COMPUTER-AIDED DESIGN & APPLICATIONS, 2018
VOL. 15, NO. 4, 556–564
https://doi.org/10.1080/16864360.2017.1419643

Extraction of vertical cylinder contacting area for motorcycle safety verification

Masatomo Inui and Nobuyuki Umezu

Ibaraki University, Japan

ABSTRACT

Japanese motorcycle manufacturers define a safety regulation regarding the motorcycle shape to prevent injury to pedestrians in the event of a crash. To satisfy the regulation, the external surface of the motorcycle must have a sufficient degree of roundness if there is any chance that it could contact a vertical cylinder having a diameter of 300 mm approximating the human body. In this paper, we propose a novel method for assisting in the safety regulation inspection. This method extracts the surface regions of the tessellated motorcycle body where the vertical cylindrical column contacts and the surface region radius is less than 4 mm. The parallel processing capability of a graphics processing unit (GPU) is used to accelerate the extraction task of identifying the column contacting area. A system using our algorithm can detect surface areas on the motorcycle body where the safety regulation is not satisfied.

KEYWORDS

Safety regulation; offset computation; curvature evaluation; parallel processing; GPU

1. Introduction

Safety is an important concern in motorcycle design. Japanese motorcycle manufacturers define several safety regulations regarding the motorcycle shape to prevent injury to pedestrians in the event of a crash. Fig. 1 illustrates one regulation regarding the motorcycle shape. A vertical cylindrical column 300 mm in diameter approximates a human body shape in the standing position. To inspect compliance with the regulation, a motorcycle is moved forward and eventually pushed into the column. In this way, surface areas on the motorcycle are detected where the column is contacted. Such surface areas must have a radius greater than R3.2. In the actual inspection, it is safer to use R4 instead, and the latter is thus adopted. According to this regulation, column contacting areas at of 400 mm above the ground level are subject to a check. This condition is not considered in our current study, and all contact areas on the motorcycle are checked.

Since the exterior strongly affects the appearance and comfort of a motorcycle it is often initially designed in terms of function and aesthetics. Currently, the safety regulation regarding the shape is inspected by specialists at the final design stage using a physical apparatus and models. This work is cumbersome, time consuming, and prone to human errors. The detection of safety issues at this stage results in costly rework. A fast and automatic inspection method that allows designers to check the regulations during the shape design process is desirable. To assist designers, we propose a novel method for extracting the surface regions on a motorcycle body where the vertical cylindrical column contacts and the radius of the surface region is less than R4. In our method, the parallel processing capability of a GPU is used for fast extraction. In the next section, the basic concept of our method is illustrated and some related studies are briefly reviewed. Details of the novel concept of our algorithm are explained in Section 3. Experimental computational results are provided in Section 4, and conclusions are summarized in Section 5.

2. Main contributions

2.1. Input and output

The input data for our method consists of a polyhedral STL model that approximates the motorcycle shape with a high degree of accuracy. The use of a polyhedral model in the manufacturing process of mechanical products is becoming more common. Many computer-aided engineering systems accept polyhedral models for the shape data in structure analysis. The aesthetics of a motorcycle body are usually evaluated by precisely rendering the body shape in a polyhedral representation. Various data formats are known for recording polyhedral models, including the OBJ, STL, PLY and 3DS formats. Among these formats, the STL format is the most commonly used. In the following discussion, the STL model of the

CONTACT Masatomo Inui masatomo.inui.az@vc.ibaraki.ac.jp; Nobuyuki Umezu nobuyuki.umezu.cs@vc.ibaraki.ac.jp

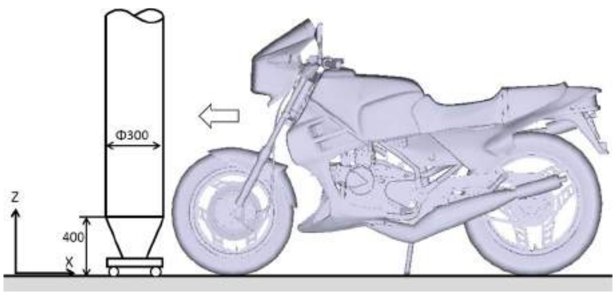

Figure 1. Safety regulation of a motorcycle.

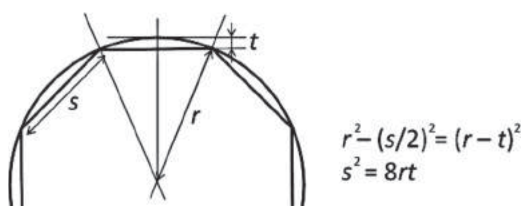

Figure 2. Geometric relationships between sag (*t*) and step (*s*) parameters in the tessellation of a sphere of radius *r*.

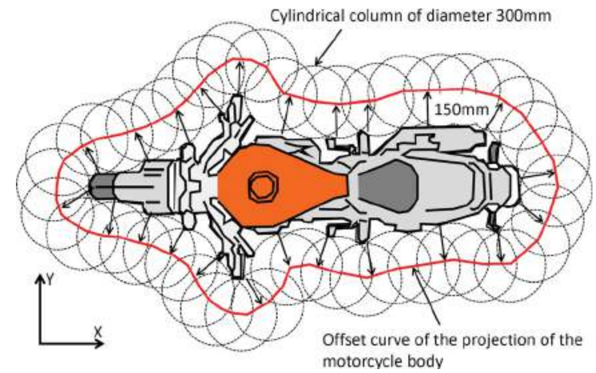

Figure 3. Cylindrical column sliding on the external surface of a motorcycle.

motorcycle is positioned so that it is on the xy-plane of the coordinate frame, and it faces the negative x-axis direction, as shown in Fig. 1. The system extracts the column contact area whose radius of curvature is less than 4 mm and outputs the area data as a set of small polygons in STL format.

In the tessellation of an object with a curved surface, two parameters, named "sag" and "step", are considered for controlling the quality of the result polyhedral model [5]. The sag parameter represents the allowance of the shape difference between the original shape and its approximated shape with polygons. The step parameter controls the maximum length of the edges of the tessellated model. These parameters are not mutually independent in the representation of a shape with a small radius. Fig. 2 illustrates a section of a sphere with radius r. Consider a tessellation of the sphere with a sufficiently small sag value t. In this case, the step value s becomes smaller than $sqrt(8rt)$, as shown in the figure. In our current implementation, $t = 0.1$ mm is used. To inspect the curved shape with a radius smaller than $r = 4$ mm, the step value s must be smaller than 1.79 mm. $s = 1.5$ mm is used as a step value in our current implementation.

2.2. Outline of algorithm

Our inspection algorithm achieves the task in the following two steps. In the first step, the column contacting polygons are extracted in the polyhedral motorcycle model. This operation is achieved by sliding the vertical column of 300 mm in diameter while maintaining its contacts to the external surface of the motorcycle (see Fig. 3). Consider a projection of the motorcycle shape onto the horizontal xy-plane. The trajectory of the center point of the vertical column in the sliding motion corresponds to the boundary curve of the offset shape of the projection by 150 mm, as shown by the red curve in Fig. 3. After the extraction of the column-contacting-polygons, polygons facing the back of the motorcycle (that is, the positive x-axis direction) are selected. Since the column never contacts these polygons when the motorcycle is moving in the forward direction, these polygons are discarded and not used in the following processing.

In the second step, the radius of curvature of the extracted polygon is evaluated and the polygons whose radius of curvature is smaller than 4 mm are selected and visualized. In the curvature evaluation of the polyhedral surface, we applied the "sphere placement method" proposed by the authors in [28]. For each column-contacting-polygon $\boldsymbol{f}$, a sphere $\boldsymbol{S}$ of radius 4 mm is placed so that it contacts the backside of the polygon at its center of gravity $\boldsymbol{g}$ as shown in Fig. 4. Intersections between $\boldsymbol{S}$ and the polygons adjacent to $\boldsymbol{f}$ (polygons shaded in the light blue color in Fig. 4) are checked. The radius of curvature of $\boldsymbol{f}$ is judged to be smaller than 4 mm if $\boldsymbol{S}$ intersects any adjacent polygons. These polygons are collected for the visualization and for outputting the data in an STL file.

The tessellated surface of a convex shape always exists on the internal side of the original surface, as shown in Fig. 2. The internal volume of the tessellated object near the convex shape becomes smaller than the internal volume of the original shape. If a sphere of radius r can be placed on the backside of a polygon $\boldsymbol{f}$ of the tessellated object, then its corresponding original convex surface can contain the same sphere within. Our sphere placement method checks the roundness of the convex shape on a safe side when the sag value t is sufficiently small.

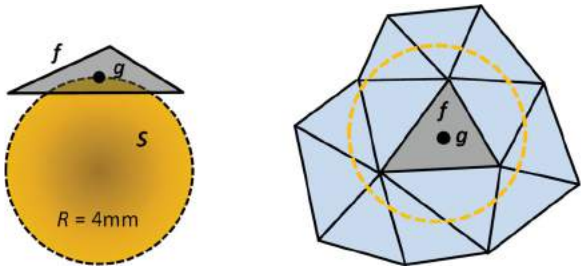

Figure 4. Evaluation of the radius of curvature of polygon ***f*** using the sphere placement method.

2.3. Related studies

One possible solution for extracting the cylinder-contacting shape is to use the collision detection technology of 3D objects. Established technologies in this field are described in [9]. [15, 16, 24] present the use of the GPU's parallel processing technology for accelerating the collision detection. In this method, a manual inspection with physical apparatus is replaced by a virtual process with solid models on computers. A solid model of the vertical cylindrical column of 300 mm in diameter is prepared in various positions. The motorcycle model is then moved in a straight line towards the column to determine its first point of collision on the body. This approach is simple and easy to implement; however, its cost is estimated to be large because a tremendous number of collision detections with column models in different positions are required.

In the automobile industry, UNECE (United Nations, The Economic Commission for Europe) safety regulations are defined by the external shape of the body. There are some software systems available for verifying the regulations (for example, CAVA system [6]). Toyota Motor Corporation submitted some patents relevant to the automatic inspection of the UNECE regulations concerning collisions of a large sphere to represent an infant's head [14]. Yamazaki et al. proposed a regulation inspection system based on detecting the intersection between a CAD model and spheres placed on the surface of the model [28]. The current authors proposed a method for the rapid detection of the sphere-contacting shape for evaluating UNECE regulations 17, 21, and 25 [11, 12]. These systems only detect the sphere contacting area on automobile parts and are not relevant to the regulations concerning the collisions of the cylindrical column.

Our inspection method uses the two dimensional offset shape of the part model. Offsetting is one of the most fundamental operations in geometric modeling. The offsetting operations for curves and surfaces are well known [10, 19, 23]. In the offsetting of 2D shapes, the system must handle both the generation of individual offset curves as well as topological reconstruction by trimming and reconnecting the curves into a complete one. New offset computation methods based on the discrete representation of 2D/3D models have become popular. Known representation schemes utilize points [8], voxels [17, 18], distance field [3, 4], Layered Depth Images (LDI) [26, 27], and triple-dexels [12] have been reported. As discrete models do not have boundary elements, the topological reconstruction step, which is the most critical process in conventional offsetting, is not necessary. In this paper, we utilize two-directional dexels for representing the offset shape of 2D closed figure. This method is an adaptation of the triple-dexel model for representing the 2D figure.

In our inspection method, the roundness of a point on the polyhedral model is necessary. The Plumber algorithm [21] can extract a cylindrical shape as a series of circular disks from the polyhedral model. This algorithm is not applicable for extracting sharp corners or fillets with small radii. Principal curvatures are known as the measures of the roundness of a curved surface. In classical differential geometry, these invariants are defined for a smooth surface of C^2 continuity. As the use of discrete triangular meshes has increased, various technologies for estimating the curvature on a polyhedral surface have been developed. They can be classified into two groups. The first is the formulation of discretized formula representing the geometric properties of the curvature (for example [2]). The second is based on the local fitting or interpolation method (for example [7]). Local geometric characteristics of the mesh are analytically computed using the fitted surface. These methods cannot evaluate the roundness at a sharp corner, which is the critical shape in the inspection of the regulation.

Our algorithm has the following three novel features that differ from the prior works;

- The column contacting shape is extracted by moving the vertical column while maintaining contact with the motorcycle body. The ruled surface of the moving column is obtained by offsetting and shrinking the projection of the body shape to the xy-plane.
- Offsetting and shrinking operations are realized by Boolean operations of 2D figures in a discrete representation, named the two-directional dexel model. The parallel processing capability of the GPU is utilized to accelerate the Boolean computations.
- In the roundness evaluation of a column contacting polygon ***f*** in a triangular mesh, we use the sphere placement method. The original method [28] cannot evaluate the roundness of a thin shape. This limitation is resolved using only polygons surrounding ***f*** in the roundness evaluation.

3. Details of the algorithm

3.1. Step 1: Detection of column contacting polygons

3.1.1. Offsetting of a projection in the xy-plane

The motorcycle shape is projected onto the xy-plane. The projection is then horizontally expanded by 150 mm to obtain its offset shape. A two-directional dexel model is used to represent the offset shape. In the original dexel model [25], the 3D object is represented by a series of vertical segments (dexels) defined at each grid point in a regular square grid in the xy-plane. Each segment corresponds to an overlapping range between the vertical ray that originates from each grid point and the object. In this method, the object shape is represented by a bundle of vertical segments, as shown in Fig. 5(a). In the dexel model, near-vertical surfaces have inevitable staircase errors caused by the finite grid resolution. A triple-dexel model was proposed to overcome this non-uniformity of the representation accuracy [1]. In this representation, the 3D shape is not only defined by the z-axis-aligned (vertical) dexels, but also the x-axis-aligned dexels based on a grid in the yz-plane and y-axis-aligned dexels based on a grid in the zx-plane (see Fig. 5(b)).

The two-directional dexel model is an adaptation of the concept of the triple-dexel model for representing a 2D figure with closed boundaries. In this method, a square mesh with x-axis-aligned lines and y-axis-aligned lines is defined in the xy-plane. For each line, the overlapping range of the line with respect to the 2D figure is computed and the horizontal segments (dexels) corresponding to the overlapping regions are obtained. Finally, the figure is represented as a set of x-axis-aligned dexels and y-axis-aligned dexels in the xy-plane. Fig. 6 illustrates an offset shape of the projection of the motorcycle body in the two-directional dexel representation. In this method, the interval distance between the parallel lines of the mesh limits the accuracy of the result shape. In our current implementation, the square mesh is defined so that the interval distance is less than 1 mm.

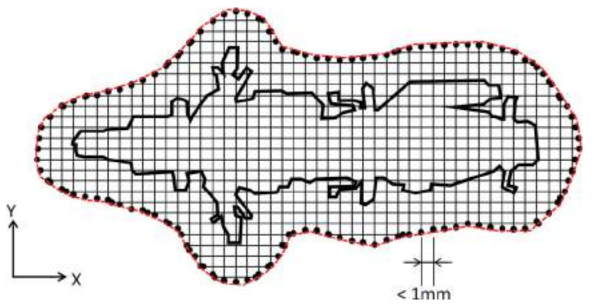

Figure 6. An offset shape of the projection of motorcycle in the two-directional dexel representation.

In the following sections, we explain the offset computation with the y-axis-aligned dexels. In the actual processing, the same operations are repeated for the x-axis-aligned dexels to obtain a complete two-directional dexel model. Before processing, a null y-axis-aligned dexel model (the Offset Dexel Model, ODM) is prepared. The ODM is updated in the following computation, and then forms the offset shape of the projection. A projection for each polygon of the motorcycle model is applied to the xy-plane (see Fig.7(a)). The projection of the whole body is thus obtained as a set of (mutually overlapping) horizontal triangles in the plane.

The offset shape of the triangles in the xy-plane by 150 mm is equivalent to a Boolean union shape of

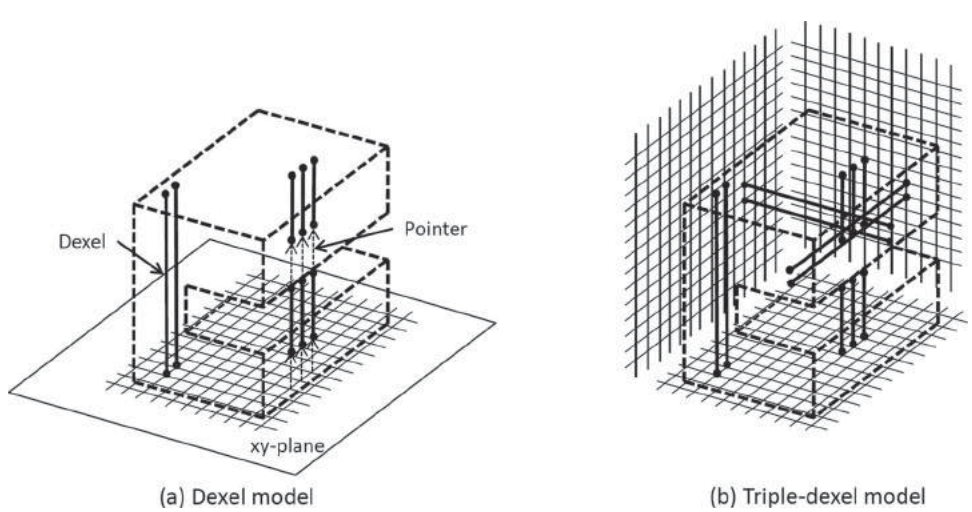

Figure 5. Dexel model and triple-dexel model for representing a 3D object.

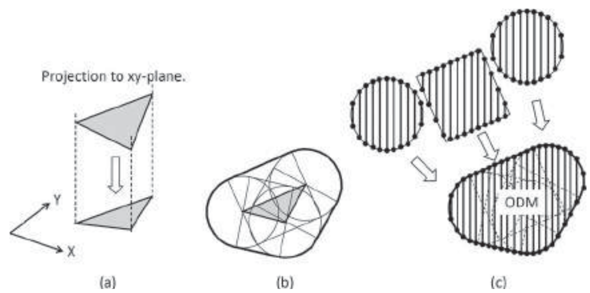

Figure 7. The offset computation of a triangle.

horizontal disks and horizontal rectangles defined as follows (see Fig.7(b)):

- Disks of radius 150 mm are placed on all vertices of the projected triangles.
- Rectangles of 300 mm in width are placed along each edge of the projected triangles, with the center line of the rectangle coinciding with the edge.

Every projected triangle is not necessary in the Boolean union computation because disks of radius 150 mm or rectangles of width 300 mm usually contain the original triangles within, as shown in the figure.

After the definition, each disk is converted to a y-axis-aligned dexel model. Each rectangle is also converted to its equivalent dexel model. The dexel-wise Boolean union operation for each dexel model and the ODM is computed, and the result gives the new ODM (Fig.7(c)). This process is iterated for all the dexel models of the disks and rectangles.

The Boolean union computation of dexels on one y-axis-aligned line is independent of those on other lines. Thus, the dexel-wise Boolean union computation can be parallelized. To implement the parallel offsetting software, we use Compute Unified Device Architecture (CUDA)[22]. Current GPUs are designed to have thousands of small streaming processors (SP) on a chip. By using CUDA, programmers can utilize a GPU as a general purpose parallel processor in which each SP executes a computation unit (or thread). Using CUDA, programmers can specify the execution of up to $65535 \times 65535 \times 512$ threads. The GPU automatically schedules the execution process of threads, and assigns as many threads as possible on available SPs.

For each circular disk, y-axis-aligned lines crossing the disk are selected based on the center point of the disk, its radius ($=$ 150 mm), and the coordinate information of the square mesh in the xy-plane. For each selected y-axis-aligned line, a single CUDA thread is assigned for computing an intersection segment between the line and the disk. The same thread then executes the dexel-wise

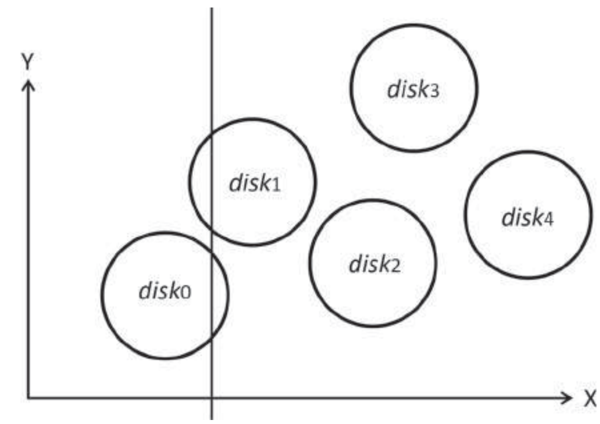

Figure 8. Arrangement of the disks for a simultaneous Boolean union computation of multiple disks.

Boolean union computation of the segment and the dexels of the ODM on the same y-axis-aligned line. After the parallel execution of all threads, a new ODM containing the disk shape within is obtained. This operation is iterated for all disks. Similar computations are executed for all rectangles and a y-axis-aligned dexel model representing the offset shape is obtained in the graphics memory.

To further utilize the parallel processing capability of a GPU, the offset computation for a single disk or for a single rectangle is extended to parallel offset computations with multiple disks or multiple rectangles. In this case, we need a mechanism to avoid conflicts in the dexel-wise Boolean operation. Consider Boolean union computations for $disk_0$ and $disk_1$ being simultaneously invoked. Fig. 8 illustrates a possible arrangement of $disk_0$ and $disk_1$ in the xy-plane. As $disk_0$ and $disk_1$ intersect a same y-axis-aligned line, a CUDA thread for $disk_0$ and another thread for $disk_1$ may update dexels on the same y-axis-aligned line simultaneously. In this case, the dexel structure constructed by a thread for $disk_0$ can be improperly modified by another thread for $disk_1$, and vice versa. To avoid such conflicts in our parallel offsetting framework, multiple disks or rectangles must be selected so that they do not intersect the same y-axis-aligned line, for example $disk_0$, $disk_2$, and $disk_4$, as shown in Fig.8.

3.1.2. Shrinking of offset figure to obtain ruled surface

After the offset computation, the vertical cylindrical column is moved along the boundary curve of the offset shape to detect the column contacting polygons on the motorcycle body. The end points of the two-directional dexels of the offset shape correspond to the points in the boundary curve. The column contacting polygon is thus obtained by checking the distance between the vertical columns placed on the points and polygons

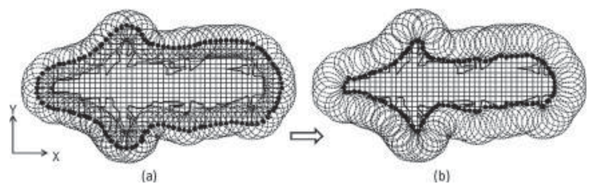

Figure 9. Shrinking operation of the offset shape by subtracting disks with radius 150 mm.

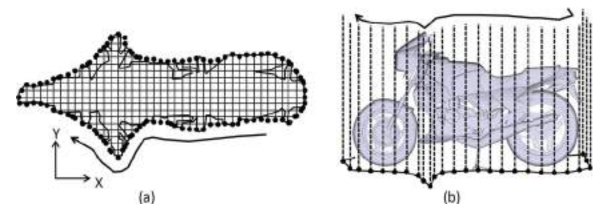

Figure 10. A ruled surface obtained by moving a vertical line along the boundary curve of the shrunken shape.

in the motorcycle body. To simplify the distance computation, we consider the swept volume of the cylindrical column moving along the boundary curve and compute the inner boundary surface of the swept volume. The distance computation is done between this boundary surface and the polygons of the motorcycle body.

There are some software packages capable of computing the swept volume of a moving object, such as IRIT [13]. In our study, only the inner boundary surface of the swept volume of the vertical column is necessary. This surface is easily obtained by shrinking the offset shape in the xy-plane by 150 mm. For each end point of the y-axis-aligned dexels and x-axis-aligned dexels of the offset shape model, a disk of radius 150 mm is applied, as shown in Fig.9(a). These disks are subtracted from the given dexel model of the offset shape. This operation is repeated for all end points of the dexel model and the shrunken shape of the offset shape is obtained (see Fig.9(b)). In the dexel-wise subtracting operation, a parallel algorithm similar to the dexel-wise Boolean union computation is used.

After the shrinking operation for the y-axis-aligned dexels and x-axis-aligned dexels, the dexel information is transferred from the graphics memory to the memory of the CPU and the boundary curve of the shrunken shape is computed by using CPU. A vertical line is moved along the boundary curve. A ruled surface organized by the moving line corresponds to the inner surface of the swept volume of the moving column (see Fig.10). For each polygon of the motorcycle body, the distance between the ruled surface and the polygon is checked. If the distance is sufficiently small, then the polygon is selected as the polygons contacting to the vertical column of 300 mm in diameter.

3.2. Step 2: Evaluation of roundness

After the extraction of the column-contacting-polygons on the motorcycle body, the radius of curvature of the polygons is evaluated. For each column-contacting-polygon $\boldsymbol{f}$, a sphere $\boldsymbol{S}$ of radius 4 mm is placed so that it contacts the backside of the polygon at its center of gravity $\boldsymbol{g}$. Intersections between $\boldsymbol{S}$ and polygons adjacent to $\boldsymbol{f}$ are then checked. The radius of curvature of $\boldsymbol{f}$ is judged smaller than 4 mm if $\boldsymbol{S}$ intersects some adjacent polygons.

In this method, polygons adjacent to $\boldsymbol{f}$ must be selected for each column-contacting-polygon. As STL models do not record such adjacency information between polygons, it must be extracted from the triangle data. In our implementation, 2 triangles, $\boldsymbol{f}_0$ and $\boldsymbol{f}_1$, are judged adjacent if they share a same line segment, more precisely, an edge e_0 of triangle $\boldsymbol{f}_0$ and another edge e_1 of triangle $\boldsymbol{f}_1$ overlap on the same line segment.

To prepare data that is suitable for efficiently detecting the adjacent polygons around a specific polygon $\boldsymbol{f}$, the surface polygons of the input model are classified into small groups according to their proximity. This classification proceeds according to the hierarchical structure of the Axis-Aligned Bounding Box (AABB) [20]. Consider n triangular polygons forming the model surface. An AABB that tightly confines the polygons is defined by measuring the coordinate ranges of the polygons in the x-, y-, and z-directions. One root, AABB, is defined that holds all the polygons of the model. Polygons in the AABB are sorted along a line parallel to its longest axis. Then, two child AABBs are formed by the first $n/2$ sorted polygons and the remaining polygons, respectively. The process of defining descendant AABBs is iterated until the number of the polygons in a group becomes less than or equal to a predetermined number $nmax$, and a binary AABB tree is obtained. $nmax$ is set to be 4 in our current implementation. This number is determined based on numerical experiments with some sample motorcycle models.

Polygons sufficiently close to a given polygon $\boldsymbol{f}$ are obtained by traversing the hierarchical AABB tree in a depth-first manner. An axis-aligned bounding box tightly contains $\boldsymbol{f}$ within in defined. This box is given to the root node of the AABB tree. At each node nd in the tree, the AABB containing $\boldsymbol{f}$ and an AABB corresponding to nd is compared. If they do not overlap, the polygons contained within the AABB at the node are not adjacent to $\boldsymbol{f}$

and stop further processing from the node. If they have potential overlap, then;

- If node *nd* is a leaf node in the tree, a set of polygons within AABB of the node is selected as candidates of the adjacent polygons of ***f***.
- Otherwise, traverse the child nodes of *nd* with an AABB of ***f***.

After the tree traversal, adjacency relationship between ***f*** and obtained candidate polygons is checked.

4. Numerical experiments

A system for extracting cylindrical column contacting polygons and for evaluating their roundness was implemented using Visual C++, CUDA 7.5, and OpenGL, and a series of computational experiments were performed using a PC with an Intel Core i7 Processor (2.6 GHz), 16 GB memory, and an nVIDIA GeForce GTX-960M GPU. We applied the system to four polyhedral models of motorcycles. Two models were provided by a motorcycle manufacturer in Japan, and the other models were selected from CAD demonstration models. The system can successfully extract the polygonal areas in the motorcycle body which contact the cylindrical column of 300 mm in diameter in the forward motion and whose radius of curvature is less than 4 mm. The required computation time is usually less than 1 min, as shown in Tab. 1. Most of the computation time is consumed in the detection of the column-contacting-polygon in step 1.

For the purpose of maintaining confidentiality, only the computation result for two CAD demonstration models is shown here. Fig.11 illustrates sample model A with 437,040 polygons (Fig.11(a)) and sample model B with 2,700,707 polygons (Fig.11(b)). The result of the offsetting and shrinking the projection of the model is also shown in blue in the same figure. A ruled surface is generated by moving a vertical line along the boundary of the shrunken figure. Fig.12 illustrates the ruled surface in blue. By using the contact analysis between the ruled surface and surface polygons of the motorcycle model, polygons contacting the cylindrical column of diameter 300 mm in the forward motion are extracted as green polygons in Fig.13. In these polygons, polygons whose radius of curvature are less than 4mm are illustrated in red in Fig.14.

Table 1. Required time for inspection.

Model	Number of polygons	Grid resolution for dexel model	Time for inspection (s)
A	437,040	2054 X 1018	24.75
B	2,700,707	2055 X 1017	63.02
C	673,958	2071 X 1001	30.10
D	429,454	2128 X 944	26.87

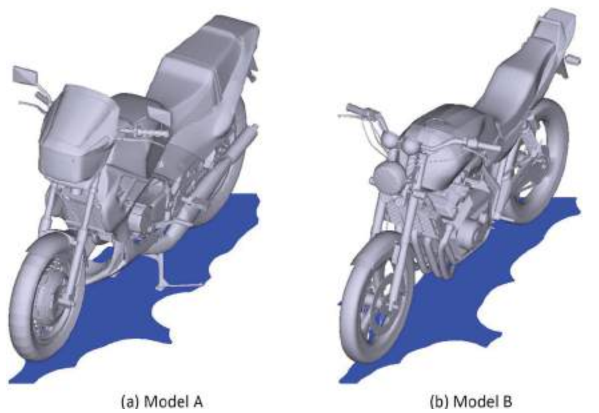

Figure 11. Sample model and offsetting and shrinking result of the projection of the model shown in blue.

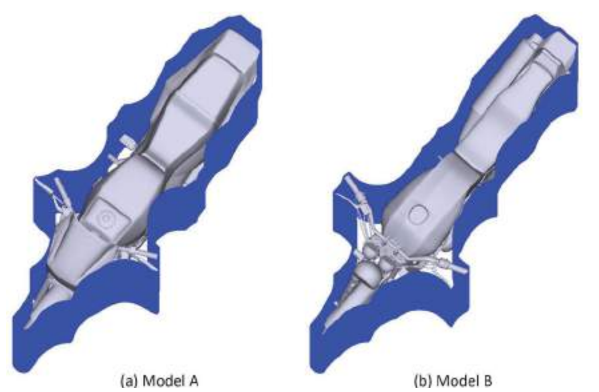

Figure 12. A ruled surface of a vertical line moving along the boundary curve of the shrunken shape.

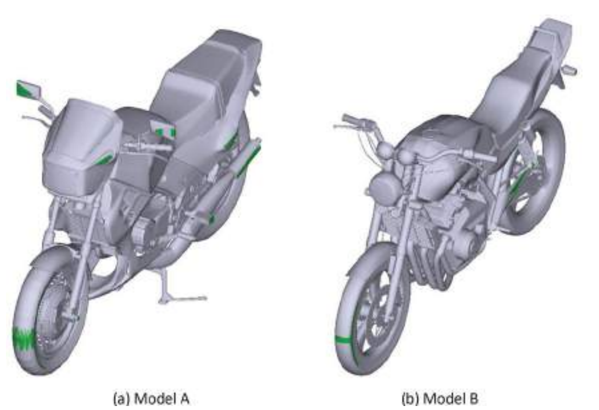

Figure 13. Polygons where the vertical column of 300 mm diameter contacts in the forward motion of the motorcycle.

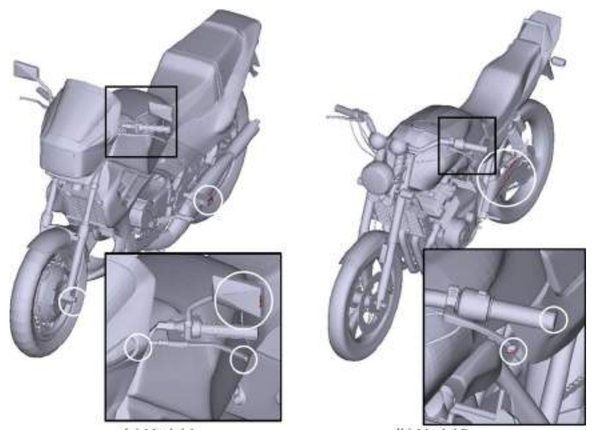

Figure 14. Column-contacting-polygons whose radius of curvature is less than 4 mm.

5. Conclusions

In this paper, we have proposed a novel method for assisting in the safety regulation inspection of motorcycles. This method extracts the surface regions on the motorcycle body where a vertical cylindrical column having a diameter of 300 mm contacts and the surface region radius is less than R4. In the first step of the algorithm, the column contacting polygons are extracted on the motorcycle model. This is achieved by offsetting and shrinking a projection of the motorcycle shape by 150 mm. Parallel processing capability of GPU is used for accelerating the operation. In the second step, the roundness of the extracted polygon is evaluated and all polygons with a radius of curvature less than 4 mm are selected. The sphere placement method is used in the curvature evaluation. In this method, polygons adjacent to a target polygon from the roundness evaluation are identified. To efficiently prepare candidates of the adjacent polygons, a hierarchical structure of the boxes bounding the surface polygons of the model is used. An experimental system is implemented and computational results are demonstrated. This system detected surface areas on a motorcycle body where the safety regulation were not satisfied within a minute in our sample cases.

Acknowledgments

This research work was supported by JSPE KAKENHI Grant Number 17K06075.

ORCID

Masatomo Inui http://orcid.org/0000-0002-1496-7680
Nobuyuki Umezu http://orcid.org/0000-0002-7873-7833

References

[1] Benouamer, M.O.; Michelucci, D.: Bridging the Gap between CSG and Brep via a Triple Ray Representation, In: *Proceedings of ACM Symposium on Solid Modeling and Applications*, 1997, 68–79. doi:10.1145/267734.267755
[2] Borrelli, V.; Cazals, F.; Morvan, J.M.: On the angular defect of triangulations and the pointwise approximation of curvatures. *Computer Aided Geometric Design*, 20, 2003, 319–341. http://doi.org/10.1016/S0167-8396(03)00077-3
[3] Breen, D.E.; Mauch, S.: Generating Shaded Offset Surfaces with Distance, Closest Point and Color Volumes, Proceedings of International Workshop on Volume Graphics, 1999, 307–320.
[4] Breen, D.E.; Mauch, S.; Whitaker, R.T.: 3D Scan Conversion of CSG Models into Distance Volumes. *Proceedings of IEEE Symposium on Volume Visualization*. 1998, 7–14.
[5] CATIA, https://www.3ds.com/products-services/catia/products/v5/portfolio/domain/Machining/product/STL/
[6] CAVA Systems, http://www.edstechnologies.com/index.php/component/content/article/86-solution/cad-quality-check/135-cava.html, EDS Technologies.
[7] Cazals, F.; Pouget, M.: Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, 22(2), 2005, 121–146. http://doi.org/10.1016/j.cagd.2004.09.004
[8] Chen, Y.; Wang. H.; Rossen D.W.; and Rossignac J.: Filleting and Rounding Using a Point-Based Method. In: *Proceedings of ASME 2005 International Design Engineering Technical Conference*, 2005, 533–542.
[9] Ericson, C.: *Real-Time Collision Detection*, Morgan-Kaufmann, 2005.
[10] Hoffman, C.M.: *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann. 1989.
[11] Inui, M.; Umezu, N.: Fast detection of head colliding shapes on automobile parts, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 7(5), 2013, 818–826. http://doi.org/10.1299/jamdsm.7.818
[12] Inui, M.; Umezu, N.; Kitamura, Y.: Visualizing sphere-contacting areas on automobile parts for ECE inspection, *Journal of Computational Design and Engineering*, 2(1), 2015, 55–66. http://doi.org/10.1016/j.jcde.2014.11.006
[13] The IRIT modeling environment, http://www.cs.technion.ac.il/ ~ irit/
[14] Japanese Patent [P2006-277304A], http://www.j-tokkyo.com/2006/G06F/JP2006-277304.shtml, Toyota Motor Corporation, in Japanese.
[15] Kim, D.; Heo, J.-P.; Huh, J.; Kim, J.; Yoon, S.-E.: HPCCD: Hybrid Parallel Continuous Collision Detection, in Eurographics'09, 2009, 1791–1800.
[16] Lauterbach, C.; Mo, Q.; Manocha, D.: gProximity: Hierarchical GPU-based Operation and Distance Queries, in Eurographics'10, 2010, 419–428.
[17] Li, W.; McMains, S.: A GPU-Based Voxelization Approach to 3D Minkowski Sum Computation. In: *Proceedings of the 14th ACM Symposium on Solid and Physical Modeling*, 2010, 31–40.
[18] Li, W.; McMains, S.: Voxelized Minkowski Sum Computation on the GPU with Robust Culling. *Computer-Aided Design*, 43(10), 2011, 1270–1283.

[19] Maekawa, T.: An Overview of Offset Curves and Surfaces. *Computer-Aided Design*. 31, 1999, 165–173. http://doi.org/10.1016/S0010-4485(99)00013-5

[20] Moller, T.; Haines, E.: *Real-Time Rendering*. A. K. Peters Ltd., Natick MA, USA, 1999.

[21] Mortara, M.; Patane, G.; Spagnuolo, M.; Falcidieno, B.; Rossignac, J.: Plumber: a method for a multi-scale decomposition of 3D shapes into tabular primitives and bodies, *Proceedings of the ninth ACM symposium on Solid modeling and applications*, 2004, 339–344.

[22] nVIDIA, CUDA Compute unified device architecture programming guide, 2007.

[23] Pham, B.: Offset Curves and Surfaces: A Brief Survey. *Computer-Aided Design*. 24(4), 1992, 223–229. http://doi.org/10.1016/0010-4485(92)90059-J

[24] Thomaszewski, B.; Pabst, S.; Blochinger, W.: Parallel Techniques for Physically-Based Simulation on Multi-Core Processor Architectures," in *Computers and Graphics*, 1(31), 2008, 25–40.

[25] VanHook T.: Real-Time Shaded Milling Display, *Computer Graphics*, 20(4), 1986, 15–20. http://doi.acm.org/10.1145/15922.15887

[26] Wang, C.C.L.; Manocha, D.: GPU-Based Offset Surface Computation using Point Samples. *Computer-Aided Design*, 45(2), 2013, 321–330.

[27] Wang, C.C.L.; Computing on Rays: A Parallel Approach for Surface Mesh Modeling from Multi-Material Volumetric Data, *Computers in Industry*, 62(7), 2011, 660–671.

[28] Yamazaki, S.; Baba, T.; Umezu, N.; Inui, M.: Fast safety verification of interior parts of automobiles, *Proceedings of IEEE International Conference on Mechatronics and Automation (ICMA)*, 2011, 1957–1962. http://doi.org/10.1109/icma.2011.5986280