A new computational approach to cracks quantification from 2D image analysis: Application to micro-cracks description in rocks

Alessio Arena 1, Claudio Delle Piane*, Joel Sarout

CSIRO Earth Science and Resource Engineering, 26, Dick Perry Avenue, Kensington 6151, Western Australia, Australia

A R T I C L E   I N F O

Article history:
Received 12 April 2013
Received in revised form
13 January 2014
Accepted 18 January 2014
Available online 28 January 2014

Keywords:
Cracks
Separation
Segmentation
Microcracks
Image analysis
Quantification

A B S T R A C T

In this paper we propose a crack quantification method based on 2D image analysis. This technique is applied to a gray level Scanning Electron Microscope (SEM) images, segmented and converted in Black and White (B/W) images using the Trainable Segmentation plugin of Fiji. Resulting images are processed using a novel Matlab script composed of three different algorithms: the separation algorithm, the filtering and quantification algorithm and the orientation one. Initially the input image is enhanced via 5 morphological processes. The resulting lattice is “cut” into single cracks using 1 pixel-wide bisector lines originated from every node. Cracks are labeled using the connected-component method, then the script computes geometrical parameters, such as width, length, area, aspect ratio and orientation. A filtering is performed using a user-defined value of aspect ratio, followed by a statistical analysis of remaining cracks. In the last part of this paper we discuss about the efficiency of this script, introducing an example of analysis of two datasets with different dimension and resolution; these analyses are performed using a notebook and a high-end professional desktop solution, in order to simulate different working environments.

Crown Copyright © 2014 Published by Elsevier Ltd. All rights reserved.

1. Introduction

2D image analysis in geosciences is mainly focused on the scrutiny of two classes of objects of distinct geometry:

1. Grains, minerals, pores, and all the objects that have a shape traceable to an ellipse. These objects are analyzed for applications including estimates of porosity and permeability (Lock et al., 2002), grain size distribution (Berger et al., 2011), plastic strain and mineral reactions estimations (Delle Piane et al., 2009);

2. Objects described by a linear shape such as cracks, hydrographical elements, roads and others. The analysis of these objects is complicated by their spatial interactions and a separation processing is necessary if the characterization of every single element of the population is required.

In this article we propose a new separation method for 2D linear objects, applied to the study of microcracks in crystalline rocks. As defined by Simmons and Richter (1976), a microcrack is:

“an opening that occurs in rocks and has one or two dimensions smaller than the third. For flat microcracks, one dimension is much less than the other two and the width to length ratio, termed crack aspect ratio, must be less than \(10^{-2}\) and is typically \(10^{-3}\)– \(10^{-5}\). The length typically is of the order of 100 \(\mu m\) or less.”

Microcracks strongly affect rocks’ physical properties, such as permeability (e.g. Fortin et al., 2011; Sarout, 2012), and static/dynamic elastic moduli (e.g. Delle Piane et al., 2011; Heap and Faulkner, 2008; Sarout and Guéguen, 2008); yet there is no uniquely accepted method to precisely quantify the density and geometrical characteristic of such microstructural features.

Automatic crack detection methods have been proposed for structural investigations of building and engineering materials: Dare et al. (2002) introduced a user-guided feature extraction application for the measurement of crack width based on digital images of single cracks in of concrete acquired with a digital camera. Chen and Hutchinson (2010) and Lee et al. (2013) also proposed a framework for the determination of the geometric quantification of surface cracks in concrete using digital images visualizing the evolution of one crack. More recently, Zhu et al. (2011) and Jahanshahi and Masri (2013) presented a method of retrieving the properties of the cracks (orientation and width) from optical images of concrete based on crack detection and crack property retrieval, using image thinning and distance transform techniques. It should be noted however that all of the above
methodologies were specifically developed for field applications focused on the detection of macroscopic surface cracks in concrete visualized using a digital camera; as such they are better suited for the recognition and characterization of a limited number of non-interacting cracks per field of view.

Microcracks in crystalline rocks are better visualized by means of optical or electron microscopy and generally constitute a network of interacting segments within a visualized field of view requiring a careful separation procedure for the statistical analysis of a large population of objects.

Most of the approaches used for this analysis tried to bypass the problem of cracks separation; some examples are:

- **Mamtani et al. (2012)** based on the analysis of the fractal geometry of the cracks pattern.
- **Chen et al. (2001)** computed the average length of cracks by estimating the number of elements and the total length of the lattice.
- **DeVasto et al. (2012)** used fitting rectangles based on the Multiple Ferret Method (Wang, 2006).

The work of **Ito et al. (2002)** was probably the first attempt to separate single cracks based on the identification of the nodes of the skeletonized pattern. These nodes were subsequently used for tracing and labeling the cracks. However, the skeletonization process used to perform this operation alters the geometry of the cracks.

Another approach suggested by **Le Roux et al. (2013)** consists in identifying and dilating the nodes of the crack pattern to separate out the cracks. The nodes are subsequently subtracted from the original black and white (B/W) image. The remaining of the image is then labeled using the "connected-component method" (Han and Wagner, 1990; Samet and Tamminen, 1988).

We propose a new quantification method for the identification of cracks and microcracks, based on an improved version of the separation algorithm originally suggested by **Le Roux et al. (2013)**.

### 2. Methodology

The new method is summarized in Fig. 1 and has been implemented in the **Matlab** environment. First, an 8-bit image of a polished rock surface, collected with a Scanning Electron Microscope (SEM), is converted into a binary image using the Trainable Segmentation method (Kaynig et al., 2010) available in **ImageJ** (plug-in developed by Schindelin et al., 2012). The binarized image is then loaded into **Matlab** and subjected to a morphological pre-processing. The resulting image is then segmented using bisecting lines originating from branch points, the objects are extracted and then filtered according to user-defined parameters; this step is reiterated by changing the region considered during the generation of bisectors in order to achieve a balance between interference among objects and angular resolution. Next, the distributions of orientation, length, width, and aspect ratio of the cracks are computed. The user can supervise the results, delete specific objects not relevant for the intended application (e.g., twin lamellae, dust particles, surface scratches) before the final results are displayed. Each processing step is detailed in a separate section below. Additional flowcharts detailing the sequence of single operations for each step are also reported in the Appendices.

#### 2.1. Binarization

First, an 8-bit grayscale SEM image is binarized to yield a black and white image where the solid phase is black and the "empty" phase is white (pores, microcracks etc.). The simplest way to perform this operation is by applying a threshold to the gray level histogram. This can be done manually or by applying automatic thresholding algorithms such as the Otsu method (Otsu, 1979; Stanekova and Lapin, 2012). The so-called Trainable Segmentation method, available with the software Fiji (ImageJ based) (Schindelin et al., 2012), is used in our approach: the user "trains" the software to differentiate and classify different portions of the image by drawing lines, thus collecting information on pixels' gray level and their spatial variation. The advantage of this method is that the operator helps to finely discriminate two or more sets of pixels with different gray levels and/or different gradients by simply drawing a reasonable number of lines used to collect the necessary information. This is particularly critical in conditions of non-uniform illumination of an image where purely automatic algorithms might fail or induce biases.

**Fig. 1.** Flowchart of the crack detection and analysis methodology from 2D images.
cardinal positions (N, S, E, W, 4-connectivity), or all the pixels surrounding the center one (N, NW, W, SW, S, SE, E or NE, 8-connectivity). The Diagonal fill operation changes the value of the center pixel to 1 if two or more 4-connectivity sites have a value of 1 as in the kernel example below; the result of the operation on the image is illustrated in Fig. 2b.

\[
\begin{array}{cccc}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
\end{array}
\]

- **Morphological opening**: consists of a morphological erosion followed by a morphological dilation aimed at making the cracks more compact and easier to analyze (Pratt, 2001) (Fig. 2c).
- **Thinning processing**: used to “skeletonize” the objects under consideration. The process is iterated until there is no more modification. This technique preserves the topological properties of the objects (number of holes and connections) mathematically represented by the Euler number (Lam et al., 1992; Pratt, 2001) (Fig. 2d). This approach is preferred to the traditional “skeletonization” process (Pratt, 2001) because it gives better results when combined with the subsequent branch-point processing.
- **Cleaning processing**: used to delete isolated pixels representing noise and very small and round objects (Pratt, 2001) (Fig. 2e).
- **Branch points identification**: used to detect intersection points of branches of the skeleton; these points represent crack tips and will be used to separate individual crack segments (Fig. 2f).

### 2.3. Separation

Once the branch points have been identified they can be used to isolate single cracks. A search window, with user-defined dimensions, is created and centered at each branch point (Fig. 3a). A temporary copy of the portion of the skeleton enclosed in the search window is extracted and used to separate cracks as follows:

- only the object passing through the central branch point is considered (Fig. 3b);
- the center part of the temporary copy is erased, keeping only white pixels adjacent to the window borders (Fig. 3c);
- the coordinates of these pixels and those of the branch point are collected by the program and used to build a simplified crack pattern approximating the original skeleton by straight lines;
- based on this simplified geometry, the algorithm detects and ignores “false” node (e.g. aligned branches);
- every pair of lines forms an angle with the vertex located at the branch point. A bisector line is created for each of these angles (Fig. 3d); and
these bisector lines are subtracted from the original image which leads to a separation of the cracks (Fig. 3e and f).

The relevance of the crack separation process strongly relies on the size of the scanning window relative to the image resolution and to the microstructural features it contains. Larger windows allow for a better angular resolution for constructing the bisector lines; on the other hand, excessively large windows may include more than a single branch point and therefore induce inconsistencies in the results. Therefore, the adequate size of the searching window is potentially different for every investigated image. In order to ease the control of the user on this aspect, the whole process (including the subsequent filtering) is semi-automatically repeated for different searching window sizes; in addition the user have the possibility to quickly evaluate the average distance between cracks in pixels using a dedicated script, useful to choose.

Fig. 3. Segmentation method. (a) Original image with searching window; (b) portion of the skeleton copied using the searching window; (c) temporary copy with the center portion erased; (d) bisector lines; (e) subtraction of bisector lines from the original image; and (f) resulting image.
the correct range of searching window sizes. For a given image, the user can then select among the different outputs the most appropriate one. Once cracks are separated, the “connected object” method (Han and Wagner, 1990; Samet and Tamminen, 1988) is used to label the cracks and work with one crack at a time.

In comparison to the algorithm used by Le Roux et al. (2013) (Fig. 4), our method ensures less geometric alteration; in fact it should be noted that this algorithm uses 1 pixel-wide lines, instead of a square object that has to be at least as wide as the crack. Moreover our method is able to identify and ignore the “false nodes”, resulting in a more realistic separation (Fig. 4d and f, note the long crack close to the left bottom corner). A brief numerical comparison between these two separation algorithm is reported in the table below: initially the proposed separation algorithm finds more possible cracks, but after applying a filtering on both results, based on a minimum aspect ratio (0.5, and 0.05 for objects touching borders) and area, the output of our algorithm consist of significantly less cracks covering a bigger area.

<table>
<thead>
<tr>
<th></th>
<th>Elements before filtering</th>
<th>Time (s)</th>
<th>Elements after filtering</th>
<th>Percentage of cracks’ area</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeRoux</td>
<td>3330</td>
<td>435</td>
<td>240</td>
<td>2.12%</td>
</tr>
<tr>
<td>Proposed method</td>
<td>4380</td>
<td>890</td>
<td>194</td>
<td>2.72%</td>
</tr>
</tbody>
</table>

The flowchart of the segmentation algorithm is reported in Appendix B.

2.4. Geometrical analysis and filtering

Once the objects have been separated and labeled, their geometrical attributes are computed:

- Area: the total number of pixels.
- Length: the sum of pixels of the skeletonized object, multiplied by a factor derived from the orientation (ranging from 1 to $\sqrt{2}$); this factor is used to convert our values from pixels to “linear units”.
- Average width: the area divided by the length.
- Aspect ratio: the width divided by the length.

It should be noted that our approach to calculate the length and therefore the aspect ratio of the objects is different from other common methods (DeVasto et al., 2012; Le Roux et al., 2013; Stanekova and Lapin, 2012) that attempt to fit an ellipse to a given object. The latter methods are particularly unsuitable for non-straight flat/elongated objects such as cracks. If a crack is more or less straight, both methods would yield similar results. However, if a crack is curved (e.g. follows a grain boundary), the ellipse-fitting approach would yield unrealistic results in terms of geometrical attributes such as width and aspect ratio.

Based on the computed geometrical attributes (aspect ratio and area), objects are then filtered to remove spurious measurements such as pixel noise, impurities, surface dust and all objects that do not fit the criteria of the specific analysis being conducted (e.g. if one is looking for cracks, near-circular objects with aspect ratio close to 1 would have to be ignored).

Fig. 4. Comparison between the separation method from Le Roux et al., 2013 and the proposed one. (a, b) SEM image and zoom, (c, d) results from our algorithm and zoom; (e, f) results using Le Roux et al. and its zoom.
Moreover, objects touching the borders of the image that have an aspect ratio bigger than a user-defined value are also ignored. The purpose of this filtering step is to avoid the inclusion of partially imaged objects, the attributes of which are inherently “wrong”, which would skew the results of the statistical analysis.

The flowchart of this module is reported in Appendix C.

2.5. Crack orientation

After the filtering phase, orientation of every crack is computed. Cracks are not straight in 2D sections and usually show up as winding lines; for this reason, a geometrical approximation is required when calculating their overall orientation. This is done by subdividing each object in a number of straight segments defined by the intersection of the crack with a grid composed of equidistant parallel lines (Fig. 5). The quality of this approximation depends on the user-defined spacing of the grid, and consequently the number of intersection points collected.

The proposed method starts collecting the coordinates of the tips of the objects, identified using the “endpoints function” in Matlab (Fig. 5a). Based on this information, the program calculates the major Cartesian component of the object (Fig. 5b). A set of parallel and equidistant lines, having a user-defined spacing, are constructed perpendicular to this major component (Fig. 5c), and the coordinates of the intersection points between this grid and the object are collected (Fig. 5d). These points are then used to compute the orientation of every segment, and the final orientation of the crack is the average of these values. This orientation in 2D ranges from $+90^\circ$ to $-90^\circ$, where $0^\circ$ is a line parallel to the x-axis, and the angles are counted positive in a counterclockwise rotation.

The flowchart of this analysis is reported in Appendix D.

2.6. Statistical analysis

The output of this software consists of three histograms and a rose diagram. Each histogram represents the probability distribution of length, width and aspect ratio, and the rose diagram represents the probability distribution of orientation.

For the illustration example of Carrara marble, the crack lengths can be well represented by a unimodal log-normal distribution. Subsequently the software fits a log-normal function to the histogram and extracts the mean and standard deviation that best describe the cracks’ length distribution. Note that this distribution is found to be asymmetric (skewed towards the smallest lengths). The observed asymmetry is probably accentuated by the limited resolution of the acquired image.

For this same data set, the probability distribution of widths displays unimodal and symmetric characteristics, well described by a simple normal distribution function. A fit of the histogram yields the mean and standard deviation that best describe the cracks’ width distribution.

The aspect ratio distribution is somehow more complex; therefore no fitting is attempted. Nevertheless, a mean aspect ratio and an uncertainty are estimated based on the following:

![Fig. 5. Computing orientation. (a) Isolated object with recognized endpoints; if three or more points are recognized, the most distant couple is preserved; (b) Cartesian components of the object; (c) grid used to approximate the crack; and (d) intersection points between object and grid.](image-url)
The mean value of aspect ratio $\overline{\alpha}$ is calculated as

$$\overline{\alpha} = \frac{w}{l}$$

where $w$ is the mean width and $l$ is the mean length as derived from the corresponding distribution functions. The uncertainty associated with the aspect ratio $\Delta \alpha$ is derived by propagating the uncertainties associated with the width and length distribution functions (e.g. Fornasini, 2008):

$$\Delta \alpha = \frac{\Delta w}{w} + \frac{\Delta l}{l}$$

where $\Delta w$ and $\Delta l$ represent uncertainties on width and length estimations, respectively, defined as twice the standard deviation obtained from the corresponding distribution functions (i.e. $\Delta X = 2 \times \text{standard variation}$).

Fig. 6. Visual and numerical comparison between the proposed quantification method and the Fitting Ellipse one. Red bars represent the width computed using our method, while the blue bars represent the width obtained from the fitting ellipse. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).
Lastly, the orientation distribution of cracks is summarized in a standard rose diagram.

2.7. Check of results

After an analysis is performed with this algorithm, the user has the opportunity to perform a check of the results, and manually delete the objects that are not relevant to the analysis. This final step is often necessary when dealing with cracks because the original dataset is usually not perfect and as such contains linear elements that are not cracks (e.g. lamellae, scratches) that can be picked up by the algorithm despite the precautions taken. Furthermore having this possibility to check results make the program more flexible and adaptable to different situations/applications.

To perform this operation the user can simply point and click to select the unwanted elements using the cursor, our software will automatically delete them and recompute the statistical properties of the new image.

Once the image has been cleaned from all unwanted elements, results can be saved as 4 images:

- A black and white image with separated cracks;
- an image representing the filtered elements;
- an image with elements labeled; and
- the statistical diagrams.

In addition, a comma separated values (csv) file containing the statistical data and a Matlab’s variable file “.mat” of the labeled image are saved.

3. Comparison of quantification methodologies

Microcracks quantification through image analysis is a methodology adopted by a large number of scientists and researchers, but only few of them tried to quantify individual cracks’ geometric properties. This operation was previously performed using the fitting ellipse method (also known as Ferret method), that is a well-established methodology mainly adopted for grain and mineral analyses, but not fully compatible with the analysis of a large population of linear objects like microcracks, due to the difference in shape from the ellipse that this method try to fit to the object.

Our approach instead is based on simple geometric calculations, based on the information provided by the skeletonized and the original image. Obtained results were more accurate compared to the fitting ellipse algorithm. However this method was developed specifically to analyze cracks, and is suitable only for objects with a linear shape, like streets or rivers (from remote sensing images).

In Figs. 6 results from both methodologies are numerically and visually compared: length and orientation values are always comparable between these two approaches, but the width obtained from the fitting ellipse algorithm is always overestimated by the fitting ellipse method. Values obtained from the fitting ellipse algorithm are always two to five times larger than our values and they are never compatible with the area of the objects. The reason for this discrepancy has to be searched in the fitting procedure: given the major axis of the ellipse, the algorithm computes the width (minor axis of the ellipse) trying to cover the whole extension of the object, that is heavily affected by the winding of the crack. The obtained value does not represent the width of the object, but is more similar to the minor dimension of the bounding box. Instead our algorithm computes this descriptor considering the object as a rectangle with the long side equal to the length of the skeleton of the object; the average value of the width is then computed dividing the area by its length.

4. Example of analysis of a thermally-cracked Carrara marble

Analyzing a large image file can be very demanding for computers, and some operations can take hours or even days to
be completed. For this reason when a new algorithm is proposed, it is important to provide some examples of application under various conditions, specifying machine characteristics, running time and associated results. In this study, we used two reference computers and two image examples to test the efficiency of the new algorithm/software.

The images were collected from a sample of thermally-cracked Carrara marble. The specimens were ground, polished, and coated with a conductive layer of carbon before surface visualization with SEM has been carried out. Images were collected in back-scattered electron mode; therefore contrast in gray level is attributed to contrast in atomic number in the investigated region.

The first reference computer is a Windows OS based notebook (Windows 8), with a CPU Intel i5 2450M (35W, 2.50 GHz up to 3.10 GHz, 2 physical core but 4 logical processors), 12 Gb of DDR3 RAM (PC3-10700, 666 MHz) and an 1 Tb HDD (Samsung Spinpoint M8 @ 5400 rpm).

The second reference computer is a Windows OS based, top-end professional desktop (Windows 7), equipped with 2 × CPU Intel Xeon 2687 W (150W, 3.10 GHz up to 3.80 GHz, 8 physical core but 16 logical processors), 256 Gb of DDR3 RAM (PC12800, 1600 MHz) and 256 Gb SSD + 2.5 Tb HDD.

An example of the user interface of the script is given in Fig. 7; the central part of the interface displays the binarized input image while the user controlled operations are available on the right hand panel.

4.1. High resolution SEM image

The first example is a large image (4873 × 5420; 26.4 Megapixel) generated through mosaicking of 35 SEM images (Fig. 8), where 1 pixel corresponds to 0.119 μm.

Fig. 8. Example of a high resolution image; (a) gray level image, made through a mosaicking of SEM images; (b) B/W image of cracks, plus filtered elements after processing (in red); and (c) labeled image, every color represent one isolated cracks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 9. Example of a normal resolution image; (a) gray levels image; (b) B/W image plus filtered elements (in red); and (c) labeled image, every color represent one isolated crack. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
1) On the first computer the operation was performed choosing a range of 3 search window’s size, with 3 Parallel Computing Matlab workers, and took 25 min to complete.
2) Using the second computer the operation was performed using 12 reiteration of the separation/filtering algorithm, with 12 Parallel Computing Matlab workers; it took 16 min to complete.

Circa 2800 objects were originally identified in the image, these reduced to 195 after filtering, mainly due to the presence of noise in the image. Manual check and filtering further reduced the number of objects to 161 cracks (Fig. 10a).

4.2. Lower resolution SEM image

The second example is the analysis of an average size SEM image (1024 × 800; 0.8 Megapixels, Fig. 9), where 1 pixel corresponds to 1 μm.

1) The operation was performed using 6 reiterations of the separation/filtering algorithm, with 3 Parallel Computing Matlab workers, and took 81 s to complete.
2) Using the second reference computer the computation was performed using 24 reiterations of the separation/filtering algorithm, with 12 Parallel Computing Matlab workers, and took 51 s to complete.

Circa 1700 objects were originally identified in the image, which reduced to 275 after filtering and to 239 after manual verification of results (twin lamellae and holes were deleted) (Fig. 10b).

5. Discussion

This manuscript outlines a new quantification method for the analysis of gray level images of cracked rocks. It is based on an

Fig. 10. Examples of resulting diagrams and statistical distributions from the 2D image analysis. (a) Results from the high resolution image example, from top to bottom the histograms or showing distributions of microcracks’ length (in micrometers), width (in micrometers) and aspect ratio. (b) Results from the normal SEM image example, from top to bottom the histograms or showing distributions of microcracks’ length (in micrometers), width (in micrometers) and aspect ratio.
automatic separation algorithm developed for isolating cracks, followed by a filtering and quantification algorithm.

This method has several advantages:

- **Separation algorithm**: compared to recently published separation algorithms for analyzing cracked rocks (e.g. Le Roux et al., 2013), the use of our method of bisector lines 1 pixel-wide instead of dilated nodes ensures a similar separation efficiency, while maintaining an almost unaltered original image.

- **Geometrical analysis**: a classical approach involves the fitting of an ellipse or rectangle to the objects to be quantified (DeVasto et al., 2012; Le Roux et al., 2013). The geometric distortions of these approximations are negligible if used for the analysis of grains and minerals, but can be considerable for cracks,
especially non-straight cracks. Our method uses a totally
different approach: the length is computed using the skeleton
of the elements, while the width is derived using length and
area. The result is more reflective of the actual micro-structure,
with virtually no extraneous distortions introduced, and is
independent from the shape of the cracks, and therefore more
reliable.
- Our method includes a check of results at the end of the
  computation. Fully automatic algorithms are certainly faster, but
  not completely adaptable and reliable for every situation. For this

Fig. A2. Segmentation algorithm.
reason it is believed that allowing the user check and modify results accordingly at the end of the automatic process is necessary.

It is however recognized that the algorithm has a tendency to over-fragment cracks due to the high sensitivity to the winding of lines while performing the branchpoint function. This problem was partially solved by implementing an algorithm to identify these false nodes, based on the orientation of cracks connected to the node.

It should be also noted that results are heavily affected by the quality and resolution of the original dataset: microcracks are cracks

---

**Fig. A3.** Filtering and analysis algorithm.
with a very low width and aspect ratio, therefore a high resolution image is required in order to correctly quantify geometric characteristics. Ideally, it is suggested to achieve an image resolution that can represent the width of microcracks with at least 5 pixels. Finally, while the case study presented here focuses on micro-cracks analysis from SEM images of polished rock surfaces, the algorithm can potentially be used to analyze any type of crack or linear object at any scale though its accuracy will have to be tested.

Acknowledgments

This work was performed during Alessio Arena’s internship supported by a CSIRO Internal Funding Scheme; iVEC is thanked for granting access to the computing facilities. Mike Verrall is thanked for his support and feedback on the image acquisition procedure. Two anonymous reviewers and editor Jef Caers are thanked for their critical reading of the original manuscript.

Fig. A4. Orientation analysis algorithm.
学霸图书馆

www.xuebalib.com

本文献由“学霸图书馆-文献云下载”收集自网络，仅供学习交流使用。

学霸图书馆（www.xuebalib.com）是一个“整合众多图书馆数据库资源，
提供一站式文献检索和下载服务”的24小时在线不限IP图书馆。
图书馆致力于便利、促进学习与科研，提供最强文献下载服务。

图书馆导航：

图书馆首页 文献云下载 图书馆入口 外文数据库大全 疑难文献辅助工具