

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



3D Surface Analysis for Automated Detection of Deformations on Automotive Body Panels

Arjun Yogeswaran and Pierre Payeur

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/45790>

1. Introduction

1.1. Context and motivation

Quality control in the manufacturing industry has traditionally been performed manually by workers. As manufacturing increases in speed and volume through the introduction of automation, the human worker becomes a limiting factor in speed, accuracy, and consistency. In the automotive industry, quality control is critical to ensure that automotive body parts meet predefined standards. Identifying deformations, such as undesired dings and dents on panels, and marking them so that they are repaired while still on the assembly line is essential. In current industrial settings, the procedure for identifying surface defects on automotive body panels often requires a laborious manual surface rubbing operation. This time-consuming process is difficult for a human, especially when dealing with small deformations that require close inspection, and may result in a decreased accuracy when the repetitive task is performed over the course of an entire work shift. Automation of quality control could significantly improve the accuracy and speed of the assembly line, thus increasing the number of panels inspected within an allotted time, maximizing the number of accurately detected defects, and minimizing the number of false detections.

To fully automate this process, a system would have to analyze the surface of the body part to be inspected, determine the position of deformations, and mark those deformations on the body part. This chapter focuses on the analysis of 3D surfaces and automatic detection of deformations. One important contribution of this work comes from the imposed requirement that this system must be able to detect deformations without knowledge of the ideal shape of the part, meaning it cannot use a master work or CAD model for comparison. Some automated deformation detection techniques focus on the difference between the scanned model and an existing ideal model or master work (Newman & Jain 1995; Lilienblum et al. 2000). Certain challenges lie within this approach. The first constraint is

that a very precise ideal model must be available, because small faults in the master work can result in erroneously detected defects during execution. The second challenge is related to the registration of the scanned part with the ideal model. Due to vibrations on the assembly line and slight inconsistencies in the acquired model, inaccurate registration may occur resulting in incorrectly detected defects. Also, if panels of different models are processed on the same assembly line, or a new piece is introduced to the system, significant calibration and set up is required to synchronize the master work with the acquired model. Because of these difficulties, this chapter introduces a more generic and robust technique which does not require an ideal model.

1.2. Objectives

This chapter deals primarily with the design of a deformation detection system. Its requirements are to identify deformations of interest over the surface of automotive body parts, with minimal human interaction and independently from the type of acquisition system used. The deformations of interest are dings and dents, where dings are surface deformations which protrude from the surface and dents are depressions into the surface. This chapter focuses on deformation detection when no ideal model of the automotive part is provided, similar to an approach which is alluded to by Döring *et al.* (Döring *et al.* 2004) and explored by Chen (Chen 2008). Since there is no CAD model of a master work to compare the measured model to, the deformation detection must be done without knowledge of the expected surface and requires certain assumptions to be made based on common characteristics of surface deformations compared to the characteristics of an undeformed surface. However, not all characteristics can be assumed, known, or easily defined. Therefore some basic parameters need to be set by the operator to provide the system with a minimal knowledge of the approximate size or scale of the deformations that the manufacturer wants to detect and eliminate from its products. This is not unrealistic, as the operator generally has a clear idea of the approximate range of sizes for the deformations to be detected.

Given these requirements, a system is proposed which analyzes the digital 3D model of an automotive part collected along the assembly line, determines the locations of only the deformations of interest, and classifies them as dings or dents. Areas of significant surface variation could be deformations. But other features of an automotive body panel such as aesthetic curves and door handles, or inaccurate surface measurements such as acquisition artifacts and noise, also represent surface curves that must not be falsely detected as dings or dents. The deformation detection system is comprised of a surface shape analysis phase to extract areas of interest, a segmentation phase to group areas containing pieces of deformations together into segments, and a classification phase to determine which segments contain deformations and which contain design features.

A deformation detection pipeline is proposed, which combines an enhanced octree-based feature extraction, with segmentation and classification to extract deformations from a 3D mesh of an automotive surface panel. This pipeline supports multi-resolution analysis of 3D

models, providing the capability of extracting deformations regardless of the resolution or scale of the model, and relies on intuitively adjustable parameters for the operator to target the feature extraction towards desired characteristics of the deformations.

2. Literature review

The process to automatically determine the location of a defect on an automotive body part requires several steps, some of which are still complex research topics. This section reviews important research that is relevant to the topic of automated deformation detection proposed in this chapter. In order to analyze the surface for automotive body parts, the latter must first be digitally represented as a 3-dimensional object. Various techniques in 3D acquisition are explored in section 2.1. To determine the location of deformations in the digitized 3-dimensional surface, the surface must be analyzed for certain characteristics. Surface shape analysis is discussed in section 2.2.

2.1. 3D acquisition

In order to analyze the surface of a real-world object in 3-dimensions, it must be scanned and converted into digital 3-dimensional data. Laser scanners are very common and highly accurate 3D acquisition tools (Parthasarathy et al. 1982; Sequeira et al. 1995; Marszalec & Myllyla 1997; Gokturk et al. 2004; Blais et al. 2007). The latter are able to produce high resolution, high accuracy scans. However they are usually expensive systems that take a long time to complete a full scan, and often require some mechanical system to move the laser and acquire readings before accumulation into a point cloud. For lower cost, lower scan times, and minimal mechanical complexity, stereoscopic vision systems are a very popular way of digitizing a 3-dimensional scene (Murray & Jennings 1997; Murray & Little 2000; Se et al. 2001). If prominent features are lacking in a scene, such as on the surface of a smooth automotive body part, these construction techniques may fail due to a lack of usable points. One popular technique to overcome the limitations of using traditional stereoscopic imaging is to acquire 3-dimensional models using structured light scanners. This type of sensor projects a set of artificial features onto a model or scene that is being scanned, and then uses a vision system to acquire the model in 3D. Most structured lighting systems use a single camera along with a projector to acquire the 3D points (Rocchini et al. 2001; Zhang et al. 2002), or combine a pattern projector with a standard stereo pair of cameras to avoid calibration with the projector (Payeur & Desjardins 2009).

A recent trend in 3D sensing is the use of Microsoft's Kinect, which is a low-cost portable sensor that provides 3D visualization of a scene. Using structured light principles, an infrared laser projector generates artificial features onto a scene (which are invisible to the human eye), and a CMOS sensor reconstructs a scene through vision techniques. High quality scene reconstruction using the Kinect sensor has been studied (Shahram et al. 2011; Yan & Didier 2011). Relatively high quality reconstruction of real-world scenes can be achieved, yet its accuracy is still too low to detect the slight variations in an automotive

panel surface that constitute deformations. Also, due to possible holes and inaccuracies in reconstruction using a single image, several frames of reconstructed scenes must be stitched together and heavily post-processed to provide a full reconstruction of an automotive part. Though preliminary work with the Kinect is promising, more research must be done to adapt its use to detecting the fine contours of an automotive panel for deformation detection. Given the current state of the technology, and the purposes of this work, the laser scanner remains to be the most accurate way to scan 3D models.

2.2. 3D surface shape analysis

The most critical component of a surface deformation detection system is surface analysis to locate the defects in question. In the current context, no ideal model of the automotive part is provided, therefore the algorithm has no *a priori* knowledge of what the surface should look like without deformations. Advanced surface shape analysis techniques must be performed to determine the locations of probable deformations.

Given that the 3-dimensional data can be converted from a range image to a 2-dimensional image where each pixel intensity represents the depth of that point on the object from the viewpoint, features can be extracted and images can be segmented using traditional 2D image processing techniques. Well-known edge detectors, such as the Sobel and Canny operators, can highlight the areas that belong to features (Faugeras 1993). The efficacy of such algorithms varies greatly, since determining the peaks and valleys in histograms with significant noise or varying characteristics is difficult.

The k-means algorithm is a very well-known clustering algorithm that partitions a dataset into a specified number, k , of clusters (Plataniotis & Venetsanopoulos 2000). However, selecting the value of k is most important, and in the case of an unknown number of deformations, this value cannot be known for sure. Unseeded region growing (Plataniotis & Venetsanopoulos 2000) can overcome some of the problems with k-means algorithms by not requiring any initial knowledge. Similar to the limitations of thresholding in edge detection, gradually changing pixel intensities between actual regions of the image may not be sufficient for accurate segmentation.

These techniques can all be extended to 3 dimensions by using points or voxels instead of pixels, and adjacency can be determined by distance or connectivity in a grid or tree, as is done by Palagyi and Kuba (Palagyi & Kuba 1999). Also, the data being used as the intensity value in an image can be redefined as distance in a range image or 3-dimensional surface deformation metrics such as standard deviation of normals or a curvedness value (Koenderink & Doorn 1992; Dorai & Jain 1997).

Various techniques from the field of 3D data analysis can be used for the purpose of deformation detection. Simple deformations in a mesh can resemble outliers on a smooth surface. Using noise removal techniques to identify areas of noise-like characteristics can be beneficial to determining the location of the defect. Schall *et al.* propose a noise removal

method that also provides applications in outlier removal (Schall et al. 2005). The statistical method estimates the density of each area of the point cloud, and uses the neighbouring points to adapt a probable surface to each point. Since points are moved to their most probable location along a surface, the spatial density of the resulting point cloud is relatively consistent throughout the surface of the object. An outlier point exists where the spatial density in a surrounding is too low. Therefore a basic threshold can determine outliers. Though effective, this algorithm is dependent on the sampling density, its parameter selection is not intuitive and may cause unpredictable performance, and will fail when deformations do not resemble outliers.

The moving least-squares surface reconstruction technique proposed by Mederos *et al.* (Mederos et al. 2003) uses a hierarchical segmentation technique that finds redundant points such that the point cloud density can be reduced before surface reconstruction. This segmentation technique results in clusters of points, where the surface variation within each cluster is minimal and the boundaries between those clusters could define a significant deformation on the surface. Though computationally expensive, analyzing the eigenvalues and eigenvectors of the covariance matrix of a cluster of points can estimate local surface properties (Hoppe et al. 1992; Shaffer & Garland 2001). A binary space partitioning tree is used to segment the model into clusters of points that lie on surfaces of low variation, where subdivision is based on the flatness criterion, which represents variation within a group of points, as described by Pauly *et al.* (Pauly et al. 2002). Such an algorithm is effective at determining the characteristics of a model for surface reconstruction or resampling, but requires an extension to be used for efficient feature extraction, since the boundaries must be determined instead of just the clusters. The use of a binary space partitioning tree is very effective to separate the mesh, but tends to become too deep of a tree to traverse efficiently, since each node can only be subdivided into 2 nodes at a time.

Woo *et al.* (Woo et al. 2002) introduce a technique based on octree structures, and use recursive subdivision of the volume of a 3D mesh to identify features. It removes segments of the mesh as the octree is generated, and leaves parts of the mesh that belong to features in the final octree data structure. It requires a model with a reconstructed surface and partitions the model into subsections which represent varying levels, or scales, of features. Surface normal vectors can be calculated for all triangles composing the surface, and ultimately for each point by averaging the normals of the triangles that the point belongs to. Variations in the orientation of the surface within a given region are estimated from the standard deviation of normal vectors within that region. This method facilitates the partitioning process. All of the points that make up the surface of the object are initially added to the root of the tree structure. The standard deviation of their normals is calculated, and compared to a threshold. First the mean normal is computed:

$$\bar{N} = \sum_{i=0}^n N_i / n \quad (1)$$

where n is the number of points at the node, \bar{N} is the mean normal, and N_i is the unit normal of point i . Then the standard deviation, σ , of the normals can be estimated as:

$$\sigma = \sqrt{\frac{\sum_{i=0}^n (N_i - \bar{N})^2}{n}} = \sqrt{\frac{\sum_{i=0}^n (x_i - \bar{x})^2 + \sum_{i=0}^n (y_i - \bar{y})^2 + \sum_{i=0}^n (z_i - \bar{z})^2}{n}} \quad (2)$$

A threshold must be defined for the subdivision as the maximum standard deviation allowed in a volume before it should be further divided. This threshold is defined by the user. If the standard deviation is larger than the threshold, the volume represented by the root is divided into 8 octants represented by 8 children being added to the root. The points from the root are redistributed based on their spatial location into each of the 8 octants, and thus into each of the 8 children of the root. This process is repeated recursively for each of the children, and for their children, and so on until either there are no children remaining, or a sufficient level of feature details is discovered. The depth of the tree determines how detailed the feature level is. Figure 1 details the recursive process of the feature segmentation at various scales. The final structure provides a tree where the points are distributed amongst the tree nodes. Leaves at greater depths represent finer detailed features of the mesh contained in smaller volumes. Leaves at lesser depths represent larger scale features contained in larger volumes.

Woo *et al.*'s technique is effective, yet because it uses a single threshold value throughout the entire tree, its ability to detect features can be unpredictable. A feature has to sufficiently affect the standard deviation of the surface normals across the selected volume for the method to investigate the mesh at a higher resolution. If this is not the case then the feature is not identified. The criteria for setting the threshold is that it must be high enough such that smooth curvatures and noise are not detected, but low enough such that the deformation features are detected. This remains a subjective criterion that varies with the point cloud. Since standard deviation is used, it is hard to find values which meet the defined criteria.

On the other hand, the technique generates broad shallow trees which are easier to traverse, as opposed to deep narrow trees generated by binary space partitioning methods such as those in (Shaffer & Garland 2001; Mederos *et al.* 2003). This allows analysis at higher resolutions, with reduced computational load. The octree representation allows features to be represented in the point cloud dataset as well as in a volumetric grid, giving the flexibility of using a variety of techniques for added segmentation.

Pauly *et al.* present a technique that allows feature extraction from a 3D object composed of surfaces, at various detail levels (Pauly *et al.* 2003). Weights are assigned to each point in the point cloud, representing the amount of local variation in the surface normals. At different scales, different local neighbourhood sizes are used. Introducing the idea of *feature persistence*, a threshold can be selected, such that local maxima weights over that threshold can be considered feature nodes. As a feature persistently exceeds that threshold, across multiple scales, it can be classified as a strong feature, rather than only a small local feature.

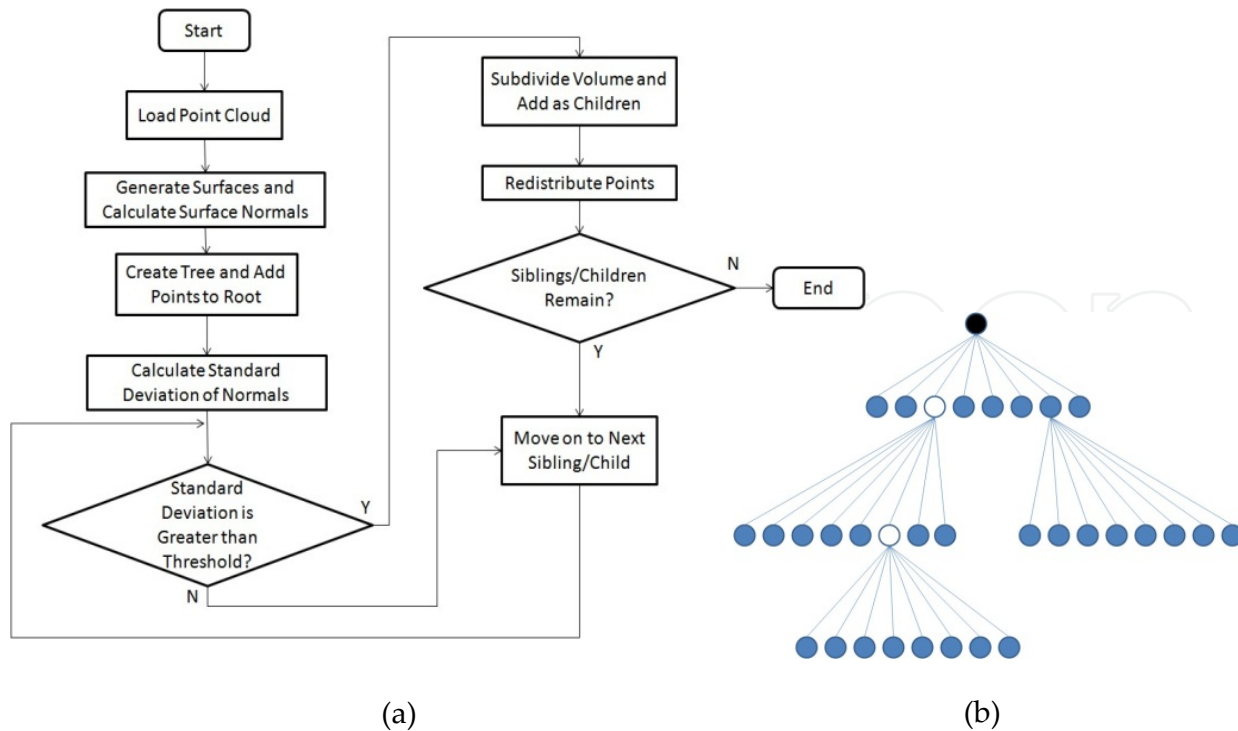


Figure 1. a) Octree segmentation subdividing flowchart, b) an example octree where the top node represents the entire mesh, and nodes at deeper levels represent subdivisions of the mesh at proportionally higher resolutions.

Results show that this method performs well under noise and is effective at identifying prominent features. Also, the idea of feature persistence is interesting, where prominent features appear over multiple scales, and can be very important in using multiresolution information to identify important features.

Vosselman *et al.* (Vosselman et al. 2004) exploit the knowledge of ordered point clouds in the form of scan lines, and combine various techniques to segment point clouds by recognizing geometric shapes and flat smooth surfaces for the analysis of industrial and city scans from LIDAR data. Each scan line is broken into line segments based on orientation and proximity, and a plane-of-best-fit equation is calculated. Adjacent scan lines are compared based on some similarity criterion to be connected as a planar surface or other shapes such as spheres and cylinders. The dependency on ordered point cloud data is a limitation of the technique, since data can come from various sources and may not always be in the form of scan lines. Also, the very distinct shapes that are being extracted are effective in scans of a city or in an industrial setting, but the techniques are less suited to the more curved and variable surfaces of automotive body parts, since such shapes do not fall into the category of basic geometric primitives.

Jagannathan and Miller (Jagannathan & Miller 2007) use a metric known as curvedness (Koenderink & Doorn 1992; Dorai & Jain 1997) for segmentation, to extract regions of the mesh with high curvature. The curvedness is calculated for each point in the mesh. Using iterative graph dilation and filtering of outlier curvedness values, the mesh is broken up into sub-meshes with similar curvedness values. Based on the results shown, the algorithm has

great success in segmenting 3D models with very large distinct form changes. However, it might be difficult to predict the success of this algorithm when faced with finding subtle shallow deformations on the surface of a flat or curved mesh, especially when dealing with significant noise and acquisition artifacts.

Döring *et al.* tackle a similar problem to this work, by detecting deformations on car body panels (Döring et al. 2004). The deformation extraction is only briefly explained as finding the differences between the point cloud and an inertial surface approximation of a low polynomial degree. The experiments in this paper work under similar assumptions to this chapter, in that there is no ideal model or *a priori* knowledge to compare to the model being analyzed. Surface deformations must be extracted by analysis of the model surface against what is assumed to be a smooth ideal surface instead of being compared to an existing model of what the surface should look like. This chapter is more concerned with the extraction of surface deformations than the classification, while Döring *et al.*'s work emphasizes the classification of the feature as one of many types of known deformations.

3. Automated surface deformation detection

3.1. General deformation detection framework

The proposed system takes a 3D mesh as an input, and outputs the sections of the mesh which are deformations of interest along with whether they are a ding or a dent. Given that no CAD model of the ideal surface is considered available, the proposed system must locate and classify the deformations of interest using assumptions based on common characteristics of dings and dents. Since some assumptions regarding size and scale of deformations cannot be made without more information, a minimal and intuitive set of parameters must be set by the operator to ensure accurate detection with minimal human intervention. This also ensures that design features of the automotive panel are not accidentally extracted as deformations, since they are generally much larger than the deformations of interest and can easily be separated by size and scale. The outputs of the proposed system are passed onto a robotic deformation marking system briefly discussed in section 3.2.

The proposed system contains 3 major components, as shown in Figure 2. The surface shape analysis component is tasked with dividing the 3D mesh into sections and analyzing each one for the magnitude of the deformation contained in that section. The segmentation component combines sections from the surface shape analysis which seemingly belong to the same deformation. The classification component classifies each segment from the segmentation as either a ding or dent, and removes segments which do not meet the criteria of being a deformation of interest, such as vehicle design features and acquisition noise.

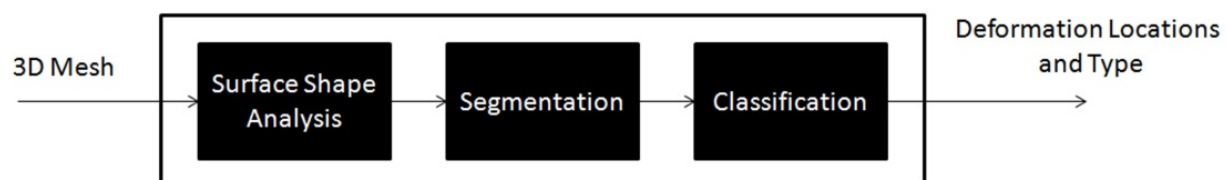


Figure 2. System diagram of proposed deformation detection system.

3.2. Experimental platform and setup

The more extensive research project that this work is part of involves the development of an automated deformation detection and marking system (Borsu et al. 2010). The primary objective is to identify deformations over an automotive panel and physically mark those deformations while the automotive part moves along an assembly line.

The 3D acquisition subsystem provides the deformation detection subsystem with 3D point cloud or mesh information. The deformation detection subsystem, which is the focus of this chapter, analyzes the surface of the automotive body panel and determines the locations and type of all deformations of interest. The robotic marking subsystem tracks the moving automotive panel along the assembly line, and marks the deformations with a robotic arm (Borsu 2010). The relationships between the subsystems are shown in Figure 3.

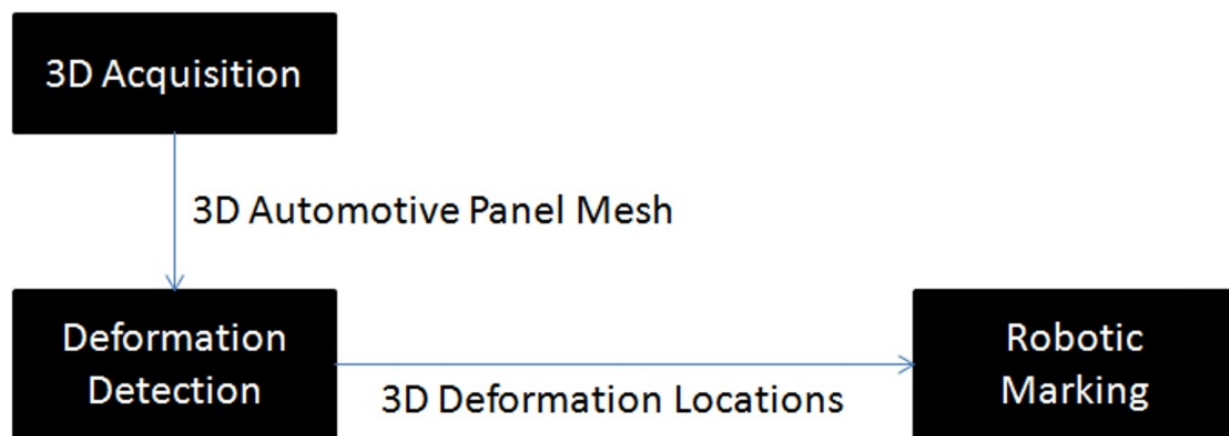


Figure 3. Relationship between subsystems for automated deformation detection and marking.

The automated deformation detection and marking system is created on a smaller scale in a lab setting. This serves as a test bed for the developed techniques, and demonstrates that they can work in a real-world setting. An image of the setup is shown in Figure 4.

To represent the idea of a moving assembly line, a PC-operated sled system is used and simulates a shortened conveyor in a lab setting. One of several real or imitation automotive panels is mounted on the sled system to imitate a real automotive panel. At the beginning of the assembly line, when the automotive panel is static, a structured light sensor is used to generate a dense 3D reconstruction of the surface of the automotive panel (Boyer 2009; Boyer et al. 2009). The deformation detection subsystem processes this 3D data, and acquires the location of the deformations. The panel continues moving along the sled system and is tracked by the robotic marking system (Borsu & Payeur 2009; Borsu 2010). Then, based on the locations automatically provided by the deformation detection subsystem, the robotic manipulator is positioned to smoothly mark deformations on the automotive panel surface.

3.3. Data sets

A 3D acquisition system provides the only input used by the deformation detection system to identify the location of deformations of interest. A detailed discussion of the 3D sensing

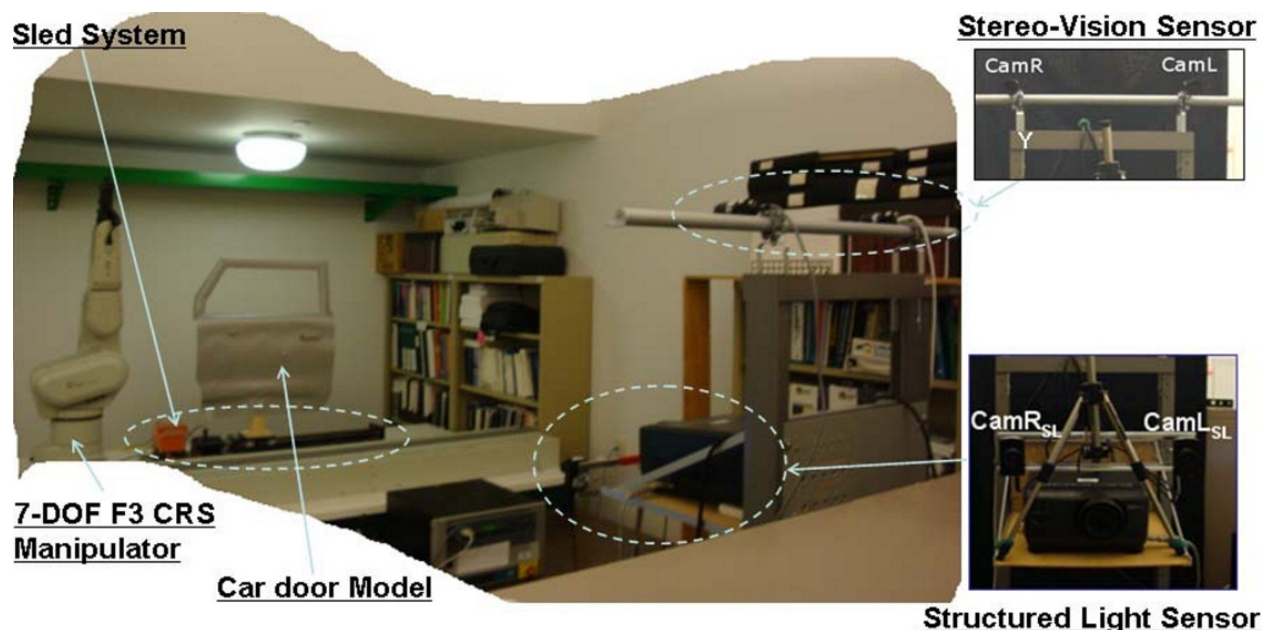


Figure 4. Experimental lab setup.

systems used for acquiring the shape of the automotive body panels is beyond the scope of this chapter. For laboratory evaluation, a custom structured light sensor (Boyer 2009; Boyer et al. 2009; Payeur & Desjardins 2009) is used in combination with slightly enlarged dings and dents deformations artificially affixed on the test panels. Alternatively, higher resolution datasets collected by industrial partners with an active laser range sensor on real automotive panels is also used to demonstrate the capability of adaptation of the proposed approach to different scales and its independence from the 3D acquisition system. Surface reconstruction, performed with the ball-pivoting algorithm proposed by Bernardini *et al.* (Bernardini et al. 1999), is used to generate a mesh triangulation out of the acquired 3D points. The output of this module is provided as input to the deformation detection subsystem, which is the starting point for the original work presented in this chapter.

Real-world test data is important to determine the effectiveness of the approach, since any 3D acquisition system does not provide ideal meshes for this application. The reflective characteristics of the surface, the subtle variations in its shape, and the large distance the panel is positioned from the sensor, cause the acquisition system to introduce an abundance of noise and acquisition artifacts. These real-world meshes serve as test cases for non-optimal acquisition and surface characteristics resulting from the acquisition errors.

The first real-world mesh is a desktop computer casing panel modified by hammering 3 dents into it. Though this is not an automotive part, it simulates real-world deformations on a relatively flat surface. The panel is 20cmx15cm and each dent is circular, with dimensions

of approximately 2cm in diameter and 0.25cm in depth. The computer casing panel was scanned at two different resolutions, with the high resolution version containing 14 626 points and the low resolution version containing 3647 points. Since it might be unrealistic to expect an accurate extraction of deformations from this mesh, a filtered version of the low resolution scan is created using a Laplacian smoothing filter to remove the noise while maintaining the deformations. The meshes are shown in Figure 5 with deformations circled. The amount of surface variation, holes along the boundaries, and noise are all visible in the images of the computer casing panel meshes.

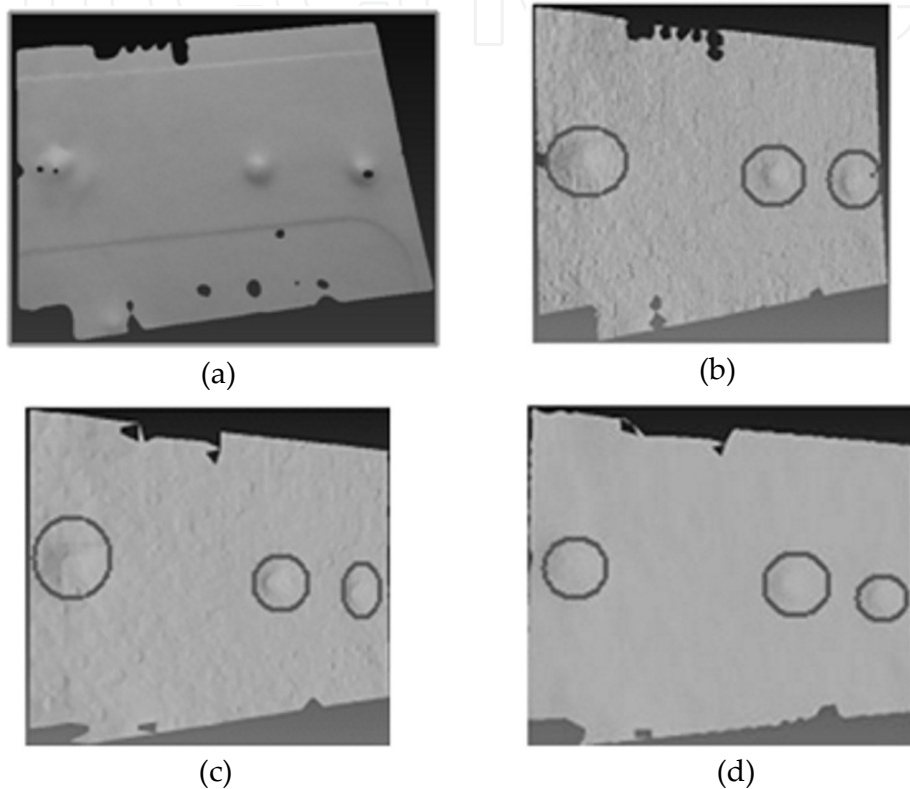


Figure 5. a) Indented computer casing panel, b) high resolution scan, c) low resolution scan, and d) filtered low resolution scan, with deformations circled.

A mock car door was crafted out of cardboard, consisting of a curved body, a door handle, and a window frame. The door is approximately 70cmx78cm. Three dings, made up of paper were stuck to the door at various positions, where each ding is circular and approximately 1cm in diameter and 1cm in depth. The scanned car door contains 32 202 points. A Laplacian filtered version is also used. The filtered and unfiltered versions are shown in Figure 6 with deformations circled.

The car door is acquired well, with the deformations, door handle, surface variation and window frame all appearing. There is lots of surface variation along the borders due to acquisition errors, and a significant amount of noise, which may interfere with isolating the deformations from parts of the noise as the peaks of the noise are almost as high as the peaks of the deformations. The filtered version reduces noise levels by minimizing the peaks and further separating them from the deformation peaks.

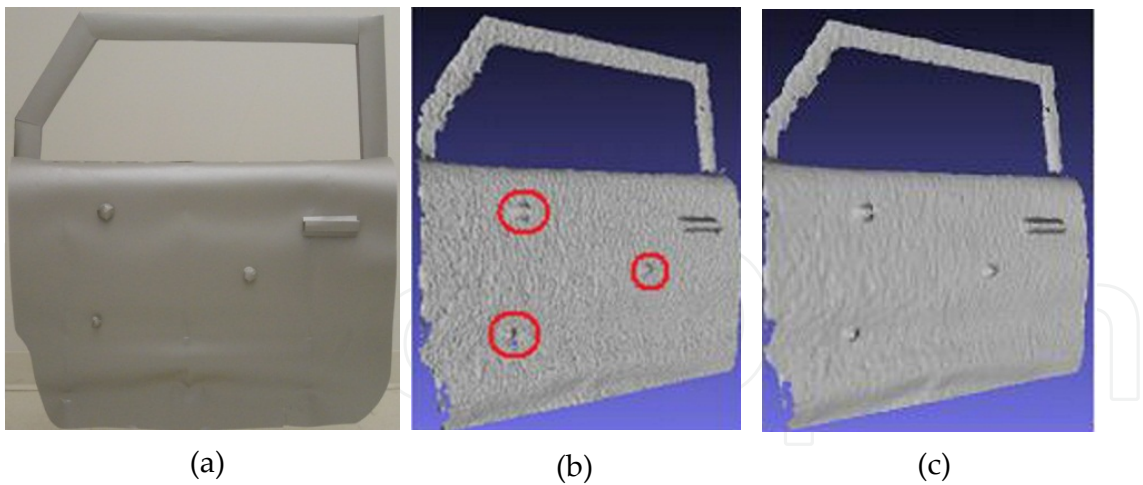


Figure 6. a) Car door sample, b) unfiltered mesh with deformations circled, and c) filtered mesh.

These real-world meshes are adequate to test the proposed system’s behavior when applied upon meshes with noise, acquisition artifacts, and real-world characteristics. However, it does not provide a comprehensive enough set of data to test the various situations that the system might be exposed to. For this reason, a set of artificial test meshes were generated, with characteristics that were not found in the acquired real-world meshes, but that may occur in other real-world meshes. These artificial meshes resemble deformations of interest, of various sizes and scale, under different surface conditions. These meshes also attempt to test the functionality of the system while working under ideal acquisition scenarios where there is no noise and no acquisition artifacts.

A flat mesh with a small dent was created as well as a flat mesh with a large ding, as shown in Figure 7. Similarly, a curved surface mesh with a small dent and a curved mesh with a large ding were created, as shown in Figure 8. The flat meshes are used to determine if the designed algorithms can detect small scale as well as large scale deformations. The curved meshes help determine if the designed algorithms can detect a deformation in spite of a curved or uneven surface around it.

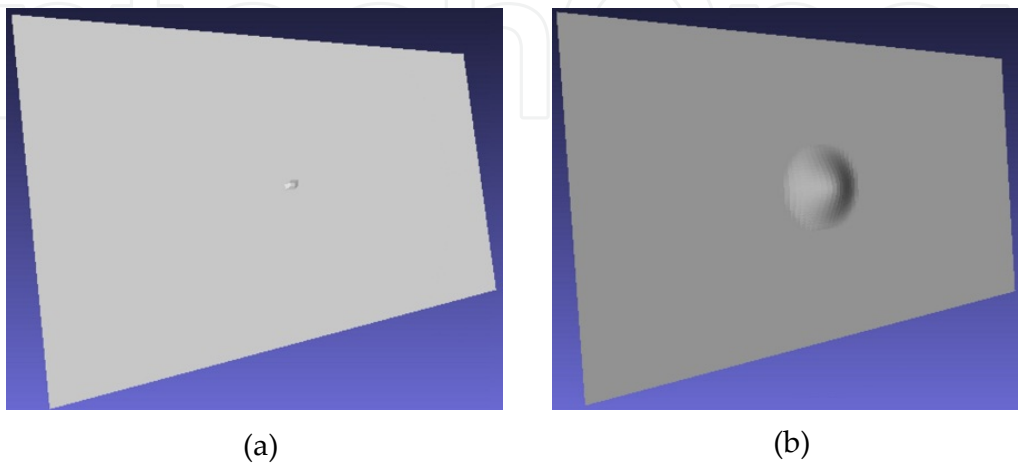


Figure 7. a) Flat mesh with small dent, b) flat mesh with large ding.

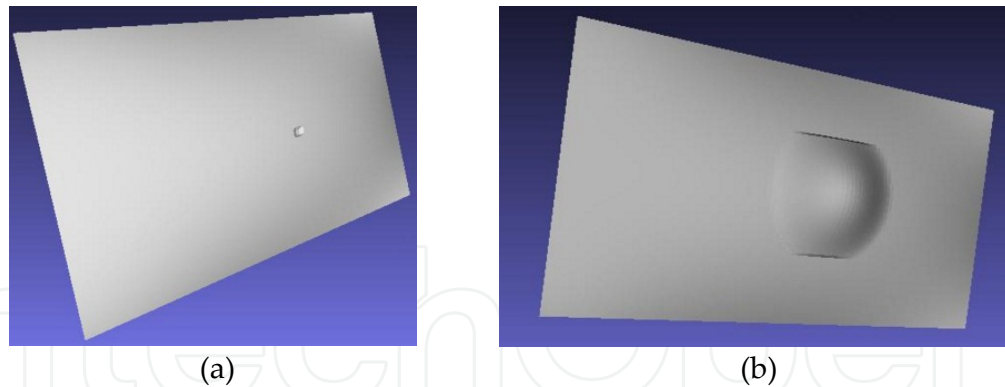


Figure 8. a) Curved surface with small dent, b) curved surface with large ding.

4. Octree-based surface shape analysis

The critical component of the proposed defect detection system is its surface shape analysis module. The goal of the latter is to break the mesh up into pieces and determine which of those pieces likely belong to the defects in question, as shown in Figure 9.

The outcome of the methods presented in this section is the labeling of all parts of the mesh as either belonging to a feature or not. This module does not determine what collection of mesh pieces define a deformation, however it will determine which mesh pieces likely contain part of a deformation. The output of the shape analysis module is therefore passed to the segmentation phase, as depicted in Figure 2, to determine which collection of mesh pieces defines a deformation.

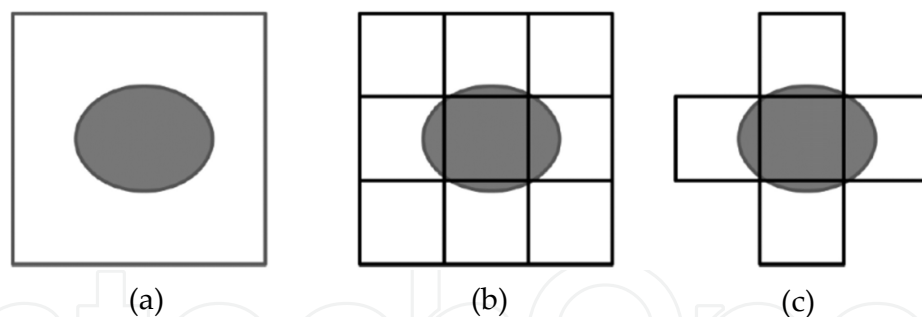


Figure 9. a) Mesh with oval feature, b) mesh broken into pieces, and c) feature extraction results.

This chapter proposes an original surface shape analysis technique that is based on octrees for the automated deformation detection framework. The octree-based technique divides the mesh into cubic volumes and analyzes the mesh contained in those volumes to determine if they belong to a feature. Taking inspiration from the octree-based segmentation method proposed by Woo *et al.* (Woo *et al.* 2002), as explained in section 2.2, a number of improvements are proposed.

The original technique represents the entire volume surrounding a point cloud in the root node of a tree. Then, by evaluating the standard deviation of the point normals, σ , against a threshold, it determines whether there should be a subdivision of that volume into eight octants. When subdivided, each of the resulting octants is a volume, and is represented by a

child node. The points in the original volume are redistributed into the new octants, based on their position, and stored in the child node which represents the new volume it belongs to. This process is repeated for each node, until a tree of sufficient depth is generated or no more volumes require subdivision.

The original technique is designed to find significant changes in a 3D point cloud, such as sharp edges. However, deformation detection for dings and dents over automotive body panels requires detection of slight variations over a smooth surface, which may not be consistent over multiple resolutions. For this reason, the original technique must be revisited. Two major aspects are introduced in this work to enhance the algorithm's flexibility and performance for the purposes of the application considered in the present work: *i)* using a triangle-based analysis rather than a point-based analysis of surface shape, and *ii)* defining non-uniform weighting of surface normals. These enhancements will be discussed in sections 4.1 and 4.2, respectively. A third improvement is also proposed, that uses the octree to aggregate multi-resolution information into performing the feature extraction after the tree generation is complete. This will be discussed in section 4.3.

4.1. Triangle-based analysis

The original method (Woo et al. 2002) operates directly on the 3D point cloud, with knowledge of the reconstructed surface, to calculate the appropriate values for subdivision of the octree. The calculation of the point normal uses all the triangles surrounding the point, and the subdivision of the octree relies on the standard deviation of the point normals contained inside each node. The triangles surrounding a point provide several pieces of information about the surface, yet reducing this information to a point normal using an averaging calculation acts as a smoothing filter, inherently inducing a loss of valuable surface information as shown in Figure 10. In the context of the present work where deformations to be detected are of a small size compared to the remainder of the panel surface curves, such a filter should not be included directly in the feature extraction

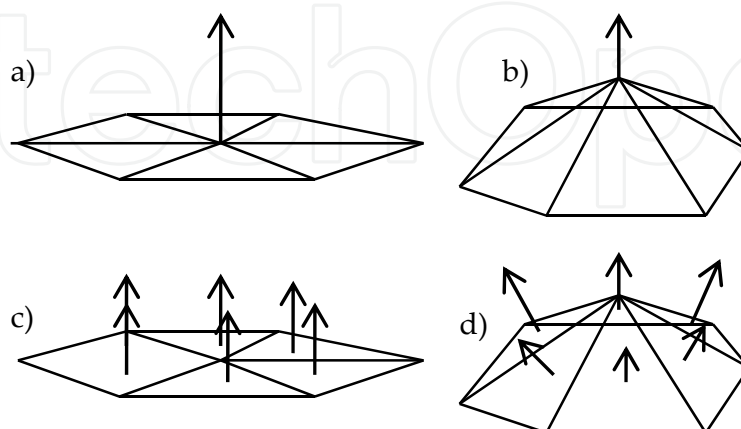


Figure 10. a) Point normal describing a flat surface, b) same point normal describing a non-flat surface, c) triangle surface normals describing a flat surface, and d) different triangle surface normals describing a non-flat surface.

technique. For these reasons, a first improvement that is proposed consists of using more information to describe the surface of a model by relying on the triangles for surface calculations rather than on points.

4.2. Non-uniform weighting of surface normals

The second strategy to enhance the performance of the algorithm proposed in (Woo et al. 2002) consists in using information about the size of the triangles represented by the surface normals to improve the standard deviation, σ , calculation from Eq. 2.

When using 3D scanning that provide a non-uniform sampling density, the original technique will assign equal weight to every point. As a result, there might be more points to describe a certain region within a volume, and the variation in that region is more strongly accounted for in the σ calculation than in other regions, even if there is no disparity in the surface area represented by these regions, as shown in Figure 11.

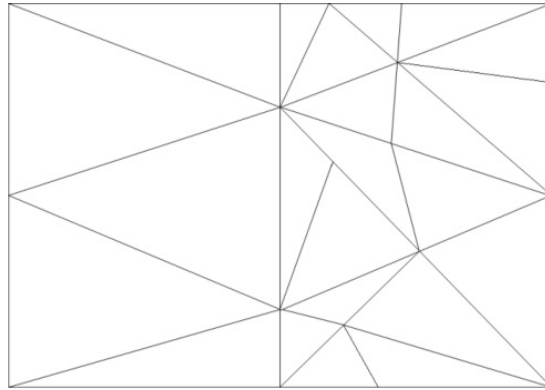


Figure 11. Non-uniformly distributed scan points. The right half of mesh contributes more to the standard deviation value than the left half due to higher density of points/triangles.

A solution to this problem is achieved by using the area of each triangle as a weight to calculate the mean normal and σ values. This approach helps to minimize the effect of small noisy areas, to overcome the effect of non-uniformly distributed points, and to provide a more accurate representation of the surface variation over the region being analyzed.

First, the area of each triangle is calculated:

$$a_i = \frac{1}{2} |v_{i1} \times v_{i2}| \quad (3)$$

where v_{i1} and v_{i2} are any two of the edge vectors that define a triangle, T_i . Then the weighted average normal is calculated over all the triangles that are contained within a given node of the octree, with the area of each triangle serving as a weight:

$$A = \sum_{i=0}^n a_i \quad (4)$$

$$\bar{N} = \sum_{i=0}^n \frac{a_i}{A} N_i \quad (5)$$

where n is the number of triangles contained in the volume represented by the node of interest in the octree, \bar{N} is the weighted average normal, and N_i is the unit normal of each triangle. Finally, the weighted standard deviation, σ , of the normals can be estimated as:

$$\sigma = \sqrt{\frac{\sum_{i=0}^n a_i (N_i - \bar{N})^2}{A}} = \sqrt{\frac{\sum_{i=0}^n a_i (x_i - \bar{x})^2 + \sum_{i=0}^n a_i (y_i - \bar{y})^2 + \sum_{i=0}^n a_i (z_i - \bar{z})^2}{A}} \quad (6)$$

As in the original method, if σ is greater than the defined threshold for that resolution, the volume is subdivided for further investigation at a higher resolution, until the entire tree has been generated. This improvement makes the standard deviation values more accurately represent the amount of deformation within a volume. Figure 12 compares non-uniform weighting to uniform weighting. Detected regions containing deformations are marked in red over surface maps corresponding to various levels of resolution in the octree. Deeper levels in the octree correspond to finer details in the 3D surface mesh.

It can be seen that uniform weighting (upper line) is not as effective at extracting the complete set of deformations. By octree resolution level 10, none of the deformations are extracted with uniform weighting. The non-uniform weighting scheme, with similar thresholding, more consistently extracts the deformations across all resolution scales.

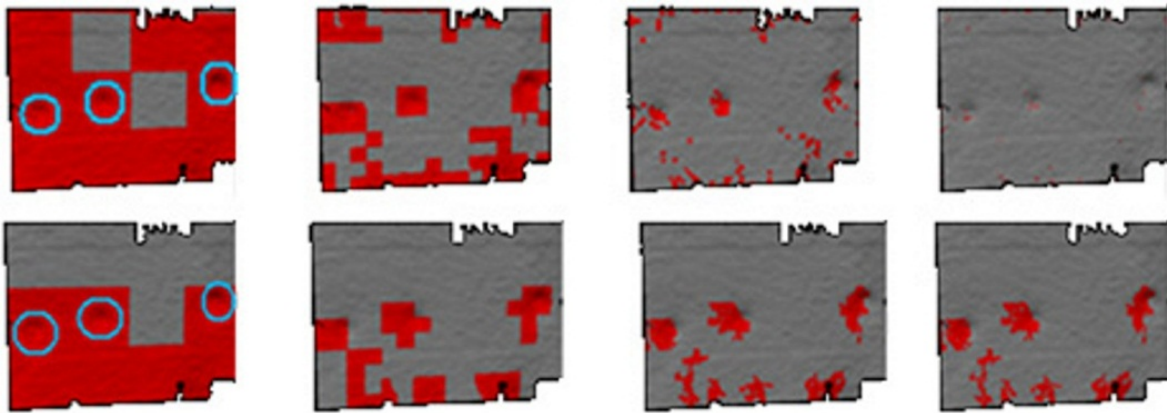


Figure 12. Deformation detection performance at octree resolution levels 4, 6, 8, and 10 under: uniform weighting (top), and non-uniform weighting (bottom). Actual deformation locations are circled on left hand-side images, and detected deformations are marked in red.

4.3. Aggregate standard deviation

Even with the aforementioned improvements of triangle-based analysis and non-uniform surface normal weighting, deformations extraction using an octree-based distribution of 3D scan points requires the determination of the appropriate thresholds. Based on the thresholds alone, it is difficult to predict what features will remain and what features will be removed by the time the deeper and higher resolution levels of the octree are reached. A slight threshold change can produce drastically different results. In general, a more robust technique is required to deal with meshes with varying characteristics, involving an intuitive relationship between the threshold and the results.

The proposed aggregate standard deviation variation of the octree-based technique allows the generation of the tree in its entirety, without limiting subdivision to only nodes that meet a certain threshold. Then, all nodes at selected resolutions of the fully generated tree can be extracted and those with low surface variation can be removed. A new metric is introduced to measure multi-resolution surface variation.

Before the new metric is detailed, it is important to show how the algorithm can be applied using the standard deviation, σ , as the main metric. Analyzing the histogram of the σ values can help in selecting a proper threshold to isolate feature nodes. Histograms are computed to separate the range between the lowest σ and highest σ values at a given octree resolution level into 256 equally divided bins. In the histograms, the x-axis represents the bins, where bin 1 is the lowest range of σ values and bin 256 is the highest, and the y-axis represents the number of nodes with σ values in each bin. In Figure 13, standard deviation values are mapped as a grayscale representation. Black pixels represent low deviation, and white pixels represent large deviation. Therefore, the nodes containing deformations appear in the higher bins. Over most surfaces, the majority of the nodes are mapped to non-feature nodes, such that the bulk of the normal distribution gets classified as non-feature nodes, that is nodes with low σ values. Using the computer casing panel scan as an example, Figure 13 represents the σ values on the mesh as intensity values, along with its histogram, a selected threshold, and the thresholding results. Nodes with σ values above the threshold are extracted as features, which leads to the extraction of the three deformations along with some extra noisy areas.

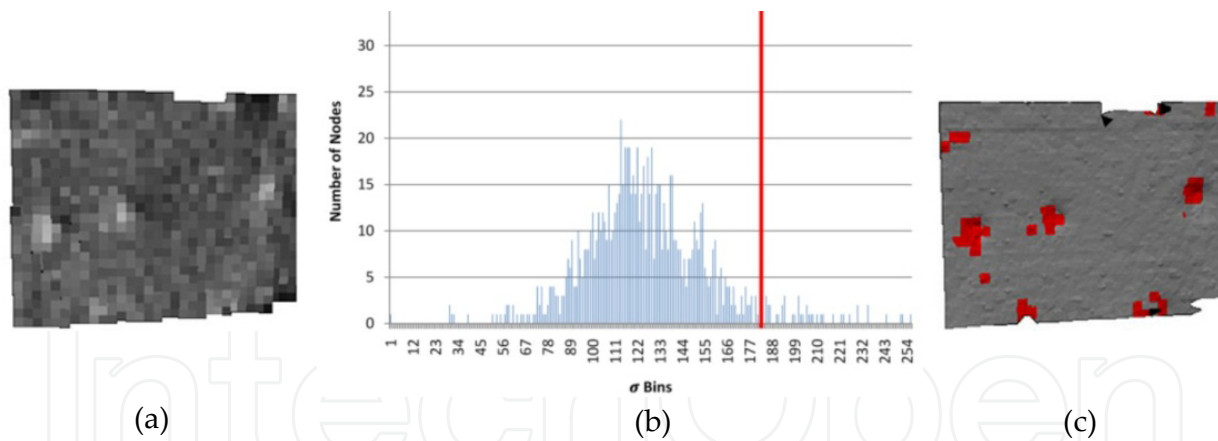


Figure 13. a) Image of surface shape standard deviation mapped as intensities for the low resolution computer casing panel sample encoded in octree at resolution level 6, b) corresponding histogram of σ values with selected threshold, and c) extracted features including three deformations.

Relying on σ values alone does make use of the multi-resolution capabilities of the octree structure and places a lot of emphasis on the proper selection of the threshold at a given resolution. As described in section 2.2, Pauly *et al.* (Pauly et al. 2002) proposed the idea of multi-resolution feature persistence, where a strong feature can be retained only if it is persistently detected across multiple adjacent scales. In order to combine some of the key concepts of multi-resolution feature persistence with the octree-based feature extraction technique, it is proposed in this work that the characteristics of nodes are accumulated

across multiple resolution levels of the octree. The accumulated standard deviation of the surface normals for an octree node is estimated as follows:

$$\sigma_{\text{children}} = \sum_{i=0}^n \frac{\sigma_i}{n} \quad (7)$$

$$s = (\sigma)(\sigma_{\text{parent}})(\sigma_{\text{children}}) \quad (8)$$

where s is the aggregate standard deviation value, σ is the standard deviation value of surface normals in the current node, σ_{parent} is the standard deviation value of surface normals in the parent node, σ_i is the standard deviation value of surface normals in the i^{th} child, and n is the number of children that are not empty, such that only nodes that contain 3D points are considered not to bias the metric. σ_{children} is calculated as the average standard deviation value of the node's non-empty children. Note that σ values are calculated using non-uniform weighting as detailed in section 4.2. At any given scale, each node will contain a value representing the accumulated standard deviation, s , of a certain volume of the mesh located under itself in the octree structure.

Using the computer casing panel scan as an example, Figure 14 shows how thresholding the s values at a single resolution can be successful in isolating areas of interest. The histogram shown is in the same format as the histogram of Figure 13, but maps s values into pixel intensities instead of σ values.

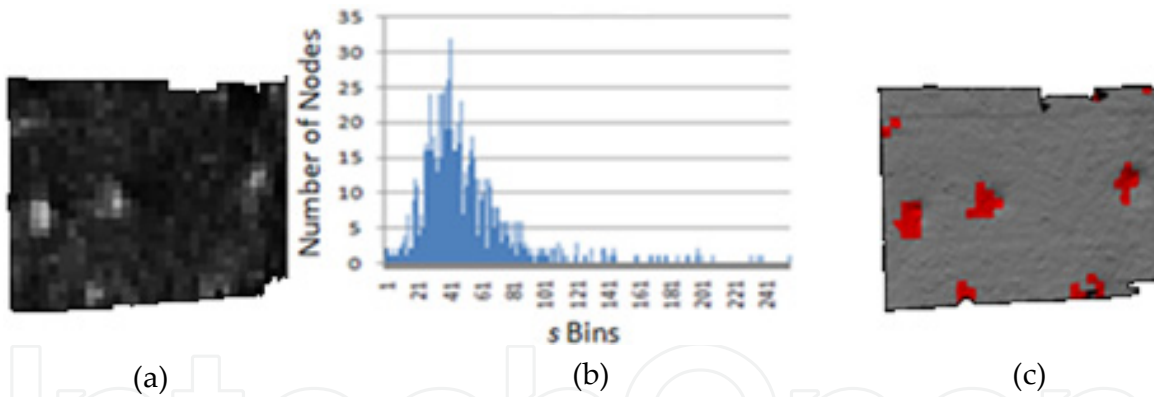


Figure 14. a) Intensities corresponding to aggregate standard deviation, s , in low resolution computer casing panel sample at resolution level 6 of the octree, b) s histogram with threshold, and c) extracted features including three deformations.

The aggregate standard deviation, s , value provides a greater separation between feature nodes and non-feature nodes than the local standard deviation, σ , only. A more accurate separation also adds tolerance to non-optimal thresholds. To compare the tolerance to non-optimal thresholds when using the σ value against using the s value for feature extraction, the algorithm is applied on the filtered low resolution computer casing panel mesh with both metrics at octree resolution level 5. A suitable threshold was determined such that the results are comparable between using the σ and s values respectively. Then, using the histogram, thresholds which are 50 bins in either direction of the selected threshold are used to determine

how that affects the extraction results. When using the σ metric, a threshold of 0.223 is used, which corresponds to bin 163. Then, a threshold of 0.157, corresponding to bin 113, and a threshold of 0.289, aligned with bin 213 are also applied. When using the s metric, a threshold of 0.295 is used, which corresponds to bin 176. Then, a threshold of 0.214, corresponding to bin 126, and a threshold of 0.377, aligned with bin 226 are also applied. Figure 15 shows the results using σ for feature extraction, and Figure 16 shows the results using s for feature extraction.

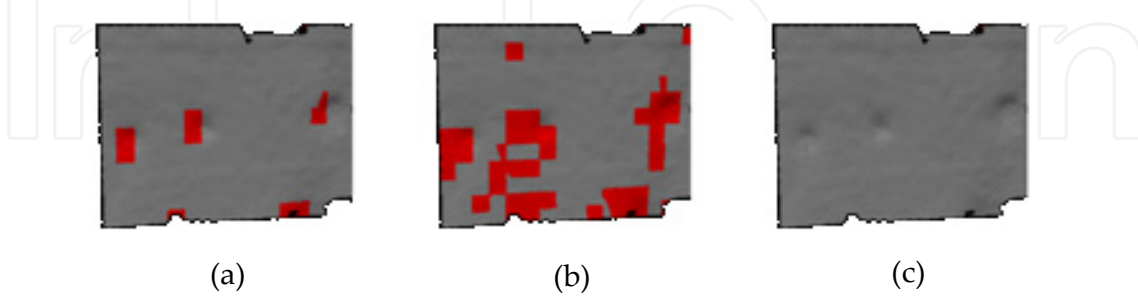


Figure 15. a) Deformed regions detected over computer casing panel at resolution level 5 of the octree with a) optimal σ threshold, b) optimal σ threshold minus 50 bins, and c) optimal σ threshold plus 50 bins.

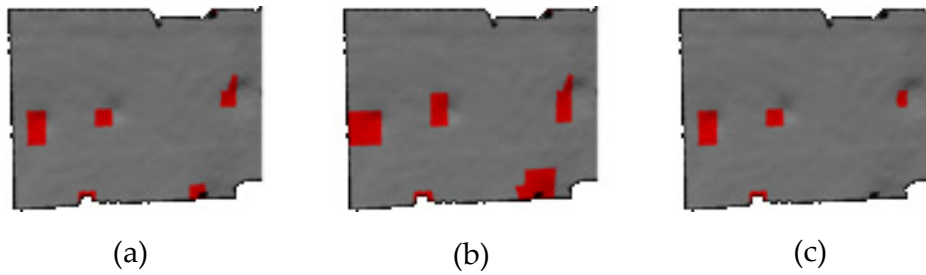


Figure 16. a) Deformed regions detected over computer casing panel at resolution level 5 of the octree with a) optimal s threshold, b) optimal s threshold minus 50 bins, and c) optimal s threshold plus 50 bins.

This case demonstrates that a change in threshold setting affects the deformation detection method more extensively when using the local standard deviation, σ , value as a metric than when using the proposed aggregate standard deviation, s , value. When using the σ value as a metric, the surface analysis captures more noise and transient features when the threshold is lowered, and removes all of the deformations when the threshold is increased. When using s as a metric, the outcome of the surface analysis does not change significantly with the different thresholds, as the deformations are all still present and no significant additional surface variation is detected. As a consequence, when thresholds need to be selected from experimentation, the expected results with aggregate standard deviation are much less sensitive to changes in threshold setting than with the enhancement described in section 4.2 alone. The increase in robustness when compared to non-optimal thresholding, and the significant separation between non-deformation areas and deformation areas in the s metric, justify the use of s over σ as a metric.

Also, since only three levels of the octree need to be analyzed for thresholding, in accordance with Eq. 8, only the level of the octree at which the nodes are being extracted

and thresholded, along with the levels immediately above and below, must be generated. By selectively generating only the necessary levels of the tree from the 3D points cloud, the efficiency of the algorithm is improved significantly. The increase in intuitiveness of the thresholding parameter setting and the efficiency over the original octree-based method of Woo *et al.* (Woo *et al.* 2002), which thresholds nodes during tree generation as opposed to after tree generation, also support the development and application of the algorithm introduced in this work.

To demonstrate the effectiveness of the proposed algorithm, in addition to the results presented above on the computer casing panel, the method is applied to the more challenging unfiltered mesh of the car door with the aggregate standard deviation threshold set at 0.155, for the octree resolution level 6. The results are shown in Figure 17. The same algorithm is also applied to the artificial curved meshes, with the threshold set at 0.002, at resolution level 5 of the octree, and the results are shown in Figure 18.

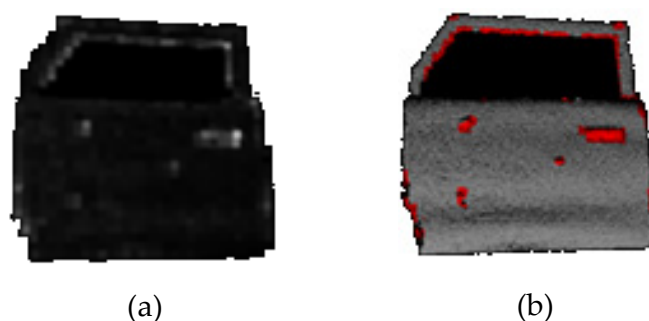


Figure 17. a) Intensity map corresponding to s values on unfiltered car door octree at resolution level 6, and b) features extracted including the three deformations of interest.



Figure 18. Feature extracted, at resolution level 5, on artificial curved mesh with a) small dent, and b) large ding.

In terms of deformation extraction effectiveness, provided that the correct threshold is selected, the algorithm performs similarly on both the filtered and unfiltered car door meshes. It also extracts many of the edges around the door and window frame. These edges are very rough areas in the meshes due to the limitations of the acquisition system, and generate large s values in their surroundings, resulting in them being extracted. Despite these small issues with the noisy data, the algorithm isolates the deformations well while increasing robustness and decreasing memory usage when compared to using only the previous octree-based method enhancements.

5. Segmentation and classification

When using the octree-based surface shape analysis technique described in section 4, each node records information determining whether it belongs to a feature or not. Among the nodes that correspond to a deformation, those that contain pieces of a same deformation must eventually be grouped together to segment the deformation from the rest of the panel surface scan, as shown in Figure 19.

Finally, before the entire system outputs the segments that contain deformations, each of them must be classified as a ding or a dent as that is one of the primary objectives of the proposed solution. The octree-based feature extraction requires the classification component to handle this task by receiving the segments containing the deformations of interest as an input and labeling them as dings or dents for the output. A two-step segmentation and classification strategy is proposed to achieve this goal.

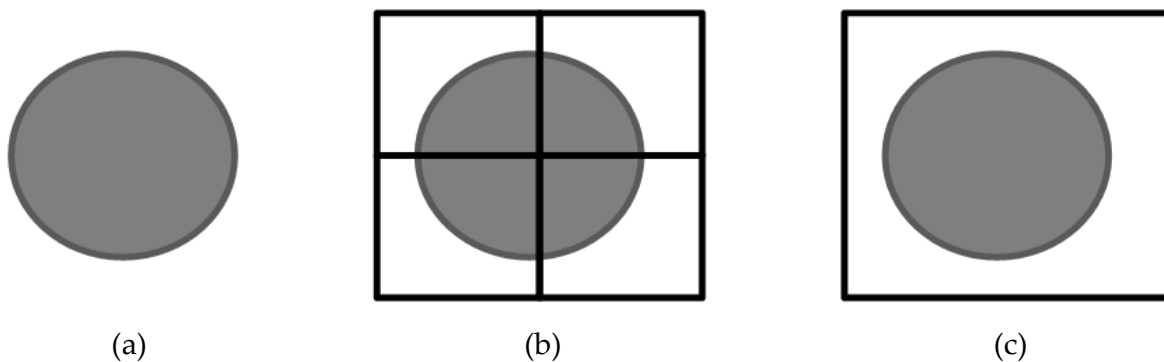


Figure 19. a) Original deformation, b) octree-based surface shape analysis results, and c) octree-based segmentation.

5.1. Single-resolution segmentation based on octrees

If the scale of the desired deformations is known, an appropriate resolution of the octree can be selected to extract those deformations. Since different depths of the octree correspond to different spatial resolutions, selecting all nodes at a certain depth (defined as octree levels in the previous section) will provide a voxel representation of the object at that scale. However, the appropriate resolution level to segment the deformation must generally be lower than the resolution level considered for the surface shape analysis described previously. Indeed small discontinuities in the deformation should not be detected and segmented as individual deformations based on the connectivity between nodes in the higher resolution version of that deformation. On the other hand, the segmentation resolution must be sufficiently high to avoid deformations being grouped with non-deformations, and to reduce the size of small segments defining features such as noise, in order to avoid confusion with the actual deformations during the classification phase.

After the feature extraction removes all non-feature voxels at the desired resolution, grouping is performed. These remaining voxels are denoted as *feature voxels*. Sets of feature voxels are grouped together to define a deformation based on adjacency, since each voxel

contains a piece of a deformation. Since the voxels are cells of a 3-dimensional grid, adjacent voxels can be determined based on their coordinates in the 3-dimensional grid and looking for voxels at adjacent coordinates to the current one. By extending the idea of blob extraction, which is a well-known 2-dimensional image processing algorithm, to three dimensions, adjacent feature voxels can be grouped together. The final result is a set of voxel groups, where each group represents a segment containing a deformation.

The proposed octree-based method is applied on the flat mesh with a small deformation of Figure 7a to extract the small deformation until octree resolution level 8. The segmentation results are shown in Figure 20. Figure 21 shows the segmentation applied at octree resolution level 6 on feature extraction results on the indented computer casing panel high resolution surface mesh.

These results demonstrate that the segmentation can group the required voxels to properly define the deformations. On the artificial mesh, applying segmentation at resolution level 6 segments the deformation clearly. On the other hand, the segmentation at resolution level 4 shows that the deformation is still located, but covers a larger surface than the actual deformation. This is because the resolution considered is lower, therefore the voxel containing the deformation is larger and entirely marked. Similarly successful results are achieved on the computer casing panel, with all deformations being successful grouped.

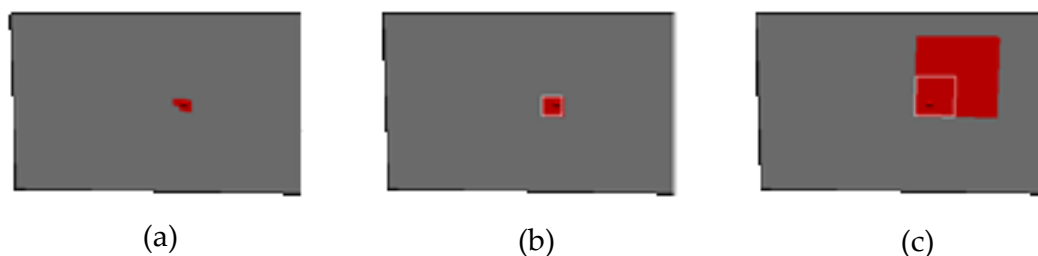


Figure 20. a) Octree-based feature extraction, at octree resolution level 8, on the flat mesh with small deformation, b) bounding box of segmented deformation at octree resolution level 6, and c) bounding box of segmented deformation at octree resolution level 4.

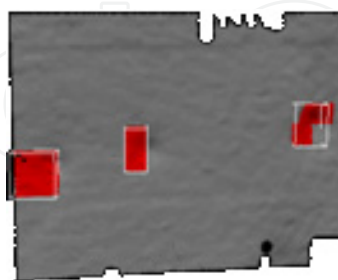


Figure 21. Bounding boxes defining the areas segmented as actual deformations on the filtered high resolution computer casing panel surface mesh at resolution octree level 6.

5.2. Classification

Classification represents the final phase of the proposed deformation detection process. It helps determine whether the identified segments are dings or dents. It also provides abilities

to ensure that the extracted segments are indeed deformations of interest. Ideally, the previous steps of surface shape analysis and segmentation have already removed most non-deformation areas. But in case some erroneous deformation areas remain, the classification phase provides the necessary filtering stage to remove those areas and reduce false positives. Complex classification methods, such as using neural networks as proposed by Döring *et al.* (Döring *et al.* 2004), or spin image signatures as attempted by Assfalg *et al.* (Assfalg *et al.* 2007), could be implemented at this stage. However this work focuses on simpler and more computationally efficient solutions that take advantage of the fact that accurate results have already been obtained by the surface shape analysis and segmentation components.

5.2.1. Classification of the type of deformation

To measure the shape characteristics of the segments, a basic understanding of the orientation of the segment must be determined. A least-squares plane-of-best-fit fitted to the 3D points contained in a segment, specifically the boundary points, is used to determine the orientation of the shape represented by a given segment. Since the boundary points are on the outside edges of the segment, they would more likely belong to the regular surface of the automotive panel than to the deformation. This leads to a plane best fitted to the surface of the automotive panel around the deformation, and determines the general orientation of the surface shape that is contained in the segment. A descriptor, called the point-count descriptor, uses the number of points that share a similar positional relationship to the plane-of-best-fit in estimating the direction of variation of the surface contained in the segment. If a majority of the points contained in the segment are above the plane-of-best-fit, that is, in the direction of the normal vector, the deformation is classified as a ding. If a majority of the points are below the plane-of-best-fit, that is in the opposite direction to the normal vector, the deformation is classified as a dent.

In any classification, a certainty measure is also important. The percentage of the points that are above the plane-of-best-fit in the case of a ding, or below the plane-of-best-fit in the case of a dent, provides the certainty measure on the classification. This way, if there are a similar number of points that are above and below the plane-of-best-fit, the certainty measure is close to 50%, indicating uncertainty.

To test the classification technique's ability to determine whether a deformation is a ding or a dent, it is applied on deformation segments of every mesh using the point-count descriptor, and the results are compared. Non-ideal extraction and segmentation results are presented in Figure 22, while the resulting classifications are presented in Table 1.

Over the artificial flat and curved meshes, it can be seen that the classification is correct. These results show that the classification behaves well on artificial models, corresponding to an acquisition system with minimal noise and acquisition artifacts. For the real world meshes (car door and computer panel), the descriptor accurately classifies each of the dents on the computer casing except for one, which is recognized as a ding rather than a dent. This

can be attributed to the non-ideal feature extraction, as classification is dependent on the quality of the latter step. However, the certainty measure reflects the inaccuracy of the classification by being close to 50%, lower than that of the correctly classified deformations. Overall, it can be seen that the classification provides proper results even when feature extraction and segmentation results are non-ideal.

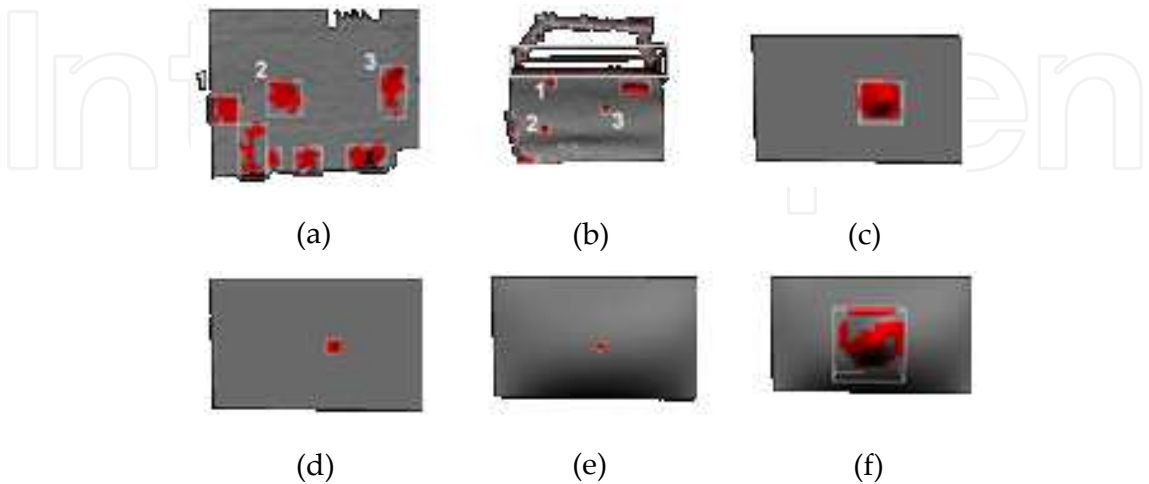


Figure 22. Octree-based feature extraction and single-resolution segmentation applied on a) computer casing panel mesh with dent segments labeled, b) car door mesh with ding segments labeled, c) flat mesh with large ding segment labeled, d) flat mesh with small dent segment labeled, e) curved mesh with small dent segment labeled, and f) curved mesh with large ding segment labeled.

Model	Actual	Point-Count Descriptor Estimates	
	Type	Type	Certainty
<i>Car Door</i>			
Def 1	Ding	Ding	0.636
Def 2	Ding	Ding	0.644
Def 3	Ding	Ding	0.609
<i>Computer Casing Panel</i>			
Def 1	Dent	Ding	0.502
Def 2	Dent	Dent	0.511
Def 3	Dent	Dent	0.546
<i>Flat Mesh</i>			
Small	Dent	Dent	0.583
Large	Ding	Ding	0.896
<i>Curved Mesh</i>			
Small	Dent	Dent	0.667
Large	Ding	Ding	0.558

Table 1. Comparison of actual deformation characteristics and the results of classification following octree-based feature extraction and segmentation.

5.2.2. *Additional classification*

Though it is not the primary goal of the classification stage, the latter also allows fine-tuning of certain parameters by the operator as a final effort to ensure that only deformations of interest are outputted as marked segments. In the process of analyzing descriptors to determine whether a segment is a ding or a dent, segments that do not meet the characteristics of either can be removed. Certain well-known characteristics of deformations can be taken into account to remove non-deformation areas that remain.

The combined surface area of the mesh contained in a segment makes a suitable descriptor of deformations of interest. Thresholding the surface area is another effective strategy for the removal of noise and acquisition artifacts, as those erroneous extracted segments typically cover only very small surface area. Similarly, thresholding surface area also proves effective in removing large surface features that do not correspond to deformations expected over an automotive body panel at the assembly stage, such as actual door handles or aesthetic curves. The latter cover much larger surface areas than dings and dents.

After the orientation of the shape contained in the segment is determined, as detailed in section 5.2.1, and the plane-of-best-fit provides the shape its own local coordinate system, characteristics such as the deformation size in the x , y , and z directions can be measured relative to the shape's local coordinate system to provide an estimate of the shape's height, width, and depth. Noise typically has a small depth, while features like door handles have a larger depth and width relative to the deformations of interest. Applying thresholds on these parameters further increases the reliability of isolating only segments that contain actual deformations.

Though a certain dependency on the setting of threshold values remains, the combination of these descriptors that provide a large amount of information about the various shapes contained in the extracted segments proves to be an excellent technique to improve the reliability of the feature extraction process. In order to demonstrate the relevance of using the final classification phase to further refine the selection of actual deformation segments, here focusing on dings and dents over smoothly curved surface meshes, some cases of poor feature extraction scenarios are artificially created using the experimental models described in section 3.3, but with non-optimal parameters for feature detection and segmentation. The resulting non-deformation areas that get included into detected segments are used to test the classification system's ability to distinguish between actual deformations and non-deformations.

Figure 23 depicts a non-optimal case where many false positives are detected and segmented as potential deformation areas over the car door surface mesh, and shows how the classification is able to remove them. By setting the parameters of the additional classification to remove depths less than 8.5mm or greater than 17mm, most of the broad curvatures on the panel, the door handle, and the small noisy areas are removed. Also,

setting the parameters to remove segments with surface area less than 2800 mm^2 or segments with surface area greater than 5200 mm^2 results in further non-deformation areas being removed. In spite of the non-optimal tuning of the quality inspection system, only one additional area, influenced by the boundary effect around the lower left-hand side of the panel, is preserved as a potential deformation segment in Figure 23b. This represents a major improvement from the 40+ erroneous segments initially identified in Figure 23a.

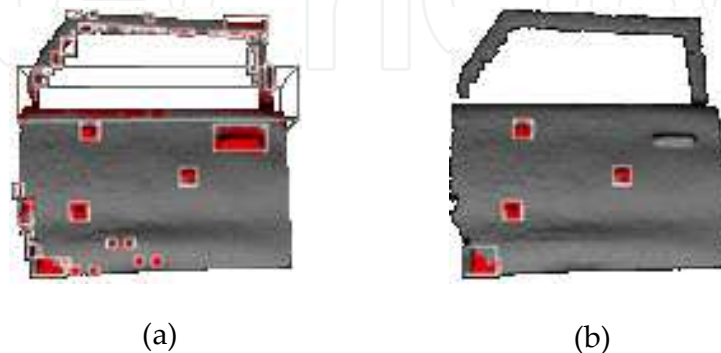


Figure 23. a) Deformation and non-deformation areas initially segmented on the car door mesh, b) additional classification removes most non-deformation areas.

6. Conclusion

In this chapter, an original feature detection, segmentation and classification framework is proposed to process 3D point clouds and corresponding surface meshes in order to meet the advanced requirements of an automated deformation detection system for use in the context of automotive panels quality control over an assembly line. The requirements in place are that such a system must be able to detect deformations of interest, using 3D analysis, without knowledge of the ideal surface and without any comparative CAD models. The deformations must also be classified as dings or dents. The proposed approach assumes that the operator possesses a minimal knowledge about the approximate size and scale of these deformations of interest in the context of the specific application. The proposed technique then makes optimal use of this additional information to refine the deformation isolation process which leads to an accurate separation of ding and dent deformations from desirable aesthetic design features that typically appear over automotive panels.

A variety of techniques were reviewed for the deformation detection pipeline. An octree-based technique is revisited and refined for surface shape analysis. A single-resolution segmentation method is also presented to refine the location of deformations. Finally, a classification approach is proposed and a complete experimental evaluation is performed on every stage of the surface inspection procedure. The complete pipeline is effective in identifying the location of deformations of interest, and classifying them as dents or dings when presented with a 3D mesh of a surface.

Experiments were conducted on both artificial and real-world test data, offering a set of meshes encompassing various characteristics. These experiments demonstrated that the proposed approach can be used in both ideal circumstances, such as finding a large deformation over a flat, noiseless mesh, as well as in more complex circumstances, such as finding small deformations over a noisy, smoothly curved surface, with acquisition artifacts and holes. The experimental results demonstrate that the proposed framework is scalable, effective and robust to meshes with noise and acquisition artifacts, along with non-ideal surfaces containing shape variations other than the deformations of interest. The proposed technique is therefore suitable for integration in an automated deformation detection and marking system for quality control on automotive panels assembly lines.

Author details

Arjun Yogeswaran and Pierre Payeur
University of Ottawa,
Electrical Engineering and Computer Science (EECS),
Canada

7. References

- Assfalg, J., Bertini, M., Bimbo, A. D. & Pala, P. (2007). "Content-based Retrieval of 3-D Objects Using Spin Image Signatures." *IEEE Transactions on Multimedia* 9(3).
- Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C. & Taubin, G. (1999). "The Ball-Pivoting Algorithm for Surface Reconstruction." *IEEE Trans. Visualization and Computer Graphics* 5(4): 349-359.
- Blais, F., Taylor, J., Cournoyer, L., Picard, M., Borgeat, L., Godin, G., Beraldin, J. A., Rioux, M. & Lahanier, C. (2007). Ultra High-Resolution 3D Laser Color Imaging of Paintings: the Mona Lisa by Leonardo da Vinci. *Proceedings of the 7th International Conference on Lasers in the Conservation of Artworks*, Madrid, Spain.
- Borsu, V. (2010). Pose and Motion Estimation of Parts Exhibiting Few Visual Features for Robotic Marking of Deformations, University of Ottawa.
- Borsu, V. & Payeur, P. (2009). Pose and Motion Estimation of a Moving Rigid Body with Few Features. *Proceedings of the IEEE International Workshop on Robotic and Sensor Environments*, Lecco, Italy.
- Borsu, V., Yogeswaran, A. & Payeur, P. (2010). Automated Surface Deformations Detection and Marking on Automotive Body Panels. *Proceedings of the 6th IEEE International Conference on Automation Science and Engineering*, Toronto, Ontario, Canada.

- Boyer, A. (2009). Adaptive structured light imaging for 3D reconstruction and autonomous robotic exploration, University of Ottawa.
- Boyer, A., Curtis, P. & Payeur, P. (2009). 3D Modeling from Multiple Views with Integrated Registration and Data Fusion. Proceedings of the 6th Canadian Conference on Computer and Robot Vision, Kelowna, BC, Canada.
- Chen, H. (2008). Automatic Dent Detection on Car Bodies, Hong Kong University of Science and Technology.
- Dorai, C. & Jain, A. (1997). "A Representation Scheme for 3D Free-Form Objects." IEEE Trans. Pattern Analysis and Machine Intelligence 19(10): 1115-1130.
- Döring, C., Eichhorn, A., Girimonte, D. & Kruse, R. (2004). "Improving Surface Detection for Quality Assessment of Car Body Panels." Mathware & Soft Computing 11: 163-177.
- Faugeras, O. (1993). Three-Dimensional Computer Vision, MIT Press.
- Gokturk, S. B., Yalcin, H. & Bamji, C. (2004). A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions. Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop, Washington D.C., USA.
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. & Stuetzle, W. (1992). Surface Reconstruction from Unorganized Points. Proceedings of SIGGRAPH '92, Chicago, Illinois.
- Jagannathan, A. & Miller, E. (2007). "Three-dimensional Surface Mesh Segmentation using Curvedness-based Region Growing Approach." IEEE Trans. Pattern Analysis and Machine Intelligence 29(12): 2195-2204.
- Koenderink, J. & Doorn, A. v. (1992). Surface Shape and Curvature Scales. Image and Vision Computing: 557-565.
- Lilienblum, T., Albrecht, P., Calow, R. & Michaelis, B. (2000). Dent Detection in Car Bodies. Proceedings of the 15th International Conference on Pattern Recognition (ICPR), Barcelona, Spain.
- Marszalec, J. & Myllyla, R. (1997). "Shape Measurements Using Time-Of-Flight-Based Imaging Lidar." Proceedings of the SPIE Conference on Three-Dimensional Imaging and Laser-based Systems for Metrology and Inspection III 3204: 14-15.
- Mederos, B., Velho, L. & Figueiredo, L. H. D. (2003). "Moving Least Squares Multiresolution Surface Approximation." Proceedings of SIBGRAPI 2003 - XVI Brazilian Symposium on Computer Graphics and Image Processing: 19.
- Murray, D. & Jennings, C. (1997). Stereo Vision Based Mapping and Navigation for Mobile Robots. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '97), Albuquerque, New Mexico, USA.
- Murray, D. & Little, J. J. (2000). "Using Real-Time Stereo Vision for Mobile Robot Navigation." Autonomous Robots 8(2): 161-171.
- Newman, T. S. & Jain, A. K. (1995). "A System for 3D CAD-Based Inspection using Range Images." Pattern Recognition 28(10): 1555-1574.

- Palagyi, K. & Kuba, A. (1999). "Directional 3D Thinning Using 8 Subiterations." *Discrete Geometry for Computer Imagery* 1568: 325-336.
- Parthasarathy, S., Birk, J. & Dessimoz, J. (1982). "Laser Rangefinder for Robot Control and Inspection." *Proceedings of SPIE Robot Vision* 1982 336: 2-11.
- Pauly, M., Gross, M. & Kobbelt, L. P. (2002). "Efficient Simplification of Point-Sampled Surfaces." *Proc. of IEEE Visualization*: 163-170.
- Pauly, M., Keiser, R. & Gross, M. (2003). "Multi-scale Feature Extraction on Point-Sampled Surfaces." *Computer Graphics Forum* 22(3): 281-289.
- Payeur, P. & Desjardins, D. (2009). "Structured Light Stereoscopic Imaging with Dynamic Pseudo-random Patterns." *Proceedings of the International Conference on Image Analysis and Recognition (ICIAR 2009)* 5627: 687-696.
- Plataniotis, K. N. & Venetsanopoulos, A. N. (2000). *Color Image Processing and Applications*, Springer Publishing.
- Rocchini, C., Cignoni, P., Montani, C., Pingi, P. & Scopigno, R. (2001). "A Low Cost 3D Scanner Based on Structured Light." *Computer Graphics Forum (Eurographics 2001 Conf. Issue)* 20(3): 299-308.
- Schall, O., Belayev, A. & Seidel, H. (2005). *Robust Filtering of Noisy Scattered Point Data. Proceedings of Point-Based Graphics, 2005, Eurographics/IEEE VGTC Symposium*, Stony Brook, New York, USA.
- Se, S., Lowe, D. G. & Little, J. (2001). "Vision-Based Mobile Robot Localization and Mapping Using Scale-Invariant Features." *Proceedings of the International Conference on Robotics and Automation* 2001: 2051-2058.
- Sequeira, V., Goncalves, J. & Ribeiro, M. (1995). "3D Environment Modelling Using Laser Range Sensing." *Robotics and Automation* 16(1): 81-91.
- Shaffer, E. & Garland, M. (2001). "Efficient Adaptive Simplification of Massive Meshes." *IEEE Visualization '01*.
- Shahram, I., Richard, A. N., David, K., Otmar, H., David, M., Steve, H., Pushmeet, K., Jamie, S., Andrew, J. D. & Andrew, F. *KinectFusion: real-time dynamic 3D surface reconstruction and interaction. ACM SIGGRAPH 2011 Talks*. Vancouver, British Columbia, Canada, ACM.
- Vosselman, G., Gorte, B., Sithole, G. & Rabbani, T. (2004). "Recognising Structure in Laser Scanner Point Clouds." *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 46: 33-38.
- Woo, H., Kang, E., Wang, S. & Lee, K. H. (2002). "A new segmentation method for point cloud data." *International Journal of Machine Tools and Manufacture* 42(2): 167-178.
- Yan, C. & Didier, S. *3D shape scanning with a Kinect. ACM SIGGRAPH 2011 Posters*. Vancouver, British Columbia, Canada, ACM.

Zhang, L., Curless, B. & Seitz, S. M. (2002). Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. Proceedings of Int. Symposium on 3D Data Processing Visualization and Transmission, Padova, Italy.

IntechOpen

IntechOpen