COMPUTER-AIDED DESIGN & APPLICATIONS, 2017
VOL. 14, NO. 1, 1–16
http://dx.doi.org/10.1080/16864360.2016.1199751

Computer-Aided Design
Taylor & Francis
Taylor & Francis Group

An innovative photogrammetry color segmentation based technique as an alternative approach to 3D scanning for reverse engineering design

David W. James, Fawzi Belblidia, Jurgen E. Eckermann and Johann Sienz

Swansea University, UK

ABSTRACT

The photogrammetry procedure is introduced as a cheap and -effective alternative to 3D scanning for constructing 3D point cloud data for reverse engineering applications. The data is then manipulated to generate CAD-ready models for FEA simulations or 3D printing purposes. A practical demonstration using model cars is conducted to show the effectiveness of the procedure in generating point cloud data, and various improvements are introduced. In particular, and innovative to this research work, a feature segmentation algorithm based on color information captured from photographs is implemented. This program, embedded within the photogrammetry process, allows for automatic selection of parts and features from an assembly model, providing a leading advantage over current 3D scanners, which capture spatial data only. Moreover, an intuitive method of enhancing the selection of required features of a triangular mesh during the segmentation process is introduced. This is achieved by manually marking the boundaries on the part of the object to be selected prior to taking photos. This has been proved to be more effective where areas of the object parts have similar colors.

KEYWORDS

Photogrammetry; 3D scanning; CAD; color segmentation; point cloud data; RGB color data

1. Introduction

Three-dimensional computational techniques play an important role in many engineering applications. Three-dimensional CAD software is commonly used to design parts and assemblies. Generated CAD files are also used in a variety of other applications to reduce time to market. For instance, finite element analysis (FEA) packages allow a range of simulations (thermal, structural, etc.) and more cost-effective optimisation on the CAD design prior to prototyping. Alternatively, CAD can be automatically converted to machine paths for computer aided manufacturing processes. In addition, rapid prototyping and 3D printing are commonly being adopted to aid visualisation of parts and provide the designer with a better perspective on the physical size and shape of the final product, allowing properties such as ergonomics to be quickly assessed.

Over the past two decades there has been increasing interest in reverse engineering methodologies, in which CAD geometry is generated by converting 3D geometrical data in the form of point clouds, obtained by 3D scanning real objects. In 3D scanning, 3D point cloud data is generated by measuring the exact location of points on the object generally using a laser or structured-light-type system. In most cases the point cloud data is converted into a triangular mesh and then surface or solid CAD geometry. An alternative to 3D scanning is close-range digital photogrammetry. In essence, photogrammetry consists of compiling a set of 2D photographs of an object taken from different locations around it to generate 3D geometrical data. This is made possible by considering the camera positioning relative to the location of matched key features, used as markers, on the object in different photographs. A range of software has been developed to automate this process, giving rise to digital photogrammetry.

By using photogrammetry software, a set of digital photographs is transformed into 3D point cloud data. Once the point cloud data is generated by photogrammetry, the remaining processing steps to generate CAD are similar to those in the 3D scanning process. In addition to the point cloud data, red, green and blue (RGB) color data of the object is also captured from the photographs processed during digital photogrammetry.

In this work, the digital photogrammetry technique is presented as a cost-effective and time-efficient method of producing point cloud data used to generate 3D computational models for engineering applications as an alternative to 3D scanning. The main focus of the work is the practical consideration of creating point cloud via

CONTACT David W. James D.W.James@swansea.ac.uk

photogrammetry and the conversion of this data into triangular meshes using available commercial software. Further to this, a novel procedure to segment a triangular mesh into different named features using RGB color data is implemented and fully explored, using model cars as the subjects.

The outline of this article is as follows. In the section "Background", differences between 3D scanning and photogrammetry are described in detail and the readiness of these processes for engineering applications is outlined. Moreover, the section continues with the description of different types of segmentation and the use of RGB color data. The section "Methods" focuses on the description of the photographic setup, including camera, positioning and models, and introduces the software used to generate point clouds and the algorithm developed to handle the segmentation feature. "Photogrammetry – practical considerations" presents practical considerations to achieve the best results. This is followed by the section "Results and discussion".

2. Background

The steps involved in generating point cloud data through either 3D scanning or photogrammetry are compared and discussed. This is followed by a detailed discussion of the novel segmentation process developed as a means of separating an object assembly into useful parts.

2.1. 3D scanning and photogrammetry

Photogrammetry generates 3D geometry from a series of 2D photographs of an object taken from different positions. Starting from the position of the camera, and by identifying and matching corresponding points in each image, a 3D model is constructed through a triangulation procedure, the principle of which is shown in Fig. 1. Unique 3D locations are calculated by using the known translations and rotations of the camera positions for multiple photographs and identifying and matching key features in the photographs. Fig. 1(a) shows that the location of a point in a single photo may be anywhere on the line tracing the path of a light ray intersecting the image of the point on the camera sensor and the actual point in 3D space, whereas with two or more images and known camera positions, as in Fig. 1(b), the location of the point in 3D space is uniquely identified at the position where the light ray paths intersect.

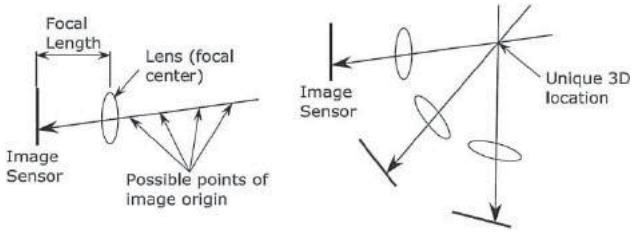

Figure 1. The principle of triangulation. (a) Using a single photo the possible point of origin of a light ray reflected from the subject is at any position on the line tracing intersecting the subject and the image sensor pixel; (b) with multiple photos the lines tracing the paths of the light rays intersect at a unique location allowing is position in 3D space to be defined.

The theory of generating 3D positions from a series of 2D images considerably pre-dates 3D scanning [12]. However, until recently the technology to automate the process to recognise and match features in images and the computational power required to generate enough points to formulate useful datasets was lacking.

Nowadays, several relatively cheap or free photogrammetry software packages are available for standard desktop PCs, including:

- Autodesk 123D Catch [1],
- Agisoft Photoscan (which is used in this work) [4] and
- Photo Modeler [18].

Computational processing power has significantly increased over the past few decades. Likewise, the development of digital cameras means that digital image formats and related computer software are commonplace. In addition, triangulation algorithms such as Lowe's Scale-invariant feature transform SIFT [14] and its variations [5] have proved to be computationally efficient in correctly identifying and matching corresponding key points in a collection of photographs with different scales and transformations.

In general, the equipment used in the photogrammetry process is far cheaper and simpler than 3D scanners. However, some care is required at the image capturing stage to acquire photographs with the required characteristics for the photogrammetry procedure. Once images have been captured, little user input is required to generate 3D point cloud data. Moreover, the learning curve for the photogrammetry procedure is not as steep as for 3D scanning. In most cases of reverse engineering applications, capturing images of an object, and especially a large object such as a full-scale car, is faster and more cost-effective than 3D scanning [10]. The steps required to generate 3D CAD data from photogrammetry and 3D scanning are shown in Fig. 2.

One criticism of photogrammetry relates to the low accuracy and resolution of the generated data in contrast to 3D scanning; fine details cannot be as accurately represented [6]. However, several authors have demonstrated that this evolving technique remains an effective tool, providing acceptable accuracy for engineering

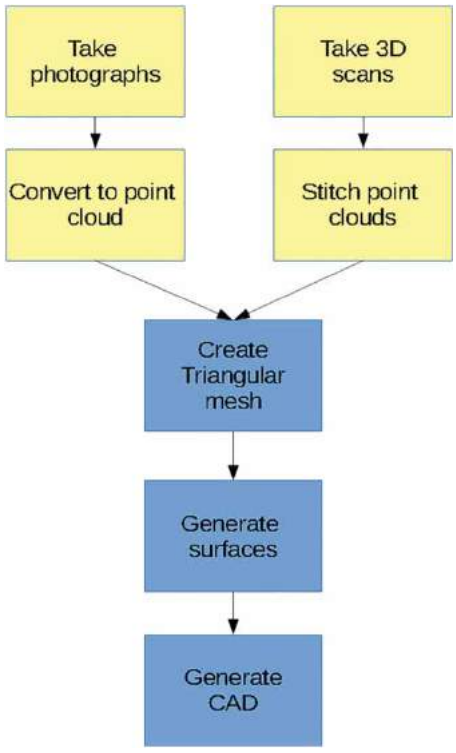

Figure 2. Steps required for generating 3D CAD data from 3D scanning and from photogrammetry (steps common to both processes are shown in blue).

applications. For instance, Luhmann [16] stated that, for industrial applications, length measurement errors (LMEs) are the largest source of measurement error in photogrammetry. An LME is the error in the maximum distance between two points on an object. With a camera of sufficient resolution, LME values below 80 μm from objects with maximum dimensions of 2 m in orthogonal directions are possible. Golparvar and his co-authors [9] found that while the accuracy of point clouds resulting from images is lower than from laser scanning, an advantage of using images is that useful semantic information can be extracted about progress on construction sites in an easy-to-use and time-efficient manner. Furthermore, González-Jorge and his colleagues [10] demonstrated that both laser scanning and photogrammetry procedures are suitable for obtaining geometrical parameters in a car-testing environment in accordance with UNE and ISO standards. The authors cited lower equipment cost, portability and lower data-acquisition times as significant advantages of photogrammetry when compared with 3D scanning.

2.2. Segmentation

Segmentation is the process of separating the complete scan of an object assembly into constituent components [3]. In reverse engineering applications there are two types of segmentation, generally referred to as: a) geometric segmentation; and b) feature segmentation. In the case of geometric segmentation, scanned objects are segmented into different 3D CAD primitives such as planes, surfaces and cylinders; see, for instance, the work of Várady *et al.* [23], Vosselman *et al.* [25], Rabbani *et al.* [19, 20] and Masuda and Tanaka [17]. Software such as Geomagic Studio [8] and Rapidform [21] contain algorithms that recognise and segment point cloud and mesh data into primitives and surfaces. In addition, whilst most of the geometric segmentation procedure is automatic, there is still some user control for efficient 3D CAD primitive recognition

Feature segmentation separates an object assembly into named features representing specific parts and/or functionalities. For instance, named features recognisable on the outside of a car would include: doors, windows, lights, tyres and number plates. When a part consisting of multiple features has been scanned or photographed, feature segmentation and/or removal of some features is required in most situations [24]. It has been observed that when geometric segmentation techniques are adopted as a route to feature segmentation, over-segmentation often occurs, owing to each feature being composed of several geometric primitives [26]. In many cases, therefore, it is desirable to perform feature segmentation prior to geometric segmentation, firstly dividing the scanned part into useful components and then geometrically segmenting each component independently.

It is a simple task for a software user to identify named features belonging to a scanned object. Nonetheless, manually segmenting a point cloud or triangular mesh with well-defined boundaries is an extensively time-consuming and tedious task. Tools to aid the user recognise the features boundaries', especially features that do not have pronounced geometrical deviations at boundaries, are therefore extremely desirable.

In using photogrammetry to create 3D point cloud geometry, additional valuable information in the form of color is gathered at the same time as the geometrical data, and is stored either as a set of RGB color data values at each point or as a texture map. Software such as Photoscan [4] and 123D Catch [1] can assign RGB color values from the photographs to points and triangles created. The RGB-D images created from the gathered color information are useful in generating 3D data and segmenting different objects [13]. Color-based segmentation on the point cloud has been demonstrated by Zhan and co-workers [26]. Their procedure is fully automated, as the user has no input into selecting the set of starting points for the segmented regions.

In the present work, two model cars are used as objects on which to apply and demonstrate the photogrammetry technique and the novel color segmentation procedure. Practical steps are introduced, supported by the use of numerical photogrammetry software and algorithmic programming, to generate feature-segmented 3D triangular meshes from 2D images. These steps demonstrate the ease and convenience of photogrammetry as a means of generating 3D geometric datasets in the form of point clouds and triangular meshes. The data is then used to create CAD using reverse engineering software. The application of the color information in the segmentation process is novel and is implemented (algorithm chart presented in Fig. 9 below) using the Python language in the Geomagic scripting environment to feature-segment triangular mesh data. The developed algorithm is designed as an interactive tool to aid the user in selecting regions from a set of initial triangles on the mesh. A pre-photography step is also suggested, in which colored boundaries are manually marked out on the object prior to photographing for quick-and-easy feature segmentation during the post-processing of the mesh data. Although an example of CAD surfaces is presented, the focus in the work remains on the generation of the point cloud and feature segmentation of the triangular mesh by color. Additionally, methodologies for generation of CAD from point clouds are common to 3D scanning and photogrammetry and are well documented; see, for example [7, 22].

3. Methods

In this section, the methods used in this work are discussed. The first sub-section describes the photographic setup, including details of the models, camera and camera positioning. The second sub-section lists the software adopted for each stage of the point cloud creation and feature segmentation and gives details of the color-segmentation algorithm developed.

3.1. Photography procedure adopted

The subjects of this work are a 1/42 scale model Jaguar E-type car and a 1/32 scale model Ford Gran Torino (see Fig. 3). In order to reduce the shininess of the surfaces while taking photographs, the models are coated with talcum powder applied by hand, with the excess then being shaken off to ensure that the underlying colors of the models could be clearly seen.

Photographs are taken at thirteen equally rotationally transformed locations (about the center of the model) at three height levels, giving thirty-nine images in total for each model. The first photo at each height is in line with the rear of the model and the final photo is in line with the front of the model, with images taken of the front, rear and one side of the car. Between each photo, the camera position is rotated by fifteen degrees around the center point of the model with the camera focused approximately at the model's center. A template marked with the angular increments is provided to align the camera frame. For each photo, the marker is located approximately at the bottom of the center of the frame, therefore keeping the distance from the camera to the center of the model roughly constant. A white screen is placed around the model to give a featureless background, in order to minimise the contribution of background features to the digital photogrammetry procedures and aid masking (see section 5.1). The positioning of the camera at a single height relative to the model is shown in Fig. 4.

Fig. 5 shows typical photographs of one of the models at a single height with a systematic camera movement to a new location.

Similarly, Fig. 6 shows three images of one of the models taken at the same camera position but with the camera at different heights. The camera is angled so that it is focused on the center of the model. In the first image the camera is approximately parallel with the door of the car model, in the second it is angled at approximately twenty-five degrees to the horizontal and in the third it is at approximately forty-eight degrees to the horizontal.

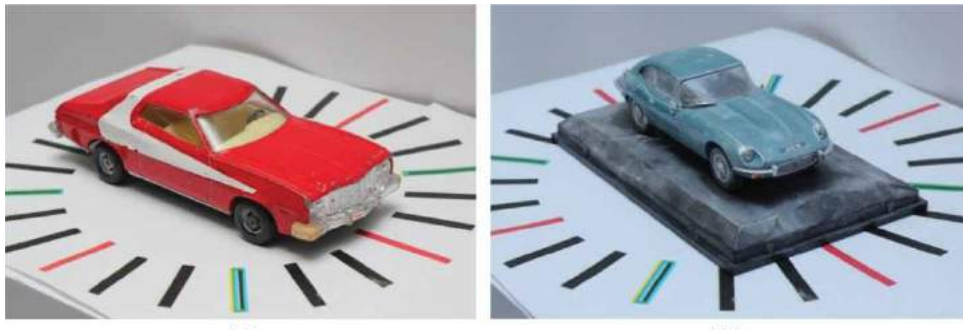

Figure 3. Model cars used in this work a) Ford Gran Torino, b) Jaguar E-type.

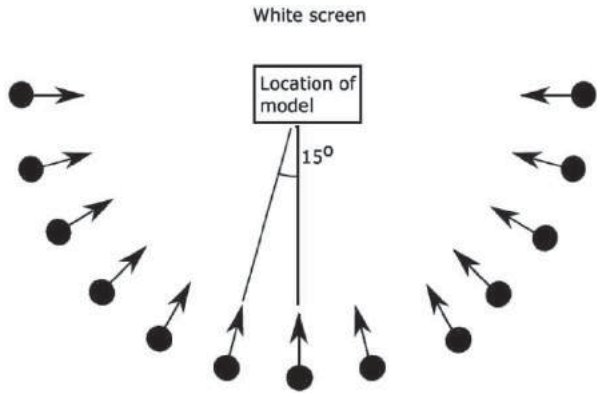

Figure 4. 2D schematic showing camera positions with angler translation for a complete set of photos at a single height.

A Canon 550d 18.9 MP digital SLR camera with a Canon EF 50 mm F/1.8 II prime lens is used in this work. The camera is mounted on a tripod. The total cost of the camera, lens and tripod is less than £500. The camera is operated in aperture priority mode with the aperture set to F/11, the ISO set to 200 and the shutter speed selected automatically. With the relatively low aperture size and low ISO, the resulting shutter speeds are long to compensate for the low exposure, so it is crucial that the camera is kept steady to avoid blurring. Using a two-second delay between pressing the camera trigger and taking the photo allows the camera to stabilise and minimise blur. Images are saved in JPEG format. With prior care and consideration, the thirty-nine photos required for reconstruction of this model can easily be captured within an hour.

3.2. Software and color segmentation

An academic version of Agisoft Photoscan software [4] is selected to generate the point clouds from the photographs, whilst point cloud cleaning and generation of triangular meshes is performed by Geomagic Studio [8]. Feature segmentation by color is implemented in the Python language within the Geomagic scripting environment.

The aim of the color segmentation is to subdivide the triangular mesh into selections of connected triangles of the same color (within a given tolerance). Practically, two triangles of the same color are considered to be connected if a path is traced between them consisting of triangles with the same color within the tolerance. In this way different areas of the same color are separated into different selections (for instance, each tyre of a car is a separate selection despite being the same color and part of the

Figure 5. Photos of the model taken with different angle translations with camera at constant height and center of model approximately centered within the frame.

Figure 6. Photos of the model taken for different height translations with camera approximately focused on the center of the model.

same mesh). By operating on the triangular mesh rather than the point cloud, connectivity between triangles is stored intrinsically through vertices shared between triangles, eliminating the need for a nearest-neighbor-type search as would be required if using point clouds [13].

The color segmentation algorithm is implemented via the Python language and uses Geomagic libraries (app.v2), which interact with the software GUI system. The libraries utilise the object-orientated programming methodology of Python, with triangles, points, meshes and models, etc. stored as objects within their appropriate classes.

In Geomagic scripting, each triangle and each vertex of the mesh is given a unique ID. Color data is stored at the vertices with each vertex having R, G and B values. The color value of each triangle drawn to the screen is the mean of the values of its vertices. Selections of triangles are treated as objects containing a reference to the mesh the selection is located on and a list of the indices of the triangles in that selection. Triangles are selected in the GUI window using the selection tools then appended to a current or new selection in the script.

Figure 7. The triangle selection (pink triangles) as an initial input of the color segmentation algorithm.

In the present work, triangles on the mesh are selected in the Geomagic GUI window located on the region to be segmented. This creates the initial triangle selection S_1. A typical triangle selection for S_1 (pink triangles) is shown in Fig. 7 within the white area of the mesh. This triangle selection is the initial input of the color segmentation algorithm.

The triangles selected in S_1 should be located on the feature to be considered and have colors typical of that feature. They are not required to be connected. The mean value of R, G and B is calculated for all vertices of triangles contained in S_1 and this is considered to be the RGB value of the selection. An RGB range is calculated by defining a lower limit of each color component by multiplying the mean RGB values by a number between 0 and 1 and the upper limit of each color component by multiplying the mean RGB values by a number greater than 1 (typical values are 0.9 and 1.1). These multiplication factors are defined by the user and may be different for each color channel depending on the specific problem being considered.

The segmentation algorithm is split into two main parts (see Fig. 8). The first part locates all triangles in the selected mesh with vertex R, G and B values within the RGB range and appends them to a selection S_2. This is shown schematically in Fig. 8(a). In contrast, the second part of the algorithm checks for triangles in S_2 (and hence

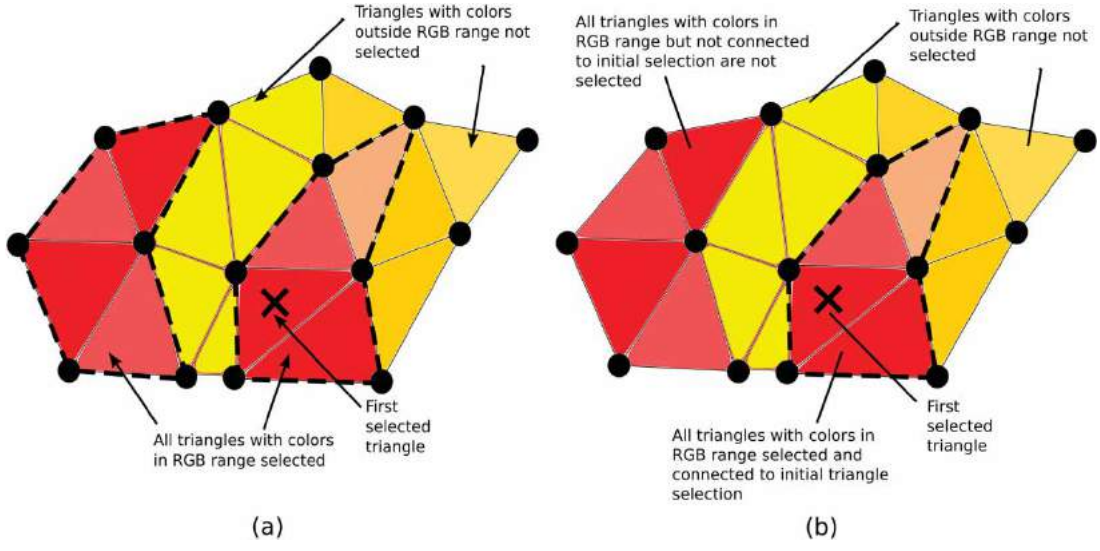

Figure 8. Segmentation algorithm schematic: (a) all triangles within the RGB range are selected; (b) only triangles connected to other triangles in the RGB range are selected from the initial selection (S_1).

in the RGB range) that are connected to an arbitrarily selected triangle t_1 in S_1 (as shown in Fig. 8(b)). These connected triangles are bounded by triangles not in S_2.

The second part of the algorithm is recursive, with triangles in the RGB range attached to triangles that have been appended to the final triangle selection added at each iteration. New selections S_3 and S_t are created, with S_3 being the final selection of connected color-matched triangles, and S_t a temporary selection, which will contain only triangles appended to S_3 in the previous iteration. Initially, a single triangle arbitrarily selected from S_1 is stored in S_t. All triangles that share a vertex with triangles in S_t are checked if they are also a member of S_2. If they are, then they are appended to S_3. The triangles stored in S_t are removed and replaced with those triangles that have just been added to S_3. If S_t is empty after a number of iterations, then no further triangles are appended to the final selection. Thus, all triangles that are connected to t_1 within the RGB range have been found and the routine ends. A flow diagram of the algorithm is shown in Fig. 9.

The algorithm is split into two separate parts, rather than checking whether a triangle is in the RGB range and connected at the same time, owing to the creation and use of a vertex-triangle list in the connectivity check, which lists triangles connected to each vertex. The list is time consuming to search; constructing and searching within a smaller list containing only triangles in S_2 is more computationally efficient.

A schematic of how the triangles are appended to selection S_3 is shown in Fig. 10. The selected region grows from t_1 until it reaches mesh members that do not correspond to the RGB criteria. The final selection grows in

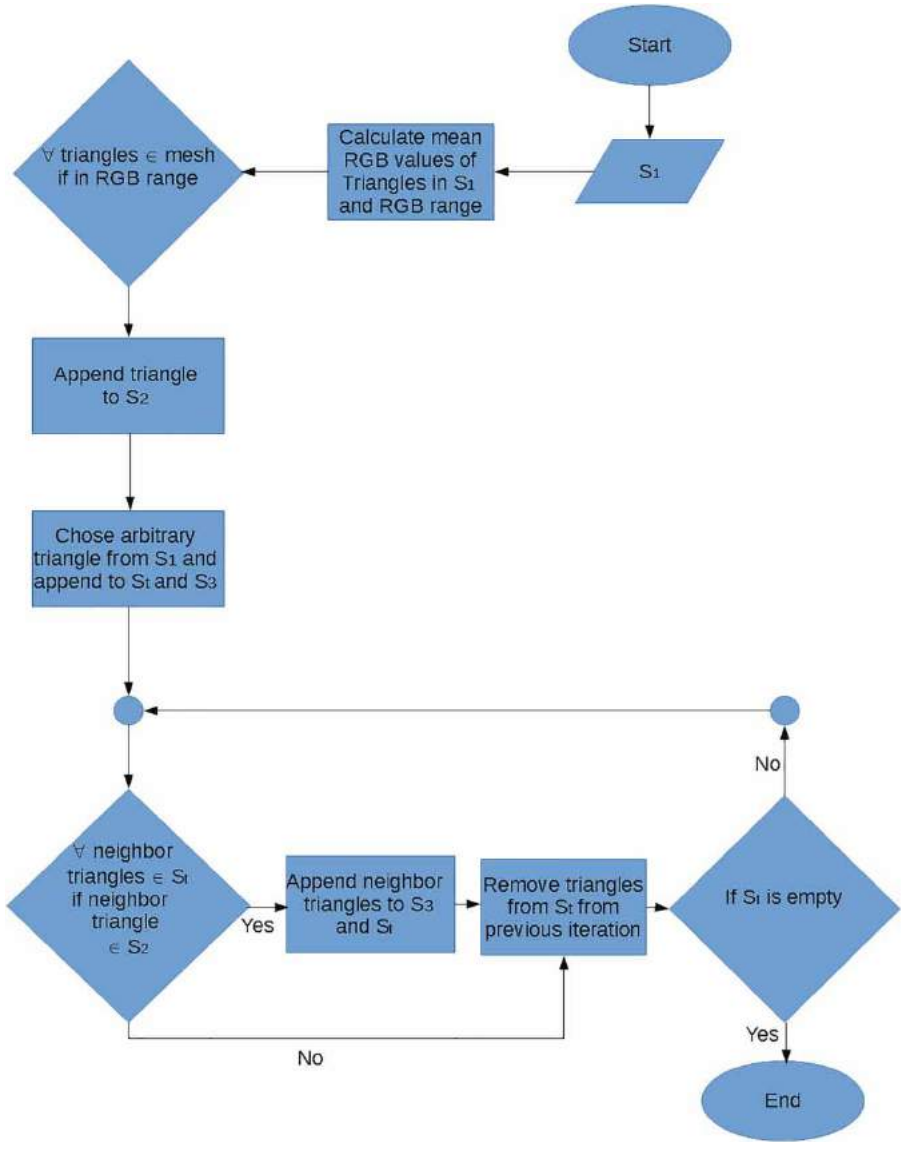

Figure 9. Color segmentation algorithm.

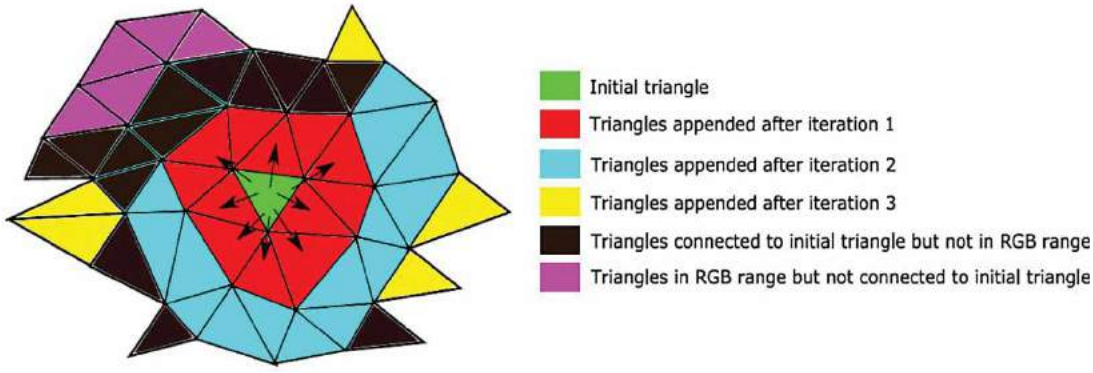

Figure 10. Color segmentation algorithm schematic; triangles selection appended for each iteration.

all directions, adding a layer of triangles at each iteration until a triangle outside the RGB range is reached.

4. Photogrammetry – practical considerations

There are guides and documentation available on the best practice for taking photographs to be used in the photogrammetry process [11]. In this section, the main steps of the photogrammetry computer program are introduced, including common issues related to constructing accurate and representative 3D point cloud data in the photography stage. Practical examples highlighting the issues and problems observed are presented, and ways to overcome these shortcomings are discussed. From a high-level overview, the main steps of a photogrammetric computer program are:

Feature recognition – Each photograph is analysed and key features are identified. Generally, variations of the SIFT algorithm developed by Lowe [14, 15] are considered. The SIFT algorithm is designed to have rotation, translation, illumination and viewpoint invariance, so a feature is identified as the same with good accuracy from different photos taken, independent of the distance and angle from the object.

Key feature matching – Key features have to be matched between photos. Key features are given a mathematical descriptor dependent on the gradient of the surrounding pixels. The design of the algorithm means that the descriptors given to different key points on the object allow them to be distinguished from each other. The descriptors are transformation invariant, thus, key points corresponding to the same feature in different photographs will have similar descriptor values, allowing them to be identified and grouped together

Alignment of cameras – The coordinates of the camera locations relative to each other and the recognised features are calculated by minimising the error between distances on images and expected distances for all camera positions for the grouped key points. This minimisation is collectively known as bundle adjustment [2].

Construction of dense point cloud – Once the camera locations are aligned and the distances between key features are known, the dense point cloud is constructed. This is the most computationally intensive step in the photogrammetry process.

Table 1 shows common problems and issues that are encountered with photogrammetric subjects and the photography environment, and gives a brief description

Table 1. Common issues with photogrammetric subjects and environment.

Problem	Symptoms	Solutions
Shiny surfaces or transparent surfaces	Reflected objects may be identified as key points, but will move relative to the object to be reconstructed with camera position	Use of talcum powder to make surfaces matt
Camera shake	Causes blurring so that features are not distinct and cannot be detected via the key feature detection	Use of a tripod / Use delay shutter to avoid camera shaking by user
Poor depth of field	Parts of the object are out of focus so cannot be detected by the feature detection	Fix aperture and ISO / Use of prime lens
Specular reflections	Specular reflections caused by spot light sources may obscure or be identified as key features. These will move with camera position, causing issues to camera alignment	Take photos in diffuse lighting environment
Background features	Program may place more emphasis on key features matching within the background features, affecting the minimisation algorithm	Use white screens to make a plain background/Mask background in Photoscan
Features not seen in many photographs	If photos do not contain enough overlapping areas, key features may not be available in enough photos to allow effective minimisation of camera positions	Take at 15^{o} increments ensuring that a feature is present in more photographs

of how they affect the photogrammetry program and the basic steps followed in this work to minimise the effects.

An example of bad practice in the photogrammetry process is presented and described below. Fig. 11(a) shows an image from a set that produced poor results, including failure to align the cameras, while Fig. 11(b) is an image from a set that produced good results.

In Fig. 11(a), the surfaces of the model are shiny, resulting in many reflections. The environment where the picture is taken is exposed to spot light sources. Moreover, parts of the model are out of focus. A partial remedy to these photogrammetry drawbacks is the use of talcum so that the surfaces are matt, as shown in Fig. 11(b). Note that the background is plain and the entire model is in focus.

Minimising shiny surfaces and avoiding specular reflections has the largest effect on the photogrammetry results[11]. Reflections in the surfaces will move relative to the object in each photo. They are particularly prevalent features and are identified by the feature recognition algorithm, so they might be matched during bundle adjustment, resulting in false key-point matching.

In our procedure, the shininess and transparency of the models' windows in particular contributed to poor results. It was found that taking photos in areas where there is a constant light source and no spotlights, such as light fittings, helped improve the quality of the generated point clouds. Coating the car with a matt powder provided the biggest improvement.

Fig. 12(a) and Fig. 12(b) illustrate images of the point clouds and the triangular meshes generated from the point clouds from two sets of photos of the Gran Torino model, without and with the model being coated with powder, respectively. From the photos without coating, holes and mismatched regions occur on the body panels, and more predominantly in the windows. Interestingly,

Figure 11. Problems with photogrammetric process, (a) a poor image (b) resulting good image.

Figure 12. Point clouds triangular meshes generated from images of model with and without matt coating demonstrating improvement in quality after coating application: (a) point cloud without matt coating; (b) point cloud with matt coating; (c) triangular mesh without matt coating; and (d) triangular mesh with matt coating.

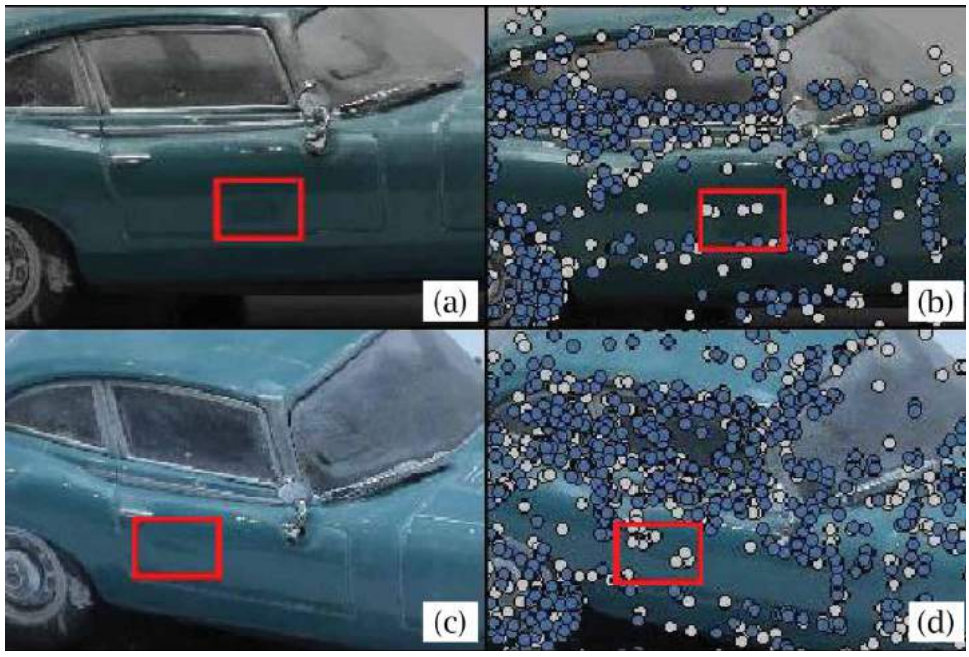

Figure 13. Movement of a false feature, recognised owing to shininess; (a) Reflection of the wing mirror in the door in the first photo; (b) key points locations of the wing mirror reflection; (c) reflection of the wing mirror in the door in the second photo after translation of the camera position; and (d) key points locations of the translated wing mirror reflection, despite the translation of position relative to the car.

for the powdered model, these problems are almost completely eliminated and reconstruction of the body panels and widows is enhanced.

Another common problem with the photogrammetry process is the movement of false key features within the model, as illustrated in Fig. 13, which shows the model's wing mirror reflected in the door in two photos from different locations when the powder coating the model has been rubbed off from the door. Clearly, the reflection of the model's wing mirror in the door is recognised in both images; however, this feature has shifted relative to the rest of the model.

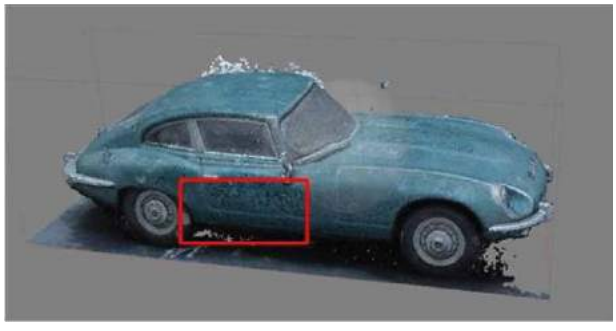

Figure 14. Poor point cloud reconstruction of the door due to reflections.

The resulting point cloud is shown in Fig. 14. The dense point cloud is a good representation for most of the model except for the door, which is badly reconstructed, demonstrating the importance of minimising surface reflections.

5. Results and discussion

In this section the utilisation of Photoscan and Geomagic Studio software to generate point clouds and triangular meshes is discussed, along with the implementation and effectiveness of the color segmentation algorithm.

5.1. Creation of point clouds and triangular meshes

Point clouds are created with Photoscan software from the photographs. The camera is calibrated using Agisoft lens software, which is packaged with Photoscan. Initially, backgrounds of photos are masked, as shown in Fig. 15, so that key features identified on background objects are not used in the camera location minimisation scheme.

Bundle adjustment is performed using the align photos option (described above), under standard settings. The align photos setting builds a sparse point cloud based on the tie points, which are the locations of key features that have been matched on two or more images. The

Figure 15. Masking of the background.

result of this procedure showing a sparse point cloud and the calculated positions of the cameras is presented in Fig. 16.

The dense point cloud is created using the build dense cloud option, with quality set to medium, producing a point cloud with approximately 2 million points for the models considered in the present work. Depth filtering, which removes points regarded as outliers, is set to mild. Complete dense clouds are shown in Fig. 17.

Smooth triangular meshes are created in Geomagic Studio software (see Fig. 18). These are filtered to remove outliers and small holes are filled. The final triangle count is set to around 200 000 triangles.

Figure 16. Bundle adjustment and creation of a sparse point cloud.

Figure 17. Final dense point clouds on: (a) the E-type model; (b) the Gran Torino model.

Figure 18. Triangular mesh created for: (a) the E-type model, (b) the Gran Torino model.

5.2. Color segmentation

The color segmentation algorithm is applied on the models to separate the body shell of the cars from other parts.

5.2.1. Color segmentation applied to the Gran Torino model

This novel procedure is particularly effective on the Gran Torino model, taking advantage of the distinct differences in color of the car body shell and other parts. The limits on the RGB values could be set quite far apart because the blue and green values are low compared with the red value for the body shell. For instance, at a randomly selected vertex near the center of the car roof, the RGB values are set to 163, 82, and 89, respectively. Other parts, such as the windows, chrome parts and wheels, have their red, green and blue channels at similar levels. At a randomly selected vertex in the center of the windscreen the RGB values are 153, 156 and 143, and at a randomly selected vertex near the center of the rear bumper they are 98, 101 and 106. Variation in the RGB values across each feature occurs for two reasons: firstly, the camera position varies for each photograph, which may result in different exposure levels and therefore affect the brightness. This is reflected in a shift in all three RGB values. Secondly, there is physical color variation within the model. Correspondingly, a tolerance on matching RGB values is required. The lower tolerance was set to 0.95 and the upper tolerance was set to 1.05 for all three values.

To demonstrate this methodology, practical examples of the color segmentation routine are shown in Fig. 19 and Fig. 20, which illustrate the removal of the side window and the removal of the trim at the bottom of the doors, respectively, from the Gran Torino model. Initially, a few sample points are selected, as in Fig. 19(a). The color segmentation routine is run and most of the triangles located on the trim are then automatically selected (those that are not are outside the RGB range); see Fig. 19(b) and Fig. 20(b). In both cases, the selected triangles are bounded at the edges of the trim by the red triangles of the car body. Fig. 19(c) shows that after removal of the triangles selected by color the edges of the window frame are well defined. Likewise Fig. 20(c) shows that, once the selected triangles have been removed, the bottom edges of the door frame are also well defined. Clearly the routine is effective owing to the difference in RGB values, although some isolated triangles still need to be removed using manual selection tools, as shown in Fig. 19(d). The resulting boundary between the car body and trim is accurate and clearly defined.

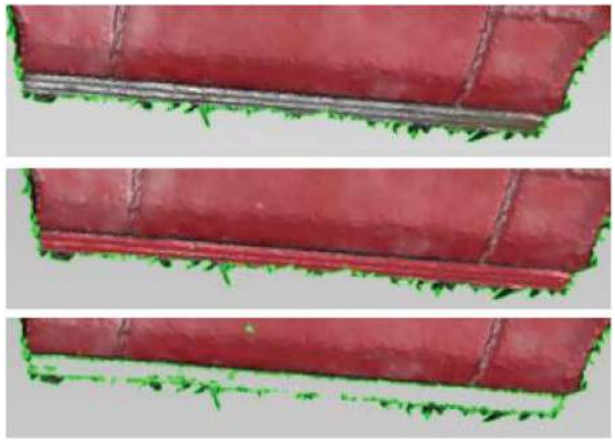

Figure 20. Removing the trim from the bottom of the car: (a) select triangles on the lower trim; (b) apply color segmentation to select all connected triangles within RGB limits; and (c) remove selected triangles.

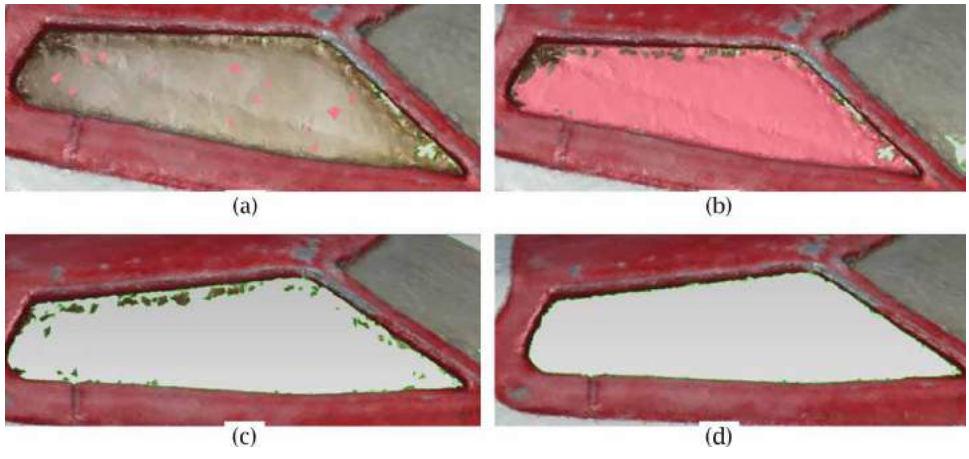

Figure 19. Color segmentation procedure for part removal. (a) Select initial triangles (pink) in window region, (b) automatically select all connected triangles within the chosen RGB limits, (c) remove selected triangles, (d) remove individual triangles missed during color selection.

Once the color segmentation algorithm is applied and specific regions are removed or separated, Geomagic's boundary modification tool is introduced to clean and straighten the new boundaries that resulted from the segmentation process, as shown in Fig. 21.

The resulting CAD surface model created in Geomagic from the above mesh is shown in Fig. 22.

Figure 21. Cleaned triangular mesh of the car body with other parts removed by color segmentation.

Figure 22. 3D CAD surfaces of body panels generated from the triangular mesh in Fig. 21.

5.2.2. Color segmentation applied to the E-type model

The color segmentation routine was less effective on the E-type model, because the colors of the body shell have RGB values close to those in other features, such as the windows. This demonstrates that for color segmentation to be effective, the features' colors should have suitably different RGB values (the shortcomings of this operation are overcome in the next section, 5.3). However, in another area within this model, the wheels and wheel arches were joined by a region of black triangles owing to cast shadows. Color segmentation proved effective in removing these areas from inside the wheel arches, as shown in Fig. 23. The final triangular mesh with wheels removed is shown in Fig. 24.

Figure 24. Cleaned triangular mesh of the E-type body with wheels removed using color segmentation.

5.3. Marking boundaries

The ineffectiveness of the color segmentation algorithm on the E-type model due to the RGB values of the model's features being too similar represents a drawback to the proposed technique. It will often be the case that different features of an object will have similar colors. An intuitive solution to this problem is the manual creation of boundaries on the object prior to photographing. The color segmentation algorithm only requires a feature to be bounded entirely by triangles with RGB values outside the limits to be effectively segmented. In these circumstances, various practical approaches are

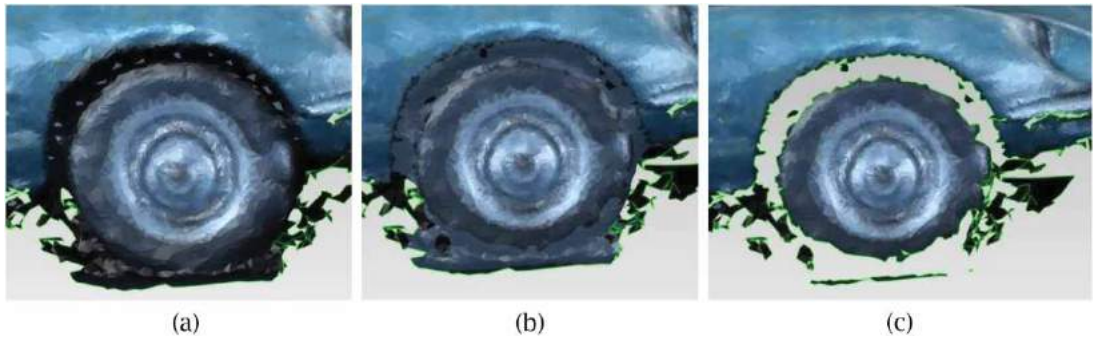

Figure 23. Successive steps to remove the shadow between the wheel and wheel arch for the E-type model: (a) Select triangles in area to remove, (b) Run color selection routine to select connected and color-matched triangles, (c) Delete selected triangles.

employed to create the boundaries, such as colored tapes or paints. In fact, marking boundaries on the object could also help to easily segment features with multiple colors, provided that the boundaries have a different color to all colors in the features. This will allow the user to manually select all triangles falling within the boundary for further manipulations.

To demonstrate the effectiveness of this intuitive approach, a further set of photos were taken with the two models marked with black lines (using a marker pen) dividing them into different features, as shown in Fig. 25. With this pre-marking of the boundaries, features of the models with the same color, such as body panels or doors, can be segmented easily.

For example, Fig. 26 demonstrates the steps followed in the removal of the rear window of the E-type using the marked boundary to isolate the window from the frame. Fig. 26(a) shows the selection of initial triangles on the black-marked boundary, whilst Fig. 26(b) depicts the application of the color segmentation algorithm and the removal of selected triangles in the boundary region. Finally, Fig. 26(c) illustrates the removal of the rest of the window, and the use of Geomagic Studio to tidy the new edge of the mesh.

Figure 25. Images of the models with feature boundaries marked manually.

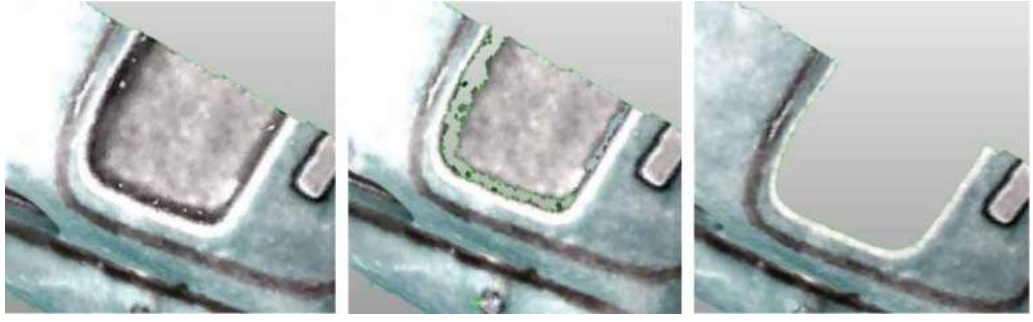

Figure 26. Selection and removal of the E-type rear window via color segmentation using marked boundaries: (a) the selection of initial triangles on the marked boundary; (b) application of color segmentation and the removal of triangles on the boundary; (c) removal of the window and tidying of the new edge.

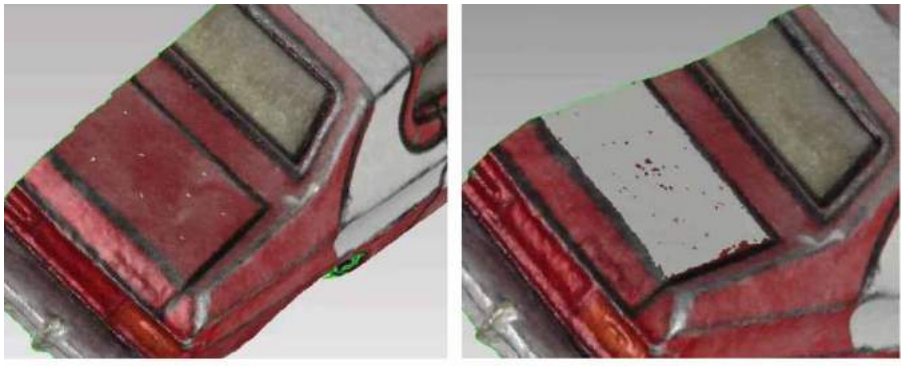

Figure 27. (a) The selection of red triangles in the black-bounded region, which was marked on the model prior to photographing (on top of the Gran Torino boot); (b) selected triangles after running the color segmentation algorithm.

Similarly, Fig. 27 demonstrates that using marked boundaries is effective in segmenting parts of the same color provided the boundaries around them are complete. In this case, the top of the boot is selected. It is worth mentioning that in cases where there are small gaps in the boundary (triangles with no boundary color, for physical or computational reasons), the RGB values of these triangles could be modified to fill the gaps using software such as MeshLab (meshlab.sourceforge.net).

6. Conclusions

In this work, the photogrammetry procedure is demonstrated as a relatively cheap and cost-effective method of constructing 3D point clouds for reverse engineering applications, leading to generating CAD ready models that could be utilised in FEA simulations or 3D printing. A practical demonstration is conducted to show that, with the affordable commercial software Photoscan effective point clouds are generated easily, quickly and accurately using standard, low-cost photography equipment.

Innovative to this approach, feature segmentation is achieved based on the color information captured from the photographs. This is an advantage over most current 3D scanners, which only capture spatial data, leaving the user with the difficult task of performing the feature segmentation. Some newer 3D scanner models are able to capture color information and the techniques developed here are equally applicable to point clouds produced by these scanners.

Additionally, we have applied the color segmentation to triangular meshes, making use of the fact that vertices are shared by a number of triangles. The aim of the color segmentation technique presented is not to completely automate the time-consuming process of feature segmentation but rather to be a tool available to the user to improve the efficiency of accurately segmenting a mesh produced by photogrammetry. For this, an example of an algorithm used for color segmentation is demonstrated on model cars with a scope for further process optimisation. The present feature segmentation approach is enhanced by marking boundaries on the object prior to photogrammetry taking place, enabling the procedure to be more effective, even with model parts of the same color.

Acknowledgement

The authors acknowledged the European Regional Development Fund (ERDF) through the Welsh Government financial contribution to undertake this study.

ORCiD

David W. James http://orcid.org/0000-0003-3951-9187
Fawzi Belblidia http://orcid.org/0000-0002-8170-0468
Jurgen E. Eckermann http://orcid.org/0000-0001-6961-0972
Johann Sienz http://orcid.org/0000-0003-3136-5718

References

[1] 123D-Catch, www.123dapp.com/catch, Autodesk software.
[2] Agarwal, S.; Snavely, N.; Seitz, S.; Szeliski, R.: Bundle Adjustment in the Large, in Computer Vision – ECCV 2010, Daniilidis, K.; Maragos, P.; Paragios, N., Editors, Springer Berlin Heidelberg. 2010, 29–42. http://dx.doi.org/10.1007/978-3-642-15552-9_3
[3] Agathos, A.; Pratikakis, I.; Perantonis, S.; Sapidis, N.; Azariadis, P.: 3D Mesh Segmentation Methodologies for CAD applications, Computer-Aided Design and Applications, 4(6), 2007, 827–841. http://dx.doi.org/10.1080/16864360.2007.10738515
[4] Agisoft-PhotoScan, www.agisoft.com, Agisoft Software.
[5] Bay, H.; Tuytelaars, T.; Van Gool, L.: SURF: Speeded Up Robust Features, in Computer Vision – ECCV 2006, Leonardis, A.; Bischof, H.; Pinz, A., Editors, Springer Berlin Heidelberg. 2006, 404–417. http://dx.doi.org/10.1007/11744023_32
[6] Bhatla, A.; Choe, S.Y.; Fierro, O.; Leite, F.: Evaluation of accuracy of as-built 3D modeling from photos taken by handheld digital cameras, Automation in Construction, 28, 2012, 116–127. http://www.sciencedirect.com/science/article/pii/S092658051200115X
[7] Chang, K.-H.; Chen, C.: 3D Shape Engineering and Design Parameterization, Computer-Aided Design and Applications, 8(5), 2011, 681–692. http://www.tandfonline.com/doi/abs/10.3722/cadaps.2011.681-692
[8] Geomagic, www.geomagic.com/en, Geomagic Software.
[9] Golparvar-Fard, M.; Bohn, J.; Teizer, J.; Savarese, S.; Peña-Mora, F.: Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques, Automation in Construction, 20(8), 2011, 1143–1155. http://www.sciencedirect.com/science/article/pii/S0926580511000707
[10] González-Jorge, H.; Riveiro, B.; Arias, P.; Armesto, J.: Photogrammetry and laser scanner technology applied to length measurements in car testing laboratories, Measurement, 45(3), 2012, 354–363. http://www.sciencedirect.com/science/article/pii/S0263224111004179
[11] Granshaw, S.I.: Close Range Photogrammetry: Principles, Methods And Applications, The Photogrammetric Record, 25(130), 2010, 203–204. http://dx.doi.org/10.1111/j.1477-9730.2010.00574_1.x
[12] Jiang, R.; Jáuregui, D.V.; White, K.R.: Close-range photogrammetry applications in bridge measurement: Literature review, Measurement, 41(8), 2008, 823–834. http://www.sciencedirect.com/science/article/pii/S0263224108000031
[13] Lai, K.; Bo, L.; Ren, X.; Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset, in Robotics and Automation (ICRA), 2011 IEEE International Conference, IEEE. 2011, 1817–1824.

[14] Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 60(2), 2004, 91–110. http://dx.doi.org/10.1023/B3AVISI.0000029664.99615.94
[15] Lowe, D.G.: Object recognition from local scale-invariant features, in Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference 1999, 1150–1157 vol.2.
[16] Luhmann, T.: Close range photogrammetry for industrial applications, ISPRS Journal of Photogrammetry and Remote Sensing, 65(6), 2010, 558–569. http://www.sciencedirect.com/science/article/pii/S0924271610000584
[17] Masuda, H.; Tanaka, I.: Extraction of Surface Primitives from Noisy Large-Scale Point-Clouds, Computer-Aided Design and Applications, 6(3), 2009, 387–398. http://www.tandfonline.com/doi/abs/10.3722/cadaps.2009.387-398
[18] Photomodeler, www.photomodeler.com/products/modeler/default.html, Eos Systems Inc.
[19] Rabbani, T.; Van Den Heuvel, F.: Efficient hough transform for automatic detection of cylinders in point clouds, ISPRS WG III/3, III/4, 3, 2005, 60–65.
[20] Rabbani, T.; van den Heuvel, F.; Vosselmann, G.: Segmentation of point clouds using smoothness constraint, International archives of photogrammetry, remote sensing and spatial information sciences, 36(5), 2006, 248–253.
[21] RapidForm, www.rapidform.com, 3DSystems Software.
[22] Varady, T.: Automatic Procedures to Create CAD Models from Measured Data, Computer-Aided Design and Applications, 5(5), 2008, 577–588. http://www.tandfonline.com/doi/abs/10.3722/cadaps.2008.577-588
[23] Vàrady, T.; Benkó, P.; Kós, G.: Reverse Engineering Regular Objects: Simple Segmentation and Surface Fitting Procedures, International Journal of Shape Modeling, 4(127), 1998, 127–141. http://dx.doi.org/10.1142/S0218654398000106
[24] Várady, T.; Martin, R.R.; Cox, J.: Reverse engineering of geometric models—an introduction, Computer-Aided Design, 29(4), 1997, 255–268. http://www.sciencedirect.com/science/article/pii/S0010448596000541
[25] Vosselman, G.; Gorte, B.G.H.; Sithole, G.; Rabbani, T.: Recognising structure in laser scanner point clouds, International archives of photogrammetry, remote sensing and spatial information sciences, 46(8), 2004, 33–38.
[26] Zhan, Q.; Liang, Y.; Y., X.: Color-based segmentation of point clouds, in Laser Scanning 2009, Bretar, F.; Pierrot-Deseilligny, M.; Vosselman, G., Editors, Paris, France. 2009, 248–252.