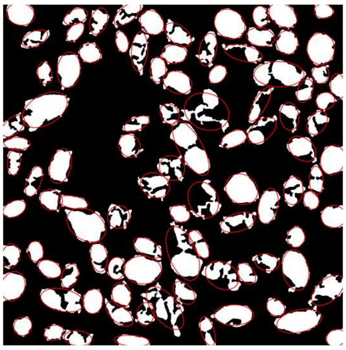
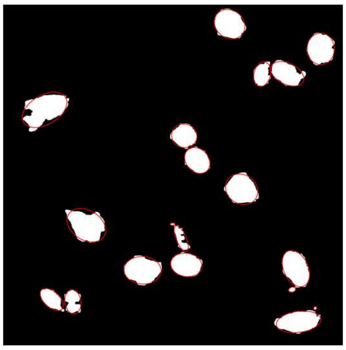
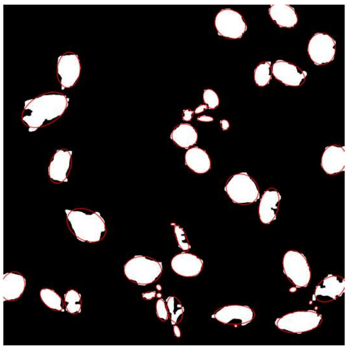
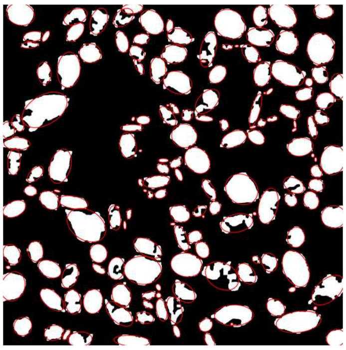
Step 1: Threshold = 6000



Step 2: Opening, square size = 15 **Step 2:** Opening, square size = 16



The table below shows the results (from Level #1) using different thresholds in **Step 1**, with the same opening size (16×16 pixels) used in **Step 2**.

Step 1	Threshold = 35,000	Threshold = 24,000	Threshold = 10,000
Result			
Step 2	Opening, square size 16	Opening, square size 16	Opening, square size 16
Result			

3.2.2 Subsequent Levels

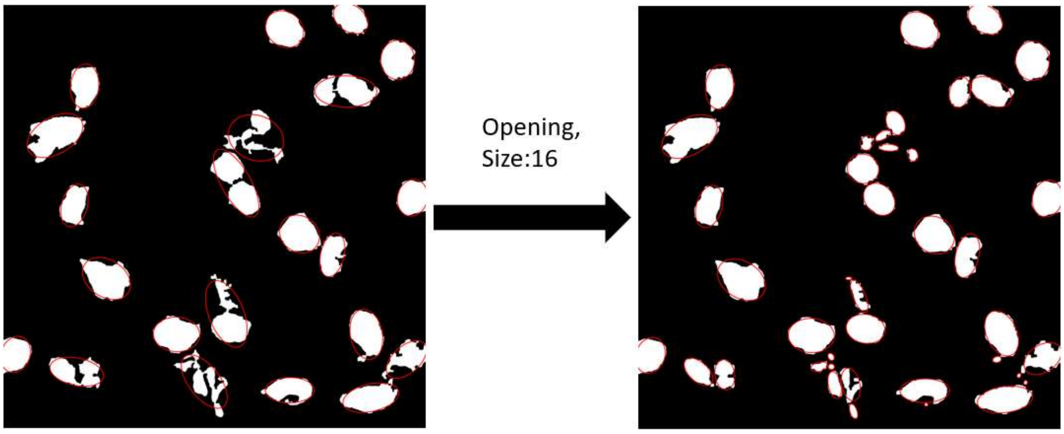
After a level of morphological grain segmentation is done, enter the next level to process smaller grain segments. Repeat for subsequent levels until a minimum resolvable size is reached.

The following table shows the thresholds (for **Step 1**) and opening sizes (for **Step 2**) used in the example demonstrated below. These parameters are not necessarily optimal. Users may need to adjust the number of levels and the parameter values to meet their specific needs.

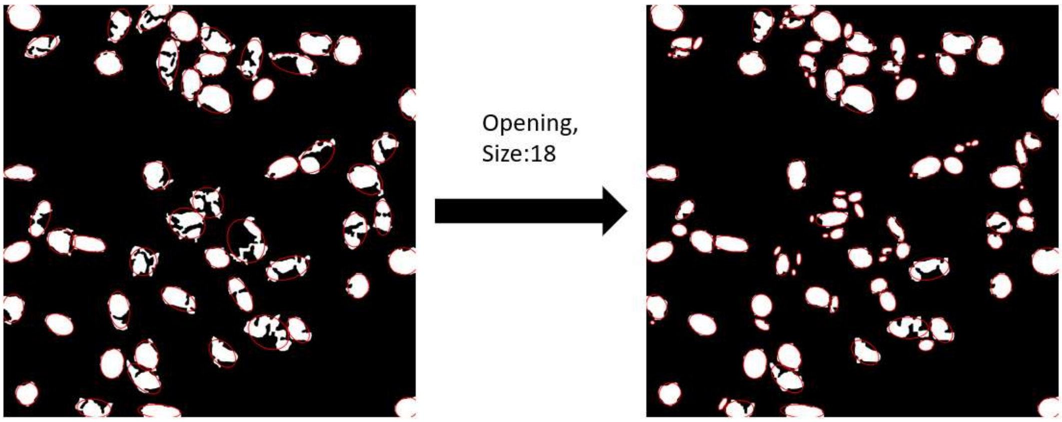
Level	**Step 1: Threshold**	**Step 2: Opening operation**
#1	24,000	Square size 16
#2	12,000	Square size 18
#3	6,000	Square size 19
#4	800	Square size 14

The outcomes are shown as follows for individual size levels.

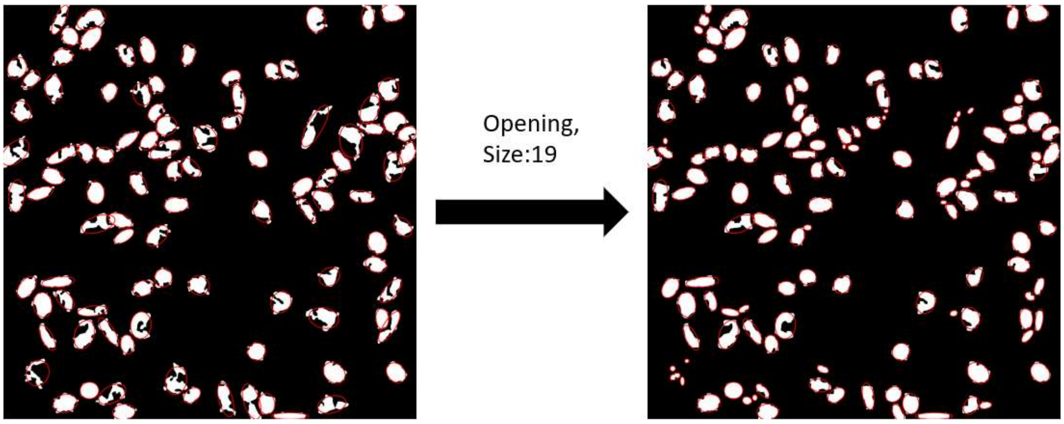
Level #1: Threshold = 24,000



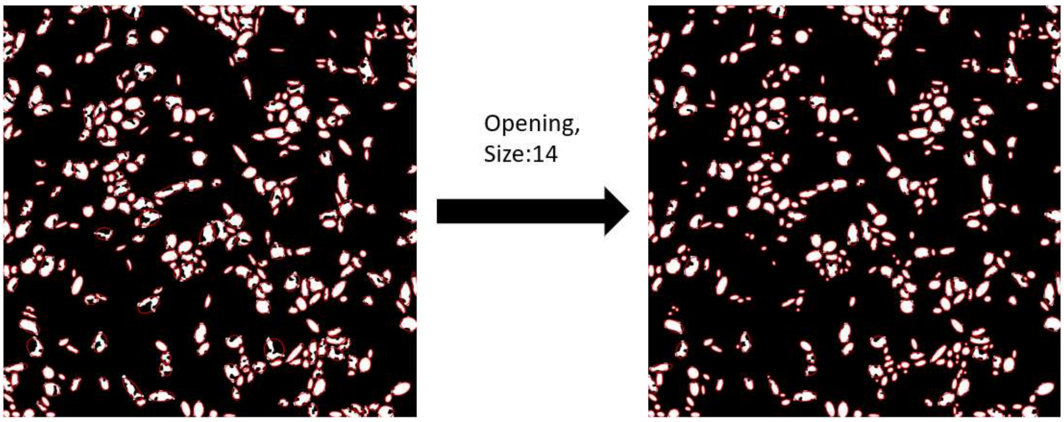
Level #2: Threshold = 12,000



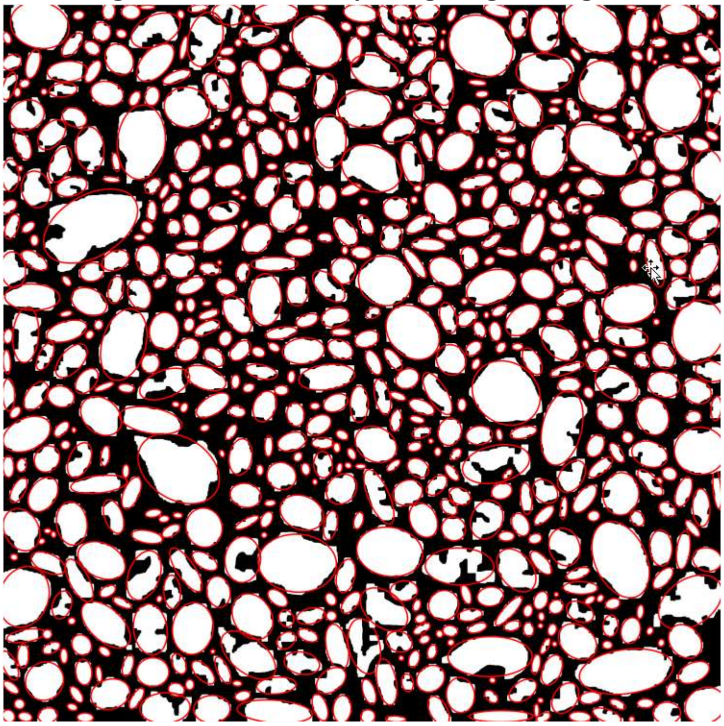
Level #3: Threshold = 6,000



Level #4: Threshold = 800

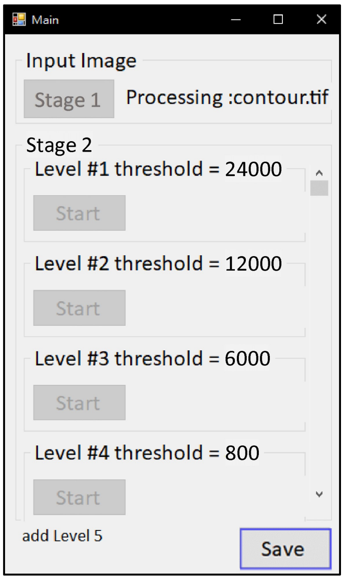


After Stage 2: Four-level morphological grain segmentation



3.2.3 *Export Results*

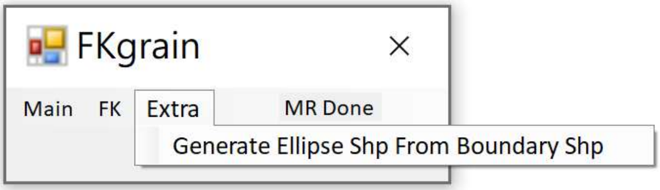
Press "Save" in the dialogue box to export the results. The output results include two shapefiles: (1) result.shp is the grain boundaries; (2) result_ellipse.shp is the ellipse fits. These shapefiles can be retrieved by the GIS software. In the attribute table of the shapefile are the coordinates (x, y) of the ellipse centers, lengths of the major and minor axes (*a*- and *b*-axes) and orientation angles of the *a*-axes. Their units are consistent with those of the input DEM.



名稱	修改日期	類型	大小
result.dbf	2021/1/25 上午 12:38	DBF 檔案	101 KB
result.shp	2021/1/25 上午 12:38	SHP 檔案	2,491 KB
result.shx	2021/1/25 上午 12:38	SHX 檔案	6 KB
result_ellipse.dbf	2021/1/25 上午 12:38	DBF 檔案	101 KB
result_ellipse.shp	2021/1/25 上午 12:38	SHP 檔案	3,653 KB
result_ellipse.shx	2021/1/25 上午 12:38	SHX 檔案	6 KB

	A	B	C	D	E	F
1	ID	x	y	a	b	tilt
2	1	0.1700	1.0450	0.5220	0.4136	108.9381
3	2	0.7275	4.1499	0.8636	0.5187	-30.3067
4	3	0.8550	0.8125	0.4992	0.2726	42.5775
5	4	1.0050	3.1574	0.6197	0.3650	105.6599
6	5	1.1525	4.8524	0.6292	0.4018	94.8379
7	6	1.2275	0.7725	0.4316	0.2817	82.1758
8	7	1.4650	2.1199	0.7357	0.5249	32.2649
9	8	2.4574	1.3275	0.6580	0.4897	3.1818
10	9	2.5699	0.8850	0.2428	0.1323	-13.0248
11	10	2.7374	1.0250	0.1104	0.0846	57.2052
12	11	2.7899	0.6250	0.3929	0.1842	77.9055
13	12	2.7424	0.8850	0.0886	0.0819	11.4348
14	13	3.0174	0.6325	0.4711	0.2647	77.1987
15	14	3.1924	3.6924	0.4507	0.4155	15.9243
16	15	2.9874	2.1374	0.0875	0.0430	0.3449
17	16	3.2299	1.4225	0.5571	0.4148	9.2431
18	17	3.0574	0.2500	0.2201	0.1047	71.4185

Alternatively, a shapefile of ellipse fits and the corresponding attribute table can be generated using an existing grain-boundary shapefile (e.g., digitized grain boundaries). This can be done by clicking Extra tab in the Menu bar and selecting an input shapefile of grain boundaries.



The maximum amount of data that can be processed by FKgrain is 70 million points, which is the upper limit of FK computation in the FORTRAN code. This limit is equivalent to a DEM of area 84 m × 84 m with 1 cm grid resolution. The processing time required, however, is determined principally by the CPU power. Two tasks in FKgrain require intensive use of CPU, namely: (1) calculation of variogram, and (2) generation of FK DEMs. Task 1 is to compute the semivariance of DEM between all paired grids for a full range of spatial lag h (or separation distance). Task 2 is to generate the short- and long-range FK DEMs by solving a system of $2(N+1)$ equations at each grid point, where N = number of data pairs separated by lag h. For instance, in the example presented above, an area of 6 m × 6 m with 1 cm grid resolution requires 17 min of CPU time (10 min for task 1; 6 min for task 2), using a PC with an Intel Core i7-7700K 4.20 GHz CPU and 32 GB RAM. With the same resolution, an area of 12 m × 12 m (4 times of 6 m × 6 m) requires 2 hr of CPU time (92 min for task 1; 31 min for task 2), while an area of 18 m × 18 m (9 times of 6 m × 6 m) requires ~9 hr of CPU time (415 min for task 1; 103 min for task 2). The CPU time required, thus, grows as a second-degree polynomial function of the number of data points.