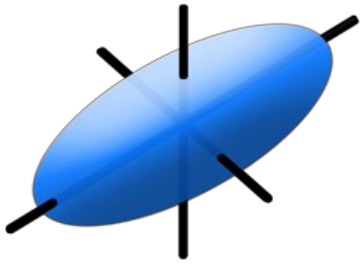
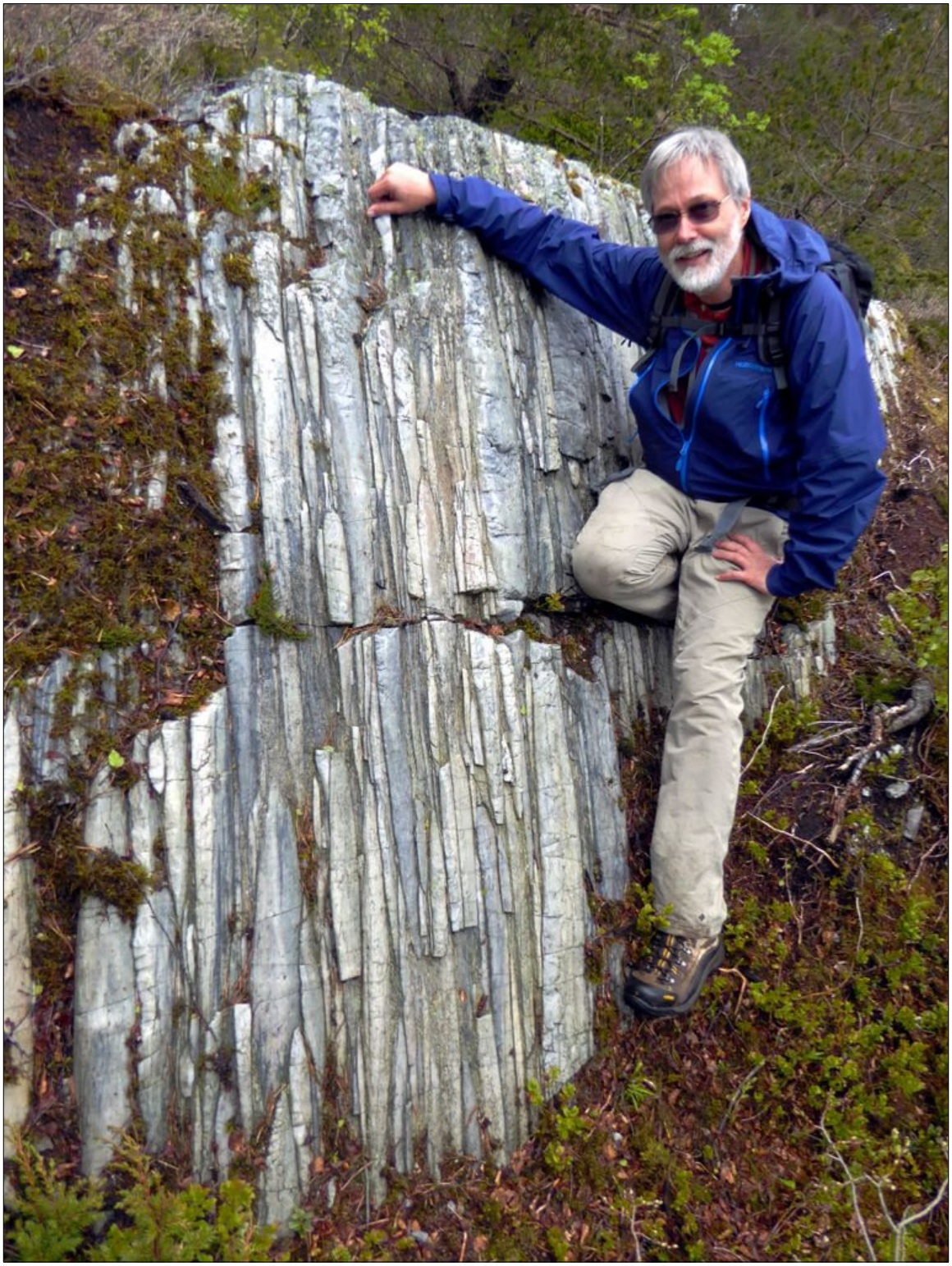


EllipseFit

EllipseFit 3.3.0
Strain Analysis Software
User Manual

Frederick W. Vollmer



Frontispiece. Strained quartz cobble conglomerate at Sandviksfjellet, Bergen, Norway. The prolate, cigar-shaped, cobbles indicate a highly constrictional strain (Holst and Fossen, 1987; Fossen, 1988). Photograph by E. Lubicich.

Table of Contents

License and Citation

License

Citation

EllipseFit is the result of many hours of work over several decades. Algorithms used in the program come from numerous sources, however most have been developed by the author, some of which have not yet been published and are the subject of papers in preparations. I have released the program publicly with the hope that the structure and tectonics community will find it useful, and ask forgiveness for the limited documentation, as well as respect for publication priority.

In return for free use, any significant use of the software in analyzing data or preparing diagrams should be cited in publications, presentations, or other works. Appropriate references for the software and user manual are:

Vollmer, F.W., 2017. EllipseFit 3.3.0 Strain Analysis Software. http://www.frederickvollmer.com/ellipsefit/.
Vollmer, F.W., 2017. EllipseFit 3.3.0 Strain Analysis Software User Manual. http://www.frederickvollmer.com/ellipsefit/.

References for specific techniques are (see *References* for citations):

Vollmer (2010) discusses ellipse and ellipse fitting techniques, including Shan's method, and their implementation in EllipseFit.
Vollmer (2011a) discusses methods for contouring finite strain on the unit hyperboloid and the use of hyperboloidal stereographic, equal-area and other projections for strain analysis.
Vollmer (2011b) discusses best-fit strain from multiple angles of shear and an analytical solution to the Wellman diagram.

An acknowledgment, such as “I thank Frederick W. Vollmer for the use of his EllipseFit 3 software.” is appreciated.

Registration

Please consider registering the software, registration is free and helps me determine the software usage and justify the time spent in it's upkeep. To register, simply send an email to me at vollmerf@gmail.com with your user name, affiliation, and usage. I will send you an email in reply with my thanks, and will not place you on a mailing list. For example, send me an email with something like:

User: Frederick Vollmer
Affiliation: SUNY New Paltz, Geology Department
Usage: Undergraduate structural geology course and research

I am happy to take emails with questions and suggestions, either at the university (SUNY New Paltz) or at the gmail address used on my website. However I am not reliable about checking email, so please forgive me if I am slow in answering, I will try to respond in as timely a fashion as possible.

1. Introduction

EllipseFit is an integrated program for geological finite strain analysis. It is used for determining two and three-dimensional strain from oriented photographs, and is designed for field and laboratory based structural geology studies. The graphical interface and multi-platform deployment also make it ideal for introductory or advanced structural geology laboratories. I use the software to teach structural geology at SUNY New Paltz, where hundreds of students have used it in laboratory and field studies. EllipseFit is currently implemented for Windows, Macintosh, and Linux platforms.

EllipseFit is suitable for determining two and three dimensional strain using various objects including center points (Fry analysis), lines, ellipses, and polygons. Polygons include ooids, pebbles, fossils, or particles of *any* initial shape. The analysis of strain from polygons is widely applicable to many rocks in thin section, hand sample, or suitable outcrops. EllipseFit allows digitizing polygons directly, or indirectly by using a flood fill method. EllipseFit converts them to moment equivalent ellipses, and the mean ellipse is equivalent to the strain (Mulchrone and Choudhury, 2004). Given three or more oriented sections EllipseFit can calculate the three dimensional strain using the method of Shan (2008).

This User Manual was initially prepared for a strain workshop at the *2014 Structural Geology and Tectonics Forum*, at the Colorado School of Mines with Paul Karabinos and Matty Mookerjee, but is still a work in progress. Version 2 is stable and has been widely used, including for a strain workshop at the *2012 Structural Geology and Tectonics Forum* at Williams College, however no updates are planned for EllipseFit 2.

I am a professor of structural geology, and have taught for over 30 years at SUNY New Paltz. I had the luck to be introduced to analytical structural geology as a student, and am grateful to my mentors Rob Twiss at UC Davis, Win Means at SUNY Albany, and Peter Hudleston at U Minnesota whose clear thinking inspired me. I was introduced to programming as a grade school student, when my dear mother required me to take a summer school course. I subsequently joined the Computer Club, one of three members, spent countless hours on the terminal connected remotely to a mainframe, and became obsessed with coding.

The final version of EllipseFit 1 was completed in the 1989 in C++ for Macintosh, in part based on code from a Fortran program written on punch cards for Win Means. Version 2 was written in cross platform RealBasic, however issues with licensing, performance, and the closed source code led me to abandon that language. Version 3 is fully rewritten, with tens of thousands of lines of code, in Free Pascal, a professional open source compiler. This allows improved code with better speed and extensibility, and the potential to port to additional platforms.

1.1 Installation

On Macintosh OS X, double click the disk image file (.dmg), and drag the EllipseFit application to your Applications folder, or other desired location.

On Windows, unzip the zip file (.zip) using the *Extract All* option, and drag the EllipseFit folder to any desired location. The EllipseFit folder contains the EllipseFit application (EllipseFit.exe), and a “Resources” folder which is required. Please make sure to entirely extract the EllipseFit folder from the zip file, this is the most common installation problem.

On Linux unpack the gzip file (.tar.gz), and copy the EllipseFit folder to any desired location. The EllipseFit folder contains the EllipseFit application (ellipsefit), and a “Resources” folder which is required. An application icon (ellipsefit.png) is included in the Resources folder if desired for installation.

There is also a folder of example data and images to show how data is formatted, these are referred to in this guide. After installing a new version it is recommended that you reset the preferences using the “Reset Preferences” command in the Help menu. This will clear any options that may have changed and set them to default values. The preferences are stored in the file EllipseFit3.xml, which is located in the folder EllipseFit in your operating system's application preferences folder. To deinstall simply delete the EllipseFit application folder, and optionally delete the preference folder. No other files are installed on your computer. No administrative permissions are required to install EllipseFit, and it is possible to keep a copy on a thumb drive to run on any computer.

1.2 Example Data Files

The included example files and images can be used to determine input data formats. These are simple files that can be generated using a text editor or spreadsheet. EllipseFit 3 will read comma separated (csv), tab separated (tsv), and Open Document (ods) formats. The header line indicates the type of data required in each column. The included example files are named to indicate their contents (this is not required, EllipseFit will examine the headers to determine the available data, and extra columns are ignored):

E2 - Ramsay and Huber 1983 (small).csv
E2 - Ramsay and Huber 1983 (small).jpg
E2 - Ramsay and Huber 1983 (large).jpg
Example ellipse data and thin section photomicrograph (from Ramsay and Huber, 1983). This data type can contain (X, Y) coordinates for Fry-type analyses, or complete ellipse data including (X, Y, A, B, R, Phi) axes data. Note that there are small and large versions, I use the large version, which does not include a data file, for teaching.

E3 - Hossack 1968.csv
Example ellipsoid data (from Hossack, 1968) with (A, B, C) axes data for Flinn and Nadai plots.

ES - Owens 1984.csv
Example ellipse section data (fron Owens, 1984) for calculating the three-dimensional strain ellipsoid from three or more faces using Shan's (2008) method. The strikes and dips of each section must be included.

LA - Ragan 1985 F10.1a.csv
LA - Ragan 1985 F10.1a.png
Example line angular shear data and image (from Ragan, 1985) for analytical Wellman-type analysis (Vollmer, 2011). Each data point requires the endpoints of two lines that originally had a constant angle. This is an analytical solution to the classic multiple brachiopod problem illustrated in a number of structural geology texts.

LS - Ragan 2009 T14.9.csv
Example line stretch data for lines with known initial and final lengths, such as boudins and folds.

EllipseFit does not yet provide digitizing of this type of data. Please contact me if this would be of interest. Note that the LS data is from fold flattening index example (Ragan, 2009), which is mathematically related.

MLLF Test 60.csv
Sample of 60 points used to test the maximum mean log likelihood function (MLLF) method of Shan and Xiao (2011).

Field	Alternate	Definition
ID	N	Unique integer id for measurement.
X		X coordinate of point or ellipse center.
Y		Y coordinate of point or ellipse center
Max	A	Maximum radius of ellipse or ellipsoid.
Int	B	Intermediate radius of ellipsoid.
Min	C	Minimum radius of ellipse or ellipsoid.
Area		Area of ellipse (read only) = ACπ
R		Ellipse ratio = A/C
Phi	ϕ	Orientation of ellipse long axis clockwise from X axis.
Strike		Strike of section.
Dip		Dip of section.
R Calc		Calculated R from ellipsoid section.
Phi Calc		Calculated ϕ from ellipsoid section.
R Res		Residual between calculated R and input ellipsoid section R.
Phi Res		Residual between calculated ϕ and input ellipsoid section ϕ.
X11, Y11		First endpoint of first line of pair.
X12, Y12		Second endpoint of first line of pair.
Alpha1		Directed angle of first line clockwise from X axis.
X21, Y21		First endpoint of second line of pair.
X22, Y22		Second endpoint of second line of pair.
Alpha2		Directed angle of second line clockwise from X axis.
Beta		Directed angle between two lines of pair.
Li		Initial line length.
Lf		Final line length.
Stretch		Length of line.

Table 1 Spreadsheet field headers used in EllipseFit.

2. Overview of Strain Analysis

When attempting to unravel the history of a mountain belt, one starts with an outcrop or a hand sample. The lithology, textures, and mineralogy give clues to the past sedimentary environment, the temperature and pressure history, and geochronology gives the dimension of time. Strain analysis gives another dimension, a measure of the deformation enjoyed during that history.

Geological strain analysis and theory is an important aspect of structural geology that is covered in numerous textbooks (e.g., Means, 1976; Hobbs, Means, and Williams, 1976; Ragan, 1985; Marshak and Mitra, 1988; Van der Pluijm and Marshak, 2004; Pollard and Fletcher, 2005; Twiss and Moores, 2007; Ragan, 2009; Fossen, 2010). Ragan (2009) and Ramsay and Huber (1983) provide excellent overviews of techniques for the analysis of strain in deformed rocks.

Strain markers can be grouped into three general categories (Ramsay and Huber, 1983; Lisle, 2010; Mulchrone, 2013):

Geometric - Objects or groups of objects with known pre-strain geometries
Ellipse fitting - Objects whose shape, including irregular polygons, can be approximated by ellipses
Point distribution - Collections of objects whose spacial arrangement can be used to determine strain

Geometric methods apply to fossils and other objects of known unstrained geometry, including angular measurements, to which equations of finite strain can be applied (e.g., Ramsay, 1967; Ramsay and Huber, 1983). These techniques are useful for specific locations or samples (e.g., Wellman, 1966; Waldon, 1988), but are less broadly applicable than the other two. EllipseFit implements an analytical Wellman method (Vollmer, 2011), and a method where multiple line stretches (as from folds and boudins) are known (Chapter 4).

Ellipse fitting methods apply to samples such as sandstones and conglomerates, as well as collections of irregular clasts or fossils (Mulchrone and Choudhury, 2004), so these techniques are very broadly applicable. EllipseFit includes numerous procedures to collect and analyze this type of data (Chapters 5). *Point distribution* methods include Fry (Fry, 1979) and nearest neighbor (Ramsay, 1967) methods, for which EllipseFit includes numerous procedures (Chapter 3).

The following chapters discuss techniques of strain analysis that are implemented in EllipseFit in terms of the type of data collected: *points, lines, ellipses*, and *polygons*. Points are the simplest type of data collected, however, as discussed in Chapter 3, *Strain from Points*, it can be difficult to objectively extract strain from point distributions. The analysis of line data depends on the known initial lengths of, or angles between, lines, and has important applications for some data as discussed in Chapter 4, *Strain from Lines*.

Chapter 5, *Strain from Ellipses and Polygons*, covers ellipse data, which is collected assuming that particles, such as sand grains, initially approximated a collection of random spheres or ellipsoids. It turns out, however, that ellipse data is a subcategory of polygon data. An important mathematical proof (Mulchrone and Choudhury, 2004) shows that all particles, *of any shape*, that can be assumed to have been initially randomly oriented, can be used to calculate strain. This allows numerous geological objects to be used for strain analysis using objective calculations developed for ellipse analysis.

Chapter 6, *Ellipse Data Plots* covers graphical techniques for two-dimensional strain plots, including R_f

ϕ plots and polar Elliott plots, which are types of *hyperboloidal projections*. Hyperboloidal projections are analogous to spherical projections, such as the stereographic and equal-area projections that are used to create stereonets and Schmidt nets respectively, familiar to students of structural geology.

Chapter 7, *Mean Ellipse Calculation*, discusses the calculation of a mean ellipse from a sample of ellipses. As discussed in Chapter 5, these calculations apply to polygons as well as ellipses, as the use of polygon moment equivalent it ellipses removes the requirement that particles were initially elliptical.

The techniques mentioned thus far are related to two-dimensional strain analysis. Chapter 8, *Ellipsoid Calculation*, covers the more complex steps involved in determining three-dimensional strain ellipsoids from oriented sections for which the two-dimensional strain ellipse has been determined. Chapter 9, *Ellipsoid Plots*, covers strain plots used to display this type of data, Flinn and Nadia plots.

Chapter 10, *Data Transformation* discuses methods for transforming data sets, including *unstraining* or *retrodeforming* data sets and images to their pre-deformation state. Chapter 11, *Data Synthesis*, covers data synthesis for making artificial samples from random populations. Chapter 12, *Image Analysis* discusses image analysis techniques, including filtering and edge finding, that can aid in highlighting particle edges prior to digitizing.

It is essential to be aware of the assumptions involved in strain analysis. Refer to the referenced texts for a complete discussion. An important consideration is whether the particles, such as fossils or clasts, record the same deformation as the rock. In general, this means whether there was a viscosity contrast between the particles and the matrix that encloses them. This is discussed briefly in Chapter 3.

A second problem to consider is whether there was an initial preferred orientation of the particles, this can be related to an initial sedimentary fabric, or compaction. Unimodal, or orthogonal, sedimentary fabrics and compaction essentially apply a “deformation” that is indistinguishable from a tectonic deformation without additional information. Detection of initial fabrics is discussed briefly in Chapter 7. Similarly, volume change is difficult to quantify, and strain is generally calculated with volume equivalent to an initial unit sphere.

This User Manual is written in a tutorial fashion, in order to become acquainted with the program, it is a good idea to work through the examples provided. This User Manual is also not yet finished, it is a work in progress.

3. Strain from Points

This chapter discusses *point distribution* methods. It is common in nature for objects to be distributed randomly, but with some minimum cutoff distance between them. A random distribution in space follows a *Poisson distribution* (e.g., Davis, 1986), which might be imagined by throwing pingpong balls into an empty room. However, in this case the centers of the pingpong balls can never touch, giving a cutoff distance of twice the radius of the balls. If the room is filled with ping pong balls, it becomes a three-dimensional packing problem.

The basic idea for methods utilizing point distributions (e.g., Ramsay and Huber, 1983) is that the distance between the initial object centers is the same in all directions, thus defining a circle in two dimensions or a sphere in three dimensions. After deformation the particle centers are closer in some directions and further in others, and the post-strain point distribution defines an ellipse or ellipsoid.

Examples of this type of data include the centers of clasts in sedimentary rocks such as sandstones and conglomerates. The centers of phenocrysts in igneous rocks, where nucleation of crystals is prevented in proximity to existing crystals due to the chemical gradient, is another example. Two dimensional examples include the centers of skolithos burrows and dewatering structures.

If the viscosity of the particles is identical to the viscosity of the enclosing matrix, point distribution methods should give similar results as ellipse fitting methods discussed in Chapter 5. If the particles have a different viscosity, or are even perfectly rigid, it is possible to get an estimate of the strain of the rock independent of the particle shape. It may be useful, then, to compare the results of both point distribution and ellipse fitting methods.

Two general point distribution methods have been proposed, a *nearest neighbor* approach (Ramsay, 1967; Ramsay and Huber, 1983), and an *all object separation* approach (Fry, 1979), commonly referred to as the *Fry* method. The latter, initially graphical, approach has many variations, one of the most common is the *enhanced Fry* method (Erslev, 1988; Erslev and Ge, 1990). It is important to note that the enhanced Fry method requires the particle shape, and therefore loses independence from *ellipse fitting* methods.

The nearest neighbor approach (Section 3.3) has been enabled computationally by the availability of Delaunay triangulation algorithms (e.g., Preparata and Shamos, 1985). This approach was initially used in EllipseFit 1 (Vollmer, 1989), and has been developed extensively by Mulchrone (Mulchrone, 2003; Mulchrone, 2013).

The main problem in point distribution analysis is to determine the strain ellipse from the *central void*. The *enhanced normalized Fry* method (Erslev and Ge, 1990) was developed to solve this, but requires the particle ellipses (Section 3.2). As discussed above, this blurs the distinction between *point distribution* and *ellipse fitting* methods. A number of solutions to this void fitting problem have been proposed (e.g., Lisle, 2010; Shan and Xiao, 2011; Waldron and Wallace, 2011; Mulchrone, 2013), EllipseFit implements a number of these methods with additional variations.

3.1 Fry Analysis

A *Fry analysis* (Fry, 1979) is an important and widely used technique for analyzing this type of data, and there is an extensive literature on it and its variations (e.g., Hanna and Fry, 1979; Crespi, 1986;

Onasch, 1986; Erslev, 1988; Erslev and Ge, 1990; Dunne, Onasch, and Williams, 1990; McNaught, 1994; McNaught, 2002; Shan and Xiao, 2011; Waldron and Wallace, 2011; Mulchrone, 2013).

A Fry analysis can be simply done with two pieces of tracing paper, by tracing all of the particle centers on one sheet, then drawing a center point on a second sheet overlain on the first, and then sequentially moving the center point to each point and trace each point. For *n* initial points, this generates:

$$n_f = n! / (2 * (n - 2)!)$$

points, which is a lot of points to draw by hand. To illustrate the use of the method in EllipseFit, start EllipseFit and open the image file (*File > Open Image*):

E2 - Ramsay and Huber 1983 (large).jpg

This is a photograph of a deformed ironstone oolith in thin section from Ramsay and Huber (1983) that is widely used as a test image for strain analysis. For point digitizing make sure the *Center Point* icon (*Digitize > Center Point*) and *Add Tool* (*Digitize > Add Tool*) icons are selected, as shown in Figures 1 and 2.

Use the *Zoom In* and *Zoom Out* tools to enlarge the image, and click on one particle center. The *Data Window* will display a highlighted line of data. Before continuing, open the Fry plot (*Analyze > Fry Plot*), as shown in Figure 2.

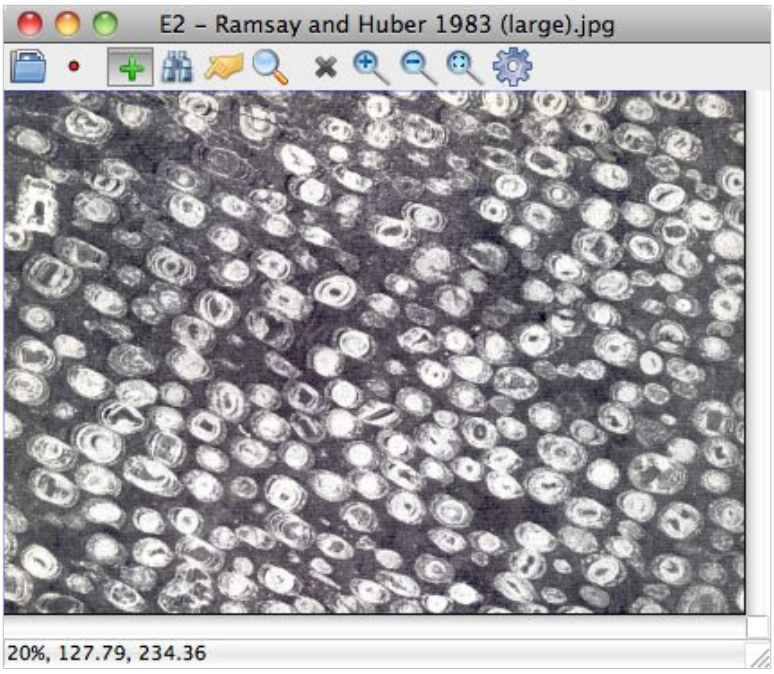


Figure 1. EllipseFit's *Image Window* used for digitizing, with photomicrograph of a deformed oolite from Ramsay and Huber (1983).

Continue digitizing point centers, you should ideally work out from one point digitizing adjacent points keeping a roughly circular area. The Fry plot will start to develop as you digitize, with each new set of

generated points highlighted (Figure 3).

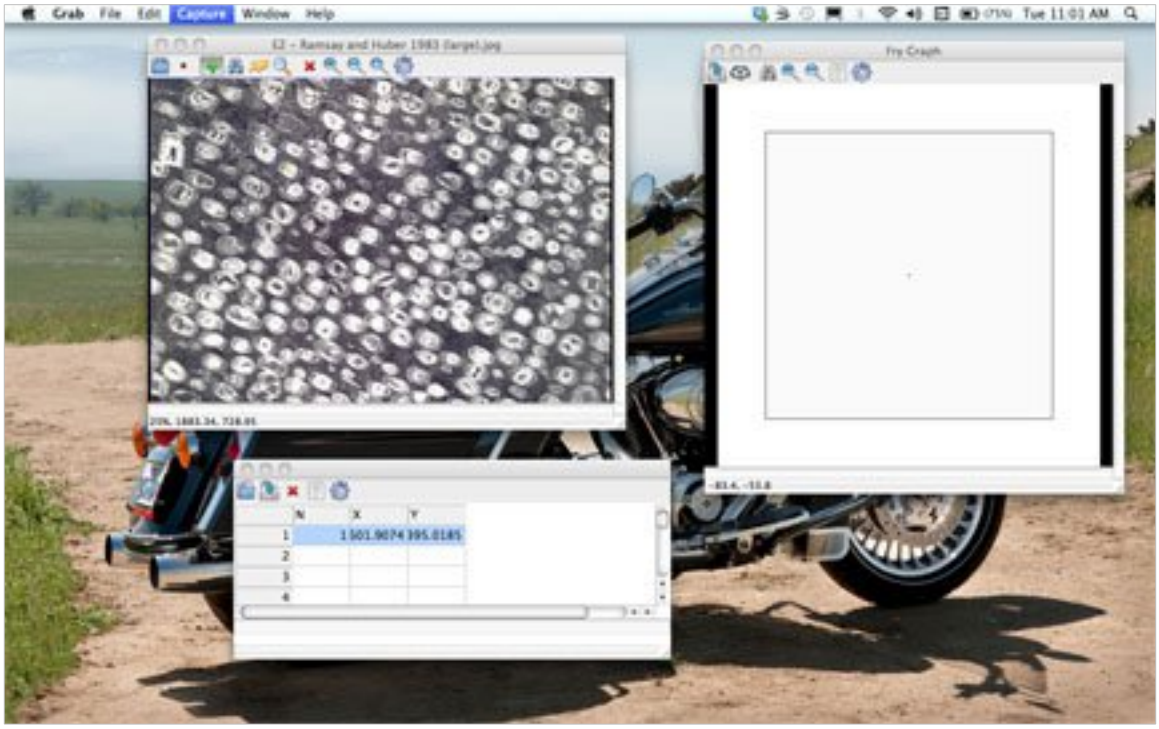


Figure 2. EllipseFit's *Image Window*, *Data Window* and *Fry Graph* displaying a single data point.

Use the *Hand Tool* (*Digitize > Hand Tool*) to scroll, and the *Zoom Tool* to zoom (*Digitize > Zoom*). You can also use the *Command* (Mac) or *Control* (Windows and Linux) + and – keys to zoom in and out. Holding down the *Shift* key allows scrolling with the cursor. Points can be deleted by using the *Find Tool* (*Digitize > Find Tool*) to highlight a point, and delete it using the *Cut* command (*Edit > Cut*). A point can also be deleted by selecting it in the *Data Window* and deleting it there. It is important to be objective, and you may wish to digitize all available points, however note that some particles may not meet the required assumptions. In particular, note that the centers of the particles in two-dimensions do not generally correspond to their three-dimensional centers, as they lie on an arbitrary plane cutting through the rock, so the assumption of of a uniform cutoff distance is weakened. This is discussed further in Section 3.2, Normalized Fry Analysis.

It is also desirable to select approximately equal numbers of particles in all directions, so the point density is not biased by direction. This is one reason to maintain a uniform point density in a circular area while digitizing, and why having the interactive Fry plot open can assist in particle selection. This is discussed further in Section 3.3.

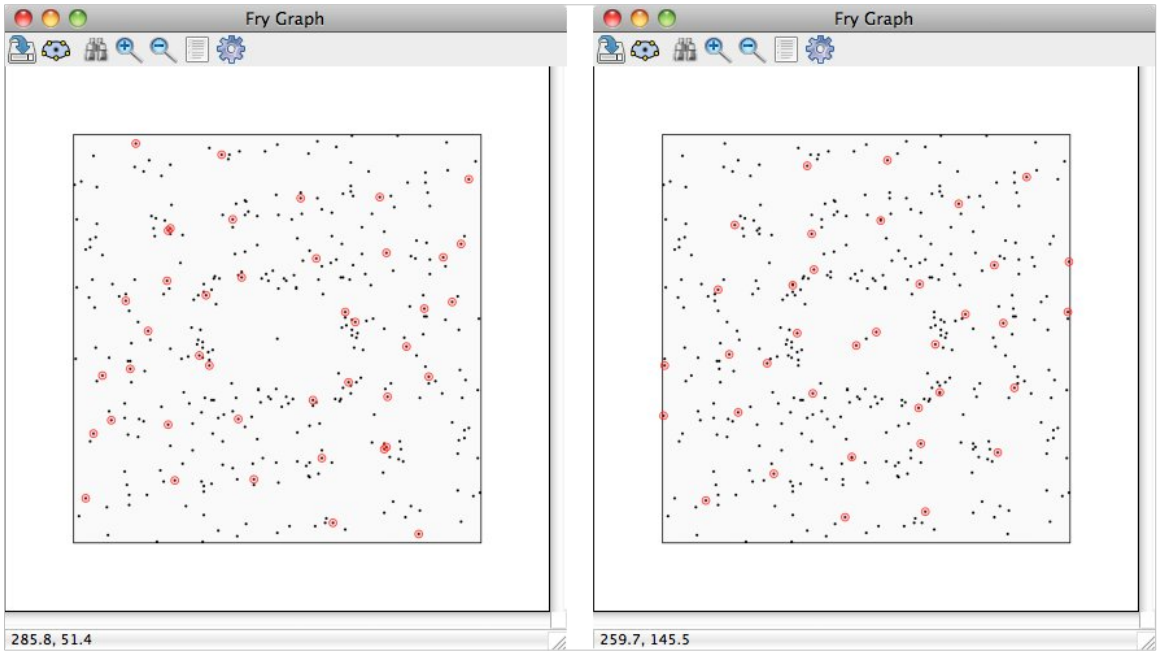


Figure 3. Fry plot after digitizing 20 adjacent particle centers. The generated points are highlighted. On the right, note the presence of the spurious data point (each point is mirrored about the center) generated by clicking too close to an existing point, i.e. an operator error which can be deleted.

If you wish to change the size of the digitized points, click the *Preferences* icon from which you can set most of the EllipseFit preferences. Note some selections have multiple pages, use the *Symbols* and *Settings* buttons toggle them. You can preview the effect of preference changes before setting then with the *OK* button.

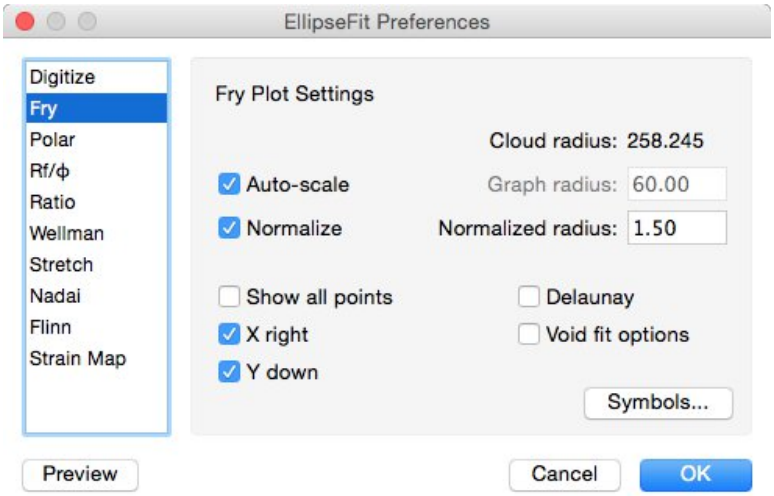


Figure 4. The EllipseFit *Preferences Dialog* where most preferences are set. The Symbols button displays an additional page.

To view the data as a *Strain Map* select *Analyze > Strain Map*. This displays the data as particle centers, this population can be strained and unstrained as described in Chapter 10, Data Transformation.

Figure 5 is the plot after carefully selecting 60 particle centers, a probable minimum number for analysis (Shan and Xiao 2011), and after digitizing 252 points, essentially all of them.

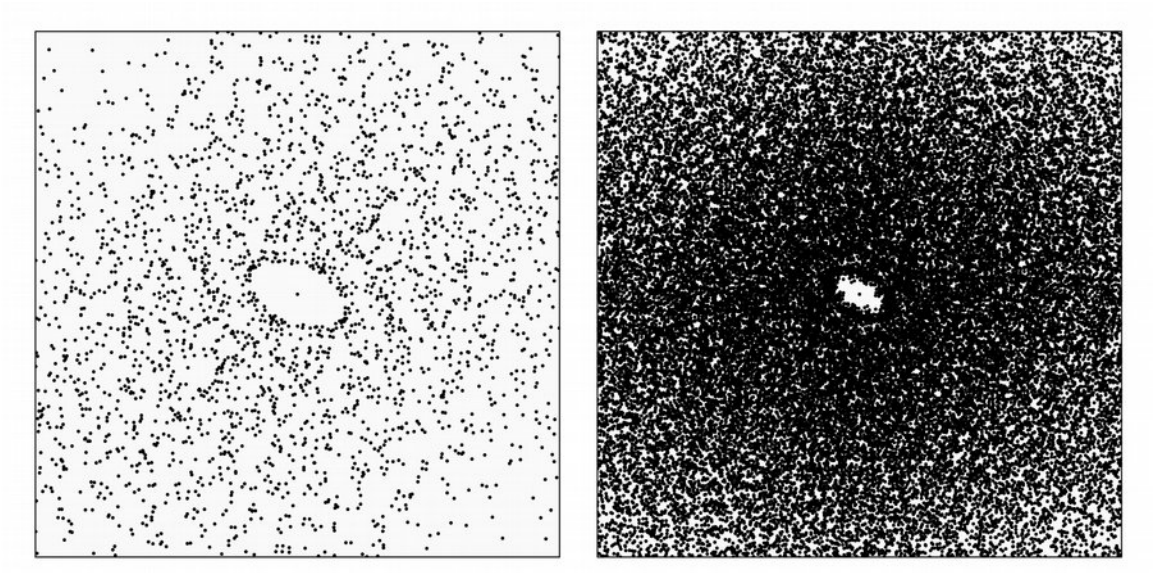


Figure 5. Fry plots after digitizing 60 carefully selected points, and after digitizing 252 points, essentially all of them. These images are PNG files as saved from EllipseFit.

To zoom in for a better image of the central void, open the *Preferences Dialog*, uncheck *Auto-scale*, and enter a number smaller than the displayed *Data radius* (Figure 6).

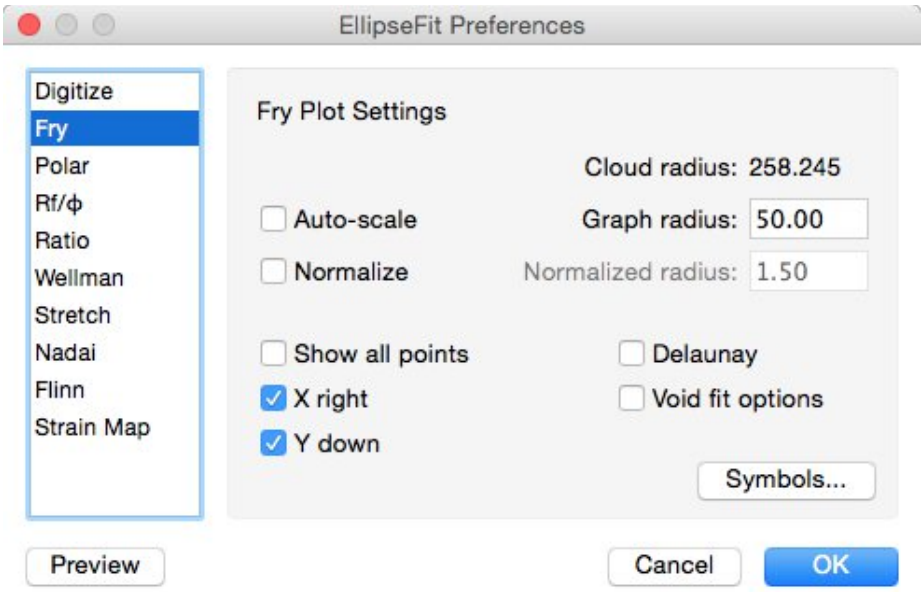


Figure 6. Set the plot radius to display the central void by unchecking *Auto-scale*, and entering a smaller radius.

Figure 7 shows the zoomed in central voids for the two examples. The next step is to determine the *best-fit* ellipse for the central void displayed in Figure 7. This can be a subjective process, and objectively choosing this ellipse is the subject of a number of papers (e.g, Erslev, 1988; Erslev and Ge, 1990; Shan and Xiao, 2011; Waldron and Wallace, 2011; Mulchrone, K.F., 2013).

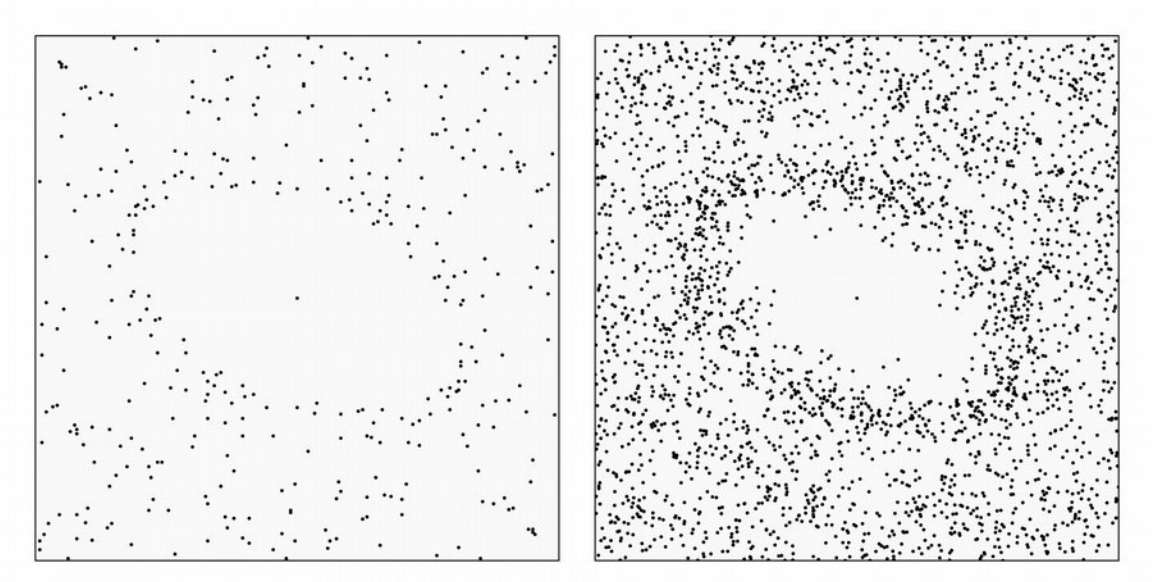


Figure 7. Close up of the central voids for the two data examples of 60 and 252 points.

The *enhanced normalized* Fry method (Erslev, 1988; Erslev and Ge, 1990) is one that is commonly employed, but requires the digitized ellipses of each particle. The normalized Fry method is the subject

of Section 3.2. Ideally a method should require only the point data (e.g., Shan and Xiao, 2011; Waldron and Wallace, 2011; Mulchrone, K.F., 2013). EllipseFit implements five different algorithms with variations as discussed in Section 2.3.

For the purposes of this section, it will be assumed that the void has been defined well enough to pick out the void by eye, which can be a good enough estimate, and also makes a good exercise for student laboratories. Click on the *Centered Ellipse* icon (*Digitize > Centered Ellipse*), and click at the edge of the void. An orange circle marks the starting point, subsequent points are marked by a yellow circle. When finished, click on the orange circle and the ellipse will be calculated and displayed in the *Log Window*.

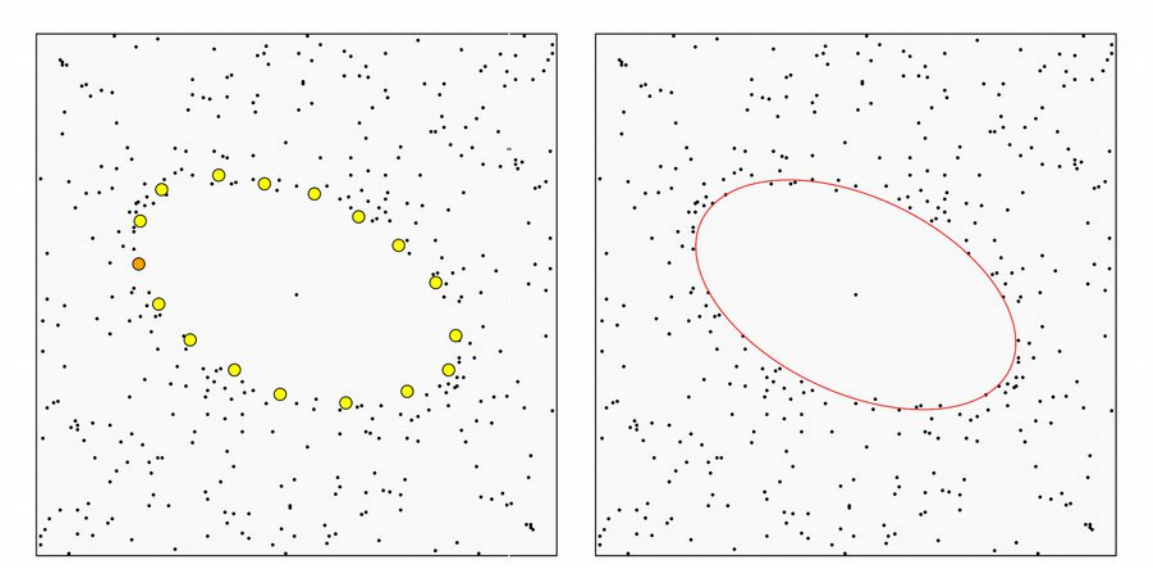


Figure 8. Digitizing the central void. The orange point is the start point, the yellow are subsequent points. Click on the orange point when finished, and the ellipse is calculated. The point size is set larger than the default size for the illustration.

For this sample, the calculated results are reported by EllipseFit as:

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-05 11:15:45
============================
N     : 60
Pairs : 1770
Time  : 00:00:00
Digitized
  Normalized : False
  Delaunay   : False
Best-Fit Ellipse
  n   = 17
  R   = 1.758
  ϕ   = 25.45°
  RMS = 0.0583
```

A centered ellipse was calculated from the 17 digitized points. The calculation is rotationally invariant, and the best fit found by minimizing the sum of the squares of the distance of the points from the ellipse, i.e., the *residuals*. The minimization is solved from the linear equations using a LU decomposition.

The RMS value is the *root mean square* measure of the variation of the residuals from the ellipse, that is the square root of the sum of the squares of the residuals of the data from the fitted ellipse. RMS is a common way to express goodness of fit of least squares solutions. It is not a measure of the error in the strain calculation, and is not technically an error. It is, however, a measure of how closely the digitized points fit the ellipse. A small RMS means that the entered points lie close to an ellipse. It makes a good class exercise for students to solve and compare their results and RMS.

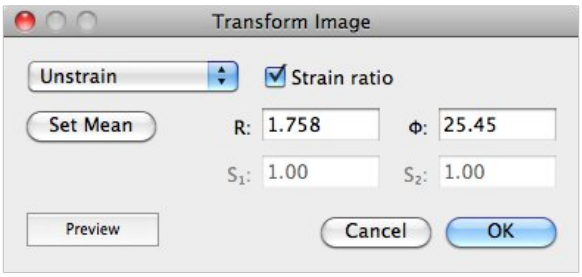


Figure 9. The *Transform Image* dialog with values entered to unstrain the mage.

As a final step in this analysis, select the *Edit > Transform Image* command and enter the results into the dialog as in Figure 9. The image will be unstrained to remove the calculated strain as shown in Figure 10.

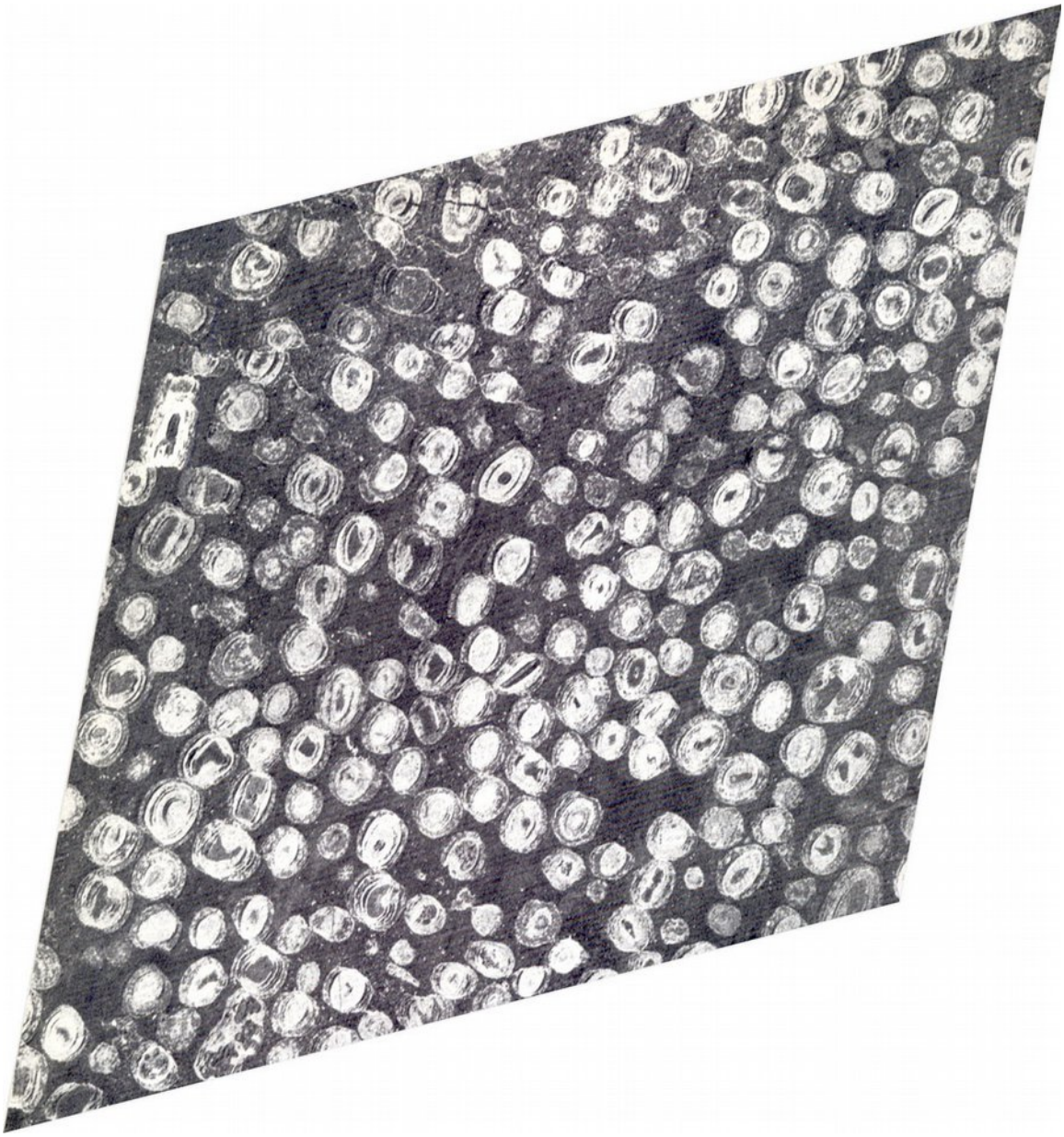


Figure 10. The oolith photomicrograph after being unstrained using EllipseFit's Image Transform command.

Next select the *Analyze* > *Transform Data* command and enter your calculated values as shown in Figure 11. Press *Transform* and then *Accept*.

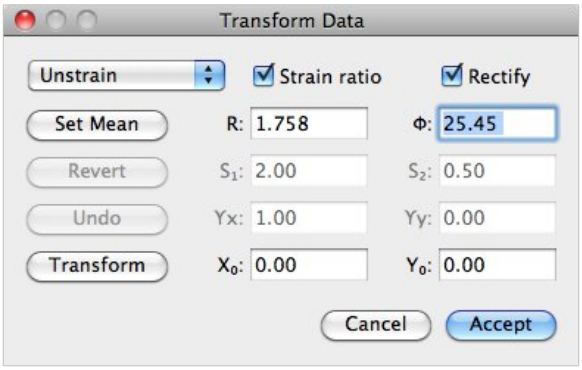


Figure 11. The Transform Data dialog with values entered to unstrain the data. *Set Mean* is only used with ellipse data. *Rectify* resolves the offsets caused by the image transformation.

The data is unstrained using the calculated values, as shown by the Fry plot in Figure 12. The *Rectify* option resolves the offsets caused by the image transformation, so the data points remain centered over the particle centers.

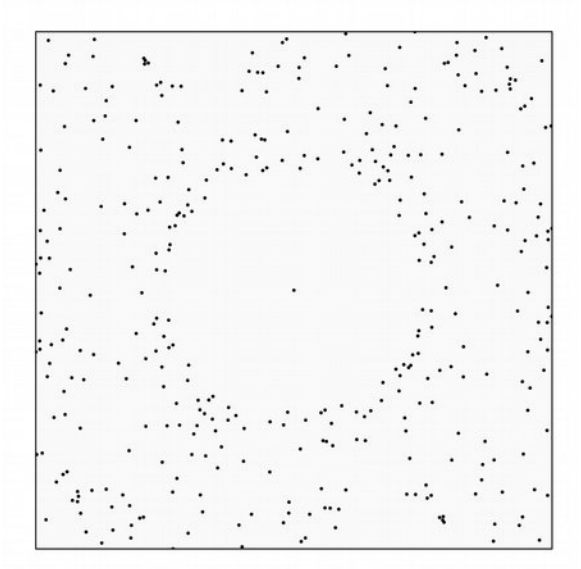


Figure 12. Fry plot of the unstrained 60 point data after using the *Transform Data* command to unstrain (retrodeform) the data using the calculated values.

3.2 Normalized Fry Analysis

As discussed in Section 3.1, the Fry analysis is a two-dimensional solution to a three-dimensional problem, since initial particles are assumed circular instead of spherical. Even if the particles have a uniform size, a section through a sample will show them as different size particles. One solution developed to overcome this is the *normalized Fry* analysis (Erslev, 1988; Erslev and Ge, 1990 McNaught, 1994; McNaught, 2002).

The distances between particles are *normalized* to account for the difference in the sizes of the particles, which can greatly improve the sharpness of the central void. Unfortunately, the ellipse sizes and orientations are required for this, and in most cases if the ellipse data is available, it should used for the ellipse fitting techniques discussed in Chapter 5, however a Fry analysis can provide different information regarding particle versus matrix strains.

The digitizing of ellipses is discussed in Chapter 5, so for an example of this analysis, open the image file:

E2 - Ramsay and Huber 1983 (small).jpg

and the data file:

E2 - Ramsay and Huber 1983 (small)

This is the 252 point data set used in Section 3.1.

The data is overlain on the image, and, if the *Find Tool* icon is selected, you can select individual particles that are highlighted in the *Data Window* and the *Fry Plot*. This selection method is implemented for most of the plots discussed in subsequent chapters. The Fry plot will look like Figure 5B.

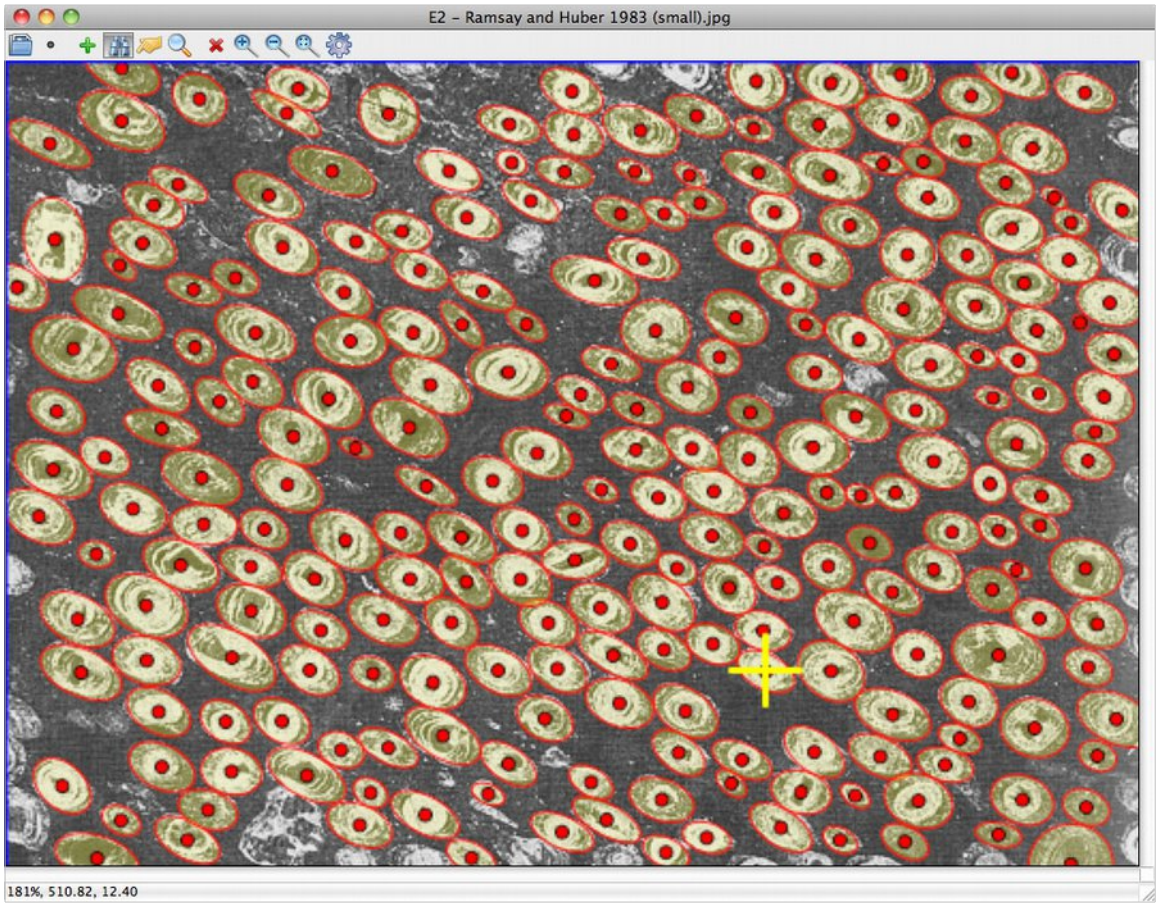


Figure 13. EllipseFit *Image* window with ellipse data overlain. Selecting the *Binoculars Icon* (as shown) allows interactive selection of particles that are highlighted in the *Data Window*, as well as on data plots including the *Fry plot*.

To zoom in on the central void, open the *Preferences Dialog* (*Gear Icon*), deselect *Auto-scale*, and enter 50 for the *Graph radius* as shown in Figure 14.

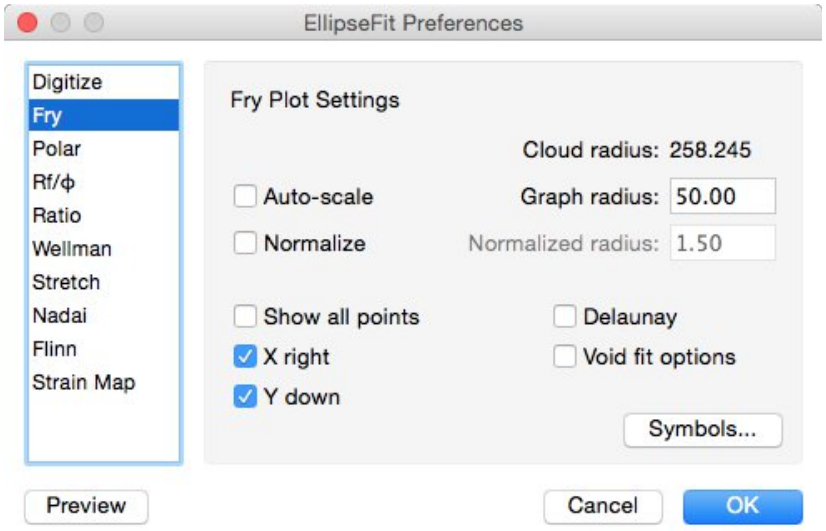


Figure 14. Settings to display the central void without normalizing.

The unnormalized plot is displayed in Figure 15.

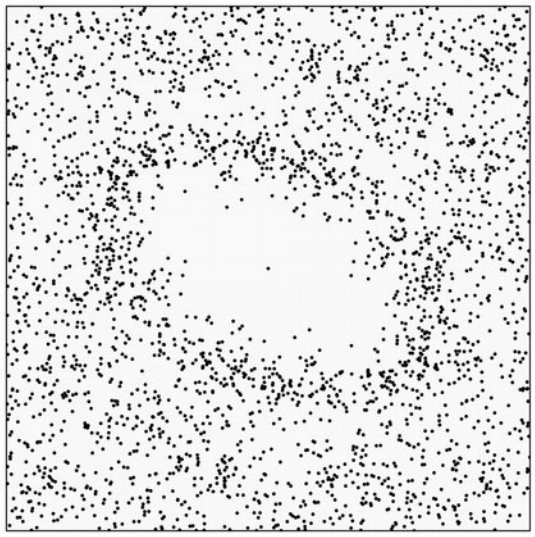


Figure 15. The Fry plot without normalizing, using the settings displayed in Figure 14.

To normalize the plot select *Normalize*, as shown in Figure 16. Note that the *Normalized radius* is now used due to the normalization to a unit circle, the default value is 1.5 as shown. The resulting normalized plot is shown in Figure 17. Note the clear sharpening of the central void.

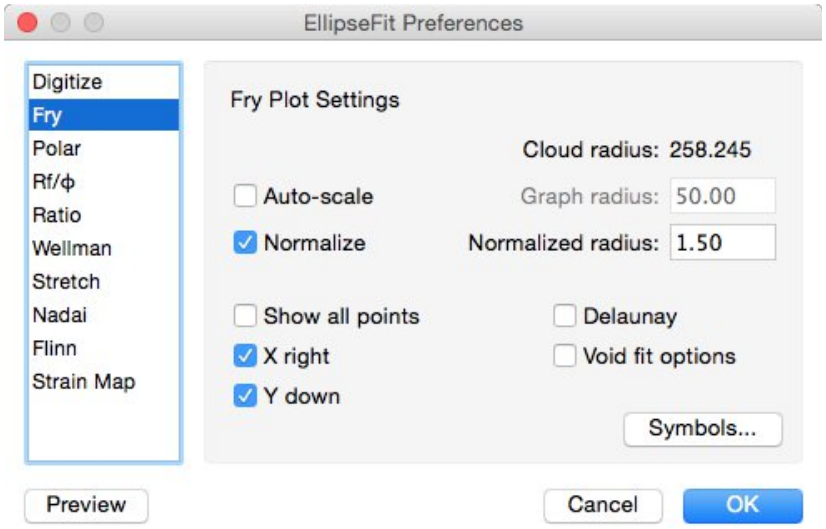


Figure 16. Settings to display a normalized Fry plot. Note that the *Normalized radius* is now used due to the normalization to a unit circle.

Normalizing the plot can be done for the *enhanced normalized*, *exponential edge detection*, *Delaunay neighbors*, and *density gradient* methods described in the next section, providing object shapes (R, ϕ) are available.

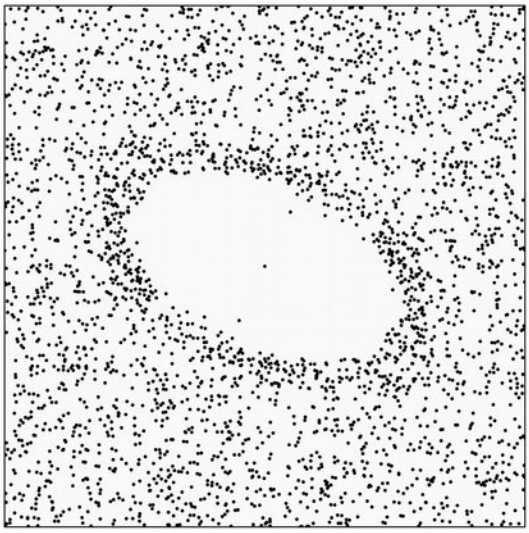


Figure 17. Plot of the normalized data. Note the better resolution of the central void.

3.3 Objective Void Fitting

Calculating the strain from a sample of points should ideally be objective, not requiring a user to select the best-fit ellipse (as in Section 3.1), and, ideally, should not require additional information about particle shape (as in Section 3.2). An objective numerical calculation is therefore desirable, and a number of methods have been proposed (Mulchrone, 2003, 2013; Waldron and Wallace, 2007; Lisle, 2010; Shan and Xiao, 2011; Reddy and Srivastava, 2012; Kumar et al., 2014). Methods currently implemented in EllipseFit, and described in this chapter, are accessed using the *Fit Void* command which displays up the *Fit Void* dialog.

Kumar et al. (2014) tested six such methods, including the *Delaunay triangulation nearest neighbors (DTNN), density contrast* (point-count), *exponential edge detection* (continuous function), and *mean log likelihood* methods, using two-dimensional simulated data sets, and concluded that of those six, *exponential edge detection* (Waldron and Wallace, 2007) is the most accurate, followed by the *DTNN* method (Mulchrone, 2013).

The *enhanced normalized* (Erslev and Ge, 1990) and related *Delaunay nearest neighbors* (Mulchrone, 2003) methods require particle shape information, $[a_i, b_i, \phi_i]$, ellipse radii and orientations, they differ in that the enhanced method uses the *entire* point cloud, that is, *all* object to object distances, while Delaunay methods use only the *nearest neighbors* distances (e.g., Ramsay and Huber, 1983).

The *exponential edge detection*, *density contrast*, and *mean log likelihood* methods do *not* require shape information. This gives them an advantage in terms of the speed of data collection, and where shape information is not available, such as in the spacing of dewatering pipes (Waldron and Wallace, 2007). Additionally, the analysis is entirely independent of particle shape, so may be used to complement shape-based methods.

Kumar et al. (2014) concluded that the *exponential edge detection* method was better than the *density contrast* method, and that the *DTNN*, *density contrast*, *exponential edge detection,* and *image analysis* methods are least affected by the degree of sorting, the nature of distortion or the amount of distortion. The *mean log likelihood* method (Shan and Xiao, 2011), gave less accurate results, possibly because the non-random point distribution in their simulated samples violated the method's requirement of a homogeneous truncated Poisson distribution.

3.3.1 Enhanced

The *enhanced* and *enhanced normalized Fry* methods (Erslev, 1988; Erslev and Ge, 1990) use a user specified cutoff radius to exclude particles beyond the a certain distance from the void center. This is set by a user defined value, the selection factor, *sf*, which has a default value of *sf* = 1.2. In the *Void Fit* dialog select options as in Figure 18, press *Preview*, and then *Run*. EllipseFit calculates the best-fit ellipse through the cloud of points using the rotationally invariant least squares method described in Section 3.1.

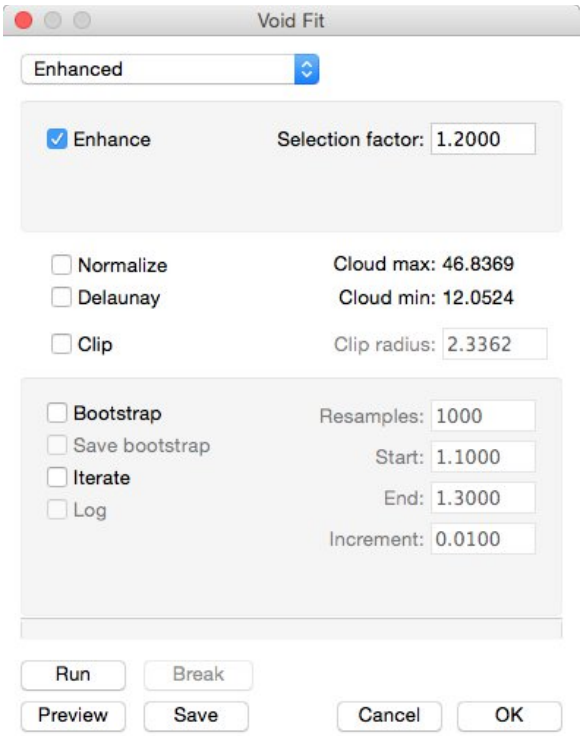


Figure 18. Settings to display an enhanced plot.

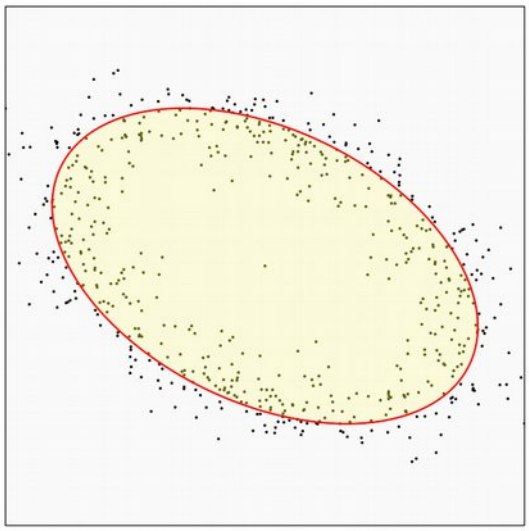


Figure 19. Fry plot with ellipse fitted to the enhanced points.

The results from the *Log w*indow are:

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-19 12:33:01
============================
N     : 252
Pairs : 31626
Time  : 00:00:00
Enhanced
  Normalized       : False
  Delaunay         : False
  Selection factor : 1.200
  Enhanced pairs   : 306
Best-Fit Ellipse
  n   = 306
  A   = 35.023
  B   = 21.071
  R   = 1.662
  ϕ   = 25.83°
  RMS = 0.2889
```

RMS (root mean square) is a measure of the deviations of the residuals, and can be used to refine the selection factor. However, note that smaller number of points will generally have a smaller RMS. For example, three points give RMS = 0, so finding the minimum RMS is not a valid strategy.

Selecting *Normalize* gives the following results and plot (Figure 20).

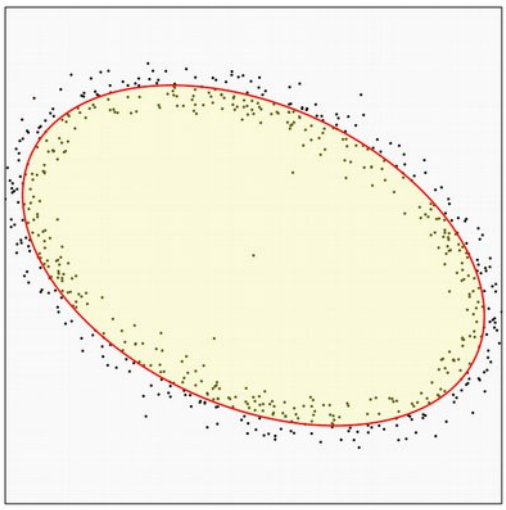


Figure 20. Fry plot with ellipse fitted to the enhanced normalized points.

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-19 12:41:08
==============================
N     : 252
Pairs : 31626
Time  : 00:00:00
Enhanced
  Normalized       : True
  Delaunay         : False
  Selection factor : 1.200
  Enhanced pairs   : 306
Best-Fit Ellipse
  n   = 306 pairs
  A   = 0.919
  B   = 0.566
  R   = 1.622
  ϕ   = 24.43°
  RMS = 0.1582
```

The *selection factor* (*sf*) is a user defined parameter that controls how many of the points are to be included in the selection. The value should be a small number greater than 1, the default is 1.2. Mulchrone (2002) studied the effect of the *sf* on error minimization, and concluded that errors were minimized using *sf* between about 1.1 and 1.3.

Select the *Bootstrap* option to do an error analysis by resampling the data set with replacement to create resamples. The following are example results from an enhanced plot bootstrap error analysis. Figure 21 is a plot of the resample means used to determine the confidence intervals. Note that, due to the random sampling, the error estimates will vary a small amount for each run.

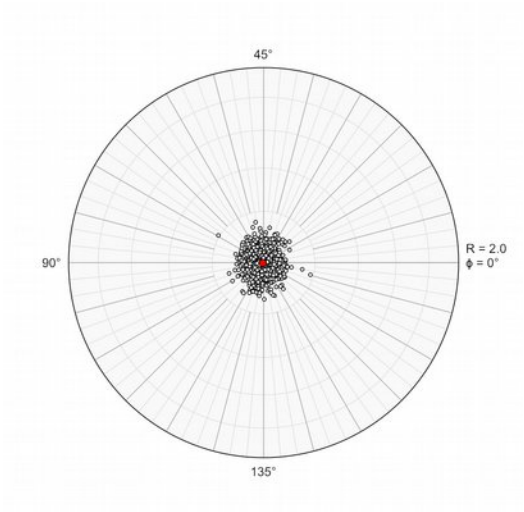


Figure 21. Polar plot of the means of 1000 bootstrapped resamples used for the confidence intervals using the *enhanced* method. The resample means are rotated to the sample mean, the red circle is the mean of the resample means.

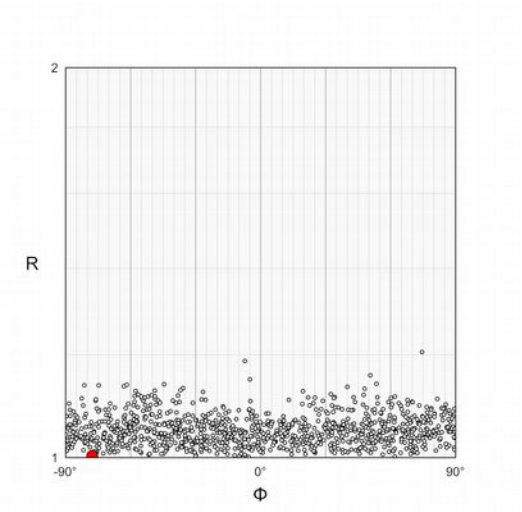


Figure 22. Cartesian R_f-ϕ plot of the means of 1000 bootstrapped resamples used for the confidence intervals using the *enhanced* method. The resample means are rotated to the mean, the red circle is the mean of the resample means. Due to distortion inherent in R_f-ϕ plots, the position of the mean on the ϕ axis has little significance.

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-19 14:39:40
==============================
N     : 252
Pairs : 31626
Time  : 00:00:21
Enhanced
  Normalized       : False
  Delaunay         : False
  Selection factor : 1.200
  Enhanced pairs   : 306
Best-Fit Ellipse
  n   = 306 pairs
  A   = 35.023
  B   = 21.071
  R   = 1.662
  ϕ   = 25.83°
  RMS = 0.2889
Bootstrap Confidence
  n     = 1000 resamples
  R     = 1.662
  ϕ     = 25.83°
  R 95% = 0.134
  ϕ 95% = 3.34°
```

The bootstrap resample means for an enhanced normalized analysis are shown in Figure 23, and the

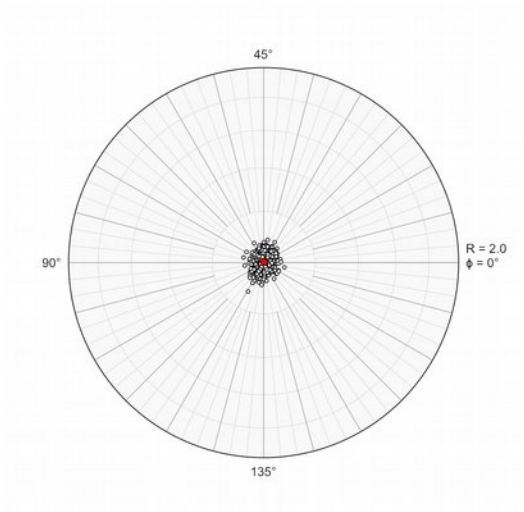


Figure 23. Polar plot of the means of 1000 bootstrapped resamples used for the confidence intervals using the *enhanced normalized* method. The resample means are rotated to the sample mean, the red circle is the mean of the resample means. Note the tighter clustering over Figure 21.

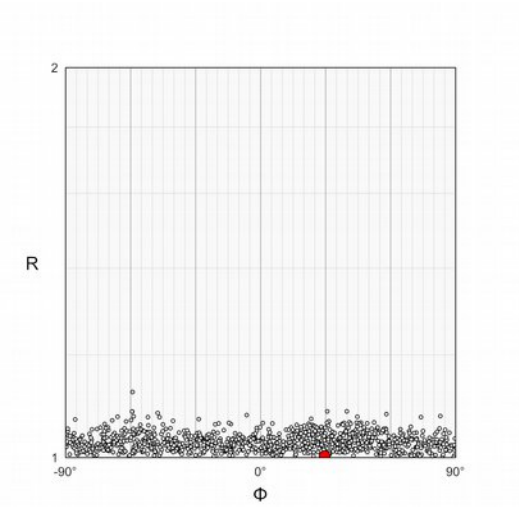


Figure 24. Cartesian R_f-ϕ plot of the means of 1000 bootstrapped resamples used for the confidence intervals using the *enhanced normalized* method. The resample means are rotated to the sample mean, the red circle is the mean of the resample means. Due to distortion inherent in R_f-ϕ plots, the position of the mean on the ϕ axis has little significance.

output is:

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-19 17:24:09
============================
N     : 252
Pairs : 31626
Time  : 00:00:26
Enhanced
  Normalized       : True
  Delaunay         : False
  Selection factor : 1.200
  Enhanced pairs   : 306
Best-Fit Ellipse
  n   = 306 pairs
  A   = 0.919
  B   = 0.566
  R   = 1.622
  ϕ   = 24.43°
  RMS = 0.1582
Bootstrap Confidence
  n     = 1000 resamples
  R     = 1.622
```

```
  ϕ     = 24.43°
  R 95% = 0.078
  ϕ 95% = 1.84°
```

This gives results close to the enhanced analysis, [R, Φ] = [1.622, 24.43°] versus [1.662, 25.83°], with a smaller confidence intervals. Figure 23 shows the tighter clustering reflected by the smaller confidence intervals.

An additional option to aid in evaluating the effect of the value of *sf* is given by the *Iterate* and *Log* options. If selected, the value of *sf* is iterated by an increment, and the associated mean and confidence calculated. By default, the iteration steps through values from 1.1 to 1.3 by an increment of 0.1. The output for an enhanced normalized analysis gives:

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-19 18:37:15
============================
N     : 252
Pairs : 31626
Time  : 00:00:00
Enhanced
  Normalized       : True
  Delaunay         : False
  Selection factor : 1.200
Iterated Ellipses
  sf      R       ϕ        Pairs
  1.1000  1.6064  24.668°  218
  1.1100  1.6064  24.402°  230
  1.1200  1.6044  24.152°  239
  1.1300  1.5958  24.217°  251
  1.1400  1.5962  24.418°  258
  1.1500  1.5995  24.338°  265
  1.1600  1.6079  24.533°  277
  1.1700  1.6182  24.467°  283
  1.1800  1.6124  24.054°  291
  1.1900  1.6130  24.235°  298
  1.2000  1.6224  24.431°  306
  1.2100  1.6316  24.755°  314
  1.2200  1.6474  24.765°  323
  1.2300  1.6411  25.093°  336
  1.2400  1.6433  25.394°  341
  1.2500  1.6441  25.395°  352
  1.2600  1.6492  25.489°  359
  1.2700  1.6659  25.367°  368
  1.2800  1.6698  25.770°  375
  1.2900  1.6790  25.719°  381
  1.3000  1.6765  26.220°  391
Mean of Iterated Ellipses
  n     = 21 ellipses
  R     = 1.677
  ϕ     = 26.22°
  R 95% = 0.012
```

```
ϕ 95% = 0.37°
```

3.3.2 Delaunay Neighbors

A Fry plot contains information about the distances between *all* data points. An alternate methodology is to use only the *nearest neighbors* (Ramsay, 1967; Ramsay and Huber, 1983; Mulchrone, 2002, 2013). A Delaunay triangulation is a triangulation of points in a plane such that no point is inside the circumcircle of any triangle, the edges of the triangles therefore define the nearest neighbor distances (Figure 25). There can be edge effects related to the shape of the sampled area, discarding the *bounding hull* helps to minimize this. Additionally, particularly at higher strains, the nearest neighbors are not guaranteed to be the same as the nearest neighbors prior to deformation. However, this subset of points can provide a less numerically intensive set for function or search minimization procedures.

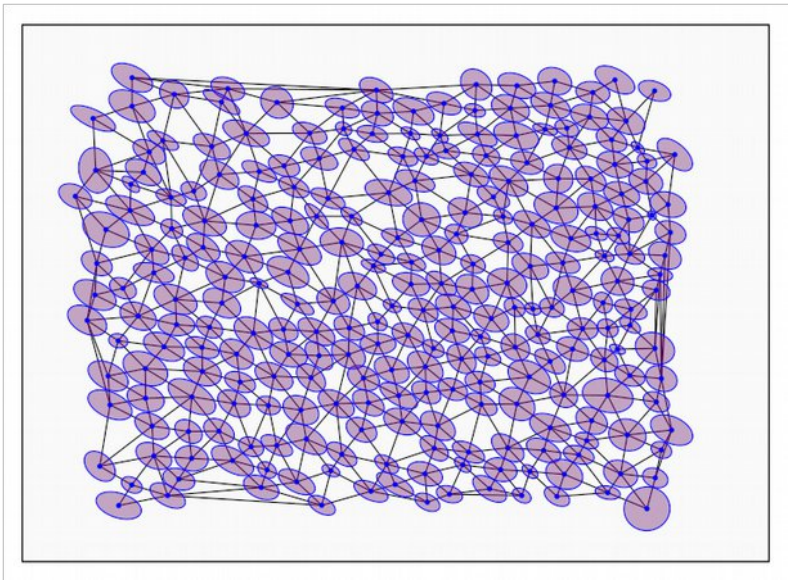


Figure 25. Strain map with Delaunay triangulation excluding the bounding hull. The triangle edges are the nearest neighbor distances.

In EllipseFit the use of Delaunay nearest neighbors is an *option* in the *enhanced*, *exponential edge detection*, *density contrast*, and *weighted least-squares* methods. In each case the point cloud is reduced to the nearest neighbors using the edges of a Delaunay triangulation, reducing the number of required calculations in a search or minimization. The method described in this section is *enhanced normalized Delaunay* method, which is similar to the method of Mulchrone (2002).

Figure 26 shows the *Void Fit* dialog *Enhanced normalized* pane with the settings to display Delaunay neighbors. The resulting plot is shown in Figure 27. Note the presence of the horizontal and vertical point “wings” due the rectangular shape of the digitized area.

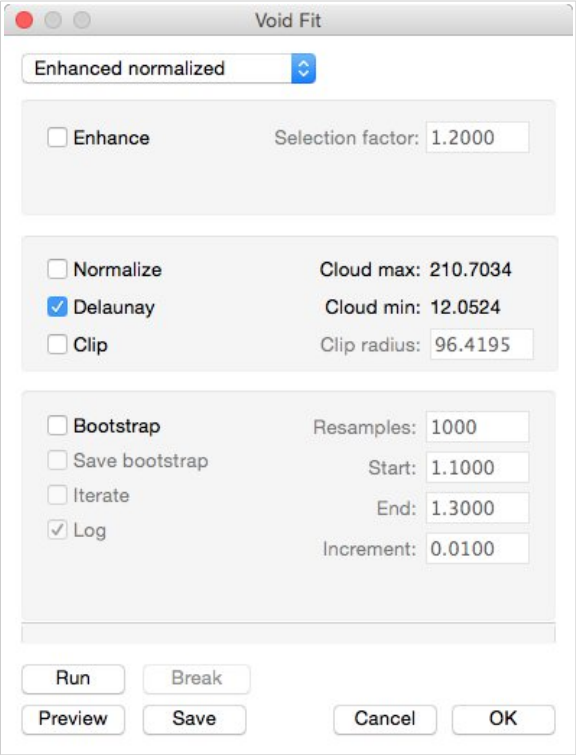


Figure 26. Enhance normalized settings pane with the Delaunay option selected.

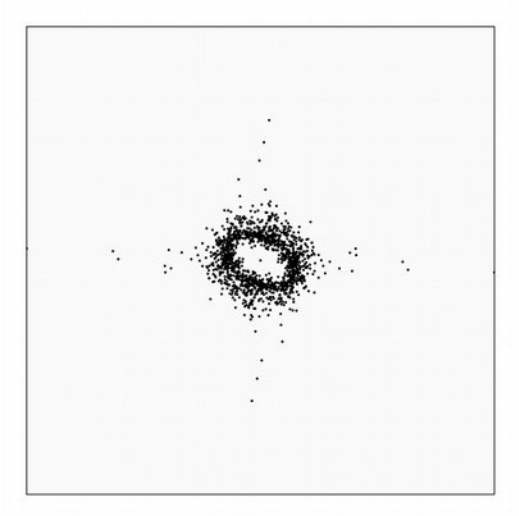


Figure 27. Fry plot showing Delaunay nearest neighbor points. The vertical and horizontal "wings" are due to the rectangular shape of the digitized area.

Figures 28 and 29 are plots using the *enhanced Delaunay*, and the e*nhanced normalized Delaunay* options using the default selection factor (*sf*) of 1.2. The results of the enhanced normalized Delaunay

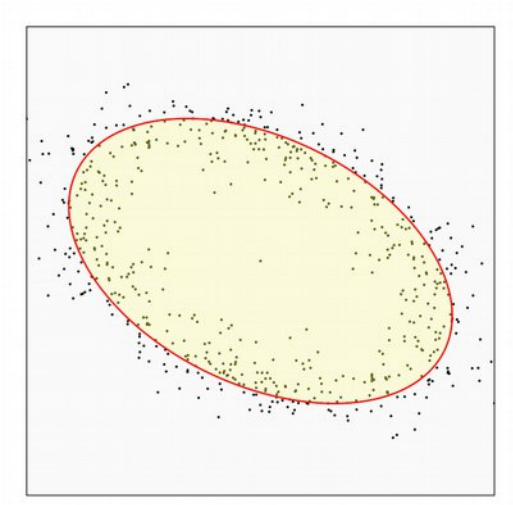


Figure 28. Fry plot showing the results of the *enhanced Delaunay* method.

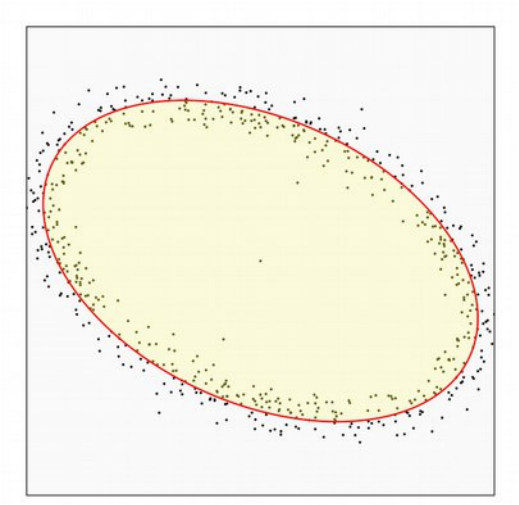


Figure 29. Fry plot showing the results of the *enhanced normalized Delaunay* method.

method are:

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-19 18:47:39
============================
N     : 252
Pairs : 717
Time  : 00:00:00
Enhanced
  Normalized       : True
  Delaunay         : True
  Selection factor : 1.200
  Enhanced pairs   : 305
Best-Fit Ellipse
  n   = 305 pairs
  A   = 0.918
  B   = 0.566
  R   = 1.621
  ϕ   = 24.45°
  RMS = 0.1584
```

Comparing the results to the *enhanced normalized* method, the Delaunay neighbors option has used one less point pair, 305 as opposed to 306, and the results are nearly identical, [R, Φ] = [1.621, 24.45°] as opposed to [1.622, 24.43°]. Figure 30 shows the means of 1000 bootstrap resamples used to determine confidence intervals for the *enhanced normalized Delaunay* method. Using the *Bootstrap* option to calculate confidence gives:

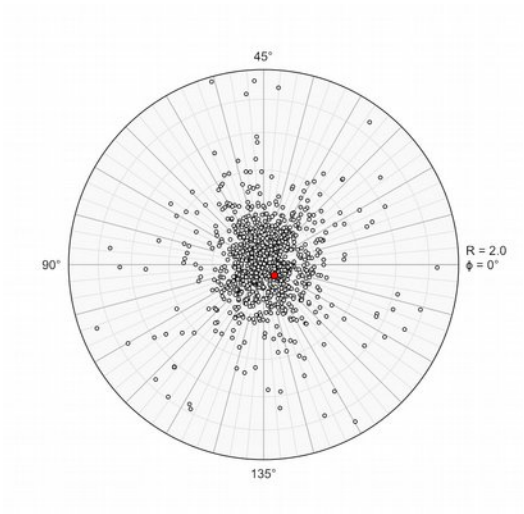


Figure 30. Polar plot of the means of 1000 bootstrapped resamples used for the confidence intervals using the *enhanced normalized Delaunay* method. The replicate means are rotated to the sample mean, the red circle is the mean of the replicate means.

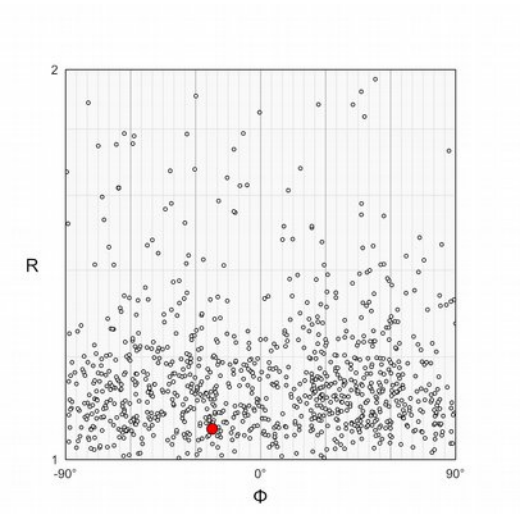


Figure 31. Cartesian R_f-ϕ plot of the means of 1000 bootstrapped resamples used for the confidence intervals using the *enhanced normalized Delaunay* method. The resample means are rotated to the sample mean, the red circle is the mean of the resample means. Due to distortion inherent in the R_f-ϕ plot, the position of the mean on the ϕ axis has little significance.

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-19 18:53:37
==============================
N     : 252
Pairs : 717
Time  : 00:00:00
Enhanced
  Normalized       : True
  Delaunay         : True
  Selection factor : 1.200
  Enhanced pairs   : 305
Best-Fit Ellipse
  n   = 305 pairs
  A   = 0.918
  B   = 0.566
  R   = 1.621
  ϕ   = 24.45°
  RMS = 0.1584
Mean and Bootstrap Confidence
  n     = 1000 resamples
  R     = 1.621
  ϕ     = 24.45°
  R 95% = 0.621
  ϕ 95% = 14.75°
```

The *enhanced normalized Delaunay* method bootstrapping is faster that the *enhanced normalized*, less than one second versus 26 seconds on a 3.06 GHz machine, however gives less confidence, [R 95%, ϕ 95%] = [0.621, 14.75°] versus [0.078, 1.84°] as illustrated in Figures 23, 24, 30, and 31,.

As discussed in the previous section, the *selection factor* (*sf*) is a user defined parameter that controls how many of the points are to be included in the selection. The value should be a small number greater than 1, the default is 1.2. The *Iterate* and *Log* options can be used to evaluate the effect of *sf* on the analysis, and to select one that minimizes the error. The reduction of the data cloud by using Delaunay triangulation can similarly be applied to the *density contrast, exponential edge detection, and weighted least squares* methods.

3.3.3 Density Contrast

Waldron and Wallace (2007) suggested a point counting method that does not require any information about particle shape. This gives an advantage in terms of the speed of data collection, and where shape information is not available, such as in their example of the spacing of dewatering pipes. Additionally, the analysis is entirely independent of particle shape, so may be used to complement shape-based methods. If the object shape parameters R and ϕ are available, however, the plot can be normalized.

The method uses two annular ellipses with the same aspect ratio. The parameters are the minimum radius of the inner void fit ellipse, its aspect ratio R, and orientation ϕ, which are varied to minimize the density contrast, *C*. The *Void Fit Density contrast* dialog pane is shown in Figure 32. Options include minimum, maximum, and increments of the three search parameters. These can be automatically set to suggested values by pressing the *Set* button, or can be set manually. Figure 32 shows a preview of automatically selected density contrast options. The two inner blue rings represent the minimum and maximum search values for the minimum (short) void fit ellipse radius. The outer yellow circle is the maximum search radius. The default optimized search settings increment R by 0.01 from 1 to 3, Φ by 0.1° from 0° to 179.9°, and B by 100 increments over a selected range.

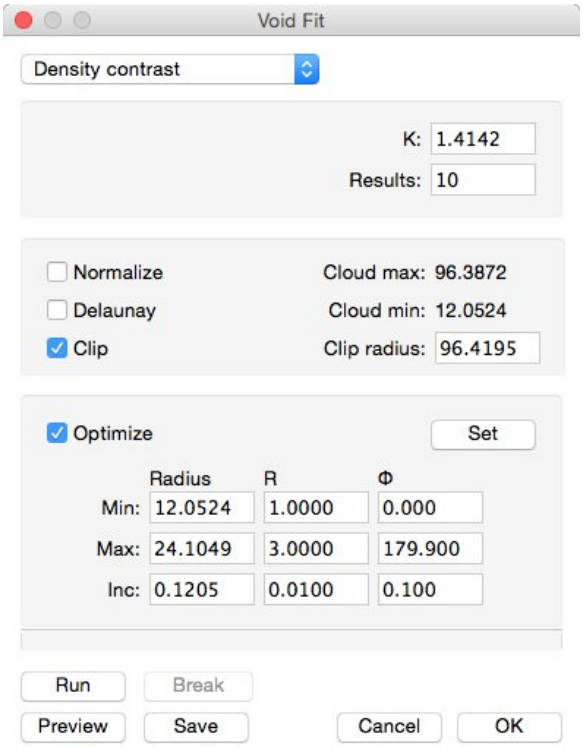


Figure 32. Void Fit Density contrast settings pane after having set the automatic options.

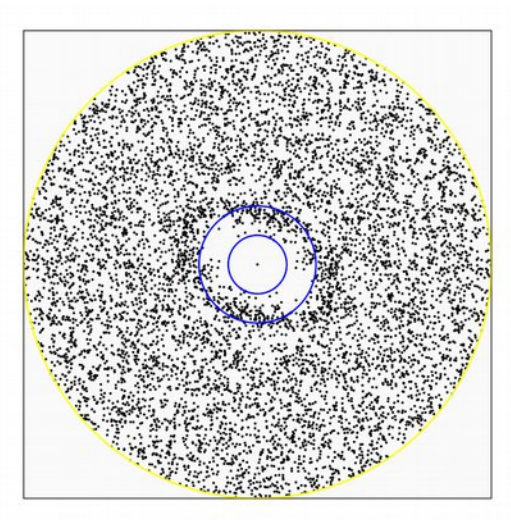


Figure 33. Preview of automatic density contrast options. The two inner blue circles represent the minimum and maximum search values for the short axis of the void fit ellipse. The outer yellow circle is the maximum search radius.

The resulting solution is shown graphically in Figures 34 and 35, and the results, showing the 10 best

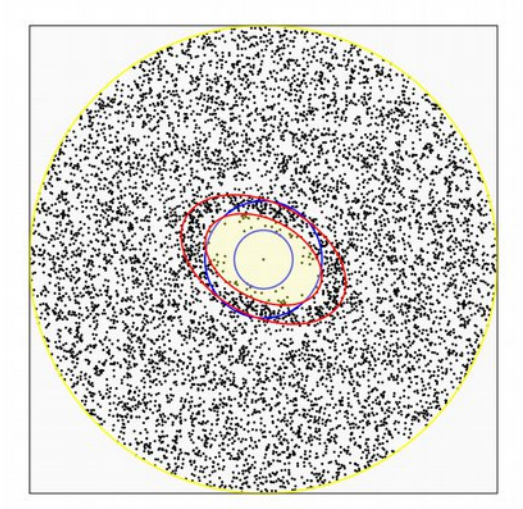


Figure 34. Plot with best-fit ellipse using the automatically selected density contrast options. The red ellipses define the best-fit annulus, both have the same aspect ratio and orientation, the inner defines the void fit.

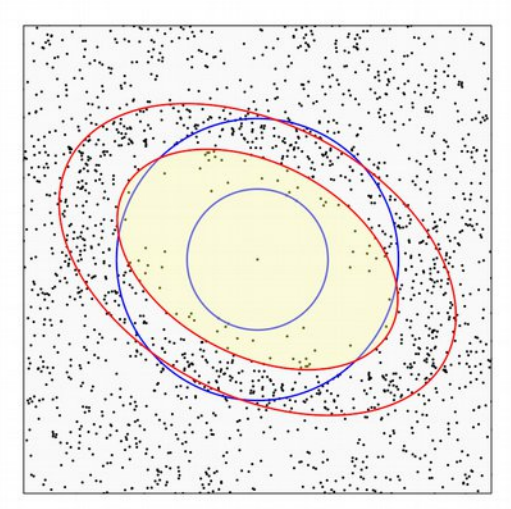


Figure 35. Zoomed in view of Figure 34 showing the void best-fit ellipse and outer annulus ring. The clip radius is 40.

solutions are as follows:

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-19 21:34:51
============================
N     : 252
Pairs : 3985
Time  : 00:00:14
Density contrast
  Normalize        : False
  Delaunay         : False
  Iterations       : 46521
  K                : 1.414
Best-Fit Ellipses
  Max      Min      R      ϕ       C
  25.7329  16.3904  1.570  28.00°  1.691497
  25.8691  16.2699  1.590  28.50°  1.689513
  25.7064  16.2699  1.580  28.50°  1.689057
  25.7064  16.2699  1.580  28.30°  1.683483
  25.5437  16.2699  1.570  29.40°  1.677375
  25.8390  16.1494  1.600  28.00°  1.676166
  25.6775  16.1494  1.590  28.00°  1.675463
  25.6462  16.0289  1.600  28.40°  1.673105
  25.6462  16.0289  1.600  28.00°  1.656090
  25.3086  16.8724  1.500  29.00°  1.654347
Mean of Best-Fit Ellipses
  n     = 10
  R     = 1.578
  ϕ     = 28.40°
  R 95% = 0.022
  ϕ 95% = 0.70°
```

The solutions are in decreasing order of C, so the best is the first in the list. The default value for search parameter, K = √2, gives an equal area to the inner ellipse and the search annulus, this was suggested by Waldron and Wallace (2007), and affirmed by Kumar et al. (2014).

3.3.4 Exponential Edge Detection

Waldron and Wallace (2007) suggested a second method that also does not require any information about particle shape. Again, this gives an advantage in terms of the speed of data collection, and where shape information is not available, such as in their example of the spacing of dewatering pipes. Additionally, the analysis is entirely independent of particle shape, so may be used to complement shape-based ellipse fitting methods. If the object shape parameters [a_i, b_i, ϕ_i] are available, however, the plot can be normalized.

The method maximizes an exponential function, $Z' = f[A, B, \Phi]$, that is calculated for all points in the search radius. In contrast to the enhanced method, which fits an ellipse through a subset of points *near* the void edge, this procedure locates the ellipse that defines the *edge* of the void by looking for the steepest density gradient. The parameters are the void ellipse long radius, short radius, and orientation. An additional user defined parameter, *k*, is by default set to $k = 3$, as suggested by Waldron and Wallace (2007), and affirmed by Kumar et al. (2014). The *Exponential edge* dialog pane is shown in Figure 36.

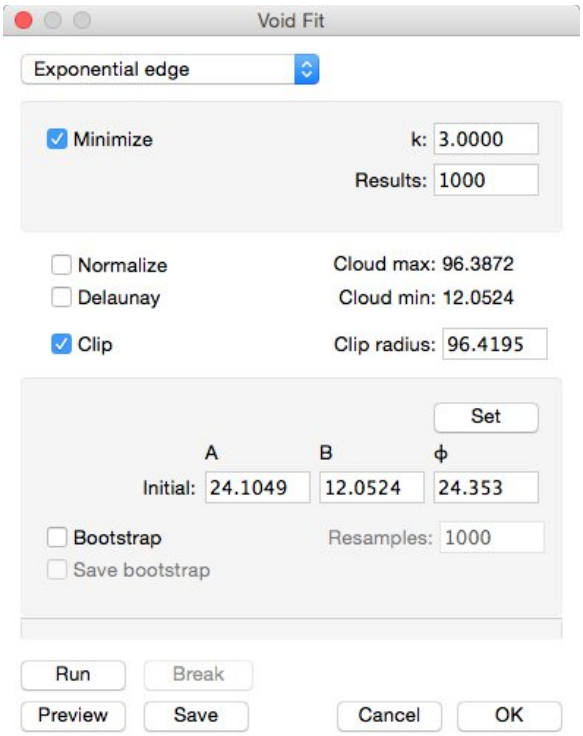


Figure 36. Exponential edge detection settings pane after having set the automatic options.

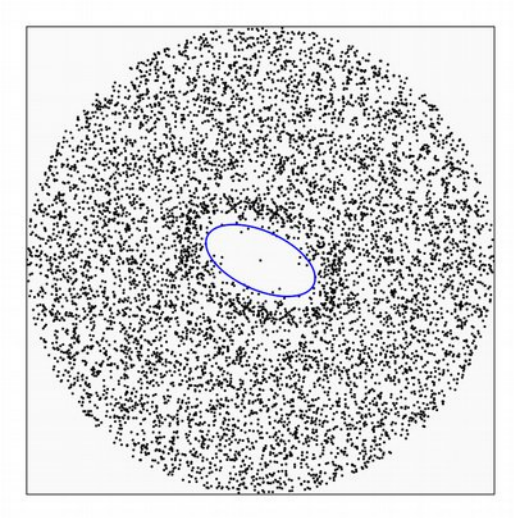


Figure 37. Preview of automatic exponential edge detedtion options. The blue ellipse is the initial estimated solution automatically chosen using the *Set* button.

The search can be conducted either by an optimized parameter search, as was done for the *density contrast* method, or more efficiently, by using function minimization (e.g., Press et al., 2007), as used in this example.

The parameters [A, B, Φ] must be set to initial estimated values. These can be automatically set to suggested initial values by pressing the *Set* button, or can be set manually. Figure 37 is a Fry plot with suggested initial parameters automatically selected by pressing the *Set* button. The resulting solution is shown graphically in Figures 38 and 39, and the results are as follows:

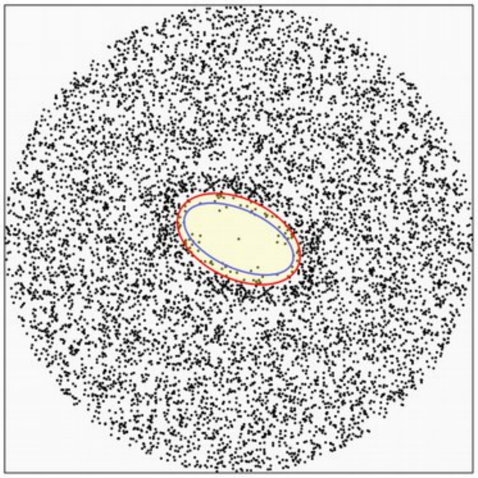


Figure 38. Plot showing best-fit ellipse determined from the automatically selected exponential edge detection options. The red ellipse defines the best void fit ellipse.

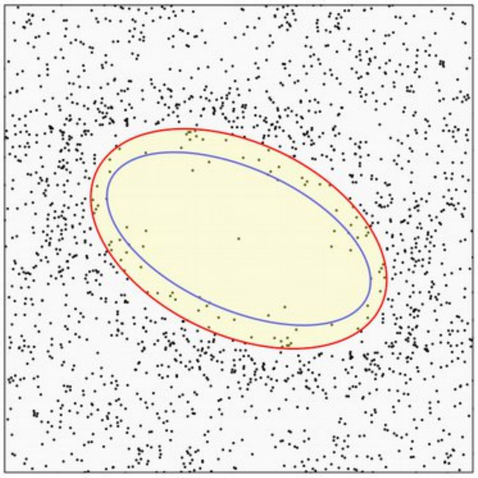


Figure 39. Zoomed in view of Figure 38 showing the void best-fit ellipse in red. The blue ellipse is the initial estimate. The clip radius is 40.

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-20 10:29:46
==============================
N     : 252
Pairs : 3985
Time  : 00:00:00
Exponential edge detection
  Normalize        : False
  Delaunay         : False
  Minimize         : True
  Iterations       : 204
  k                : 3.000
Best-Fit Ellipses
  Max       Min       R       ϕ        Z'          Z*
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
Mean of Best-Fit Ellipses
  n      = 10 Best-fit ellipses
  R      = 1.637
  ϕ      = 25.53°
  R 95% = 0.000
  ϕ 95% = 0.00°
```

The minimization procedure converges in less than one second (3.06 GHz iMac) after 204 iterations, all of the 10 best solutions are identical at the output precision. The solutions are in decreasing order of Z', so the best is the first in the list.

Confidence intervals for [R, Φ] can be calculated using bootstrapping, as in the *enhanced*, *enhanced normalized*, and *enhanced normalized Delaunay* methods. Prior to bootstrapping, it is suggested that the minimization be done once, then repeated by entering the solution as the new initial parameters. In this example, there is no change in the output values. Then, after refining the solution, select the bootstrap option to calculate the confidence intervals. Figure 40 is a polar plot of 1000 resample means used for the confidence intervals, and the results are:

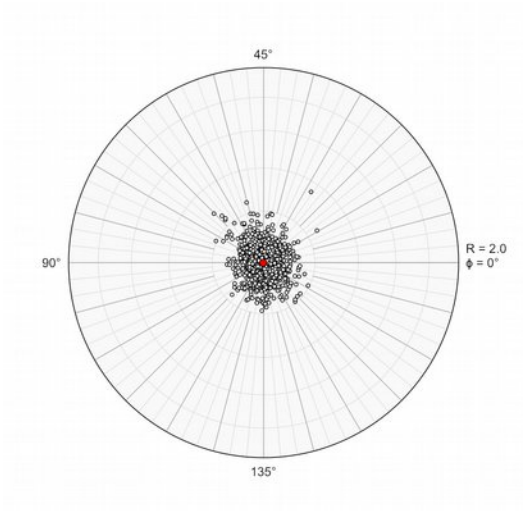


Figure 40. Polar plot of the means of 1000 resamples used to calculate the confidence intervals using exponential edge detection. The resample means are rotated to the sample mean. The red circle is the mean of the resample means.

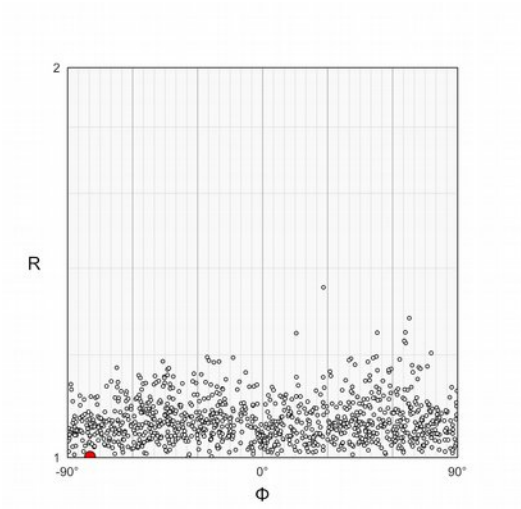


Figure 41. Cartesian R_f-ϕ plot of the means of 1000 resamples used to calculate the confidence intervals using exponential edge detection. The resample means are rotated to the sample mean. The red circle is the mean of the resample means. Due to distortion inherent in the R_f-ϕ plot, the position of the mean on the ϕ axis has little significance.

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-20 11:45:49
==============================
N     : 252
Pairs : 3985
Time  : 00:04:46
Exponential edge detection
  Normalize        : False
  Delaunay         : False
  Minimize         : True
  Iterations       : 1000
```

```
  k                 : 3.000
Best-Fit Ellipses
  Max      Min      R      ϕ       Z'          Z*
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
  26.8921  16.4278  1.637  25.53°  33.205496  1.579820
Bootstrap Confidence
  n     = 1000 Resamples
  R     = 1.637
  ϕ     = 25.53°
  R 95% = 0.165
  ϕ 95% = 4.83°
```

The bootstrapping process is relatively slow, compared to the *enhanced* methods bootstrapping, but is less than 5 minutes on a 3.06 GHz iMac. For comparison with previous sections, the following are the results for a *normalized exponential edge detection* analysis:

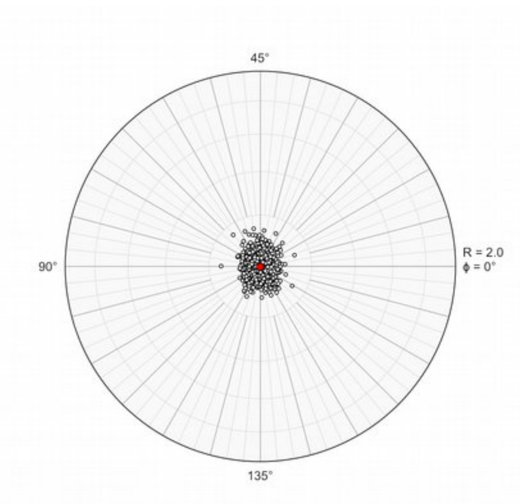


Figure 42. Polar plot of the means of 1000 resamples used to calculate the confidence intervals using *normalized exponential edge detection*. The resample means are rotated to the sample mean. The red circle is the mean of the resample means.

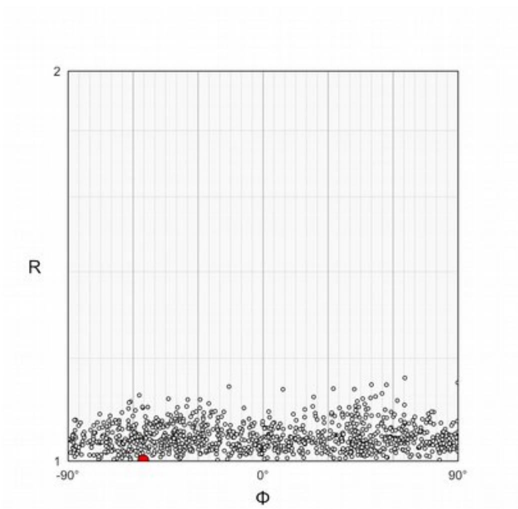


Figure 43. Cartesian R_f-ϕ plot of the means of 1000 resamples used to calculate the confidence intervals using *normalized exponential edge detection*. The resample means are rotated to the sample mean. The red circle is the mean of the resample means. Due to distortion inherent in the R_f-ϕ plot, the position of the mean on the ϕ axis has little significance.

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-20 15:15:30
==============================
N     : 252
Pairs : 3952
Time  : 00:05:27
Exponential edge detection
  Normalize        : True
  Delaunay         : False
  Minimize         : True
  Iterations       : 1000
  k                : 3.000
Best-Fit Ellipses
  Max      Min      R      ϕ       Z'          Z*
  0.7841  0.4506  1.740  24.85°  1.162810  1.956365
  0.7841  0.4506  1.740  24.85°  1.162810  1.956365
  0.7841  0.4506  1.740  24.85°  1.162810  1.956365
  0.7841  0.4506  1.740  24.85°  1.162810  1.956365
  0.7841  0.4506  1.740  24.85°  1.162810  1.956365
  0.7841  0.4506  1.740  24.85°  1.162810  1.956365
  0.7841  0.4506  1.740  24.85°  1.162810  1.956365
  0.7841  0.4506  1.740  24.85°  1.162810  1.956365
  0.7841  0.4506  1.740  24.85°  1.162810  1.956365
  0.7841  0.4506  1.740  24.85°  1.162810  1.956365
Bootstrap Confidence
  n     = 1000 Resamples
  R     = 1.740
  ϕ     = 24.85°
  R 95% = 0.124
  ϕ 95% = 2.71°
```

3.3.5 Mean Log Likelihood

Shan and Xiao (2011) suggest a *mean log likelihood function (MLLF)* method where they examine the statistics of a truncated Poisson distribution, and define the MLLF as the average sum of the log probability distribution function (PDF) of each individual point in the deformed state. This is related to the density distribution around each point.

The PDF in the deformed state is related to the pre-deformation PDF by the shape and orientation of the central void, giving as parameters a cutoff distance, the ratio R, and the orientation ϕ. The function is complex however, and is solved using a gird search to locate the maximum MLLF. The search is over the range $\phi = 0°$ to 179° in steps of 1°, and $R = 1$ to 20 in steps of 0.1. Once R and ϕ are determined, the sample is retro-deformed, and a 50 step search is done to locate the cutoff radius.

Shan and Xiao (2011) further suggest an approach to improve the results using a cross validation technique for detecting spurious points by sequentially removing up to 10 points, and repeating the search. These algorithms were implemented by Y. Shan in a Fortran program which was reimplemented in EllipseFit with careful testing to insure that identical results are obtained.

The results are the best estimates values of R, ϕ. initial cutoff distance, and a set of neighborhood points. This method has advantages in that it is a robust numerical solution, and one that uses all of the points to define the central void. A disadvantage of the method is the computing time required to calculate the solution. In particular the cross-validation can take several hours.

To run a test sample open the file MLLF Test 60.csv. This data is the 60 point oolith sample used in section 3.1, and was carefully selected to avoid spurious points, and to avoid a directional bias. In the *Mean log likelihood* pane of the *Void Fit* dialog, leave *Cross validate* off, and press *OK*.

The *Void Fit* dialog displays the progress in its status bar, showing the search iteration passes in degrees, and is done at 180. The process should complete in less than a minute, and the results displayed in the *Log* window, and on the *Fry Plot* (Figure 44).

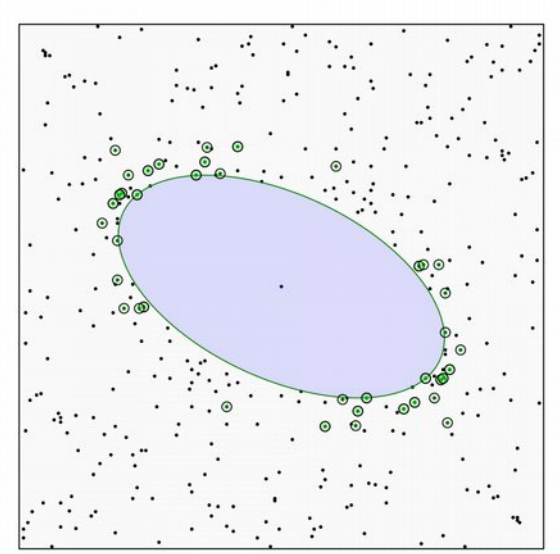


Figure 44. Fry plot with results of the mean log likelihood function (MLLF) maximization search. The ellipse is the result of the MLLF grid search. The green markers highlight the Fry neighbor points.

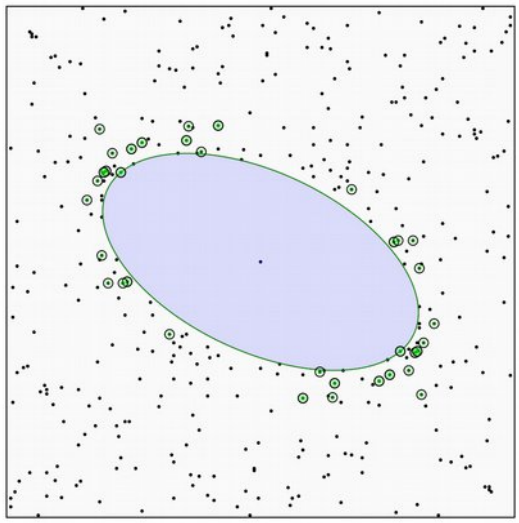


Figure 45. Fry plot of the results using the cross-validation option for mean log likelihood maximization.

The results reported in the log file are:

```
N = 60
MLLF Calculations
-----------------
Pass    Mean LL      R    Phi     Cutoff        Stat    Density
   0   -0.31829   1.90  25.00   86.98953     0.67361    0.84687
MLLF Results
------------
Point statistics:
Number                      =         60
Calculated density          =    0.00004
Real density                =    0.00000
```

```
Results:
Mean log-likelihood           =   -0.31829
R, strain ratio               =    1.90000
Phi, angle of max strain axis =   25.00000
Cutoff radius                 =   86.98953
```

In Figure 44 the ellipse is the result of the MLLF grid search. The green markers highlight the Fry neighbor points, those that maximize the MLLF after an intensive grid search.

To test the cross validation procedure, select the *Cross validation* option. There are now three iteration passes displayed, the first is 0 to 10, where 0 is the first calculation as done above. Passes 1 to 10 are the cross validation iterations, 1 to 60 are the data points, and 1 to 180 are the ϕ grid search in degrees. The R grid search values (0.1 to 20.0 by default), and the 1 to 50 distance search loops are not displayed.

The MLLF search is computationally intensive, especially for cross validation (during some test runs I set my laptop on marble coasters to keep it from overheating). After about 6 hours (on a 3.06 GHz iMac) the process completes. The search can be canceled at any time, and the results of the completed passes will be displayed.

```
Mean Ellipse Calculations
MLLF Test 60.tsv
2014-05-31 16:30:46
============================
N = 60
MLLF Calculations
-----------------
Pass    Mean LL      R    Phi     Cutoff       Stat    Density
   0   -0.31829   1.90  25.00   86.98953    0.67361    0.84687
   1   -0.31610   1.90  25.00   86.98953    0.68773    0.86122
   2   -0.31603   1.90  25.00   86.98953    0.69522    0.87607
   3   -0.31882   1.90  25.00   86.98953    0.67496    0.89144
   4   -0.31651   1.90  25.00   86.98953    0.68968    0.90736
   5   -0.31536   1.90  25.00   86.98953    0.70494    0.92386
   6   -0.32428   1.80  23.00   87.24708    0.68393    0.93542
   7   -0.31554   1.90  25.00   86.98953    0.69945    0.95872
   8   -0.31327   1.90  25.00   86.98953    0.71578    0.97716
   9   -0.31454   1.80  23.00   87.24708    0.69591    0.99044
  10   -0.31451   1.90  25.00   86.98953    0.69099    1.01624

MLLF Results
------------
Point statistics:
Number                        =         52
Calculated density            =    0.00004
Real density                  =    0.00004

Results:
Mean log-likelihood           =   -0.31327
R, strain ratio               =    1.90000
Phi, angle of max strain axis =   25.00000
Cutoff radius                 =   86.98953
```

```
Finished: 2014-05-31 22:49:58
```

The results of pass 0 are identical to the previous result, however the cross-validation procedure located a slightly better solution, in pass 8 the mean log likelihood is -0.31327, instead of -0.31829. The resulting Fry plot with 8 less neighbor points is shown in Figure 45.

Kumar et al. (2014) concluded that the maximum likelihood method gave less accurate results than other methods, possibly because the non-random point distribution in their simulated samples violated the method's requirement of a homogeneous truncated Poisson distribution.

3.3.6 Weighted Least-Squares

The *weighted least squares* method attempts to minimize a least-squares function fitting a one-sided data cloud boundary by applying a high weighting to points that lie inside the boundary, and a weight of 1 to points that lie outside (Cardiel, 2009; Mulchron, 2013). The method used closely follows that of Mulchron (2013) and Mulchron et al. (2013), however using Delaunay neighbors is optional.

Two options are provided for locating a solution. The first is an optimized search through the parameters, [B, R, Φ], the void fit ellipse's short radius, axial ratio, and orientation of the long axis. The second option is to use function minimization (e.g., Press et al., 2007) which minimizes a function g[A, B, Φ], where A is the void ellipse long axis radius. The parameters are selected as described in the sections on the *density gradient* and *exponential edge detection* methods respectively. The function g[A, B, Φ] is the polar version of Mulchrone, 2013 (and Mathematica code of Mulchrone et al., 2013).

The minimization procedure, as is used in the *exponential edge detection* method, does not converge robustly for the *weighted least-squares* function, so an optimized parameter search should be used. The default optimized search settings increment R by 0.01 from 1 to 3, Φ by 0.1° from 0 to 179.9, and B by 100 increments over a selected range.

The first example uses a *weighted least squares search without* using Delaunay nearest neighbors. Figure 46 shows the settings pane and Figure 47 is the preview of automatic function search options. The blue circles are the minimum and maximum search radii for the void ellipse short axis.

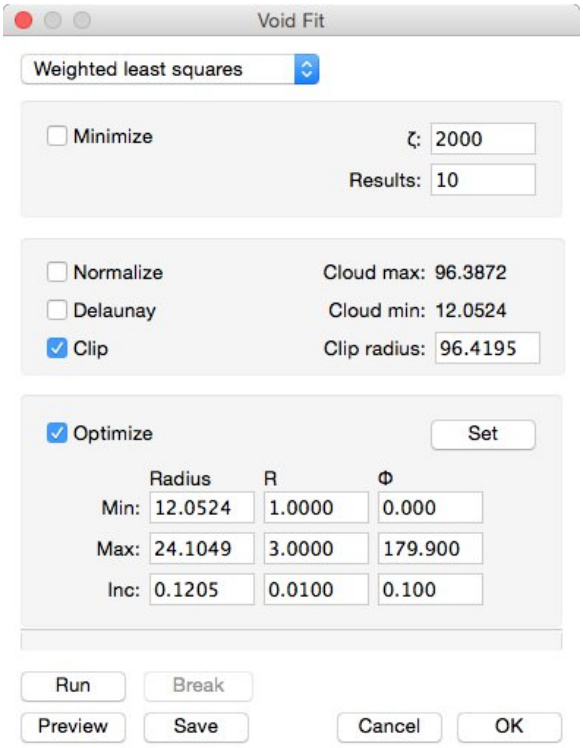


Figure 46. Weighted least-squares settings pane after having set the automatic options for a search.

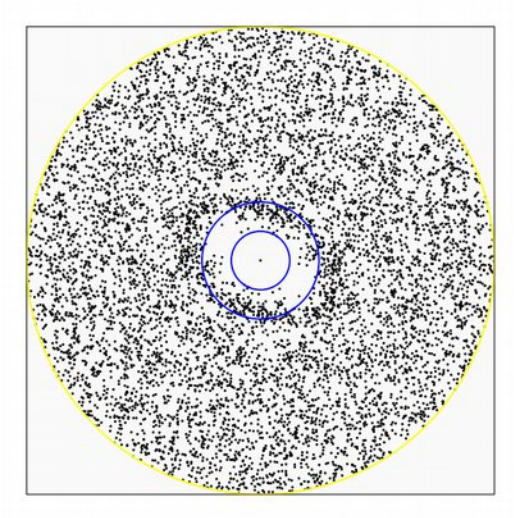


Figure 47. Preview of automatic *weighted least-squares* function search options. The blue circles are the minimum and maximum search radii for the void ellipse short axis.

The results of the weighted least-squares optimized search *without* using Delaunay nearest neighbors are plotted in Figures 48 and 49, and are as follows:

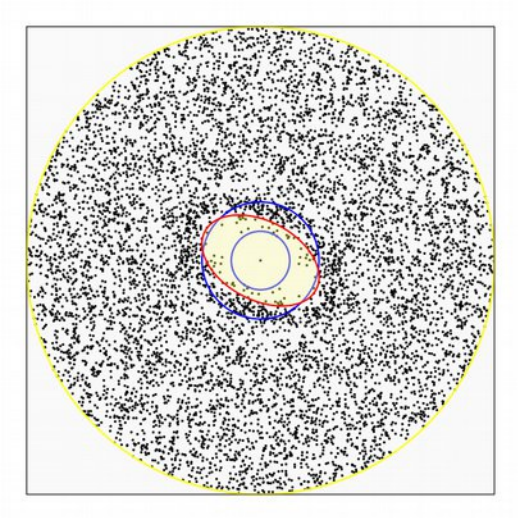


Figure 48. Plot showing best-fit ellipse determined from the automatically selected *weighted least-squares* function search options. The red ellipse defines the best void fit ellipse. Delaunay nearest neighbors are *not* used.

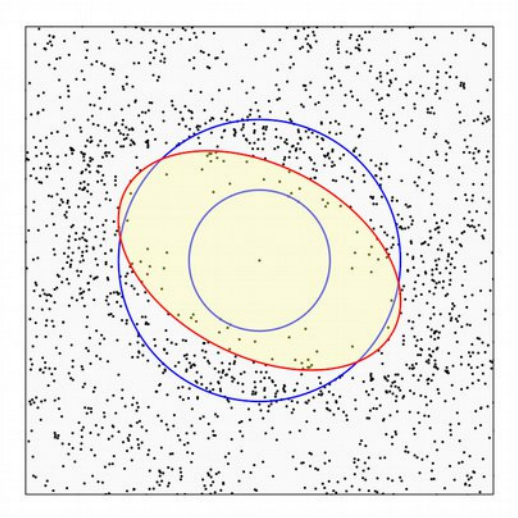


Figure 49. Zoomed in view of Figure 48 showing the void best-fit ellipse in red. The blue circles are the search range for the ellipse short radius. The clip radius is 40.

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-20 16:24:07
============================
N     : 252
Pairs : 3985
Time  : 00:00:21
Weighted Least Squares
  Normalize        : False
  Delaunay         : False
  Minimize         : False
  Iterations       : 47061
  ζ                : 2000
Best-Fit Ellipses
  Max      Min      R      ϕ       g
  25.8691  16.2699  1.590  27.60°  10193725
  25.8691  16.2699  1.590  27.50°  10193745
  25.8691  16.2699  1.590  27.40°  10193795
  25.8691  16.2699  1.590  27.30°  10193875
  25.7064  16.2699  1.580  27.60°  10193964
  25.7064  16.2699  1.580  27.50°  10193969
  25.7064  16.2699  1.580  27.40°  10194003
  25.7064  16.2699  1.580  27.30°  10194065
  25.7064  16.2699  1.580  27.20°  10194156
  25.7064  16.2699  1.580  27.10°  10194275
Mean of Best-Fit Ellipses
  n     = 10 ellipses
  R     = 1.584
  ϕ     = 27.39°
  R 95% = 0.004
```

```
ϕ 95% = 0.13°
```

The second example is a weighted least-squares optimized search using Delaunay nearest neighbors. Clipping is turned off to use all points. These results are plotted in Figures 50 and 51, and are as follows:

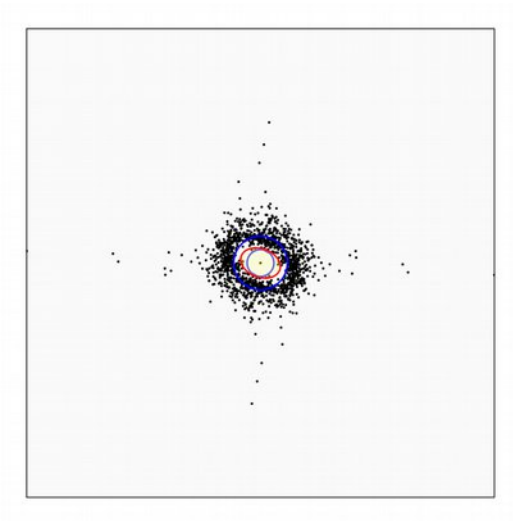


Figure 50. Plot showing best-fit ellipse determined for *weighted least-squares Delaunay nearest neighbors search* options. The red ellipse defines the best void fit ellipse. *are* used. Clipping is off.

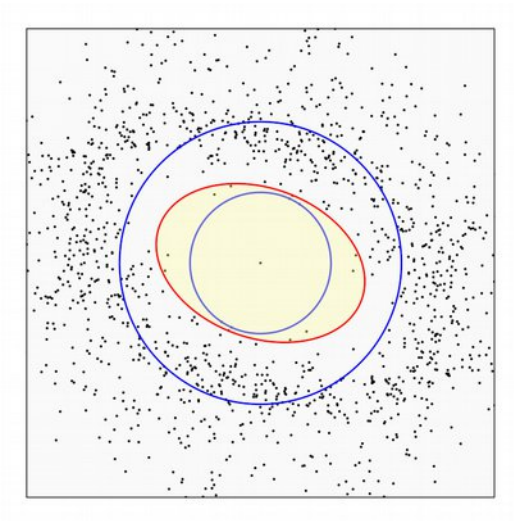


Figure 51. Zoomed in view of Figure 50 showing the void best-fit ellipse in red. The blue circles are the search range for the ellipse short radius. The clip radius is 40.

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-20 16:58:51
============================
N     : 252
Pairs : 717
Time  : 00:00:04
Weighted Least Squares
  Normalize        : False
  Delaunay         : True
  Minimize         : False
  Iterations       : 46620
  ζ                : 2000
Best-Fit Ellipses
  Max      Min      R      ϕ       g
  18.3966  12.7754  1.440  19.90°  351694
  18.3966  12.7754  1.440  19.80°  351695
  18.3966  12.7754  1.440  19.70°  351697
  18.3966  12.7754  1.440  19.60°  351702
  18.3966  12.7754  1.440  19.50°  351707
  18.3966  12.7754  1.440  19.40°  351714
  18.3966  12.7754  1.440  19.30°  351723
```

```
  18.2688  12.7754  1.430  19.70°  351730
  18.2688  12.7754  1.430  19.60°  351730
  18.2688  12.7754  1.430  19.50°  351731
Mean of Best-Fit Ellipses
  n     = 10 ellipses
  R     = 1.437
  ϕ     = 19.60°
  R 95% = 0.004
  ϕ 95% = 0.17°
```

```
Identical results are obtained if B is incremented by 1000 increments over the
selected range.
```

The last example is a weighted least-squares optimized search using *normalized* Delaunay nearest neighbors. This requires the particle shape parameters [a_i, b_i, ϕ_i], so loses independence from ellipse fitting methods. Again, the optimized search settings increment R by 0.01 from 1 to 3, Φ by 0.1° from 0 to 179.9, and B by 100 increments over a selected range, and clipping is off. The results are plotted in Figures 52 and 53, and are as follows:

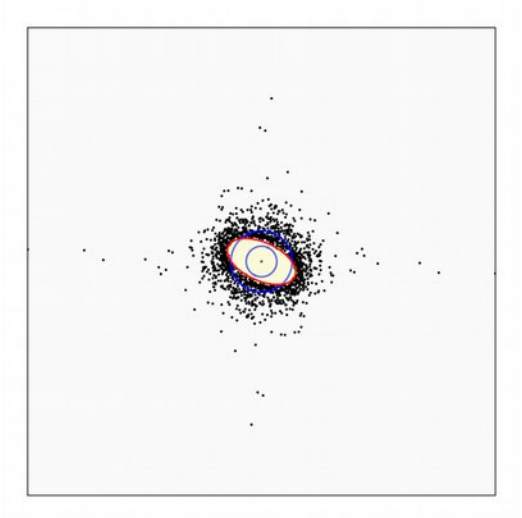


Figure 52. Plot showing best-fit ellipse determined for *weighted least-squares normalized Delaunay nearest neighbors.* The red ellipse defines the best void fit ellipse. Clipping is off.

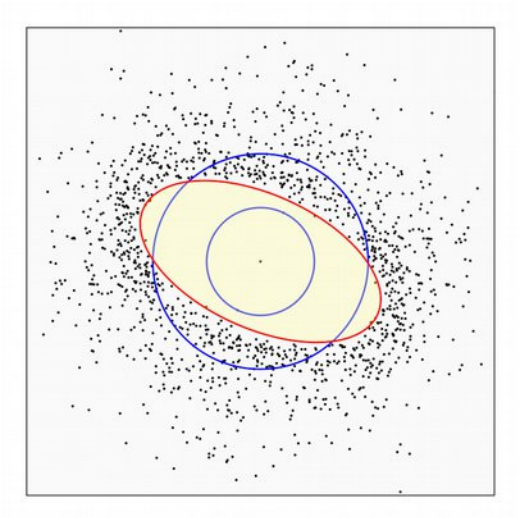


Figure 53. Zoomed in view of Figure 50 showing the void best-fit ellipse in red. The blue ellipse is the initial estimate. The clip radius is 1.5.

```
Void Fit Results
E2 - Ramsay and Huber 1983 (small)
2017-02-20 17:04:58
============================
N     : 252
Pairs : 717
Time  : 00:00:04
Weighted Least Squares
  Normalize        : True
```

```
  Delaunay         : True
  Minimize         : False
  Iterations       : 47061
  ζ                : 2000
Best-Fit Ellipses
  Max     Min     R      ϕ       g
  0.8243  0.4293  1.920  24.40°  209
  0.8243  0.4293  1.920  24.30°  209
  0.8243  0.4293  1.920  24.20°  209
  0.8243  0.4293  1.920  24.10°  209
  0.8243  0.4293  1.920  24.00°  209
  0.8243  0.4293  1.920  23.90°  209
  0.8243  0.4293  1.920  23.80°  209
  0.8243  0.4293  1.920  23.70°  209
  0.8261  0.4258  1.940  24.20°  209
  0.8261  0.4258  1.940  24.10°  209
Mean of Best-Fit Ellipses
  n     = 10 ellipses
  R     = 1.924
  ϕ     = 24.07°
  R 95% = 0.006
  ϕ 95% = 0.16°
```

3.3.7 Comparison of Void Fitting Methods

As discussed previously, Kumar et al. (2014) reviewed six void fitting methods, including *Delaunay triangulation nearest neighbors (DTNN)*, *density contrast* (point-count), *exponential edge detection (*continuous function*)*, and *mean log likelihood* methods, using two-dimensional simulated data sets, and concluded that of those six, the *exponential edge detection* method is the most accurate, followed by the *DTNN* method. The *weighted least-squares method* with the *Delaunay* option implemented here follows DTNN.

Table 2 is a summary of the results of the example analyses presented in this chapter. The results of additional tests on three-dimensional natural and simulated samples are in preparation (Vollmer, in preparation). While the examples presented here are a limited set of tests, some conclusions were drawn after extensive time implementing and testing the various methods. Note that while the code has been carefully checked, it is possible that there are implementation errors.

Of the methods that are independent of particle shape [a_i, b_i, ϕ_i], and *only* depend on [x_i, y_i] (Table 2), the *exponential edge detection* method is robust (Figures 40 and 41), and fast so bootstrap confidence intervals can be calculated in a few minutes. The *density contrast* method requires a parameter search, so is relatively slow and less suitable for bootstrapping. Kumar et al. (2014) concluded that it was less accurate. The *weighted least-squares* function method was found to be difficult to minimize robustly, so a parameter search is used. The maximum likelihood method (Shan and Xiao, 2011) is slow, and was concluded by Kumar et al. (2014) to gave less accurate results.

Of the methods that also require particle shape, [a_i, b_i, ϕ_i], the *enhanced normalized* method is fast, robust, and suitable for bootstrapping (Figures 23 and 24). Restricting the method to Delaunary nearest neighbors speeds up the analysis further, but reduces confidence (Figures 30 and 31, Table 2). The *normalized exponential edge detection* method is also fast and robust (Figures 42 and 43). This method,

as does the weighted least-squares method, fit the void edge, rather than points near the cloud around the edge, which may provide an advantage. The *normalized Delaunay weighted least-squares* method was found difficult to minimize robustly, so the slower parameter search is used.

Method	**R**	**Φ**	**R 95%**	**Φ 95%**	**XY Only**
Enhanced	1.662	25.83°	0.134	3.34°	
Enhanced Normalized	1.622	24.43°	0.078	1.84°	
Enhanced Normalized Delaunay	1.621	24.45°	0.621	14.75°	
Density Contrast	1.570	28.00°			√
Exponential Edge Detection	1.637	25.53°	0.165	4.83°	√
Normalized Exponential Edge Detection	1.740	24.85°	0.124	2.71°	
Mean Log Likelihood (N = 60)	1.900	25.00°			√
Weighted Least-Squares	1.590	27.60°			√
Delaunay Weighted Least-Squares	1.440	19.90°			√
Normalized Delaunay Weighted Least-Squares	1.920	24.40°			
Eigenvector	*1.627*	*25.74°*	*0.035*	*0.84°*	

Table 2 Summary of example results of void fitting methods on data file *E2 - Ramsay and Huber 1983 (small)*. Bootstrap confidence intervals are given if implemented, bootstrapping was not implemented if deemed impractical due to the required processing time. The eigenvector results for the particle mean ellipse are given for comparison.

4. Strain from Lines

4.1 Analytical Wellman Analysis

The Wellman method can be applied to objects in which *two lines* can be identified that have *constant initial angles*, such as brachiopod hinge and medial lines which are initially perpendicular (Wellman, 1962; Ramsay, 1967). For brachiopods not parallel to a principal strain, this angle will be distorted by shear strain.

Wellman's graphical technique is illustrated in many structural geology laboratory manuals (e.g., Ragan, 2009). An analytical solution to the problem was given by Vollmer (2011), which is implemented in implemented in EllipseFit. To try the method, open the file

LA - Ragan 1985 F10_1a.png

as an image. This is from Ragan (1985), and is used in many structural geology classes as an exercise. To begin click on the digitizing icon until the *Line Pair* icon is displayed, or use the menu command *Digitize Line Pair*. For each brachiopod click on the endpoints of each of the two lines, the hinge line and medial line. When done the lines appear in red, and the yellow cursor appears at the intersection. Mistakes can be corrected by using the *Cut* icon, or by deleting the line pair in the *Data Window*.

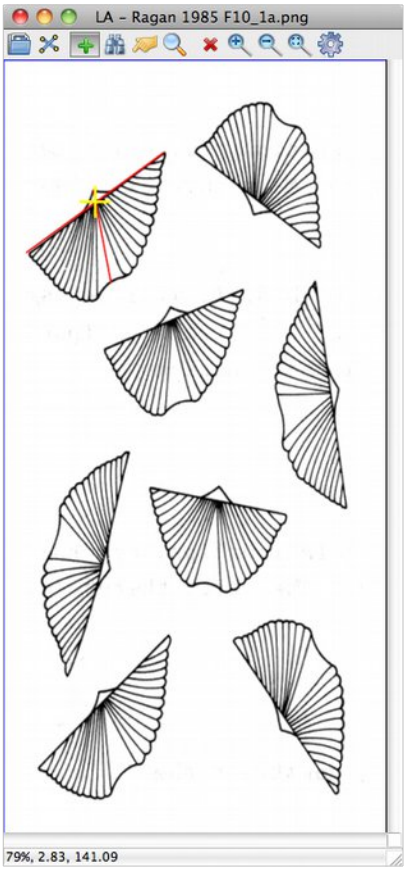


Figure 54. The *Image Window* after opening the example data from Ragan (1985). The hinge and medial lines are assumed initially perpendicular. One line pair has been digitized. Note the *Line Pair* icon is visible.

After digitizing one line pair, open the *Wellman Plot* using the menu command *Analyze > Wellman Plot*. The plot shows the parallelogram corresponding to the brachiopod (Figure 55). The parallelogram sides parallel the line pair. Note the two additional points used for the construction.

Continue digitizing the remaining line pairs. Figure 56 shows the plot after three line pairs. The yellow cross cursor highlights the corresponding data point intersection and parallelogram, and the data is selected in the *Data Window*. If the *Find* icon is pressed, as in Figure 56, you can search on the plot to locate the corresponding data. As in digitizing points, this allows the identification of outliers or spurious data.

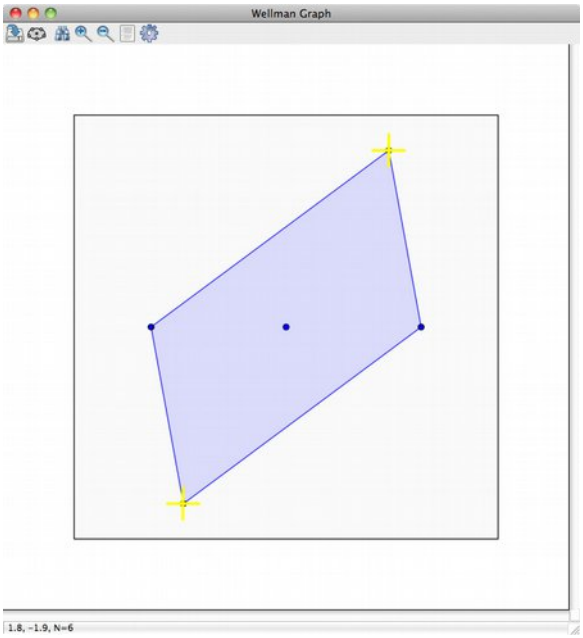


Figure 55: The analytical Wellman plot after digitizing one line pair as in Figure 26. Note the *Find* icon is selected and that the parallelogram and corresponding brachiopod are selected with the yellow cursor.

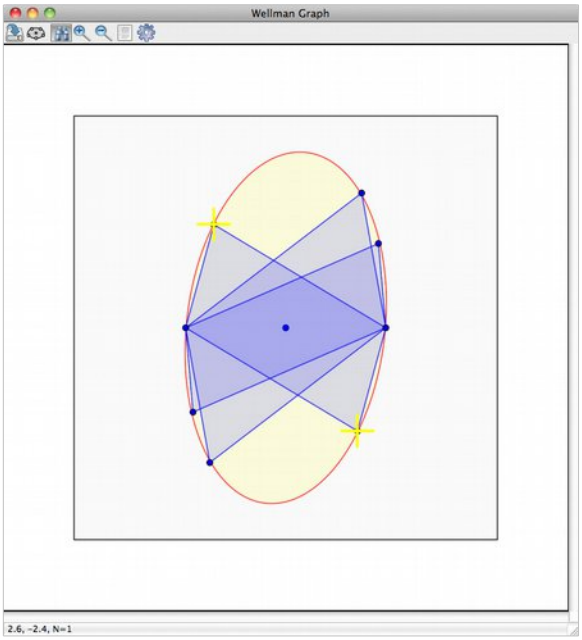


Figure 56: The analytical Wellman plot after three line pairs have been digitized.

Figure 57 shows the final analytical Wellman plot after all 8 line pairs have been digitized. Examine the

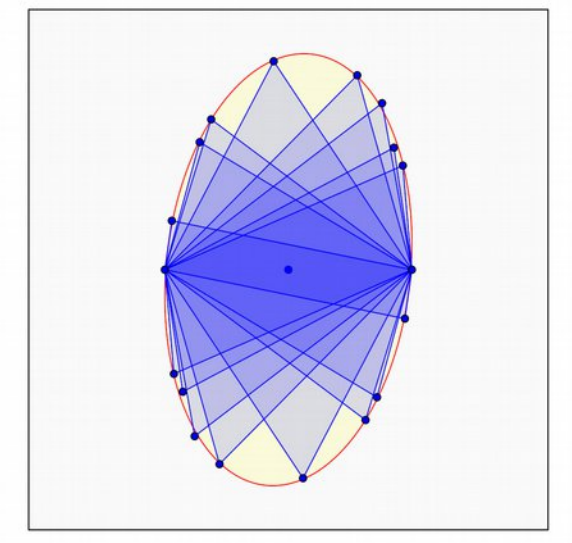


Figure 57: The final analytical Wellman plot after all 8 line pairs from the brachiopods in Figure 27 have been digitized.

Log Window (*Window > Log*) and note that at each step EllipseFit calculated the best-fit ellipse.

```
Analytical Wellman Ellipse Results
Wellman Data.tsv
2014-06-01 21:39:47
============================
N           = 8
Point pairs = 9 (symmetric)
R           = 1.773
ϕ           = 96.10°
n           = 9
RMS         = 0.025
```

The calculation is the same as described in Sections 3.1 and 3.2, minimizing the sum of the squares of the residuals the points from the ellipse using a LU decomposition. Similarly, the RMS value is the *root mean square* measure of the variation of the residuals from the ellipse, that is the square root of the sum of the squares of the residuals of the data from the fitted ellipse. It is a measure of goodness of fit of the ellipse, but is not technically an error. The RMS will be zero for two line pairs. The calculation includes the constriction line, so the ellipse has 9 point pairs including the 8 data points.

In theory, objects like graptolites that have a constant, non-perpendicular, angle between stipe and thecae, can be treated in the same fashion (Ramsay, 1967). Dirringer and Vollmer (2013) compared the automated Wellman method and the *mean polygon moment ellipse* method (Section 5.1) using a sample of slate with deformed Ordovician graptolites. The sample was oriented with the slaty cleavage as the X axis. The center lines and lower thecae lines were digitized in 120 locations for the Wellman test, only one species had clearly defined thecae lines. The outlines of 31 whole graptolites and 38 partial graptolites were digitized for the polygon method test.

The *mean polygon moment ellipse* was R = 2.079 ± 0.122, ϕ = 177.48° ± 4.57°, parallel to the slaty cleavage. The polygon method does not require assumptions about initial shapes, only that they are initially random. Interpreting the data for the analytical Wellman method was problematic, as it many outliers around a central ellipse. Removal of 77 outliers, believed to be due to initial variations in thecae angle, was required before the ellipse could be clearly resolved. While most outliers could be clearly identified, the process was subjective, and single outliers significantly effected the result. The result for 43 data points was R = 2.761, ϕ = 0.50°, RMS = 0.294, parallel to cleavage.

They concluded that the necessary assumptions about initial geometry for the analytical Wellman method were not met, and the polygon method, with no such required assumptions about initial geometry, was preferred.

Figure 58. Sample of deformed graptoliferous slate used by Dirringer and Vollmer (2013) for comparison of the automated Wellman and mean polygon moment ellipse methods.

Figure 59. The graptoliferous slate sample of Figure 24 after retrodeforming to remove the strain calculated by the mean polygon moment ellipse method, R = 2.079, ϕ = 177.48°

4.2 Line Stretch Analysis

5. Strain from Ellipses and Polygons

5.1 Digitizing Ellipses

5.2 Moment-Equivalent Polygons

6. Ellipse Plots

6.1 Elliott Polar Plot

The polar Elliott plot (Elliott, 1970) is a polar plot of the natural log R and 2ϕ. This is a natural parameter space for strain, and the plot is a simple hyperboloidal projection that gives an undistorted representation (Yamaji, 2008; Vollmer, 2011), It is therefore generally preferred over the R_f ϕ plot of the next section.

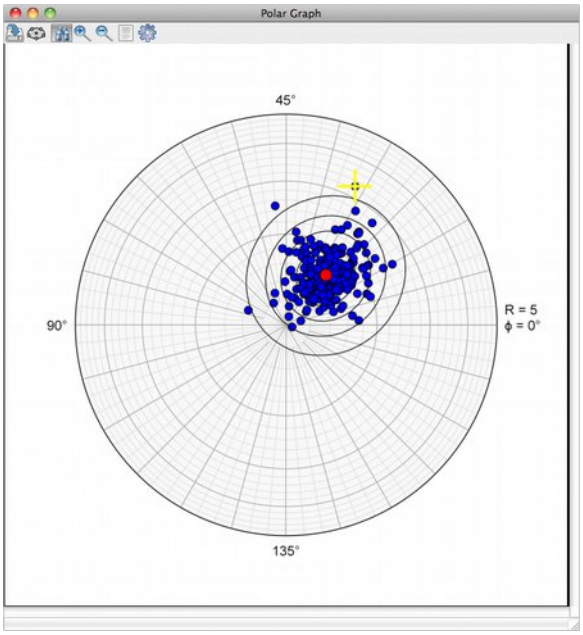


Figure 60. Polar Elliot plot with digitized data from the oolith photomicrograph in Figure 1. One outlier is selected.

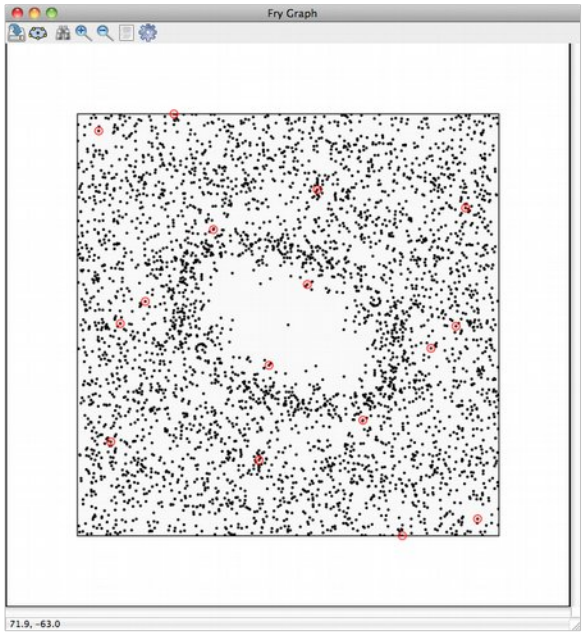


Figure 61. Fry plot with data generated from the oolith photomicrograph in Figure 1. The selected points are those generated by the outlier selected in the polar plot of Figure 32

Most of the plots in EllipseFit are interactive. When the *Binoculars Icon* is selected, points can be selected and the selection will automatically update on other plots and in the *Data Window*.To illustrate, Figure 61 shows a Fry plot with the points generated by the outlier selected in Figure 60. This outlier falls well inside the central void, and probably does not meet the assumptions necessary for a Fry analysis, i.e., a truncated Poisson distribution.

6.2 R_f ϕ Plot

The R_f ϕ plot (Dunnet, 1969) is a Cartesian plot of R_f , or more commonly natural log R_f , versus ϕ, and is widely used in strain analysis (e.g., Lisle, 1985; Ramsay and Huber, 1983). Although the origional plot had R_f, as the ordinate, it has been presented with either variable as the ordinate. This plot is probably more widely recognized and used than the polar Elliott plot, but it has more distortion at low strains.

The R_f ϕ plot distorts the strain space, especially at low strains (Vollmer, 2011). By analogy, a Mercator projection of the Earth projects the North and South Poles as lines, causing great distortion in polar regions, Greenland appears larger than South America on such a map, although it is one eight the size (Snyder, 1987). Similarly, the R_f ϕ plot projects the point of zero strain to a line, effectively stretching it along the R_f = 1 axis.

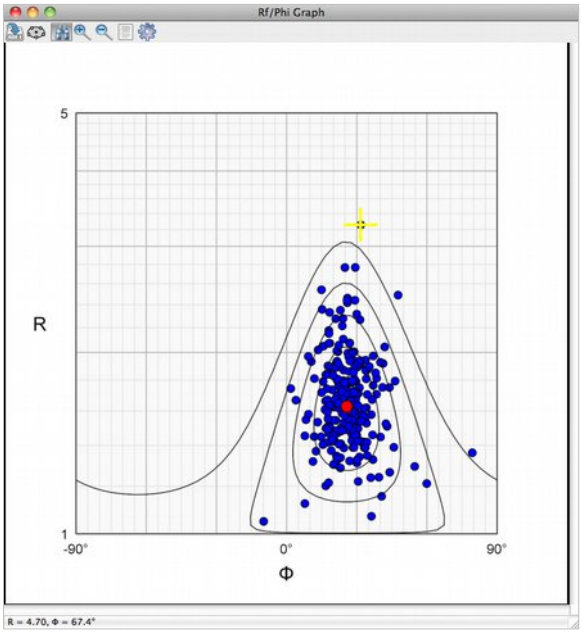


Figure 62. R_f ϕ plot with digitized data from the oolith photomicrograph in Figure 1, note the stretching near R = 1. One outlier is selected, the same as in Figures 32 and 33, all of which are automatically updated interactively.

6.3 Hyperbolic Plots

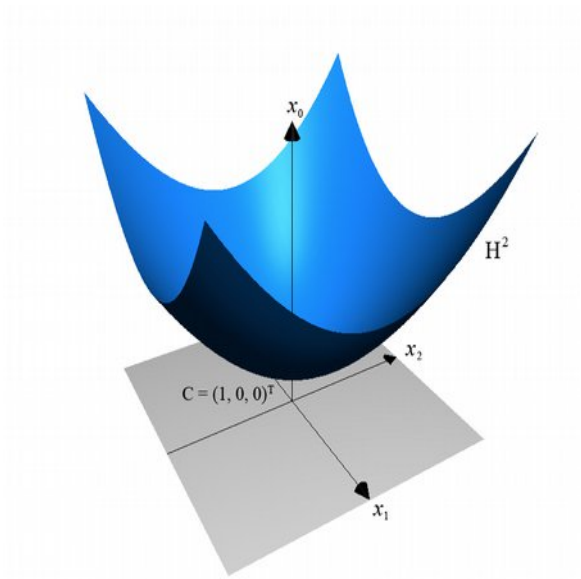


Figure 63. The unit hyperboloid, H^2, showing cartesian axes, x0, x1, x2, and point C = (1, 0, 0), which corresponds to the circle R = 1. The plane x_1x_2 is the projection plane for azimuthal projections, the polar strain plot. Points on H^2 are x = $(x_0, x_1, x_2)^T$, with origin C. If strain is represented by (ρ, ψ) = (log R, 2ϕ), then an ellipse is x = (cosh ρ, sinh ρ cos ψ, sinh ρ sin ψ)^T

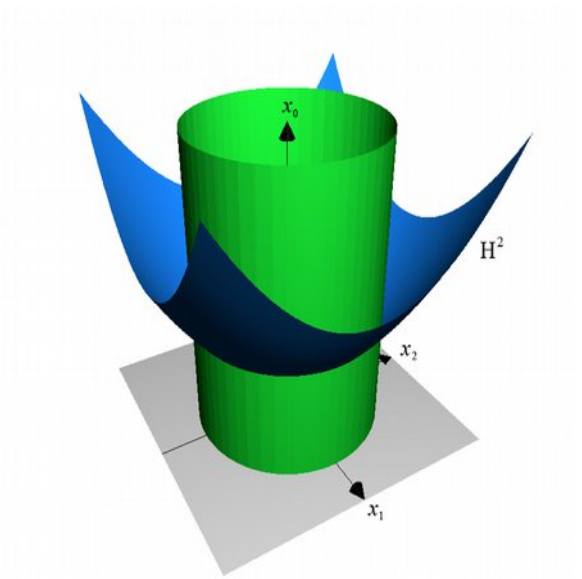


Figure 64. The unit hyperboloid with superimposed cylinder with axis x0. The cylinder is the projection surface for cylindrical projections, as the R_f ϕ plot.

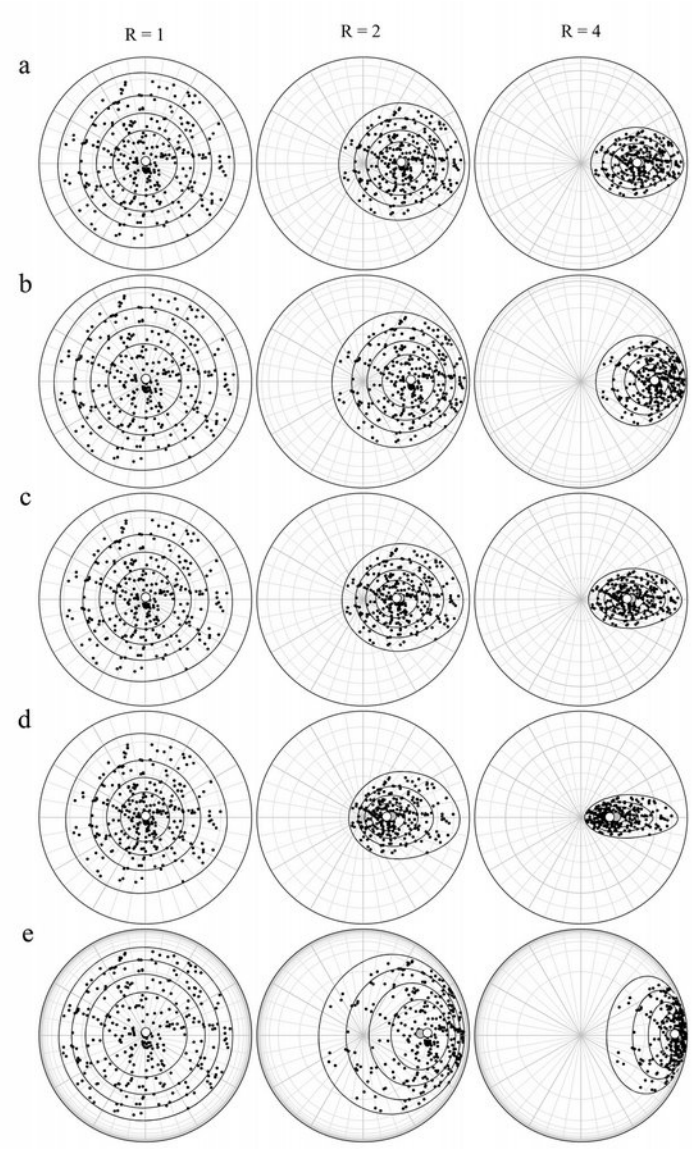


Figure 65. Synthetic data of 300 ellipses strained to values of R = 2 and R = 4 displayed on hyperboloidal azimuthal projections: (a) equidistant, (b) stereographic, (c) equal-area, (d) orthographic, and (e) gnomic.The best-fit ellipse is plotted as a white circle, the centroid of the projected data is plotted as a gray circle.

7. Mean Ellipse Calculation

Determination of the finite strain from a sample of initially randomly oriented ellipses deformed homogeneously with their matrix is of considerable importance for geological strain analysis, and numerous graphical and mathematical techniques exist (e.g. Ramsay, 1967; Ramsay and Huber, 1983; Mulchrone and Choudhury, 2004). The most direct method is a numerical calculation of the mean (Shimamoto and Ikeda, 1976; Mulchrone et al., 2003; Yamaji, 2008), which is discussed here.

Mulchrone and Choudhury (2004) showed that the method can be extended from ellipses to arbitrary shaped objects by using moment equivalent ellipses. This allows randomly oriented objects of *any* initial shape to be used in strain analysis, thus extending this technique considerably. This is implemented in EllipseFit using the *Polygon* and *Filled polygon* digitizing options.

7.1 Simple Means and Centroids

The *Centroids* and *Simple means* options give approximations less accurate than the true mean, and are provided for comparison only. The centroids of the polar and R_f ϕ plots can be plotted for comparison, but should not be confused with the true mean. Note in particular, that the centroid of the polar Elliot plot is close to, but is *not* equivalent to the true mean. The centroid of the R_f ϕ plot can deviate significantly from the true mean.

7.2 Mean Ellipse

Shimamoto and Ikeda (1976) devised a direct numerical solution involving the determination of the eigenvectors of the mean shape matrix, a two by two matrix representing an ellipse, which are summed, normalized, and the *eigenvectors* determined giving the mean ellipse, essentially a way of determining a mean for a matrix quantity.

Mulchrone et al. (2003) devised an equivalent method by determining the *mean radial length* (MRL), that does not require eigenvector calculation, and gives mathematically equivalent results. Yamaji (2008) showed that using a hyperbolic geometry, the mean can be calculated as a *hyperbolic vector mean*. Vollmer (2010) implemented these three methods independently in EllipseFit, and verified that these give numerically identical results (Table 1).

When calculating the mean ellipse using the *Calculate Ellipse* command, the method is selected in the *Calculate Ellipse* dialog, from the *Eigenvector*, *Mean radial length*, and *Hyperbolic mean* options. These three methods give identical results, and only one need be selected. EllipseFit uses the eigenvector method (Shimamoto and Ikeda, 1976), which is marginally faster, by default.

Data Set	Imposed (R, ϕ)	Eigenvector	Mean Radial	Hyperbolic
Oolith	1, 0	1.628, 25.74	1.628, 25.74	1.628, 25.74
n = 252		± 0.018, 0.73	± 0.018, 0.62	± 0.013
	0.614, 25.74	1.000, 113.32	1.000, 113.32	1.000, 113.32
		± 0.007, 55.27	± 0.011, 633.74	± 0.013
Synth 1	1, 0	1.031, 40.20	1.031, 40.20	1.031, 40.20
n = 300		± 0.021, 33.24	± 0.025, 22.81	± 0.030
	2, 0	2.012, 1.16	2.012, 1.16	2.012, 1.16
		± 0.048, 1.16	± 0.050, 0.92	± 0.032
	4, 0	4.023, 0.46	4.023, 0.46	4.023, 0.46
		± 0.101, 0.53	± 0.099, 0.37	± 0.031
Synth 2	1, 0	1.016, 146.03	1.016, 146.03	1.016, 146.03
n = 1000		± 0.012, 35.35	± 0.014, 24.51	± 0.016
	2, 0	2.012, 179.46	2.012, 179.46	2.012, 179.46
		± 0.026, 0.71	± 0.27, 0.51	± 0.016
	4, 0	4.024, 179.78	4.024, 179.78	4.024, 179.78
		± 0.052, 0.30	± 0.053, 0.21	± 0.017

Table 3. Comparative results for ellipse-fitting techniques implemented in EllipseFit. Eigenvector = Shape matrix eigenvectors (Shimamoto and Ikeda, 1976). Radial = Mean radial length (Mulchrone, et al, 2003; Mulchrone, 2005). Hyperboloidal = Hyperboloidal vector mean (Yamaji, 2008). From Vollmer (2010).

7.3 Bootstrap Error Analysis

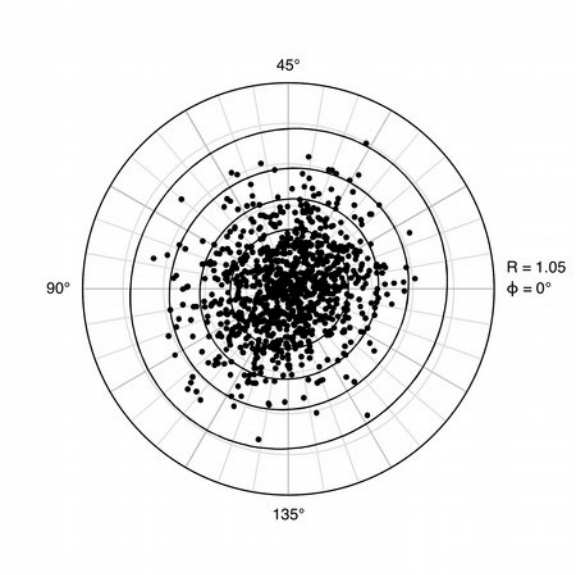


Figure 66. Error analysis is shown by an equidistant azimuthal plot of bootstrap results of 1000 resamples from oolite data. The mean vector of the bootstrap mean vectors is rotated to C. The dispersion of the points is a measure of the error in the best-fit ellipse.

8. Ellipsoid Calculation

For regional strain studies it is generally necessary to determine the three-dimensional strain ellipsoid, with three stretches and their orientations, normally expressed as trends and plunges. This can be simplified if assumptions can be made about the relationship between foliations and strain, for example slaty cleavage is commonly assumed perpendicular to the minimum stretch. However, in the general case it is necessary to determine the two-dimensional strain on a number of different planes through a sample (or outcrop where it can be considered homogeneous), and combine them to determine the strain ellipsoid in three dimensions. This is a difficult mathematical problem, and numerous solutions have been proposed (e.g., Shimamoto and Ikeda 1976; Owens, 1984; Robin, 2002; Shan, 2008; Mookerjee and Nickleach, 2011). EllipseFit implements the method of Shan (2008) as discussed in Section 8.2.

8.1 Global Coordinates and Sample Collection

The two-dimensional strain ellipses considered thus far have been referred to X, Y coordinates, where X is to the right, and Y is down the image. These coordinate axes are indicated by the blue lines on the top and left of the *Image Window*. The angle ϕ is the positive angle (clockwise) from X. This coordinate system was chosen to simplify the relationship to the *global coordinates* referred to here as X', Y', Z', and to simplify the calculation of the three-dimensional strain ellipsoid. The global coordinates are equivalent to North, East, Down (NED).

In Figure 67 the gray plane is a *section plane* that corresponds to an image analyzed for two-dimensional strain as discussed in earlier chapters. The X axis is parallel to the strike of the plane, using the standard *right hand rule* (e.g., Pollard and Fletcher, 2005), as shown in Figure 67. The strike is given by θ, the clockwise angle from North, the standard azimuth in degrees. The dip of the plane is the angle δ. The calculated strain ellipse is given by $R = A/B = L_{Max}/L_{Min}$, and ϕ, the angle from X, which is its *pitch* in global coordinates. This is referred to here as a *section ellipse*.

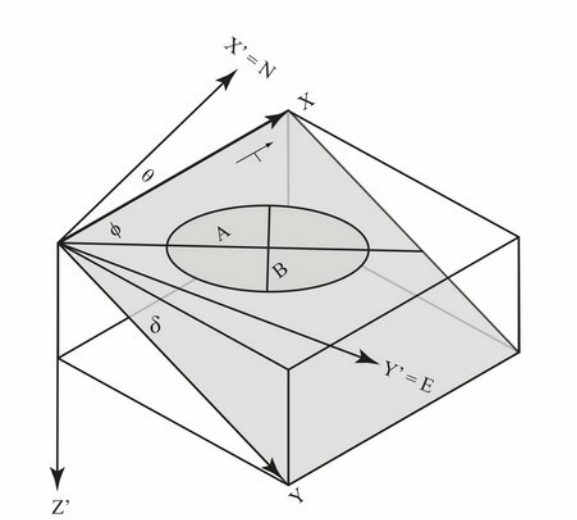


Figure 67. Coordinate system for section ellipses. The global coordinates are X' = North, Y' = East, and Z' = Down (NED). The plane with the section ellipse has a strike, θ (using the right hand rule), and dip, δ. The section ellipse has a pitch, ϕ, and R = A/B, where A and B are the maximum and minimum axes. A suggested strike arrow and dip tick marking is shown.

In order to calculate the strain ellipsoid from the section ellipses, each section ellipse must undergo a coordinate transformation from local X, Y coordinates to global X', Y', Z' coordinates. This is done automatically by EllipseFit, but the user must take great care to properly prepare samples. Time taken at this stage will save much aggravation later on. A sample collected in the field must be carefully oriented, recording its strike and dip (other conventions are fine, but the strike is the X coordinate axis so is used here). A suitable marking is a *strike arrow* and a *dip tick* (Figure 67), if possible on a surface that is not overhanging.

A minimum of three sections must be made through the sample, although more is preferred. Shan's method (Section 8.2) relaxes this requirement if lineation data is used as well, but Vollmer (2010) showed that the error range in natural samples can be large, so a minimum of three sections is recommended. If available, lineation data can supplement the section ellipses (Section 8.2).

The sections should be made at high angles to each other, but it does not need to be 90°, a restriction of some methods (e.g., Shimamoto and Ikeda, 1976). In making the sections be careful not to destroy the strike arrow and dip tick (it happens). The sample can then be taken outside, away from magnetic fields, and reoriented. The strikes and dips of the section planes can then be measured, and a strike arrow and dip tick marked on each face. The faces can then be photographed, or thin sections made, and photographed. Keeping thin sections correctly oriented is challenging, keep the strike arrow parallel to one side and pointing right.

To minimize confusion, make sure each photograph is oriented with the *section strike to the right*, and

with the *dip line down*. Careful photography is best, but EllipseFit can rotate an image an arbitrary amount if necessary (see Chapter 12 Image Analysis). It is better to do it now than after digitizing the data, although EllipseFit can rotate the data if needed (see Chapter 11 Data Transformation).

One last *important* detail is to keep track of the viewing direction. The *strike arrow must point to the right* in the section image. This means it is dipping towards you. If the strike arrow points left, you are looking at the *underside* of the section and it is dipping away from you. If so, you need to *flip the image horizontally* about a vertical axis. EllipseFit can do this (Edit > Rotate Image > Flip Horizontal), and it is better to fix the image before digitizing. Vertical sections are not a problem if the recorded strike is kept to the right in the images.

If one is lucky to have outcrops with well exposed sections the process is greatly simplified, but the same principles apply.

Fields	Alternate	Symbol	Definition
ID	N		Datum identification number
X', Y', Z'			Global coordinates (North, East, Down)
X, Y			Local coordinates, normally strike and dip line
Strike	Theta	θ	Strike of section following right-hand rule
Dip	Delta	δ	Dip of section plane from horizontal
Max, Int, Min	A, B, C		Axes of an ellipsoid
Max, Min	A, B		Axes of a sectional ellipse
R			Strain ratio, Max/Min
Phi	Pitch	ϕ	Angle in XY from X to ellipse axis Max
R*			Best-fit estimate of R
Phi*		ϕ*	Best-fit estimate of ϕ
Delta R		ΔR	Misfit between R* and R
Delta Phi		Δϕ	Misfit between ϕ* and ϕ
S1, S2, S3		S1, S2, S3	Principal stretches
Trend		t1, t2, t3	Trend of ellipsoid axis
Plunge		p1, p2, p3	Plunge of ellipsoid axis

Table 4. Data file field headers and corresponding symbols. The headers define columns in data files read and written by EllipseFit. .

8.2 Ellipsoid Calculation

Shan's method for determining the strain ellipsoid from section ellipses has similarities to the methods of Owens (1984) and Robin (2002), as they are all direct non-iterative calculations. Shan's method, however, also allows the inclusion of stretching lineation data, so has additional flexibility. Ellipsoids can be represented by *shape matrixes*, and the solution desired is the optimal shape matrix. Each section ellipse, or section lineation, adds one or two linear equations describing the shape matrix, which can be solved as an eigenvalue problem. Shan solved the problem by assuming the matrix can be

located on a six-dimensional hypersphere centered at the origin, and recognized that the smallest eigenvector of the data matrix is an optimal solution.

Before giving an example calculation, it is useful to compare it with some other methods. Shan's method has been tested on synthetic and natural samples, the following are some of the results of Vollmer (2010). Owens (1984) tested his method on a sample of slate from Dinorwic North Wales, for which the strains had been calculated from reduction spots on 8 sections. His data was also used by Launeau and Robin (2005) to test Robin's (2002) method. Table 3 shows results of Vollmer's (2010) tests on Shan's method using Owen's data.

j	θ	δ	A	B	R	ϕ	R*	ϕ*	ΔR	Δϕ	RT*	ϕT*	ΔRT	ΔϕT
1	302	78	16.5	4.5	3.670	165	3.083	165.700	0.587	0.700	3.082	165.700	0.002	0.000
2	301	77	9.5	3.5	2.710	166	3.076	165.380	0.366	0.620	3.075	165.380	0.005	0.000
3	302	75	20.5	6.8	3.010	166	3.024	165.310	0.014	0.690	3.023	165.310	0.003	0.010
4	201	71	37.0	6.0	6.170	173	6.418	172.780	0.248	0.220	6.420	172.780	0.001	0.000
5	178	71	7.5	1.5	5.000	0	4.618	179.090	0.382	0.910	4.618	179.090	0.002	0.000
6	18	79	16.7	3.0	5.570	10	5.923	7.870	0.353	2.130	5.924	7.870	0.004	0.000
7	17	78	22.0	4.0	5.500	8	5.792	7.710	0.292	0.290	5.793	7.710	0.003	0.000
8	19	78	18.0	3.0	6.000	7	5.987	8.200	0.013	1.200	5.989	8.200	0.001	0.000

Table 5. Results of test of Shan's (2008) method using data from Owens (1984). R*, ϕ* are the calculated b* (Table 4) section ellipses. Misfits ΔR, Δϕ indicate the error between calculated and measured ellipses. Calculated section ellipses were used to back-calculate bT* (Table 4) and RT*, ϕT*. Misfits ΔRT, ΔϕT indicate that the method does retrieve b*. From Vollmer (2010).

The test involves calculating the strain ellipsoid from the section ellipses, then from the calculated ellipsoid, determining the two-dimensional sections corresponding to the input data. These are reported as R*, ϕ* in the table. The difference is a *residual*. These are reported as ΔR, Δϕ in the table. An additional result is shown by using the calculated section ellipses to calculate an ellipsoid. These are reported as ΔRT, ΔϕT, and are negligible indicating success in retrieving the ellipsoid. Table 4 shows the results of the ellipsoid calculation from this sample as calculated using the methods of Owens (1984), Robin (2002), and Shan (2008). The results are compared graphically in Figure 68. The calculations and plots were done in EllipseFit 2 (Vollmer, 2011) and Orient 2 (Vollmer, 2012). There negligible differences between the results using the methods of Robin and Shan, the results using the method of Owen deviate a small amount from them.

Axis	Owens	Robin	Shan (b*)	b**
S1	2.340	2.626	2.565	2.567
t1	29.000	37.100	34.960	34.970
p1	10.000	11.300	10.890	10.890
S2	1.197	1.112	1.131	1.131
t2	122.000	129.500	127.350	127.360
p2	14.000	11.700	12.230	12.230
S3	0.357	0.343	0.345	0.345
t3	265.000	264.500	264.440	264.440
p3	73.000	73.600	73.510	73.510

Table 6. Comparison of calculated strain ellipsoids. Owens from Owens (1984). Robin from Launeau and Robin (2005), unweighted method of Robin (2002). Shan (b*) from Vollmer (2010), Shan's (2008) method. b** is a test to retrieve b*. The data is plotted in Figure 38. From Vollmer (2010).

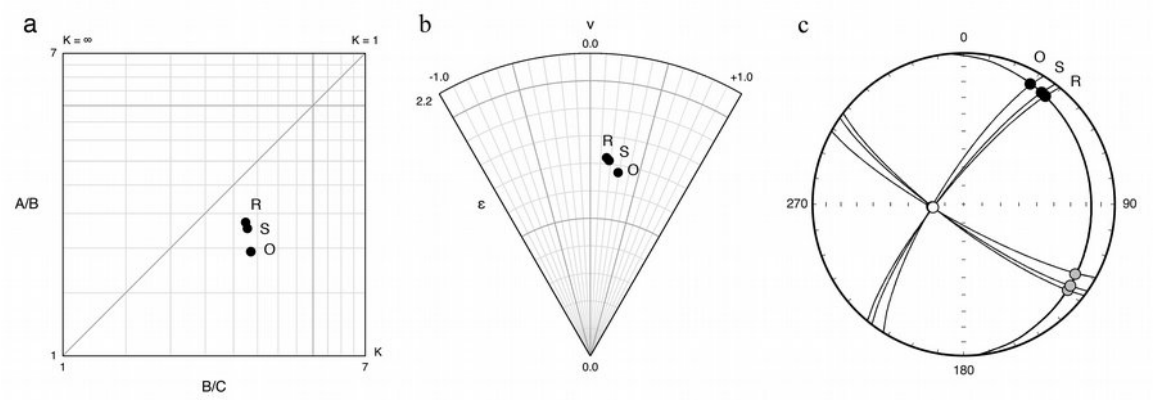


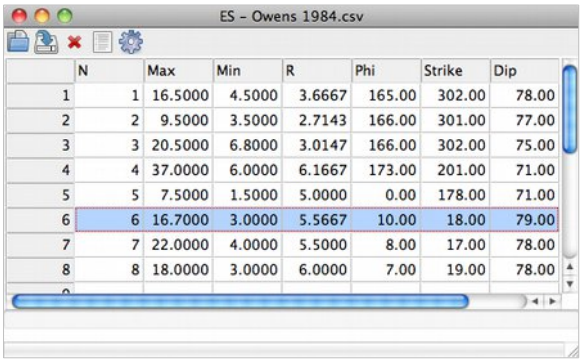
Figure 68. Comparison of calculated strain ellipsoids. O = Owens (1984). R = Launeau and Robin (2005) using unweighted method of Robin (2002). S = EllipseFit using Shan's (2008) method. From Vollmer (2010).

The file:

ES - Owens 1984.csv

contains the 8 section ellipse data from Owens (1984). Open this file in EllipseFit. The data as displayed in the Data Window is shown in Figure 69. There are 8 section ellipses, for each there is the Max, and Min (the axial lengths L_{Max}, L_{Min}), the strain ratio R = Max / Min, Phi (ϕ), the pitch of R from the X axis (X = strike), the strike angle (θ), and the dip angle (δ) (see Figure 67). This is data then, that, in EllipseFit, would be determined from oriented photographs of each of the 8 sections.

Select the command *Analyze* > *Calculate Ellipsoid* and the Calculate Ellipsoid Dialog is displayed as in Figure 70. The results will be written to the *Log Window*. Checking *Append results* will append the



	N	Max	Min	R	Phi	Strike	Dip
1	1	16.5000	4.5000	3.6667	165.00	302.00	78.00
2	2	9.5000	3.5000	2.7143	166.00	301.00	77.00
3	3	20.5000	6.8000	3.0147	166.00	302.00	75.00
4	4	37.0000	6.0000	6.1667	173.00	201.00	71.00
5	5	7.5000	1.5000	5.0000	0.00	178.00	71.00
6	6	16.7000	3.0000	5.5667	10.00	18.00	79.00
7	7	22.0000	4.0000	5.5000	8.00	17.00	78.00
8	8	18.0000	3.0000	6.0000	7.00	19.00	78.00

Figure 69. The section data from a sample of slate from Dinorwic, North Wales from Owens (1984), displayed in the EllipseFit *Data Window*.

ellipsoid results to the open *Data Window*, so it can be plotted on Flinn and Nadia plots. Check *Save orientations* to save the trends and plunges of the principal axes to a file that can be opened in Orient 3 (Vollmer, 2015) for plotting the axes on spherical projections.

The *Bootstrap* option performs a bootstrap-type error analysis, using the number of resamples specified in the *Resamples* edit box, 5000 is the default value. Finally, the *Save bootstrap* will save the 5000 results of the resampling, which is normally unnecessary. Press *OK* to start the calculation. You will be prompted to save the orientation data files, and shortly the results appear in the *Data Window* (Figure 71) and the *Log Window*.

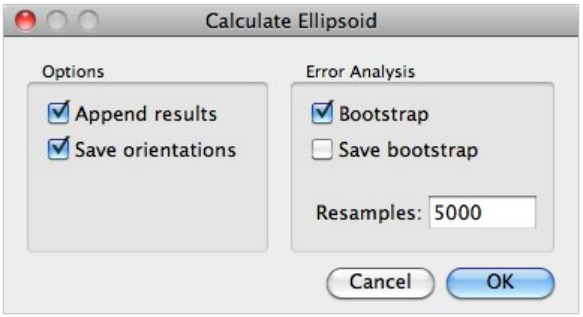
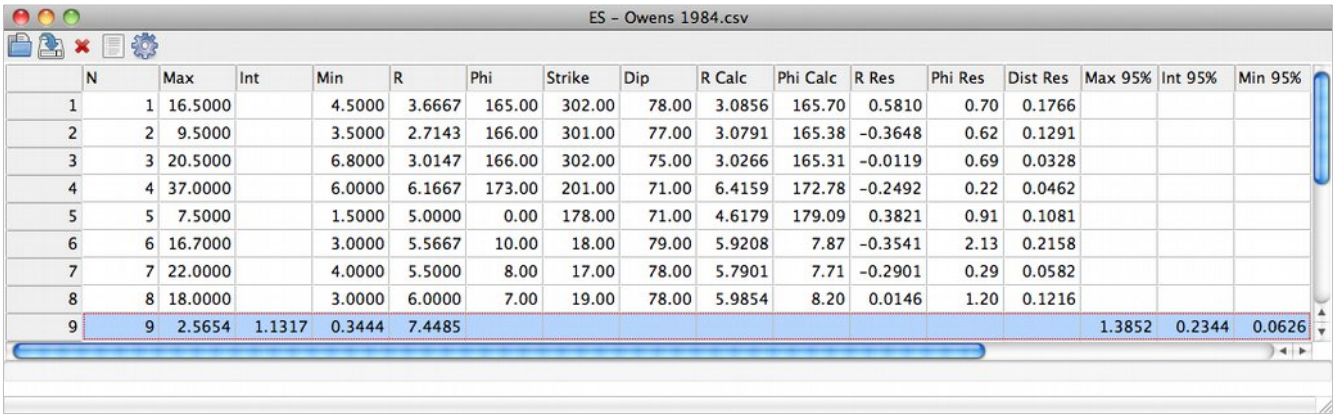


Figure 70. EllipseFit's *Calculate Ellipsoid Dialog*.

The *Data Window* now displays the ellipsoid principal axes Max, Int, Min as stretches (S_{Max}, S_{Int}, S_{Min}), and 95% confidence intervals calculated by the bootstrap. The section ellipses show the back-calculated values for R and ϕ, and the corresponding residuals. The last columns the *distance residuals*, which are the hyperbolic distance residuals.



	N	Max	Int	Min	R	Phi	Strike	Dip	R Calc	Phi Calc	R Res	Phi Res	Dist Res	Max 95%	Int 95%	Min 95%
1	1	16.5000		4.5000	3.6667	165.00	302.00	78.00	3.0856	165.70	0.5810	0.70	0.1766			
2	2	9.5000		3.5000	2.7143	166.00	301.00	77.00	3.0791	165.38	-0.3648	0.62	0.1291			
3	3	20.5000		6.8000	3.0147	166.00	302.00	75.00	3.0266	165.31	-0.0119	0.69	0.0328			
4	4	37.0000		6.0000	6.1667	173.00	201.00	71.00	6.4159	172.78	-0.2492	0.22	0.0462			
5	5	7.5000		1.5000	5.0000	0.00	178.00	71.00	4.6179	179.09	0.3821	0.91	0.1081			
6	6	16.7000		3.0000	5.5667	10.00	18.00	79.00	5.9208	7.87	-0.3541	2.13	0.2158			
7	7	22.0000		4.0000	5.5000	8.00	17.00	78.00	5.7901	7.71	-0.2901	0.29	0.0582			
8	8	18.0000		3.0000	6.0000	7.00	19.00	78.00	5.9854	8.20	0.0146	1.20	0.1216			
9	9	2.5654	1.1317	0.3444	7.4485									1.3852	0.2344	0.0626

Figure 71. The *Data Window* after calculating the optimal ellipse using Shan's method.

The *Log Window* reports the following:

```
Best-Fit Ellipsoid Calculations
ES - Owens 1984
2014-06-02 19:51:39
============================
N = 8
Ellipsoid axes as stretches:
Maximum (A)      = 2.565
Trend            = 35.02
Plunge           = 10.90
Intermediate (B) = 1.132
Trend            = 127.41
Plunge           = 12.22
Minimum (C)      = 0.344
Trend            = 264.44
Plunge           = 73.51
Root mean square of section residuals:
R +/-            = 0.333
Phi +/-          = 0.85
Distance +/-     = 0.126
See data grid for section residuals
Bootstrap confidence intervals (5000 resamples)
Maximum (A):
  Stretch +/-    = 0.973
  Stretch 95%    = 1.385
  Stretch 99%    = 3.603
  Trend +/-      = 0.186
  Trend 95%      = 0.269
  Trend 99%      = 0.369
  Plunge +/-     = 0.037
  Plunge 95%     = 0.058
  Plunge 99%     = 0.083
Intermediate (B):
  Stretch +/-    = 0.106
  Stretch 95%    = 0.234
  Stretch 99%    = 0.415
```

```
  Trend +/-      = 0.187
  Trend 95%      = 0.273
  Trend 99%      = 0.382
  Plunge +/-     = 0.041
  Plunge 95%     = 0.057
  Plunge 99%     = 0.073
Minimum (C):
  Stretch +/-    = 0.030
  Stretch 95%    = 0.063
  Stretch 99%    = 0.117
  Trend +/-      = 0.031
  Trend 95%      = 0.043
  Trend 99%      = 0.056
  Plunge +/-     = 0.014
  Plunge 95%     = 0.020
  Plunge 99%     = 0.026
```

This includes all 3 principal stretches, and their trends and plunges, with measures of error. To view the results graphically, first select *Analyze > Flinn Plot*. A Flinn plot (Section 9.1) is a plot of the ratios $A/B = S_{Max}/S_{Int}$ versus $B/C = S_{Int}/S_{Min}$. , and is commonly used for displaying strain ellipsoid data (e.g. Ramsay and Huber, 1983).

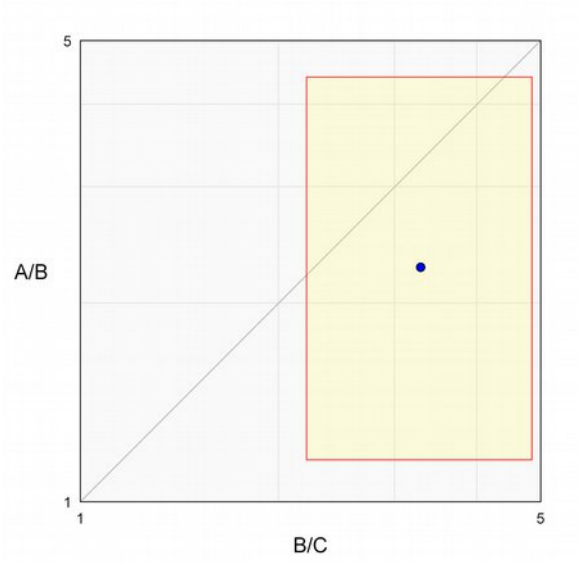


Figure 72. Flinn plot of the ellipsoid axial ratios determined from the Shan calculation, with a 95% confidence region.

Now select *Analyse > Nadia Plot*, to display the results on a Nadai plot. A Nadai plot (Nadia, 1950; Hossack, 1968; Section 9.2) is based on *natural*, or logarithmic strain, which is also the basis for the hyberboldal projections discussed in Section 6.3. This provides an undistorted representation of the deviatoric strains and is preferred by many for that reason (Brandon, 1995).

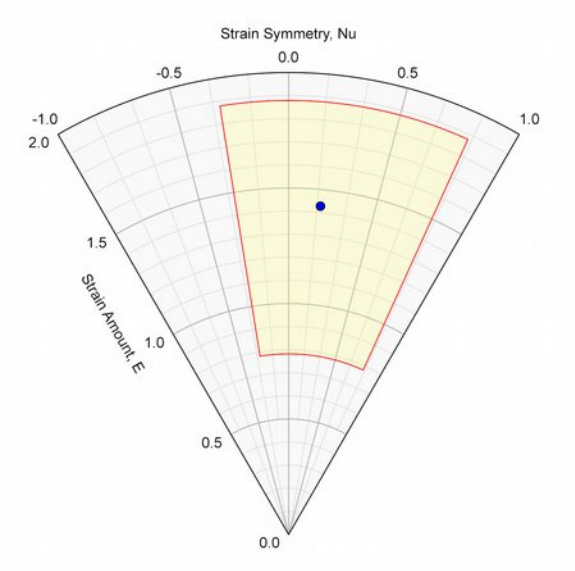


Figure 73. Nadai plot of the ellipsoid axial ratios determined from the Shan calculation with a 95% confidence region.

The calculated strain has large 95% error region as shown in both plots. Examining the data (Figure 71), shows that section 6 has the largest distance residual. Select it, delete it and preform the ellipsoid calculation again. Figure 74 shows the updated Flinn plot, which now shows both solutions.

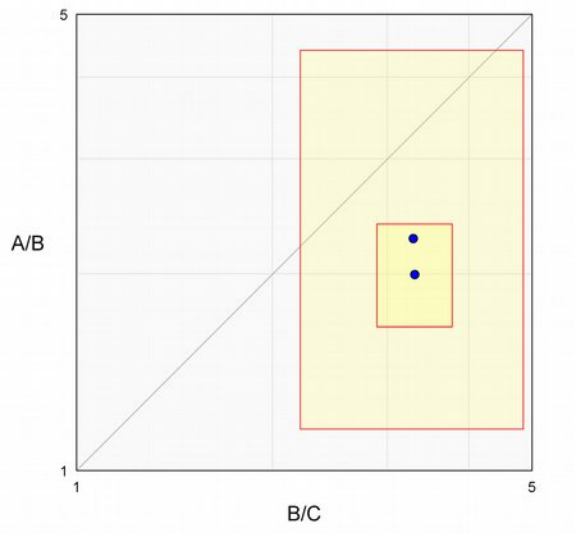


Figure 74. Flinn plot of the ellipsoid axial ratios determined from the Shan calculation, with 95% confidence regions, after deleting section 6.

Similarly the Nadia plot has been updated to reflect the newly calculated results.

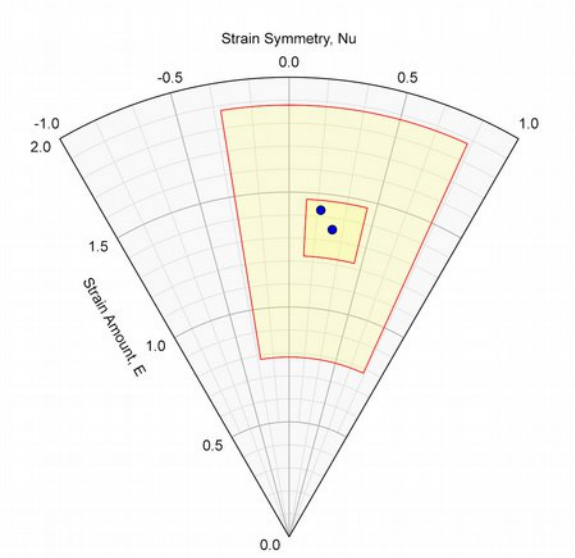


Figure 75. Nadia plot of the ellipsoid axial ratios determined from the Shan calculation, with 95% confidence regions, after deleting section 6.

Finally, the resulting axes are plotted on a lower hemisphere equal-area projection using Orient (Vollmer, 2010, 2015). The strain axes calculated from all 8 sections are plotted as circles, and the axes section 6 removed are plotted as diamonds. Red = S_{Max}, green = R_{Int}, blue = R_{Min}.

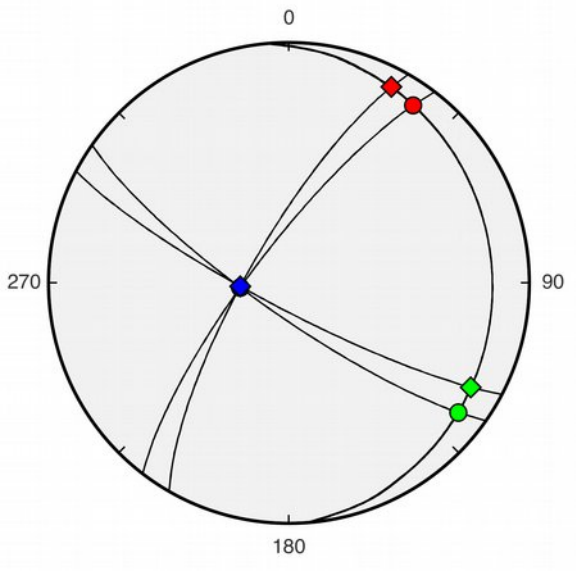


Figure 76. Lower hemisphere equal-area projection of the strain ellipsoid axes. Circles are the axes calculated from all 8 sections, diamonds with section 6 removed. Red = S_{Max}, green = R_{Int}, blue = R_{Min}.

Axis	**bT14***	**bT24***	**bT34***	**bT45***	**bT46***	**bT47***	**bT48***	**bT56***	**bT57***	**bT58***
S1	2.569	2.570	2.570	2.570	2.569	2.569	2.570	2.568	2.568	2.570
t1	35.060	35.100	35.010	35.180	35.030	35.030	35.010	35.230	35.220	35.010
p1	10.900	10.910	10.890	10.930	10.900	10.900	10.890	10.940	10.940	10.890
S2	1.130	1.131	1.130	1.132	1.130	1.130	1.130	1.133	1.133	1.130
t2	127.450	127.490	127.400	127.570	127.420	127.420	127.400	127.620	127.610	127.400
p2	12.210	12.200	12.220	12.190	12.220	12.220	12.220	12.180	12.180	12.220
S3	0.344	0.344	0.344	0.344	0.344	0.344	0.344	0.344	0.344	0.344
t3	264.450	264.440	264.450	264.440	264.450	264.450	264.450	264.440	264.440	264.450
p3	73.510	73.520	73.510	73.520	73.510	73.510	73.510	73.520	73.520	73.510

Table 7. Results of test of ellipsoid-fitting using two ellipses and six lineations from synthetic section ellipses calculated from b* (Table 4). For ten tests six of the eight RTj values were omitted. Subscripts indicate the sections with RTj data. Results are all identical down to round-off error.

Axis	b14*	b24*	b34*	b45*	b46*	b47*	b48*	b56*	b57*	b58*
S1	nan	3.422	4.379	3.196	3.389	3.371	3.469	3.126	3.301	3.127
t1	nan	41.760	47.150	43.140	20.310	20.330	20.320	42.680	45.960	37.500
p1	nan	11.690	12.580	12.310	8.060	8.060	8.060	12.240	12.790	11.280
S2	nan	0.902	0.836	1.052	0.584	0.585	0.578	0.301	1.054	1.021
t2	nan	133.950	139.230	135.430	235.100	234.930	235.570	264.470	138.190	129.850
p2	nan	10.430	9.240	10.370	80.220	80.240	80.160	73.780	9.730	11.610
S3	nan	0.323	0.273	0.297	0.505	0.507	0.499	0.561	0.287	0.313
t3	nan	264.630	264.590	264.450	111.090	111.110	111.110	264.470	264.450	264.490
p3	nan	74.230	74.300	73.800	5.510	5.470	5.600	73.780	73.830	73.700

Table 8. Test of ellipsoid-fitting using two ellipses and six lineations from eight measured section ellipses (Table 5). For ten tests six of the eight Rj values were omitted. Subscripts indicate the sections with Rj data. Results are highly variable, especially as axial ratios, which are plotted in Fig. 8.

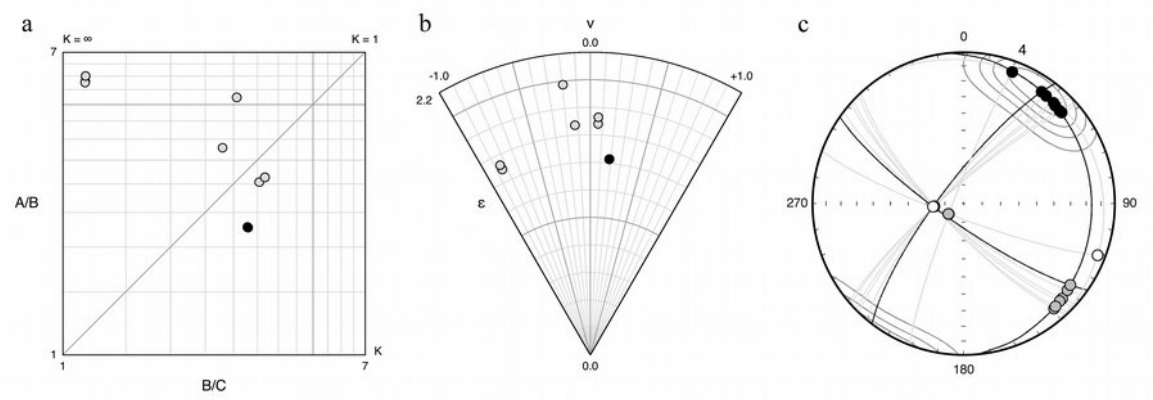


Figure 77. Test of ellipsoid-fitting using two ellipses and six lineations from eight measured section ellipses (Table 5). For ten tests six of the eight Rj values were omitted. Subscripts indicate the sections with Rj data. Results are highly variable, especially as axial ratios.

9. Ellipsoid Data Plots

9.1 Flinn Plot

A Flinn plot is a plot of the ratios $A/B = S_{Max}/S_{Int}$ versus $B/C = S_{Int}/S_{Min}$, and is commonly used for displaying strain ellipsoid data (e.g. Ramsay and Huber, 1983).

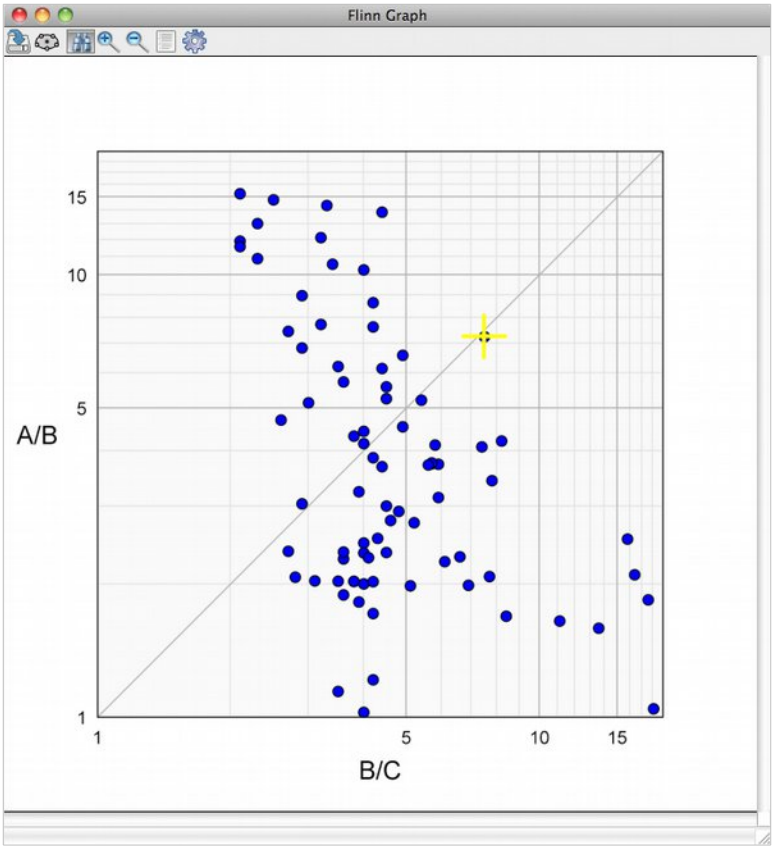


Figure 78. Log Flinn plot displaying deformed pebble ellipsoids, Bygdin area, Norway, from Hossack, 1968. This plot is interactive, with the *Binoculars Icon* selected, data points can be selected and will be simultaneously updated on the Nadai plot and in the *Data Window*, the selected data point is also displayed in Figure 51.

As with the ellipse plots, the Flinn and Nadia plots are interactive, selecting a point in one will automatically select the corresponding data point on the other plot, and in the *Data Window*.

9.2 Hsü-Nadai Plot

The Nadai plot (Nadia, 1950; Hossack, 1968; Section 9.2) is based on *natural*, or logarithmic strain, which is also the basis for the hyberboldal projections discussed in Section 6.3. This provides an undistorted representation of the deviatoric strains and is preferred by many for that reason (Brandon, 1995).

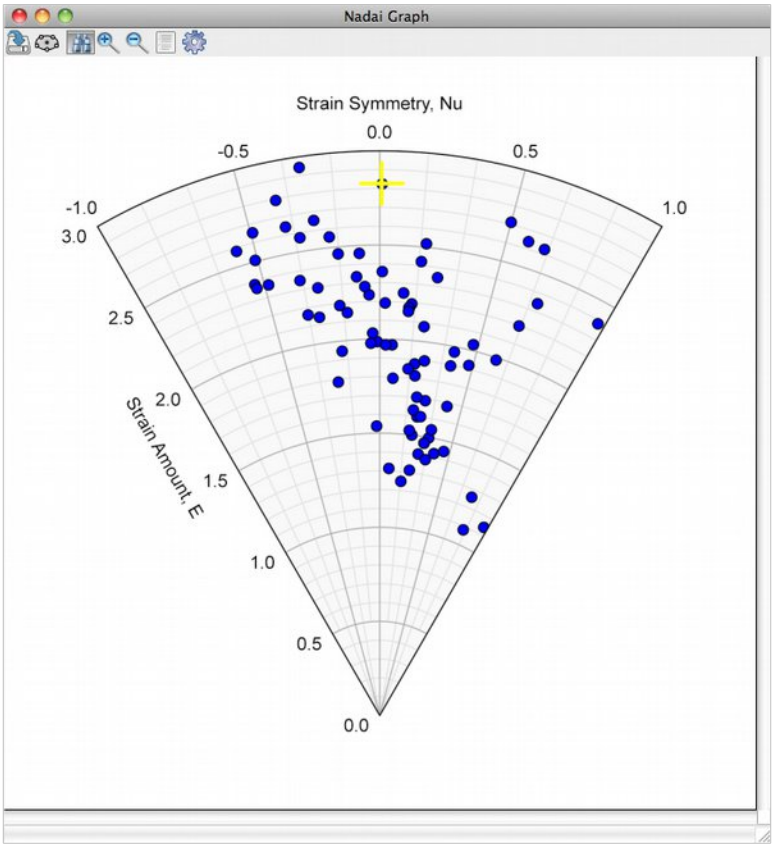


Figure 79. Nadia plot displaying deformed pebble ellipsoids, Bygdin area, Norway, from Hossack, 1968. This plot is interactive, with the *Find* icon selected, data points can be selected and will be simultaneously updated on the Flinn plot and in the *Data Window*, the selected data point is also displayed in Figure 48.

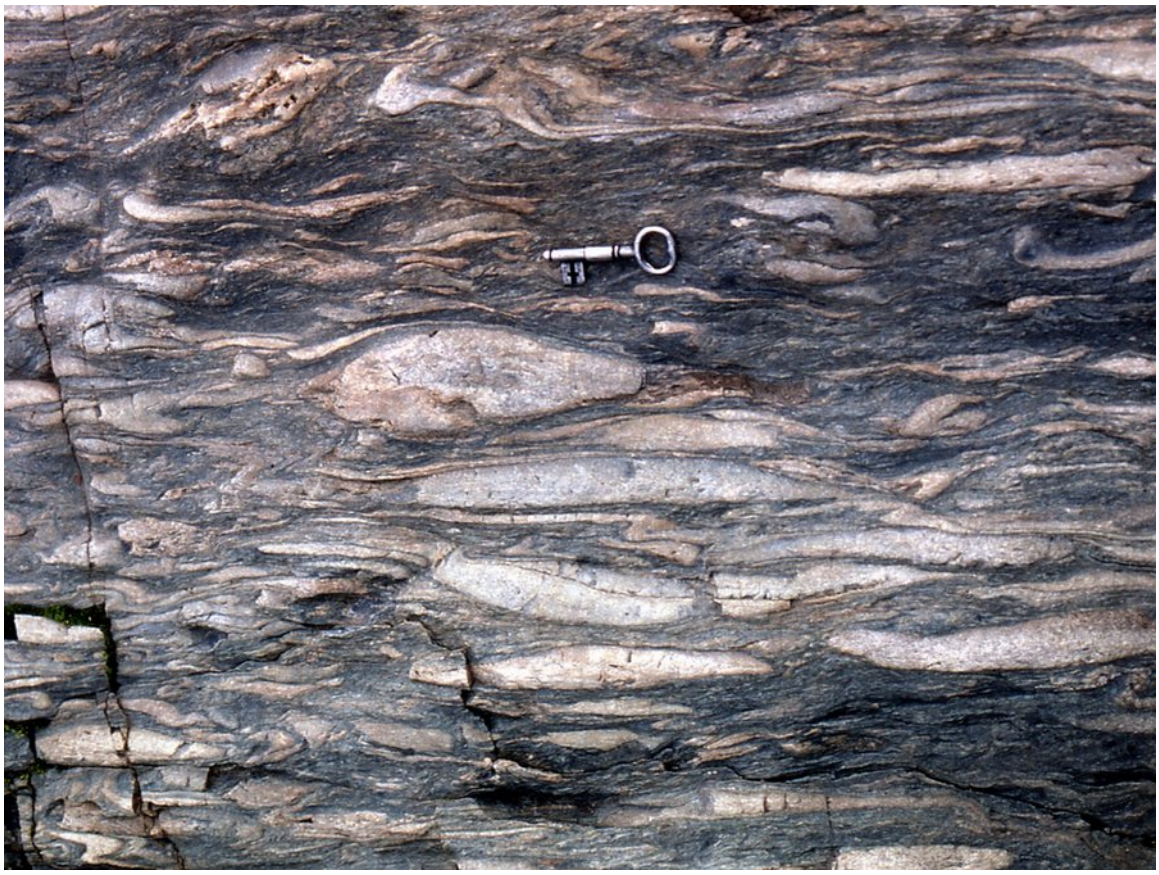


Figure 80. Deformed pebble conglomerate, Bygdin area, Norway, where the data plotted in Figures 50 and 51 was collected by Hossack (1968). Photograph by F. W. Vollmer.

10. Data Transformation

11. Data Synthesis

12. Image Processing

While the manual outlining of particles is generally necessary, image processing can assist in detecting object outlines. EllipseFit includes an *Image Filters* dialog that implements some common image filters, and an *Edge Detection* dialog that implements edge detection algorithms (e.g., Efford, 2000; Parker, 2011).

12.1 Filtering

Contrast – Linearly rescales the image pixels using *contrast* (gain) and *lightness* (bias) values. The range can also be thresholded using the black and white threshold sliders.

Normalize – Rescales the pixels to use the whole available range of values, making dark colors darkest possible and light colors as light as possible.

Threshold – Sets pixels with intensities below the black threshold to black, and pixels with intensities above the white threshold to white.

Remap – Does a linear rescaling by setting the gain and bias to rescale the image pixels to the black and white thresholding values.

Lightness – Increases or decreases the perceived lightness of the image.

Intensity – Increases or decreases the values of the pixel intensities.

Grayscale – Converts color values to grayscale.

Negative – Inverts the pixel values.

Sharpen – Adds contrast between pixels to enhance edges.

Blur – Applies a radial blur applies a blur, so each pixel is merged with pixels within the specified radius.

Median - Computes the median of colors around each pixel.

Convert BW – Converts pixels with low intensities to black, and pixels with high intensities to white.

Invert BW – Converts pixels with low intensities to white, and pixels with high intensities to black.

12.2 Edge Detection

Laplacian - Convolves the pixels by a symmetric kernel to approximate the second-order derivative of the intensity.

Contour - Computes a contour, as if the image was drawn with a 2 pixels-wide black pencil.

Prewitt – Convolves the pixels by a pair of asymmetric kernels to enhance the gradient in the X and Y

directions.

Sobel – Convolves the pixels by a pair of asymmetric kernels to enhance the gradient in the X and Y directions. This is similar to Prewitt, but gives the first-order derivatives of the intensity.

Kirsch – Convolves the pixels using eight asymmetric kernels to enhance the gradients.

DoG – Performs weighted difference of Gaussian edge detection.

Acknowledgements

I thank Y. Shan, K. Burmeister, S. Treagus, G. Mitra, S. Wojtal, H. Fossen, P. Karabinos, M. Mookerjee, J. Davis, W. Dunn, E. Erslev, Y. Kuiper, R. Bauer, D. Wise, D. Czeck, N. Mancktelow, J.M Crespi, B.M. Klemm, J. Robinson, S. Dirringer, and J. Waldron for suggestions, comments, discussions, and encouragement. Y. Shan kindly provided Fortran code for his MLLF calculation. I especially thank R. Twiss, W. Means, and P. Hudleston, mentors whose clear thinking and quantitative approaches inspired me as a student.

Thanks to Jinxi Li (University of Technology Sichuan, China), Julian Mecklenburgh (University of Manchester, UK), Scott Wilkerson (DePauw University, IN), Nate Eichelberger (Princeton University), Raphaël Gottardi (University of Louisiana), Laurence N. Warr (Ernst Moritz Arndt University of Greifswald), Chirantan Parui (IISER, West Bengal), and Christopher Dale Loyacano (University of Memphis) for suggestions and bug reports.

References

Brandon, M.T, 1995. Analysis of geological strain data in strain-magnitude space. Journal of Structural Geology, 17, 1375-1385.

Cardiel, N., 2009. Data boundary fitting using a generalized least-squares method. Monthly Notices of the Royal Astronomical Society 396, p. 680-695.

Cloos, E., 1947. Oolite deformation in the South Mountain Fold, Maryland. Geological Society of America Bulletin, 58, 843-918.

Cloos, E., 1971. Microtectonics Along the Western Edge of the Blue Ridge, Maryland and Virginia. The Johns Hopkins Press, Baltimore and London, 234 p.

Crespi, J.M., 1986. Some guidelines for the practical application of Fry's method of strain analysis. Journal of Structural Geology 16, p. 1327-1330.

Davis, J.C., 1986. Statistics and Data Analysis in Geology. Wiley, 646 p.

Dirringer, S., and Vollmer, F.W., 2013. A test of the analytical Wellman and mean polygon moment ellipse methods of strain analysis using a sample of deformed Ordovician graptoliferous slate from the Taconic orogen, New York. Geological Society of America Abstracts with Programs, 247-52.

Dunne, W.M., Onasch, C.M., Williams, R.T., 1990. The problem of strain-marker centers and the Fry method. Journal of Structural Geology 12, p. 933-1990.

Dunnet, D., 1969. A technique of finite strain analysis using elliptical particle. Tectonophysics 7, 117-136.

Dunnet, D., and Siddans, A.W.B., 1971. Non-random sedimentary fabrics and their modification by strain. Tectonophysics, 12, 307-325.

Efford, N., 2000. Digital Image Processing - A Practical Introduction Using Java. Pearson, 340 p.

Efron, B., 1979. Bootstrap methods: Another look at the jackknife. Annals of Statistics 7, 1-26.

Elliott, D., 1970. Determination of finite strain and initial shape from deformed elliptical objects. Geological Society of America Bulletin 81, 2221-2236.

Erslev, E.A., 1988. Normalized center-to-center strain analysis of packed aggregates. Journal of Structural Geology 10, 201-209.

Erslev, E.A., Ge, H., 1990. Least squares center-to-center and mean object ellipse fabric analysis. Journal of Structural Geology 8, 1047-1059.

Fisher, N.I., Lewis, T., and Embleton, B.J.J., 1987. Statistical Analysis of Spherical Data. Cambridge University Press, 329 p.

Flinn, D., 1962. On folding during three-dimensional progressive deformation. Quarterly Journal of the Geological Society of London, 118, p. 385-433.

Flinn, D., 1978. Construction and computation of three-dimensional deformations. Journal of the Geological Society of London, 135, p. 291-305.
Fossen, H. 1988. The Ulriken Gneiss Complex and the Rundemanen Formation: a basement-cover relationship in the Bergen Arcs, West Norway. Norges Geologiske Undersøkelse Bulletin 412, 67-86.
Fossen, H, 2010. Structural geology. Cambridge University Press, 463 p.
Fry, N., 1979. Random point distributions and strain measurement in rocks. Tectonophysics 60, 806-807.
Hanna, S.S., Fry, N., 1979. A comparison of methods of strain determination in rocks from southwest Dyfed (Pembrokeshire) and adjacent areas. Journal of Structural Geology 2, 155-162.
Hobbs, B.E., Means, W.D., and Williams, P.F., 1976. An outline of structural geology. Wiley, New York, 571 pp.
Holst, T.B., and Fossen, H. 1987. Strain distribution in a fold in the West Norwegian Caledonides. Journal of Structural Geology 9, 915-924.
Hossack, J.R., 1968. Pebble deformation and thrusting in the Bygdin area (Southern Norway). Tectonophysics 5, 315-339.
Jensen, 1981. On the hyperboloid distribution. Scandinavian Journal of Statistics 8, 193-206.
Launeau, L., and Pierre-Yves F. Robin. P.F., 2005. Determination of fabric and strain ellipsoids from measured sectional ellipses—implementation and applications. Journal of Structural Geology 27, 2223-2233.
Lisle, R.J., 1985. Geological Strain Analysis, A Manual for the Rf/ϕ Technique. Pergamon Press, Oxford.
Lisle, R.J., 2010. Strain analysis from point fabric patterns: an objective variant of the Fry method. Journal of Structural Geology 32, 975-981.
Mardia, K.V., 1972. Statistics of Directional Data. Academic Press, 329 p.
Marshak, S., and Mitra, G., 1988. Basic methods of structural geology. Prentice Hall, 446 p.
McNaught, M.A., 1994. Modifying the normalized Fry method for aggregates of non-elliptical grains. Journal of Structural Geology 16, p. 493-503.
McNaught, M.A., 2002. Estimating uncertainty in normalized Fry plots using a bootstrap approach. Journal of Structural Geology, 24, p. 311-322.
Mitra, S., 1978. Microscopic deformation mechanisms and flow laws in quartzites within the South Mountain anticline. Journal of Geology 86, p. 129-152.
Mookerjee, M., and Nickleach, S., 2011. Three-dimensional strain analysis using Mathematica. Journal of Structural Geology 33, p. 1467-1476.
Mulchrone, K.F., 2003. Application of Delaunay triangulation to the nearest neighbour method of strain analysis. Journal of Structural Geology 25 (5), 689-702.
Mulchrone, K.F., O'Sullivan, F., Meere, P.A., 2003. Finite strain estimation using the mean radial length of elliptical objects with bootstrap confidence intervals. Journal of Structural Geology 25, 529-539.
Mulchrone, K.F. 2005. An analytical error for the mean radial length method strain analysis. Journal of Structural Geology 27, 1658-1665.
Mulchrone, K.F., 2013. Fitting the void: Data boundaries, point distributions and strain analysis. Journal of Structural Geology 46, 22-33.
Mulchrone, K.F., McCarthy, D.J., and Patrick A. Meere, P.A., 2013. Mathematica code for image analysis, semi-automatic parameter extraction and strain analysis. Computers & Geosciences 61, p. 64-73.
Nadai, A., 1950. Theory of Flow and Fracture of Solids. McGraw-Hill, New York, 572 p.
Onasch, C.M., 1986. Ability of the Fry method to characterize pressure- solution deformation. Tectonophysics 122, 187-193.
Owens, W.H., 1984. The calculation of a best-fit ellipsoid from elliptical sections on arbitrarily orientated planes. Journal of Structural Geology 6, 571-578.
Parker, J., 2011. Algorithms for Image Processing and Computer Vision 2nd ed. Wiley, 480 p.
Pollard, D.D. and Fletcher, R.C., 2005. Fundamentals of structural geology. Cambridge University Press, Cambridge, 463 500 p.
Preparata, F.P., and Shamos, M.I., 1985. Computational Geometry. Springer-Verlag, New York, 400 p.
Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., 2007. Numerical Recipes: The Art of

Scientific Computing, 3rd Edition. Cambridge University Press, 1235 p.
Ragan, D.M., 1985. Structural Geology, An Introduction to Geometrical Techniques, 3rd Ed. John Wiley and Sons, Inc. 393 p.
Ragan, D.M., and Groshong, R.H., 1993. Strain from two angulars of shear. Journal of Structural Geology, v. 15, p. 1359-1360.
Ragan, D.M., 2009. Structural Geology, An Introduction to Geometrical Techniques, 4th Ed. John Wiley and Sons, Inc. 393 p.
Ramsay, J.G. and Huber, M. I., 1983. The Techniques of Modern Structural Geology: Volume 1: Strain. Analysis, Academic Press, London, 307 p.
Ramsay, J.G., 1967. Folding and Fracturing of Rocks. McGraw-Hill, 568 p.
Reddy Vinta, B.S.S., Srivastava, D.C., 2012. Rapid extraction of central vacancy by image-analysis image-analysis of Fry plots. Journal of Structural Geology 40, p. 44-54.
Robin, P.F., 2002. Determination of fabric and strain ellipsoids from measured sectional ellipses – theory. Journal of Structural Geology 24, 531-544.
Rogers, D.F. And Adams, J.A., 1976. Mathematical Elements for Computer Graphics. McGraw-Hill, New York, 239 p.
Shan, Y., 2008. An analytical approach for determining strain ellipsoids from measurements on planar surfaces. Journal of Structural Geology 30, 539-546.
Shan, Y., and Xiao, W., 2011. A statistical examination of the Fry method of strain analysis. Journal of Structural Geology, v. 33, p. 1000-1009.
Shimamoto, T., Ikeda, Y., 1976. A simple algebraic method for strain estimation from ellipsoidal objects. Tectonophysics 36, 315-337.
Snyder, J. P., 1987. Map projections- a working manual. U.S. Geological Survey Professional Paper, 1395, 383.
Steger, C., 1996. On the Calculation of Arbitrary Moments of Polygons, Technical Report FGBV-96-05, Forschungsgruppe Bildverstehen (FG BV), Informatik IX Technische Universitat Munchen, Germany, 18 p.
Twiss, R.J. and Moores, E., 2007. Structural geology, 2nd edition. W.H. Freeman, New York, 736 pp.
Van der Pluijm, B.A. and Marshak, S., 2004. Earth structure, 2nd edition. W.W. Norton, New York, 656 p.
Vollmer, F.W., 1998. EllipseFit 1 Strain Analysis Porgram. www.newpaltz.edu/~vollmerf.
Vollmer, F.W., 1995. C program for automatic for automatic contouring of spherical orientation data using a modified Kamb method: Computers & Geosciences 21, 31-49.
Vollmer, F.W., 2010. A comparison of ellipse-fitting techniques for two and three-dimensional strain analysis, and their implementation in an integrated computer program designed for field-based studies. Abstract T21B-2166, Fall Meeting, American Geophysical Union, San Francisco, California.
Vollmer, F.W., 2011a. Automatic contouring of two-dimensional finite strain data on the unit hyperboloid and the use of hyperboloidal stereographic, equal-area and other projections for strain analysis. Geological Society of America Abstracts with Programs, v. 43, n. 5, p. 605.
Vollmer, F.W., 2011b. Best-fit strain from multiple angles of shear and implementation in a computer program for geological strain analysis. Geological Society of America Abstracts with Programs, v. 43.
Waldron, J.W.F., 1988. Determination of finite strain in bedding surfaces using sedimentary structures and trace fossils: a comparison of techniques. Journal of Structural Geology 10 (3) p. 273-281.
Waldron, J.W.F., and Wallace, K.D., 2011. Objective fitting of ellipses in the centre-to-centre (Fry) method of strain analysis. Journal of Structural Geology 29, p.1430-1444.
Wellman, H.G., 1962, A graphic method for analyzing fossil distortion caused by tectonic deformation. Geological Magazine, 99, 384-352.
Wheeler, J., 1984. A new plot to display the strain of elliptical markers: Journal of Structural Geology 6, 417-423.
Hossack, J.R., 1968. Pebble deformation and thrusting in the Bygdin area (Southern Norway): Tectonophysics 5, 315-339.
Yamaji, A., 2008. Theories of strain analysis from shape fabrics: A perspective using hyperbolic geometry.

Journal of Structural Geology 30, 1451-1465.

History

3.3.0.62 14 Mar 2017
- Added *Apply/Undo* to *Edge Detection* dialog.
- Added *Apply/Undo* to *Image Filters* dialog.
- Added *Median*, *Convert BW,* and *Invert BW* image filters.
- Changed line pair fields to *X11, Y11, X12, Y12, X21, Y21, X22, Y22*, for line and point.
- Added *Alpha1* and *Alpha2* fields for directed angle of lines, and *Beta* for the angle between the lines.
- Fixed display of filled particle when deleted from spreadsheet.
- Added optional display of ellipse axes.
- Added *Area* field to ellipse spreadsheet.
- Fixed bug in Image Filters histogram.
- Added Windows Taskbar icons.
- Changed messaging procedures from PostMessage to Dispatch to fix display bugs (the data window was not reliably updating display of the data after opening a data file).
- Added *Fit Void* command and dialog box for void fitting routines.
- Added *exponential edge detection*, *density gradient*, *weighted least-squares* void fitting methods with variants.
- Added bootstrapping to exponential edge detection, and enhanced normalized fitting methods.
- Fixed control heights in dialog boxes.
- Added *Edit metadata* command to edit file metadata, initial lines in file beginning with "//".
- *Image Transform* command dialog rewritten with fixes.
- *Data Transform* command dialog rewritten with fixes.
- *Image Filters* command dialog rewritten with fixes.
- Reorganized menu structure.
- Implemented transform data undo stack using data copies instead of inverse transforms.
- Now reopens last open files if possible.

3.2.2 26 Oct 2015
- Fixed listbox display in Preferences dialog.
- Fixed bug with non-roman unicode characters in drive name. Workaround is to run from thumb drive.
- Maintenance upgrade compiled with Lazarus 1.4.4.
- Improved redraw speed of images when digitizing. Further optimization may be possible in subsequent versions. Thanks to Gabriele Casale, Appalachian State University, for reporting the issue.
- Fixed display of symbols on Ratio Plot. Changed labels from A, B, C to Max, Int, and Min.

3.2.1 5 Aug 2015
- Cleaned up main preferences dialog control placements, and tooltips.
- Fixed bug causing jump of preferences dialog on preview.
- Optimized data grid display scrolling.
- Changed "N" column header to "ID". ID is a unique integer assigned to identify a single particle. Files will read in correctly with either "N" or "ID", but will be written with "ID".
- Modified desktop icon.
- Replaced term "graph" with "plot" in menu and manual.
- Work on user manual.
- Implemented opening of multiple data files with associated data plots. *File > New* and *File Close* commands added.
- Default view is now the *Data Window* instead of *Image Window*.
- *Open Data* and *Open Image* shortcuts swapped.

- The last *Open Data* and *Open Image* paths ares now saved.
- Fixes to Nadai and Flinn plots.
- Optimized messaging.
- Added warning dialog to *Reset Preferences*.

3.2.0 28 Jan 2015
- Prevented redrawing of data on image when adding or undoing digitized points to speed up redraw with numerous data points or slow processors.
- Replaced StringGrid with DrawGrid and with numerous related internal modifications in viewing and updating the data grid.
- Enabled status bar in Data Window.
- Changed SendMessages to PostMessages.
- Fixed enabling of Ratio Graph.
- Added multiple selections in Data Window. Use Command/Control click for adding or removing items, and Shift click to extend selection.
- Added multiple selections in Image Window. Use Command/Control click for adding or removing items.
- Added multiple selections to Rato, Flinn, Nadia, Polar, Rf-Phi, Wellman and Stretch Graphs. Use Command/Control click for adding or removing items.
- Added multiple selections on Strain Map. Use Command/Control click for adding or removing items.
- Fixed Rf-Phi Save As and Export commands.
- Added Select All, Select None, Select Inverse commands.
- Known bug: Audio alerts do not work in Linux.
- Known bug: Menu commands do not initially update in the Data Window. Work around is to click on Image Window and back to the Data Window.
- Trying to use File > Open Image (instead of File > Open Data) to open a data file now gives a warning dialog with the option to open it as a data file.
- Numerous changes to Analyse > Synthesize Data command. Particle ratios are randomly selected from a range RMin...RMax on Ln(R), or from a Gaussian distribution on Ln(R) with a mean of Ln(RMean) and standard deviation of Sigma. Area can also be selected from a Gaussian distribution with a mean area of pi. Orientations are selected randomly from either a range in phi or from a Von Mises distribution.
- Fixed settings dependancies in Fry Panel of Preferences Dialog.
- Added Delaunay triangulation and Voronoi graphs to Strain Map options.
- Added Delaunay nearest neighbor option to Fry Graph.

3.1.1 06 Nov 2014
- Added the ability to open Microsoft Excel XLS (legacy) and XLSX formats, in addition to OpenDocument ODS spreadsheet, and delimited file (CSV, TSV) formats. In each case, a comment line starts with '//', and a header row identifying the data columns must precede the data rows.
- Fixed bug requiring “Max”, “Min” data and header as well as “R” for ellipsoid calculation. Also now allows “Pitch” header in place of “Phi”. Thanks to Kurt Burmeister for reporting this.
- Replaced timers with event messaging.
- Fixes to Analyze > Data Synthesis command, which failed in Windows. The collision tests counts have been increased to 10,000 x 10,000, which tightens adjacent particle contacts.

3.1.0 04 Jun 2014
- Added bootstrap error analysis to ellipsoid calculations. This has some similarities to the kernel density estimation approach of Mookerjee and Nickleach (2011).
- Added saving of the ellipsoid axes orientations for plotting on spherical projections in Orient.
- Changed column headers A, B, C to Max, Int, Min to clarify the axial lengths. EllipseFit will open files with

the old headers, but will save them using the new headers.
- Removed option to save files as “Space Delimited”. This format potentially causes issues parsing files with spaces in the header column. EllipseFit will still open space delimited files with recognizable headers.
- Added 95% confidence regions to Nadai graph.
- Added 95% confidence regions to Flinn graph.
- Added option to save bootstrap ellipsoid axes.
- Added numerous options to Synthesize Data command. These include generating the strain ratio from a Gaussian normal distribution, generating particle size from a Gaussian normal distribution, generating a preferred orientation from a Von Mises circular distribution, generating centers at a truncated Poisson distribution. The latter is performed by randomizing the location in x, y and discarding collisions.
- Added an option to the Strain Map command to either plot scaled strain ellipses or particle axes.
- Implemented the maximum mean log likelihood function (MLLF) search procedure of Shan and Xiao (2011). This gives a high accuracy strain estimate from Fry-type data, that is, data from truncated Poisson distributions. It does not require ellipse data, and it is not subjective and is reproducible.
- Fixed auto-scaling on Fry graphs.
- Significant progress on the User Manual.

3.0.3 13 May 2014
- Added transforms to image to rotate, flip, strain, unstrain, etc. To strain or unstrain both image and data, transform the image first. This calculates the origin offset in the new bitmap. Then transform the data at (X0, Y0) = (0.0, 0.0) with “Rectify” checked.
- Added transform data to Wellman-type data.
- Changed default bootstrap resamples from 300 to 5000.
- Rewrote ellipse standard error and confidence interval methods. Changed from using resample trials to calculate standard error and Student T for confidence interval, to use resampled data for both. Non-bootstrap MRL uses analytical error and Student T following Mulchrone (2005).
- Added option to save bootstrap resample ellipses.
- Added option to plot 95% confidence regions on Polar and Rf/Phi graphs using analytical error.
- Fixed bug that was swapping A and B radii while digitizing polygons.

3.0.2 21 Apr 2014
- Fixed bug in fill ellipse routine causing hangs at high thresholds.
- Fixed bug causing crash when opening page size dialog.
- Added strain map.
- Added synthesize data to create data sets.
- Added transform data to strain, unstrain, shear, etc., data.
- Changed names of digitize routines to reflect the objects, e.g., center points, ellipses, polygons, instead of the results (e.g., polygon moment ellipse).
- Changed names of graphs to more common specific names attributing authors, Fry, Flinn, etc., instead of generic names.
- Internal change in form communication, from flags and timers to messages.
- Numerous additional fixes and changes.

3.0.1 06 Apr 2014
- Fixed bug effecting symbol colors in svg graphics.
- Cleaned up the polar graph.
- Fixed cursor status strings on graphs.
- Fixed up contouring preferences.
- Added axial ratio Flinn type graph.

- Added octahedral Nadai-Hsu type strain graph.
- Added ellipse digitizing with polygon fill and moments.
- Fixed file save warning.
- Numerous internal changes.

3.0.0 24 Mar 2014
- First public release of Version 3.

3.0.0.28 01 Aug 2012
- Initial prerelease of Version 3.

This page intentionally left blank.