Title:
Extraction of Vertical Cylinder Contacting Area for Motorcycle Safety Verification

Authors:
Masatomo Inui, masatomo.inui.az@vc.ibaraki.ac.jp, Ibaraki University
Nobuyuki Umezu, nobuyuki.umezu.cs@vc.ibaraki.ac.jp, Ibaraki University

Keywords:
Contact Analysis, Offsetting, Curvature Evaluation, Parallel Processing, GPU

DOI: 10.14733/cadconfP.2017.149-153

Introduction:
Japanese motorcycle manufactures 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. Consider a vertical cylindrical column 300 mm in diameter. This column 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 able to make contact. According to this regulation, such surface areas must have a radius greater than R3.2 (usually, R4.0 is used instead to be conservative).

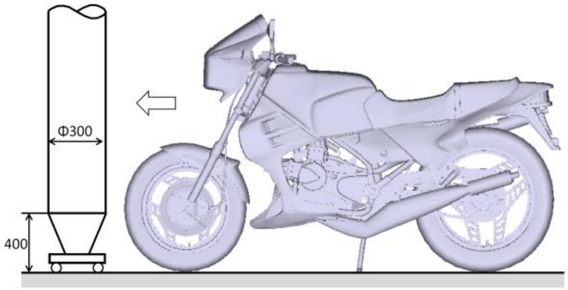

Fig. 1: Safety regulation on the motorcycle.

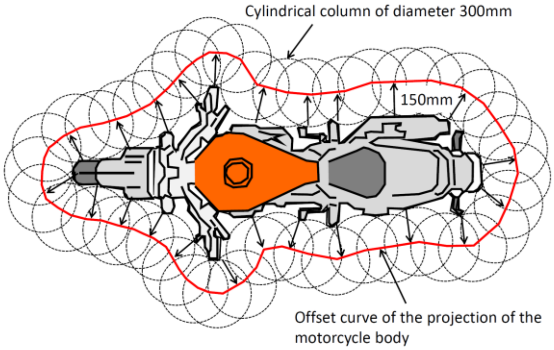

Fig. 2: Cylindrical column sliding on the external surface of a motorcycle.

Currently, the safety regulation regarding the shape is inspected by specialists at the final design stage with physical apparatus and models. This work is cumbersome, time consuming, and prone to human errors, and the detection of safety issues at this stage causes costly re-works. A fast and automatic inspection method that allows designers to check the regulations themselves during the shape design process is desirable. To assist designers, we propose a novel method for extracting surface regions on a motorcycle body where the vertical cylindrical columns collide and their radius is less than R4.0.

Our algorithm has the following three novel features that differ from the prior works. (1) The column colliding shape is extracted by moving the vertical column while maintaining contact with the motorcycle body. The ruled surface generated by the moving column is obtained by offsetting and shrinking the projection of the body shape to the xy-plane. (2) 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. (3) In the roundness evaluation of a column colliding polygon in a triangular mesh, we use sphere placement method. The original method [3] cannot evaluate the roundness of a thin shape. This limitation is resolved using only polygons surrounding the column colliding shape.

Main Idea:

Algorithm outline
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. The system extracts the column contact area whose radius of curvature is less than 4.0 mm on the display and outputs the area data as a set of small polygons in STL format. Column contacting areas higher than 400 mm from ground level are only checked in the actual inspection (see Fig. 1). To simplify the problem, this condition is not considered in this work.

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. 2). 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. 2. After the extraction of the column-contacting-polygons, polygons facing the backside of the motorcycle are discarded because the column never contacts these polygons when the motorcycle is moving in the forward direction.

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 [3]. 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. 3. Intersections between $\boldsymbol{S}$ and the polygons adjacent to $\boldsymbol{f}$ (polygons shaded in the light blue color in Fig. 3) are checked. The radius of curvature of $\boldsymbol{f}$ is judged to be smaller than 4 mm if $\boldsymbol{S}$ intersects any adjacent polygons.

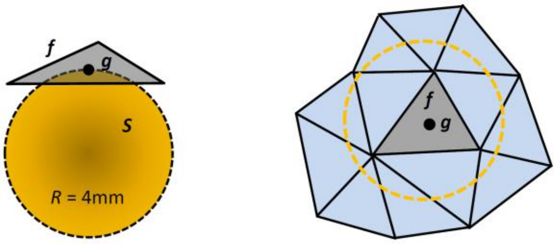

Fig. 3: Evaluation of the radius of curvature of polygon $\boldsymbol{f}$ using the sphere placement method.

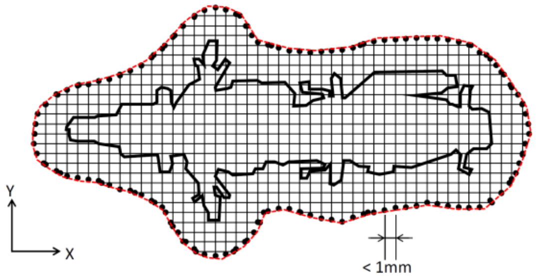

Fig. 4: An offset shape of the projection of a motorcycle in the two-directional dexel representation.

Offsetting projection figure of motorcycle (step 1)
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 [2], the 3D object is represented by a series of vertical segments (dexels) defined for each grid point in a regular square grid in the xy-plane. Each segment corresponds to an overlapping range between the vertical ray from the grid point and the object. The two-directional dexel model is an adaptation of the concept of the 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. 4 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.

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. 5(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 horizontal disks and horizontal rectangles defined as follows (see Fig. 5(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 temporal y-axis-aligned dexel model. Each rectangle is also converted to its equivalent temporal dexel model. The dexel-wise Boolean union operation for each temporal dexel model and the ODM is computed, and the result gives the new ODM (Fig. 5(c)). This process is iterated for all the temporal dexel models of the disks and rectangles.

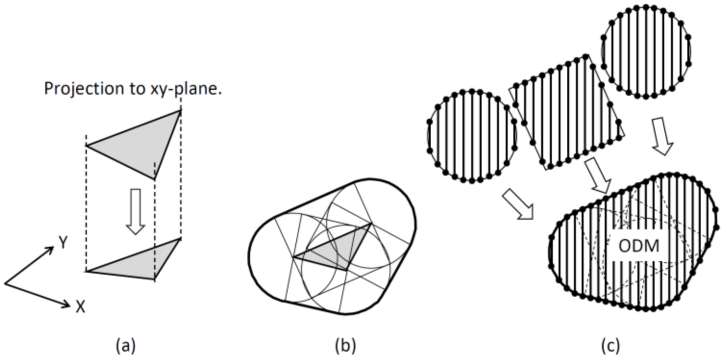

Fig. 5: The offset computation of a triangle.

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). 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 unit computation (or thread). A single CUDA thread is assigned to each y-axis-aligned line. This thread computes a set of dexels for the temporal dexel models on the line. It then executes the dexel-wise Boolean union computation of the dexels on the same line [1].

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 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.

The inner boundary surface of the swept volume of the moving column is 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. 6(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 shrunk shape of the offset shape is obtained (see Fig. 6

(b)). A vertical line is moved along the boundary curve of the shrunk shape. A ruled surface organized by the moving line corresponds to the inner surface of the swept volume of the column (see Fig. 7).

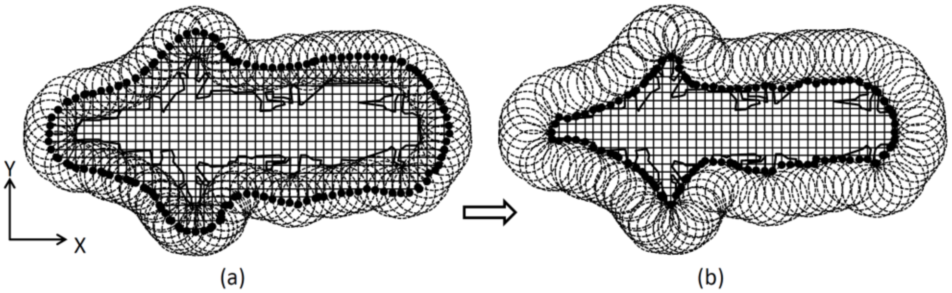

Fig. 6: Shrinking operation of the offset shape by subtracting disks of radius 150mm.

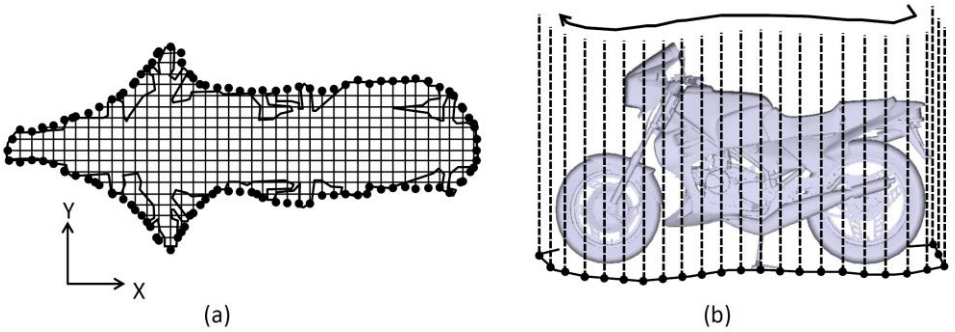

Fig. 7: A ruled surface by moving a vertical line along the boundary curve of the shrunk shape.

Contact analysis of spheres for evaluating the curvature (step 2)

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 ***f***, a sphere ***S*** of radius 4 mm is placed so that it contacts the backside of the polygon at its center of gravity ***g***. Intersections between ***S*** and polygons adjacent to ***f*** are then checked. The radius of curvature of ***f*** is judged smaller than 4 mm if ***S*** intersects some adjacent polygons. In this method, polygons adjacent to ***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, 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.

Numerical experiment

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 three polyhedral models of motorcycles. Two models were provided by a motorcycle manufacturer in Japan, and the other model was 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 less than 1 min for any models. Most of the computation time is consumed in the detection of the column-contacting-area in step 1.

For the purpose of maintaining confidentiality, only the computation result for a CAD demonstration model is shown here. Fig. 8 illustrates the sample model with 437,040 polygons. 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 shrunk figure. Fig. 9 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. 10. In these polygons, polygons whose radius of curvature are less than 4 mm are illustrated in red in Fig. 11.

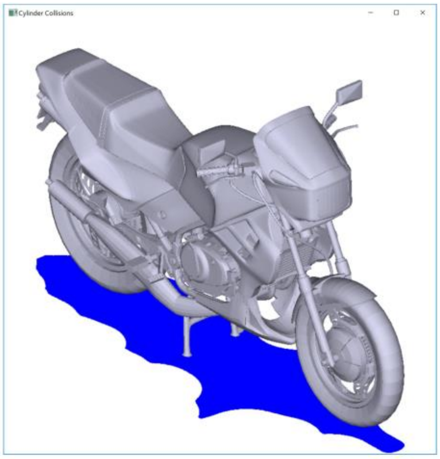

Fig. 8: Sample model and offsetting and shrinking result of the projection of the model in blue color.

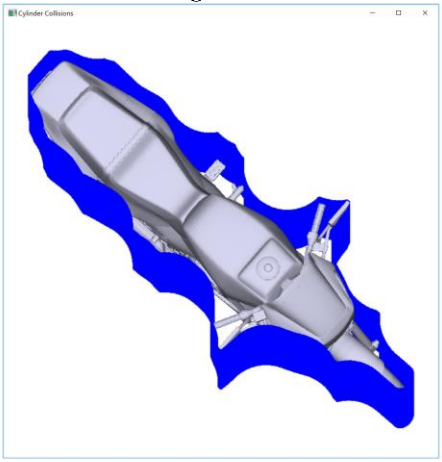

Fig. 9: A ruled surface of a vertical line moving along the boundary curve of the shrunk shape.

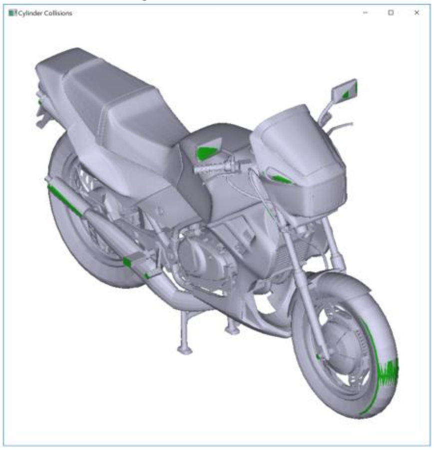

Fig. 10: Polygons where the vertical column contacts.

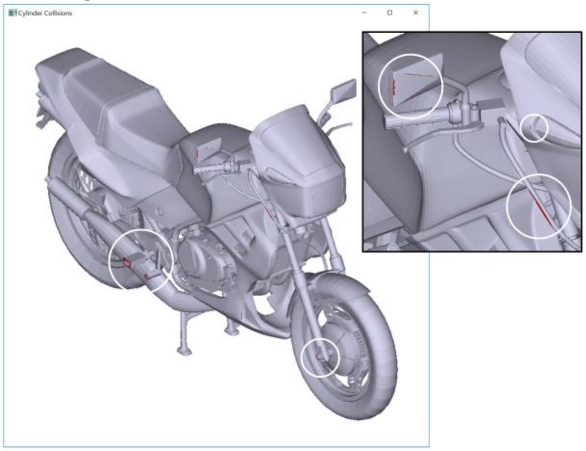

Fig. 11: Column-contacting-polygons whose radius of curvature is less than 4mm.

Conclusions:

In this paper, we have proposed a novel method for assisting a safety regulation inspection of the motorcycle. Our method can extract and visualize the surface area of the motorcycle body where the vertical cylindrical column of diameter 300mm collides and whose radius of curvature is less than 4mm. With the parallel computation capability of a GPU, our system can extract the result area in one minute.

References:

[1] 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://dx.doi.org/10.1016/j.jcde.2014.11.006

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

[3] Yamazaki, S.; Baba, T.; Umezu, N.; Inui M.: Fast Safety Verification of Interior Parts of Automobiles, Proc. of 2011 International Conference on Mechatronics and Automation, 2011. DOI: 10.1109/ICMA.2011.5986280