Building a virtual outcrop, extracting geological information from it, and sharing the results in Google Earth via OpenPlot and Photoscan: An example from the Khaviz Anticline (Iran)

S. Tavani, P. Granado, A. Corradetti, M. Girundo, A. Iannace, P. Arbués, J.A. Muñoz, S. Mazzoli

A R T I C L E   I N F O

Article history:
Received 8 May 2013
Received in revised form 17 October 2013
Accepted 18 October 2013
Available online 31 October 2013

Keywords:
Virtual outcrop
3D graphic
Google Earth
Structural geology

A B S T R A C T

Photogrammetry is becoming a highly efficient alternative technique to laser-scanning for creating virtual outcrop models. It is easy to create a 3-dimensional digital model of an outcrop and extract geological information contained in it by using photos taken from different locations and integrating few free and/or cheap software. Subsequently, both the virtual outcrop and the geological data can be easily uploaded into Google Earth for sharing purposes. This is opening a door to the use of virtual outcrops in geology, for both research and teaching, which due to the costs and computers’ skill requirements, was limited to a few.

The aim of this paper is to present methodologies involved in the creation, analysis and sharing of low-cost easily-built virtual outcrops, which can be extensively used for the introduction to the 3D geology. An example from the Khaviz Anticline (Iran) is used to create a 3D digital model from a set of non-oriented images, using Agisoft Photoscan photogrammetry software. The obtained geopositioned model is then imported into OpenPlot, from which geological surfaces can be extracted. These data, together with the 3D model, can be later exported in Google Earth format.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Virtual globes, such as Google Earth (GE), provide popular and useful platforms to integrate and share geospatial data (Butler, 2006; Chen et al., 2009; Tiede and Lang, 2010; Blenkinsop, 2012). In GE, three-dimensional (3D) surface models with associated textures can be uploaded as COLLADA file, offering the opportunity of sharing virtual outcrops (Xu et al., 2000; Pringle et al., 2001; Bellian et al., 2005; Clegg et al., 2005; McCaffrey et al., 2005, 2008; Trinks et al., 2006; De Paor and Whitmeyer, 2011; De Paor et al., 2012; Hodggets, 2013), created by laser-scanning (Buckley et al., 2008; Garcia-Sellés et al., 2011) or stereo-photogrammetry (Haneberg, 2008; Gessner et al., 2009; Favalli et al., 2012). In addition, GE KML files allow to handle 3D georeferenced objects (i.e., polygons) which can be displayed together with the COLLADA 3D models. These objects may be used to upload geological information, namely any geological surface (e.g., bedding planes, faults, fractures) that can be extracted from the virtual outcrop with different tools. Although several software for managing 3D geospatial data – for different purposes – have been created during the last twenty years (e.g. Petrel, GoCad, 3DMove, 3D Data Viewer), the introduction to the 3D virtual world for the current generation of researchers and students passes (or has passed) through GE Accordingly, making virtual outcrops easily compatible with GE is key for teaching and making geological virtual outcrops popular and accessible. In order to popularise the technique among a non-expert audience, two important additional requirements are needed: (1) providing entry-level software for making fast and easy the creation of virtual outcrops and (2) making easy and useful the extraction of geological data from virtual outcrops. All these requirements are presently satisfied, as they involve the use of existing cheap and/or free software and tools.

Low-resolution digital cameras and ordinary computers (i.e. less than 4GB of RAM), can be used for the 3D photogrammetric reconstructions of outcrops (e.g., Sturzenegger and Stead, 2009; Lato et al., 2013). The recent advances in both computational speed and resolution of digital imaging devices allows for amateur photographers to easily and rapidly produce 3D digital outcrop models (Pringle et al., 2001; Lebel et al., 2001; James and Robson, 2012; Favalli et al., 2012). As a consequence, photogrammetry has become a valid alternative to the more accurate, but yet limited to a circle of “experts”, light detection and ranging technique (LiDAR, or laserscan). In fact, although LiDAR is the most popular technique...
for producing accurate 3D digital outcrop models (Buckley et al., 2008; Garcia-Sellés et al., 2011; Hodgetts, 2013), its use still has many limitations in terms of logistics and costs. Hence, preventing its widespread use within the geoscientist community.

Moreover, the lack of software tools to analyse geological data extracted from virtual outcrops is presently a critically missing component. This apparently easy task, in fact, cannot be achieved within a single software. On the contrary, extraction and analysis of geological data require multiple steps through different tools. This prevents the use of virtual geological outcrops at the teaching and research levels, limiting their application to a restricted number of research or industry institutions. The main goal and novelty of our work is the unification of these steps into a single user-friendly, open-source software, integrating exporting tools for GE file formats. Such unification has led to the development of a simple, easy, and cheap workflow for virtual outcrop creation, analysis, documentation and sharing, which has the potential to revolutionise the geological documentation of outcrops, particularly on the teaching side.

In this contribution, it is first illustrated how to create a virtual outcrop by means of PhotoScan photogrammetry software (Verhoeven, 2011; Arbués et al., 2012). Then, the geological data extraction from a virtual outcrop using the OpenPlot software 3D tools (Tavani et al., 2011) is explained. Finally, it is shown how PhotoScan and OpenPlot allow to export the virtual outcrop and the extracted geological information in a format readable by GE. For this purpose, data from the Khaviz Anticline of the Zagros fold belt of Iran are used.

2. Virtual outcrop creation via PhotoScan

2.1. An overview on photogrammetry

3D view is ensured by the availability of at least two images of the same scene taken from different positions. Knowing position, orientation, and focal length of each image, allows for the position of any point in space to be computed from its 2D coordinates in the two images (Fig. 1). On the other hand, when camera parameters (i.e. photos’ position, orientation and focal length) are unknown, they can be derived by the 2D coordinates of equivalent points in the different photos, which in turn allows for the computation of points’ coordinates in the space. From what is stated above, it is clear the importance of detecting the position of a suite of points in different – but overlapping – photos of the same scene. This issue is a matter of study since the dawn of the digital era (Ullman, 1979), with Structure From Motion (SFM) algorithms being designed to correlate points in images of the same scene, taken from different positions and/or in different times (Ullman, 1979; Tomasi and Kanade, 1992; Gruen et al., 2004; Remondino and El-Hakim, 2006; Szeliski, 2010).

SFM algorithms are presently implemented in various software (e.g. Bundler, Microsoft’s Photosynth, Photomodeller, Agisoft-PhotoScan). For a given a set of partially overlapping images, SFM algorithms automatically detect a suite of common points in each image pairs, and subsequently, recognise the camera parameters for each photo. This, in turn, permits the extraction of the 3D coordinates of each point recognised in at least two photos and hence, the creation of a point-cloud representing the surfaces of the objects within the target scene (e.g. Gruen et al., 2004; Verhoeven, 2011; Favalli et al., 2012). Essentially, the overlapping photos should be taken from multiple points of view. It is worthwhile to use the same camera to minimize errors coming from the use of different lenses and camera sensors. It is not mandatory to keep the focal length fixed, but it is strongly recommended. Photos should also be taken consecutively, or at least under the same lighting conditions (e.g., Arbués et al., 2012). These recommendations ensure that each portion of a scene is represented by a similar pixel pattern in the different photos, easing the recognition of points by the SFM algorithms. These procedures ensure and maximise the recognition of points, allowing for the creation of denser point-clouds.

Different photogrammetry packages are available, these having a wide range of costs, ease of installation and use, and ability to export results. Agisoft Photoscan has been chosen in our workflow due to its user-friendly nature, the availability of academic licensing, and tools allowing for the export of results into GE and OpenPlot.

2.2. Building a georeferenced 3D model in PhotoScan

After uploading photos, the to-do list starts with photomasking (Fig. 2a), which consists in defining areas that will not be involved in the 3D reconstruction (i.e., the sky, vegetated areas, etc.). Photo-masking is not mandatory but it is highly recommended to ensure a substantially faster reconstruction. The subsequent photo-alignment command, attempts to recognise the position of the same points in the different overlapping photos. This allows to compute position and orientation of photos and, in turn, to create the point-cloud. At the end of the procedure, PhotoScan indicates those photos that have been aligned (according to the software) and those that have not, with wrong alignments being recognisable by either unrealistic or wrong photos’ positioning (Fig. 2b), and/or by the presence of unrealistic geometries within the point cloud, such as planes or narrow and elongated bands of points converging toward a photo. The latter can occur when two or more photos have been taken from very close locations, however with different focal lengths. Alignment can be improved by: (i) avoiding the abuse of photos taken with very different focal lengths; (ii) manually indicating the same point in the different photos (Fig. 2c); and (iii) removing photos that cannot be successfully aligned.

The subsequent “building geometry” step triangulates the obtained point-cloud and returns a mesh (or a set of meshes) made up of irregular triangles. Memory requirements for photo-alignment and building geometry steps, depends on both hardware and number of photos; details can be found in the software’s vendor wiki (http://www.agisoft.ru/wiki/PhotoScan/Tips_and_Tricks#Memory_Requirements). From our experiences building different virtual outcrop models, processing time strongly depends on the hardware but also on the quality of the dataset. Building a

![Fig. 1. Scheme showing parameters involved in stereoscopic view. When a point P is photographed from two cameras C1 and C2, the position of P in the two photos (Pc1 and Pc2) depends on the position of the two cameras, on their focal length, and on their orientation, which is defined by 4 parameters (the ijk versor and the angle α defining the amount of rotation about the ijk axis).](image-url)
small model (i.e. less than one million of triangles) using 10–20 properly-taken 1–4 Mpixel photos, is essentially a matter of some tens of minutes for a computer equipped with a 4 Gb RAM. The same 4 Gb RAM hardware requires many hours when 30–50 photos are integrated, and the output model includes some millions of triangles. Processing of hundreds of photos and the creation of models made of tens of millions of triangles (i.e. having resolutions comparable to those of laserscan models), requires at least an 8 Gb RAM working for tens of hours.

After, the “Build texture” command reconstructs a texture map that will be draped onto the triangular mesh. In detail, two coordinates (U and V) are assigned to each vertex of each triangle of the mesh, representing a point in the texture map. The U and V coordinates of vertexes permit to crop a triangle in the texture map, and to paste it onto the corresponding triangle of the mesh (Fig. 3). The higher the number of triangles in the mesh is, the higher the resolution of the texture map should be. The texture size should be set following the power of two rule, with height

---

Fig. 2. Steps involved in the photo-alignment. (A) Masking of photos allows one to remove unwanted areas from the reconstructed model. (B) Photo-alignment of unoriented photos provides as output both the reconstructed point-cloud and the position of photos. At this stage, large errors can be easily recognised, being evidenced by unrealistic photo-positioning (in the example, some photos are located more than 500 m above the ground) and by the presence of incorrectly positioned points. (C) Providing the position of known objects (markers) in the different photos can greatly help the alignment of photos.

Fig. 3. Procedure for texturing of a triangular mesh.
and width that should be set to $2^N$ (e.g. 1024, 2048, 4096 pixels). Ordinary graphic cards (for example we used an Intel HD Graphics 4400) work fine up to sizes of 4096 or 8192. This imposes limitations in the mesh size as, depending on the complexity of the target surface, a $4096 \times 4096$ texture may imply pixellation in meshes exceeding few hundreds of thousands of triangles. Excessive pixellation can be avoided by splitting the model into sub-models with associated independent $4096 \times 4096$ textures. The next step of the procedure is the georeferentiation of the model, which consists in the re-orientation and re-scaling of the 3D model. This step requires setting the coordinate system and defining the coordinates of at least three markers. Exporting the model(s) is the last step of the workflow, and it is a quite intuitive task, as it is shown at the end of the next section.

2.3. The Khaviz Anticline case study

A case study from prolific Zagros fold and thrust belt is presented. This pretends to illustrate how the photogrammetry technique can be easily used to create virtual outcrop models using commodity-off-the-shelf hardware, cost-accessible software, and archived low-resolution digital photographs (i.e., not originally taken for photogrammetrical modelling). Photoscan was used to build a point-cloud representing the western flank of a 4 km-long valley, which provides a natural cross-section for the NW–SE oriented Khaviz Anticline (Fig. 4a and b). Along this valley, the Asmari Fm. is continuously exposed along hardly accessible vertical cliffs from the forelimb to the backlimb of the anticline (Wennberg et al., 2006). A spectacular array of extensional faults are located in the fold’s crest affecting the Asmari Fm. (Fig. 4c). Panoramic photos were taken in 2003 with a Single Lens Reflex NIKON E4500 camera from different positions (Fig. 4b). These photos were taken several years before the advent of commercial software for photogrammetry and without the goal of later 3D reconstruction. The camera had a resolution of 3.8 Mpxels, and photos were taken in two different days and under different lighting conditions. Essentially, most of the basic requirements for helping the SFM algorithms working properly were not ensured. This is the reason the Khaviz example is presented, as it allows to illustrate how, even in non-ideal conditions, and for a previously acquired photographic dataset that many geologists may have, the procedure can work successfully.

38 photos were uploaded and the first automatically reconstructed point-cloud included several artefacts. The mismatch was reduced by providing the position of 13 known objects in different photos. This helped the photo-realignment and the almost correct positioning of all the photos. A relevant noise (i.e. error) still occurred in the northern portion of the point-cloud, being represented by points located up to few hundreds of meters below the ground level (Fig. 5a). Removal of photos associated with these points’ cluster ensured the reconstruction of a better, not yet error-free, point-cloud (Fig. 5b). At this stage, minor errors could not be easily detected in the point-cloud, which accordingly was triangulated to detect second-order errors. Triangulation, setting a medium target quality and without imposing the maximum number of triangles, produced a mesh made of about $4 \times 10^6$ triangles. Small errors in the photo-alignment were recognisable in the mesh by the presence of small ‘bubbles’, horizontal peaks and/or isolated triangles. These erroneous features, indicate either an inexact photo alignment or a correct photos-positioning in fact, but a wrong correlation of points between the different photos. This problem must be solved by adding additional markers and/or by individuating and subsequently removing problematic photos. This is the most time-consuming step of the workflow as it may

Fig. 4. (A) Geological map of the Khaviz Anticline. (B) Google Earth image of the study area, showing the reconstructed cliff and the shooting location of photos used to build the 3D model. (C) Panoramic view of the cliff, with faults shown.
require several loops including: (i) markers positioning, (ii) photo realignment, (iii) triangulation, (iv) errors detection, and (v) removal of problematic photos. In our case, this implied the removal of other 10 photos, with the obtained mesh fitting quite well most of the target surface (Fig. 5c). Remarkable errors still occurred in two areas, as indicated in the inset of Fig. 5c. The first area is located at the edge of the model and was finally removed. The uppermost portion of the second area was key for the purpose of the work, as it represents the link between two large areas of the model. This problematic area is covered by only three photos taken in different days and with different focal lengths. Although the use of markers allowed to correctly reconstruct the cameras’ parameters, large differences existed in the photos’ pixel patterns. Accordingly, the recognition of the same points in these three photos was a hard task for the SFM algorithm. It was not possible to improve that part of the model but the correct positioning of all the photos, ensured the construction of a single model free of any relevant distortion and/or rotation between the two correctly reconstructed portions. This was an important aspect for georeferencing the model. In fact, when dealing with ‘ancient’ photos, the absence of any strategy aimed at positioning objects of known position can imply problems. This is particularly true for small vertical cliffs, like that in the northern part of our model, which can be georeferenced only if a very detailed topographic map is available or, as in the case described in this work, if the cliff forms part of a larger model.

In order to georeference the model, two additional markers were created. Three non-collinear points were used for the georeferentiation, being located far away one from each other (Fig. 5c). X and Y coordinates of these points were provided by the high-quality image of the area available at Microsoft’s Bing maps (http://bing.com/maps/), while the Z coordinate was derived using a 1:25,000 topographic map. The adopted base maps may imply some meters of error, particularly for the Z coordinate. In our case the software indicates 26 m of error. Such a value is however small when compared to the 4 km long model, and indicates a model distortion of less than 1%. Concerning this, and the resolution of the mesh, the area of the georeferenced surface is about $6.11 \times 10^6$ m², while the number of triangles is $4.4 \times 10^6$, resulting in an average triangles’ area of about $1.38$ m². This value, the 1% distortion and a maximum error of 26 m, have to be considered proxies for the mesh resolution.

At this stage, the 3D model was built and all the points/ triangles were georeferenced. The last two steps involve texturing and exporting the model. Depending on export needs, different strategies must be adopted for texturing. First, the area that was important for the georeferentiation but not relevant for geological purposes was erased. This translated into a smaller triangular mesh (2.5 millions of triangles) which ensured the best results during texturing, lighten the CPU and GPU work and an increase of rendering speed. As GE in not able to handle meshes with more than a few tens of thousands of triangles, the resulting mesh was cloned and downsampled to about 21,000 triangles. Such a small mesh does not experience big texturing problems and, hence, mesh texturing for later GE upload was done by means of a single 4096 × 4096 texture map. In the subsequent export step, the model format was set to KMZ and the “export texture” option was activated. This generated a KMZ compressed file including: a *.dae file, which is the triangular mesh in COLLADA format, a *.jpg file representing the texture map of the mesh, and a *.kml file that provides the coordinates of the COLLADA mesh. The KMZ file can be directly uploaded in GE.

In order to export the model to OpenPlot. The use of OpenPlot is aimed at extracting as many geological features as possible, meaning that the model created in PhotoScan should not be downsampled. Depending on the computer’s hardware, the resolution of an untextured mesh supported by OpenPlot can easily exceed millions of triangles (e.g. 20 million of triangles can be rendered at an about real-time speed in a Linux machine equipped with a 8 Gb Ram and a Intel HD Graphics 4400 graphic card), whereas textured meshes have strong limitations in OpenPlot as in any other software using OpenGL graphic libraries. In order to export a model with a high resolution texture, the model splitted into six sub-models. This allowed applying a 4096 × 4096 texture map to each of the six sub-models, and to

![Fig. 5. (A) Initial photo-alignment. (B) Improved photo-alignment and point cloud reconstruction. (C) Triangulation, showing markers used for georeferentiation and problematic areas.](image-url)
3. Geological features extraction via OpenPlot

OpenPlot is an open source cross-platform (Linux, Mac OS and Windows) software for structural data analysis (Tavani et al., 2011). The software comes with a 3D environment allowing to visualize and manipulate 3D geological features. The software is written in Xojo (formerly RealBasic), and it is free for download together with its source code (http://www.openplot.altervista.org). The working philosophy and main functionalities of OpenPlot are detailed in Tavani et al. (2011). Most relevant upgrades since the 2011’s version concern the import of wavefront files created via PhotoScan, export of planar polygons in a KML format readable by Google Earth, and the replacement of Quesa graphic library with OpenGL.

3.1. Importing PhotoScan models in OpenPlot

The wavefront OBJ format is a simple data format including three files (Fig. 6): (1) The *.obj file that includes the geometric information of the mesh, and consists of: (i) a list of XYZ coordinates, (ii) a list of vertex-normal coordinates, (iii) a list of UV coordinates, and finally (iv) a list of triangles made of indexes pointing at the previously defined arrays, and defining the XYZ, UV, and vertex-normal coordinates of each vertex of the triangle (Fig. 6). (2) The *.mtl file that includes materials’ definition, and it is not used by OpenPlot. (3) The image file (either in *.jpg or *.png format), representing the texture map of the mesh.

Importing a wavefront file for its subsequent visualisation in a 3D OpenGL scene, is the most time consuming step in OpenPlot. It requires transforming data into blocks of memory readable by OpenGL functions. Without entering deep inside into the programming side, what OpenPlot does is simply reading the XYZ and UV coordinates of each of the three vertexes of each triangles, stored as string in the OBJ file, and transform them in two distinct blocks of memory. Each coordinate (i.e. X, Y, and Z in one block, and U and V in the other) of each vertex is written in 8-byte double precision format, implying 120 bytes of memory for each triangle (8-bytes X 5coordinates X 3vertexes). Similarly, red, green, and blue (RGB) values of each pixel of the texture map are read and transformed into a third block of memory, consisting of 3 bytes of memory for each RGB value. Once these blocks of memory have been created, OpenPlot can save them using the “compressed XML” format, from which they can be rapidly (i.e. few seconds) read back. Time for reading and converting wavefront files is essentially a matter of processor capability and RAM. Some examples are shown in Fig. 7 and witness how, in dual-boot the machines, Window OS is slightly faster in reading data, but Linux OS has a higher (one order of magnitude) real-time rendering capability.

The information provided above is important for planning the import of meshes. The number of triangles, in fact, determines the size of the three blocks of memory, and also that of two additional temporary blocks (serving during reading but erased after the mesh’s import). In particular, the maximum amount of RAM required by OpenPlot to import a mesh is about 0.25 Gb for each million of triangles.
3.2. Displaying data in the 3D view window and extracting geological information

After reading the wavefront mesh, it can be displayed in the 3D window of OpenPlot, via the “3D view” command in the main menu, which transforms all the selected data stored in an OpenPlot file into graphical 3D objects, adding them to the OpenGL scene of the 3D window. The Draw-polyline is one of the OpenPlot drawing tools. Once activated, clicking on the 3D scene passes the X and Y coordinates of the mouse to a function returning the X, Y and Z coordinates of the point in the space (provided there is an object under the mouse). A 3DPoint is then added to a temporary polyline, which is conceptually drawn. The geological purpose of this is the point-picking of the intersection between a target surface, such as a bedding surface, and the outcrop surface (e.g. Trinks et al., 2006). This allows the digitalised polyline to be subsequently transformed into a planar polygon, which is done by diagonalising the following 2nd order symmetric matrix:

\[ T = \begin{pmatrix}
\sum_{i=1}^{n} u_i^2 & \sum_{i=1}^{n} v_i u_i & \sum_{i=1}^{n} w_i u_i \\
\sum_{i=1}^{n} v_i u_i & \sum_{i=1}^{n} v_i^2 & \sum_{i=1}^{n} v_i w_i \\
\sum_{i=1}^{n} w_i u_i & \sum_{i=1}^{n} w_i v_i & \sum_{i=1}^{n} w_i^2
\end{pmatrix} \]

where \( u, v \) and \( w \) of the \( i \)th point of the temporary polyline are given by:

\[ u = x - Ax; \quad v = y - Ay; \quad w = z - Az \]

being \( Ax, Ay, \) and \( Az \) the coordinates of the centre of mass of the point set, and being \( n \) the number of points. Eigenvalues of \( T \) (\( \lambda_1 > \lambda_2 > \lambda_3 \)) define how much, the point set concentrates around the three eigenvectors (\( \xi_1, \xi_2, \xi_3 \)). This allowing to define and evaluate the best-fit plane that must contain the eigenvectors associated with the maximum and medium eigenvalues. The smaller is the minimum eigenvalue (\( \xi_3 \)), the lowest is the dispersion along the direction perpendicular to the best-fit plane and, accordingly, the best is the fit-plane. When the medium eigenvalue (\( \lambda_2 \)) approaches the minimum one (\( \lambda_3 \)), it means that points are clustered along a line, and the best-fit plane is becoming a geometric artefact. Therefore, the quality of the best-fit plane is defined by both \( \lambda_3 \) and \( \lambda_1/\lambda_2 \). Providing general thresholds is impossible, as they depend on the attributes of the target surface, on the required accuracy, on the scale of observation, and on several other factors. In essence, they must be decided case by case by the operator. A best-fit plane can be displayed in OpenPlot to ease this decision while digitalising points, and the operator can define threshold values for \( \lambda_3 \) and \( \lambda_1/\lambda_2 \). If the best fit plane satisfies these values, the plane is painted in blue, if not it is painted in red. Best-fit planes satisfying the user can be added to the dataset, which is done by routine taking the three eigenvectors and creating a planar rectangle oriented perpendicular to the minimum eigenvector and containing all the polyline points’ projection on the best-fit plane. The strike and dip of the plane are also derived, so that the newly created planar polygons can be treated as a structural object, filtered, analysed, plotted and so on (Tavani et al., 2011). Fig. 8 shows in a simplified scheme the steps followed to import and analyse the Khaviz’s PhotoScan model into OpenPlot.

Once the surfaces of interest have been picked and transformed in planar polygons, these can be written in KML format. The KML structure is a sub-format of the node-based Extensible Markup Language (XML). The code for writing a planar polygon in KML format is illustrated in Fig. 9, and requires writing first colours and then geometries. In KML format, the colour attribute of each polygon is expressed trough the node <styleUrl >, which points at a previously defined <stylemap > node (Fig. 9), where colours for the normal and highlighted states are defined by means of others two <styleUrl > nodes, respectively. These, in turn, point at a <Style > node, where colours are defined. Notice that colour codes of Xojo and KML language differs, so colours must be translated. Once the <Style > and <Stylemap > nodes are written, polygons’ attributes can be written. For each polygon, user’s and OpenPlot’s fields (e.g., azimuth, lithology, author of the work) are written in the <description > node, allowing these fields to be visualised in the pop-up window of GE. After this, the polygon’s points are written. UTM coordinates (used in OpenPlot) are transformed into latitude and longitude that, together with elevation, are written as a string in the <coordinates > node of the KML file.
Notice that, before writing points in KML files, the `<altitudeMode>` node must be declared and set to “absolute”, as the default value for GE is “ground”, where the Z coordinate is forced to the ground level.

4. Discussion and concluding remarks

The results of point-picking and subsequent best-fit plane extraction of bedding surfaces and fault planes in the Khaviz anticline, can be appreciated in the KMZ file provided as supplementary material, and downloadable at this site (http://www.openplot.altervista.org/Guide.html). In essence, the extracted geological data indicate that extensional faults strike parallel to the trend of the hosting anticline, consistently with detailed mesostructural studies carried out in the area (e.g. Wennberg et al., 2006). For the purpose of this work, the data themselves are the key, rather than discussing the extracted geological data. Extracted data and a ‘light’ 3D virtual outcrop model can now be easily shared in GE, which is the nowadays mostly used entry-level platform for 3D georeferenced data visualization. The presented example focuses on the creation of a virtual outcrop from tens of photos that were not collected with the aim of virtual outcrop creation. The resolution of the derived 3D model, which finally included about 2.5 million triangles, is far away from those created via laser-scanning, but also from those that can be created via photogrammetry when a strategy for photos acquisition is planned. Such a resolution allows to perform analysis of major features (namely digitalisation of polylines) but does not permit the automatic detection of features directly onto the point-cloud (e.g. Baker et al., 2008; Garcia-Sellés et al., 2011), which would require a significantly denser point-cloud. However, the Khaviz example shows that virtual outcrops are now fully accessible to common people, without requiring heavy devices, high-level hardware, expensive software and computer skills. Essentially, with a relatively minimal effort and costs it is now possible to create a 3D model of any outcrop where field data are collected. The possibility of integrating geological data into a 3D framework accessible to everybody having a computer and an internet connection, has very important implications for geoscience modelling, research and, particularly, education (De Paor et al., 2012), and this approach is expected to become the standard procedure in the near future. In fact, it has to be considered that models like the one presented in this work can establish as a standard, replacing the traditional photo-documentation of geological features. This is particularly so as it is expected that software for photogrammetry will improve and will be implemented for low-cost devices, such as smartphones or tablets, allowing an ‘on the fly’ 3D model construction directly in the field (although the resolution of these models is still not expected to be too much high). These devices, in fact, are commonly equipped with tools detecting the orientation of the device itself. As shown in Fig. 1, the parameters defining the orientation of the photo can be virtually used by the SFM algorithm to derive a model that only needs to be scaled (i.e. no rotation of the model is required), and then translated in the correct location.

At present, few limitations exist in the illustrated workflow, which are expected to be easily solved in the near future. A main concern for geoscientists is the impossibility of drawing features (like polylines) directly in software for photogrammetry. In fact, PhotoScan, as well as most of the similar software, does not allow point-picking onto the point-cloud or onto the triangular mesh, and a passage through a CAD-like software is required for such a purpose. Here, we have proposed a passage through the open-source OpenPlot software, which includes data analysis tools. Several alternative solutions exists, tough. Concerning the visualization of data into Google Earth, a list of advantages could be compiled for such a platform, however, three major problems remain: The first one is the low resolution of the COLLADA models and associated texture that...
can be supported by GE (limited to many tens of thousands of triangles) and, in essence, GE allows only the upload of draft 3D models. These preliminary low resolution models are useful for fast visualization of geological structures having a comparable size to that of the model, but virtually useless for data extraction. A second intrinsic problem of GE is the impossibility of solving the overlap between the uploaded 3D terrain model and the terrain model of GE, which requires flattening the GE elevation to properly visualize any uploaded 3D model. This must be done also because of the very different resolutions of the uploaded 3D model and the GE terrain (the latter using a 90 m DEM). This difference can give the impression of a strong misalignment, but instead is caused by the different resolutions of the two terrains. This directly brings to the next problem, which is that GE only allows zooming and rotating around portions of its terrain model. Practically, it is not possible to directly select a point of an uploaded 3D outcrop and rotate the model around it. Despite all these problems and the existence of several alternatives, which do not experience the same limitations in visualising virtual outcrops, GE still remains a valid tool, probably the best one for the introduction to the 3D world of geology.

Acknowledgements

We would like to thank Jef Caers, Klaus Gessner and two anonymous reviewers, whose comments helped us to significantly improve this paper.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.cageo.2013.10.013.

References
