Computer-AidedDesign

Taylor & Francis
Taylor & Francis Group

GPU-based visualization of knee-form contact area for safety inspections

Masatomo Inui [iD][a], Shunsuke Nakano [iD][a], Nobuyuki Umezu [iD][a] and Tetsuya Asano [iD][b]

[a]Ibaraki University, Japan; [b]Aikoku Alpha Corporation, Japan

ABSTRACT

The United Nations defines a safety regulation based on the possible collision between the driver's knee and an automobile's instrument panel. The "knee-form" apparatus used to evaluate compliance with this regulation can be modeled as a Minkowski sum shape of a vertical equilateral triangle and a horizontal cylinder with a radius of 60 mm and a thickness of 120 mm. The knee form contacting condition is geometrically equivalent to that of an equilateral triangle contacting a Minkowski sum shape of the instrument panel and a horizontal cylinder. Based on this concept, we propose a novel algorithm for extracting the knee-form contacting area on the instrument panel. With the parallel computation capability of a Graphics Processing Unit, our system can detect and output the knee-form contacting area in a practical time period.

1. Introduction

Safety is an important concern in automobile design. In driving an automobile, acceleration and deceleration are realized by the driver's leg and foot motion. If the shape of the instrument panel in front of the driver hinders the smooth motion of the knee, the operability of the automobile is significantly impaired. If the knee touches electric switches while driving, it may cause traffic accidents in the worst case. To prevent such problems, the United Nations Economic Commission for Europe (UNECE) defines a safety regulation based on the possible collision between the knee and the instrument panel [6].

Fig. 1 illustrates the UN regulation No. 21 concerning the knee collision. In CAD models of the automobile design, the instrument panel is usually positioned so that it faces the positive direction of the x-axis, and an apparatus imitating the knee shape during driving is defined. This "knee-form" has an equilateral triangle-like shape and a thickness, *t*, of 120 mm. It has a rounded corner that can be represented by a cylinder with a radius of 60 mm as shown in Fig. 1(a). To inspect for compliance with the regulation, the knee-form is represented as moving linearly in the negative direction of the x-axis, where it eventually pushes into the instrument panel. In this way, as shown in Fig. 1(b), surface areas on the panel are detected where the knee can make contact during driving. Before representing the movement, the apparatus can be rotated above and below the horizontal by up to 30 degrees to represent different orientations.

According to the regulation, any switches, nobs, buttons, and levers are not allowed in the knee contacting area. In the instrument panel design, the designer is requested to define a curve on the panel surface that subdivides the upper part and lower part of the panel. This curve is referred to as the instrument panel level or IP level. Because the knee can contact only the lower part of the panel, the knee-form contact analysis for the upper part of the panel is not necessary. During actual driving, the steering wheel also limits the knee motion. In the analysis, the linear motion of the knee-form can be disregarded when the knee-form collides with the steering wheel before contacting the instrument panel.

Because interior components, especially the instrument panel, significantly affect the appearance and comfort of the automobile, they are often designed in terms of their function and aesthetics. Knee-form contact detection is manually inspected by specialists in the final design stage with a physical apparatus and with models. This work is difficult, time consuming, and prone to human errors, and detection of safety problems at this stage causes costly re-design and fabrication. Therefore, a fast and automatic inspection method that allows the designers themselves to check the regulations during the shape design process is desirable.

Although UN regulation inspections are an important topic for automobile manufacturers, technologies for automating the inspection tasks have rarely been studied, and many manufacturers still use manual inspection
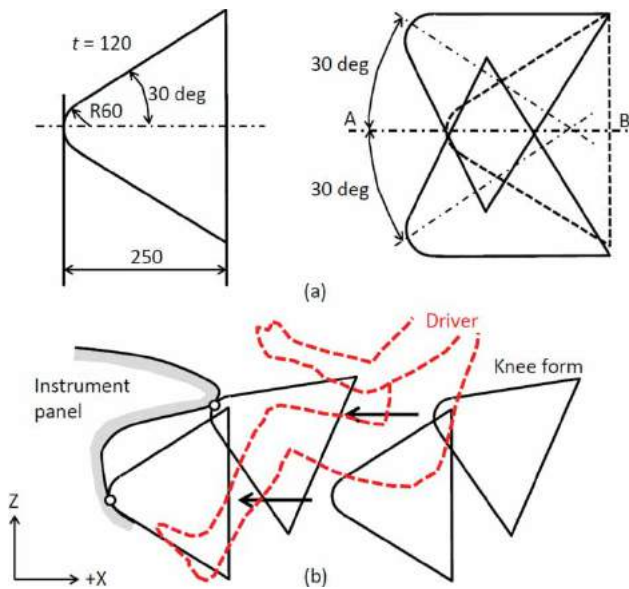

CONTACT Masatomo Inui ✉ masatomo.inui.az@vc.ibaraki.ac.jp

Figure 1. Knee-form contact analysis as defined in the UN regulation No. 21.

methods. Some commercial systems are known for verifying the UN regulations, such as the CAVA system [2]. The Toyota Motor Corporation submitted some patents relevant to the automatic inspection of UN regulations 17, 21, and 25, concerning collisions of a large sphere representing an infant's head [16]. These systems detect only the sphere contacting area on automobile parts and do not support UN regulations concerning the knee-form.

Yamazaki et al. proposed a UN regulation inspection system based on detecting the intersection between a CAD model and spheres placed on the model surface [23]. The current authors proposed an improved method for the rapid detection of the sphere-contacting shape for evaluating UN regulations 17, 21, and 25 [9]. This method uses the depth buffer mechanism of a GPU for accelerating the computation [7,8]. Similar technology is applied in our current system for detecting the knee-form contacting area. The authors developed another UN regulation inspection software based on the offset computation of the instrument panel [10]. However, this system is not applicable to the knee-form contact analysis.

One possible method of detecting the knee-form contacting area may be the application of 3D collision detection technology; established technologies in this field are described in [15], and [4] presents 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 in computers. A knee-form model is prepared in the various orientations allowed in the rotation range shown in Fig. 1(a). The knee-form model is then

moved in a straight line toward the instrument panel model to detect its first point of collision on the instrument panel. This collision detection process is repeated for knee-form models in various initial positions and a set of colliding points representing the knee-form contacting area is derived. Although this approach is simple and easy to implement, its cost is estimated to be very large because a tremendous number of collision detections are required, with knee-form models in different orientations and initial positions. In Section 4, we introduce an experimental result obtained with our system. In this experiment, possible knee-form contacts at 67 million points in the instrument panel surface are checked. If we prepare knee-forms in 60 initial orientations (for every one degree in the range from −30 to 30 degrees) for 67 million initial positions, collision detection technologies must execute 4 billion cases, which is far from feasible for the purpose of the method proposed in this paper.

To assist the designers, this paper proposes a novel method for automatically detecting the knee-form contacting area on the instrument panel. In order to reduce the computation cost, our UN regulation inspection method uses the Minkowski sum of a polyhedral object and a horizontal cylinder. Lien proposed a point-based Minkowski sum operation [13] whereby the surfaces of two objects are converted to two groups of points and then summed; points located inside the Minkowski sum object are discarded by applying a series of filters to determine the result surface. The computational cost for the filtering step can become exceedingly large in cases with a large object size, as is the case in typical UN regulation inspections. Li and McMains discussed a voxelized Minkowski sum computation with culling techniques [11,12]. Their method first generates possible surface elements of the Minkowski sum shape of two objects; voxels corresponding to the Minkowski sum shape are then selected according to the surface elements. The use of such a spatial grid-based method for UN regulation inspections would consume large amounts of memory to record the voxel model with high accuracy.

Offsetting a 3D object can be recognized as a Minkowski sum between the object and a sphere with the offset radius. Simple techniques for offsetting 3D models [5,17,18] are often computationally expensive, and model reconstruction can be unstable. To overcome these difficulties, new offset computation methods based on the discrete representation of the 3D model have become popular. Known representation schemes utilize points, voxels, and dexels [20], or rays [14], or Layered Depth Images (LDI) [19], and various improvements such as triple-dexels [1] have been reported.

The offsetting method proposed by Wang and Manocha uses LDI to record the object shape and the

temporal result of the offset computation [21,22]. Zhao et al. developed a Compact LDI (CLDI) approach, which offers improved data storage technology to reduce the amount of memory required [24]. The method proposed in this paper also uses a discrete representation for recording the Minkowski sum shape. The knee-form contact analysis requires knowledge of the surface area of the Minkowski sum shape as visible from a specific viewing direction. Therefore, our method uses a simple 2D grid-based representation for recording the Minkowski operation result. Because the amount of memory necessary for recording the result is much less than that required by other discrete representation schemes, our system can use a very high resolution grid in the computation, which is important to obtain an accurate analysis result.
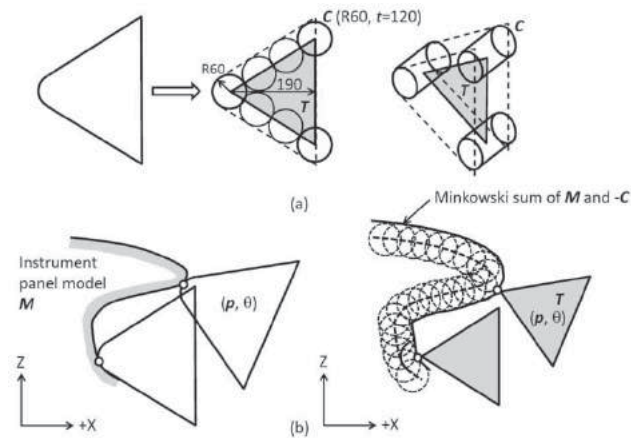
The input data for the method proposed in this paper consists of a polyhedral model M that approximates the panel shape with a high degree of accuracy. Most commercial CAD systems provide a function to output the model data as a group of triangular polygons, such as in the STL format. The system can visualize the knee-form contact area on the display and can output the area data as a set of small polygons in the STL format. To simplify the problem, the IP level and the effect of the steering wheel on the analysis are not considered in this work; therefore, only contacts between the knee-form and the upper part of the instrument panel and contacts to the panel area behind the steering wheel are allowed.

2. Contacting analysis of knee-form

The position and orientation of the knee-form can be uniquely described using a configuration (p, θ) where p represents the position of the center point of the knee-form, and θ represents the knee-form's orientation around p. The knee-form can be approximated by the Minkowski sum shape of a vertical equilateral triangle T and a horizontal cylinder C of radius 60 mm and length 120 mm, as shown in Fig. 2(a). The contacting condition of the knee-form to the instrument panel model M thus can be analyzed by evaluating the contacting condition of the equilateral triangle T with respect to the Minkowski sum shape of M and the inverted horizontal cylinder $-C$, as shown in Fig. 2(b). C and $-C$ have the same geometry if their local coordinate frames are assigned on the center of the cylinder. If the knee-form at configuration (p, θ) contacts M, then T in the same configuration contacts the Minkowski sum shape and vice versa.

Based on this concept, our method analyzes the knee-form contacting area on the instrument panel in the following three steps, as shown in Fig. 3. In Step 1, the

Figure 2. (a) Approximation of knee-form as a Minkowski sum of equilateral triangle T and horizontal cylinder C, and (b) Replacement of the contact analysis using T and the Minkowski sum of instrument panel model M and the inverted horizontal cylinder $-C$.
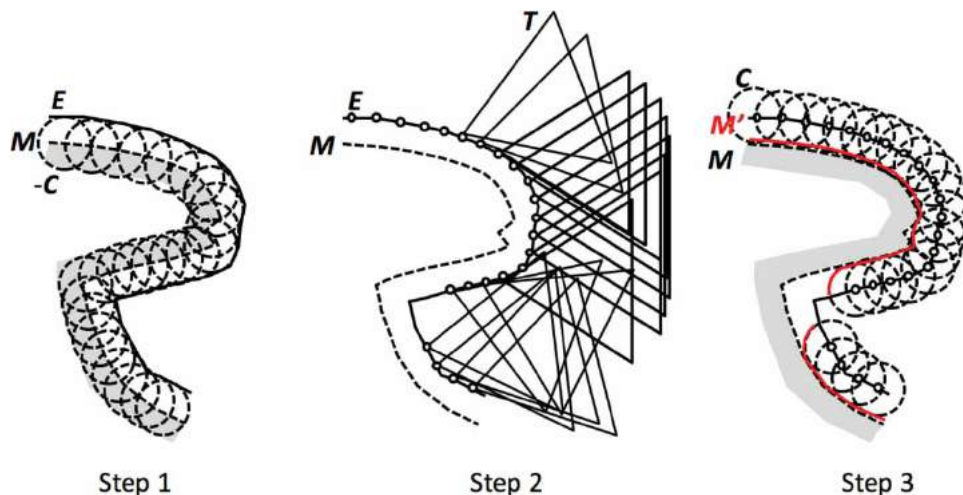

Figure 3. Knee-form contacting area extraction algorithm.

Minkowski sum shape of the instrument panel shape *M* and the inverted cylinder – *C* is computed, and the surface of the Minkowski sum shape facing the driver's side *E* is obtained. In Step 2, possible contacts between the vertical equilateral triangle *T* and *E* are analyzed, and the triangle contacting area on *E* is computed as a set of points densely covering *E*'s surface. In the final step, the Minkowski sum shape *M'* of the points obtained in Step 2 and the cylinder *C* is computed. The surface intersection of *M* and *M'* corresponds to the knee-form contacting area on *M*.

Minkowski sum computations and the detection of possible contacts between a triangle and objects are generally very time consuming. However, the parallel processing capability of a graphics processing unit (GPU) allows our system to detect and output the knee-form contacting area in a practical time period. The details of the novel contributions of our knee-form contacting area detection algorithm are explained in Section 3. Experimental detection results from complex CAD models of instrument panels are given in Section 4, and we summarize our conclusions in Section 5.

3. Details of analysis method

This section details the proposed three-step algorithm for detecting the knee-form contacting, and describes the application of the GPU's parallel processing capability for accelerating the computation.

3.1. Step 1: Minkowski sum of instrument panel model and cylinder

In the first step, the instrument panel shape is expanded by a horizontal cylinder of radius 60 mm and length 120 mm, and the surface of the expanded shape facing the positive direction of the x-axis is computed. The cylinder is oriented so that its center axis is in the y-axis direction. The Minkowski sum shape of one triangle *f* and the cylinder is obtained as a combination of the following two component shapes.

1. For each edge *e* of the triangle, the linear swept volume of the cylinder moving along the edge is defined. The swept volume is a combination of two cylinders placed on two end vertices of the edge, two slant cylinders corresponding to the linear swept volume of two circular plates of radius 60 mm at the end of the cylinder, and a prismatic shape, as shown in Fig. 4(a). The final prismatic shape corresponds to the linear swept volume of the rectangle defined by the center axis of the cylinder and an axis perpendicular to both the center axis and the edge.
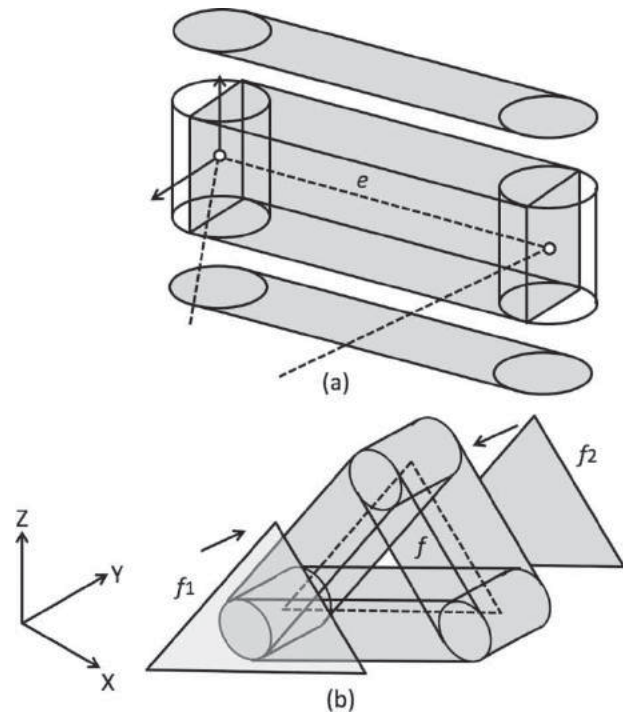

Figure 4. Minkowski sum shape of a triangle *f* and a horizontal cylinder.

2. Three swept volumes on the edges of triangle *f* form an approximately doughnut shape, as shown in Fig. 4(b). The hole of the doughnut is closed with two triangles *f*1 and *f*2. One triangle *f*1 is obtained by moving *f* in the axis direction of the cylinder by 60 mm (half of the cylinder's length), then in a direction corresponding to the outward projection of the normal vector of *f* to the zx-plane by 60 mm (the cylinder's radius). Another triangle *f*2 is obtained by moving *f* in the opposite manner from *f*1.

The Minkowski sum shape of the entire object is obtained as a Boolean union of the small Minkowski sum shapes defined for all polygons of the object. An exact computation of the Boolean union of many component shapes is time-consuming and unstable because of inevitable numerical errors in the geometric computation. Because the knee-form linearly approaches the instrument panel in the negative direction of the x-axis, only the part of the Minkowski sum shape visible from the negative direction of the x-axis is necessary. This visible surface of the Minkowski sum shape is obtained using the hidden surface elimination mechanism for rendering 3D objects [15], as illustrated in Fig. 5. In preparation, the coordinate frame of the orthogonal projection is placed so that its x-axis and y-axis are aligned with respect to the y-axis and z-axis of the object's coordinate frame. The negative z-axis direction of the projection's coordinate
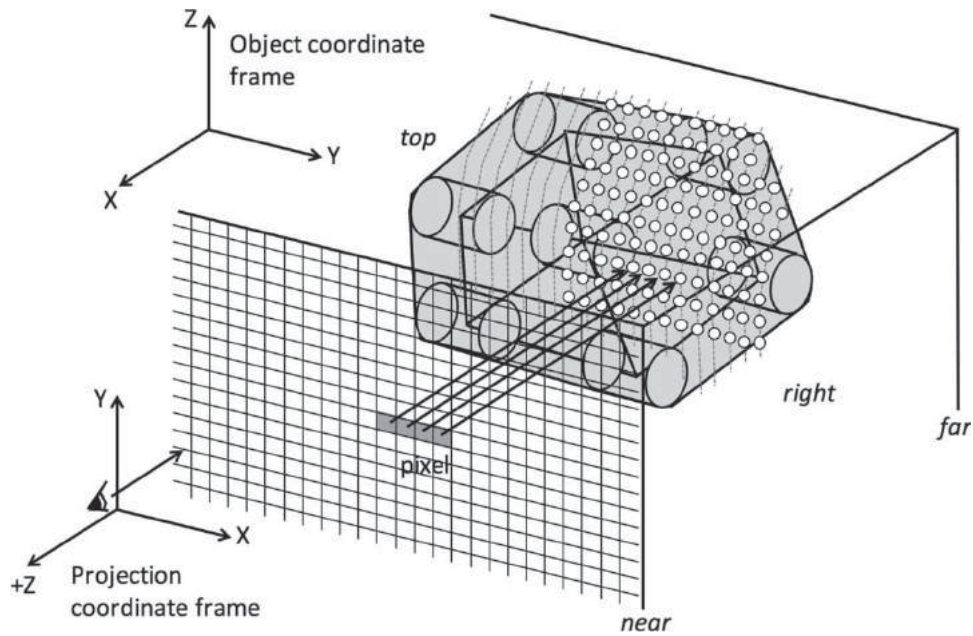
Figure 5. Depth buffer-based method for computing the visible surface of the Minkowski sum shape.

frame corresponds to the viewing direction, and six clipping planes defining the visible regions in the object's coordinate frame are given such that they tightly confine the Minkowski sum shape.

A depth buffer with the same number of entries as the pixels of the frame buffer is prepared. All the entries of the depth buffer are initialized to a sufficiently small value. After the preparation, the polygonal approximation of the Minkowski sum shape's component objects are rendered into the frame buffer in an arbitrary order. During the rendering, the color and the depth value (defined here as the x-coordinate) of points on a polygon corresponding to each pixel of the frame buffer is checked. If the depth value is larger than the value currently in the depth buffer, then the color and depth value of the new point replace the old ones in the buffers. This process is repeated for all polygons of the component objects, and the result is the pixels with the colors of points on the visible surfaces of the Minkowski sum shape and the entries of the x-coordinates of the points in the depth buffer.

Data stored in the depth buffer can be transferred to a 2D array in a C language program using a buffer manipulation function from the graphics library. Based on the clipping plane definition and the pixel position in the display, the y- and z-coordinates of the visible point at the pixel can be computed. The depth buffer value at the pixel corresponds to the x-coordinate of the point. The Minkowski sum shape visible from the negative x-axis direction is thus derived as a set of grid-like points aligned with respect to the pixel grid of the display [8,9].

In Fig. 5, the white points represent the points sampled on the visible surface of the Minkowski sum shape.

The computation result is highly dependent on the resolution of the depth buffer, especially for the y- and z-coordinates of the visible points. A Minkowski sum computation based on a higher resolution depth buffer enables the generation of more points on the surface of the Minkowski sum shape, enabling greater accuracy in the triangle contact analysis and the knee-form contact analysis in the following steps. The maximum pixel resolution of current computer displays is not larger than 3840×2160 (4 K). This resolution limits the accuracy of the computation of our method; however, this limitation can be overcome by splitting and tiling the computation. Based on a virtual depth buffer with very high resolution, the depth buffer grid is split into a set of small depth buffers with the pixel resolution available on the display. For each small depth buffer, the Minkowski sum operation is done; then each result is copied to its corresponding grid portion in the large depth buffer.

3.2. Step 2: contact analysis of an equilateral triangle

This step evaluates the contact condition between the vertical equilateral triangle corresponding to the core part of the knee-form and the surface of the expanded instrument panel model obtained in Step 1. The shape of the expanded instrument panel model is defined as a set of grid-like points aligned with respect to the y-axis and z-axis in the object coordinate frame. Because the vertical
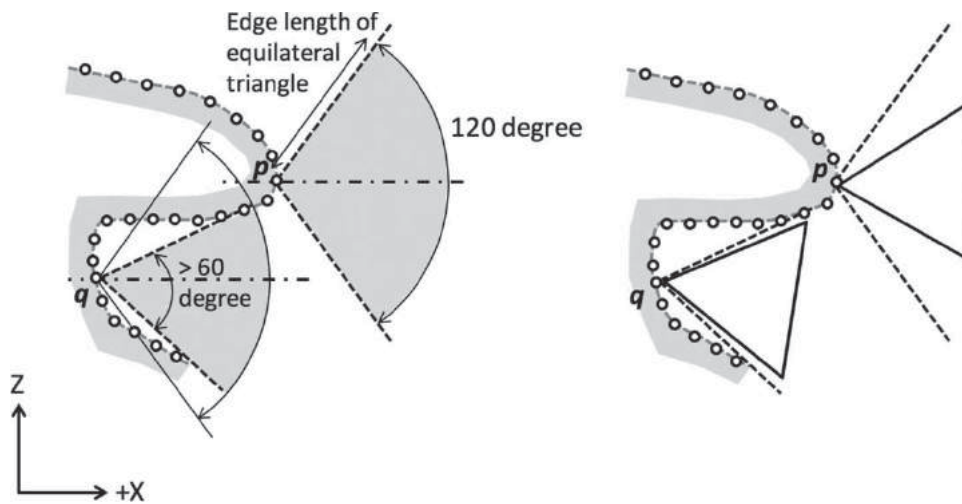
Figure 6. Contact analysis of an equilateral triangle at its vertex.

triangle is positioned to be parallel to the zx-plane in the object space, the evaluation of the contact condition can be decomposed into a contact analysis between the triangle and a series of points corresponding to each column of grid-like points, and can thus be realized in 2D.

Fig. 6 illustrates the polygonal line obtained by connecting the series of points and an equilateral triangle in the same plane. The triangle linearly approaches the polygonal line along a vector parallel to the negative direction of the x-axis. Consider that the triangle contacts the polygonal line at its vertex. To realize such contact, there must be a fan-like space adjacent to the contacting point that contains an arc of more than 60 degrees and that has a radius equal to the edge length of the triangle. Before the approach, the triangle can be rotated around its center point. Because the rotation angle is in a range from −30 to +30 degrees, the fan-like space must be contained within a larger fan containing an arc from −60 to +60 degrees, with respect to the horizontal center line. For each point on the polygonal line, the placement condition of an equilateral triangle mentioned above is checked, and points where the triangle can make contact are marked.

Because this contact analysis can be executed for multiple points on the polygonal line simultaneously, we accelerate the computation using the parallel processing capability of a GPU. Compute Unified Device Architecture (CUDA) is used to implement the parallel contact analysis software [3]. The GPU is designed to have hundreds of small streaming processors (SP) on a chip. These SPs can execute the same instructions with different data in parallel, and with CUDA, programmers can use a GPU as a general purpose Single Instruction / Multiple Data (SIMD)-type processor.

The main accelerating factor gained by the GPU is the replacement of the iterative execution of a function in a loop with a parallel execution of its equivalent threads. CUDA is designed to provide a parallel execution framework to treat these threads in a C program. To properly manage the threads, CUDA provides grid and block structures. A block is a 1D, 2D, or 3D array of threads; the maximum number of threads in a single block must be less than or equal to 512. A grid is a 1D or 2D array of blocks. In our current implementation, the contact analysis of a single point in the polygonal line is defined as a thread. Each block is designed to execute a maximum 256 threads in parallel, representing parallel contact analysis for 256 points in our implementation. For each column of grid-like points on the expanded instrument panel model, one polygonal line is defined. All points on the polygonal line are divided into groups of 256 or fewer points; then, each point group is assigned to a block of the CUDA program.

After checking the contact of the triangle at its vertex, possible contact of the triangle on its edge is evaluated. Convex points on the polygonal line unmarked in the previous step, where contact at the triangle's vertex is not possible, become candidates for contact on the triangle's edge. For each candidate point, a pseudo-normal vector is defined as the bisector of its supplementary angle. A tangent line through the candidate point and perpendicular to the pseudo-normal vector is defined, and the triangle is placed on the line as shown in Fig. 7. The triangle is translated along the tangent line in small intervals, searching for a position of the triangle where the triangle does not intersect the polygonal line. If such a collision-free position is identified, its corresponding point on the polygonal line is marked as a point where
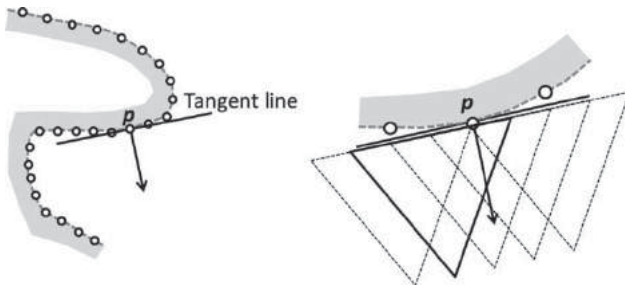
Figure 7. Contact analysis of an equilateral triangle on its edge.

the triangle can realize an edge contact. Because the orientation of the triangle must be in the range from −30 to +30 degrees with respect to the horizontal line, the inclination of the tangent line must be in the range from −60 to +60 degrees.

3.3. Step 3: evaluation of the knee-form contact

In the final step, actual contact points between the knee-form and the instrument panel model are computed based on the analysis result from Step 2. A set of points obtained in Step 1 and marked in Step 2 represents the points where the equilateral triangle corresponding to the core part of the knee-form contacts the Minkowski sum shape of the instrument panel and horizontal cylinder. Consider one such point p marked in Step 2. The actual contact point P between the knee-form and the instrument panel is obtained by replacing p to the horizontal cylinder and computing the intersection between the cylinder and the surface of the original instrument panel model, as shown in Fig. 8. In Step 3, the Minkowski sum shape of the marked points and the cylinder is computed. The intersecting area between the original instrument panel model and the shape resulting from the Minkowski

sum computation corresponds to the knee-form contact area on the instrument panel.

The Minkowski sum computation in Step 3 uses the same depth buffer-based method described in Step 1. For each marked point, the horizontal cylinder is placed so that its center point is located on the marked point. After the cylinder has been placed at all marked points, their hidden surface eliminated picture is generated. The grid-like points on the surface of the Minkowski sum shape are obtained using the buffer manipulation function from the graphics library. In Step 1, a surface point with the largest x-coordinate is selected for each pixel. Such point is visible from the negative direction on the x-axis. On the other hand, in Step 3, a point with the smallest x-coordinate is selected to obtain the point for each pixel on the opposite side of the surface. As a result, grid-like points on the Minkowski sum shape closest to the original instrument panel model are obtained.

The grid-like points are then properly connected based on the adjacency information of the grid, and the points are converted to an equivalent polyhedral surface with small triangular polygons. The intersection of the surface of the original instrument panel model and the surface of the Minkowski sum shape is derived by selecting the surface polygons of the Minkowski sum shape for which the distance to the original model is sufficiently small. As a result, the knee-form contacting area on the instrument panel is obtained as a set of small triangles. In our current implementation, $\varepsilon = 0.1$ mm is used as the small distance value for selecting the intersecting polygons. This value is determined based on a discussion with UN regulation inspection specialists in an automobile manufacturer.

To accelerate the distance computation, surface polygons of the original instrument panel model are classified
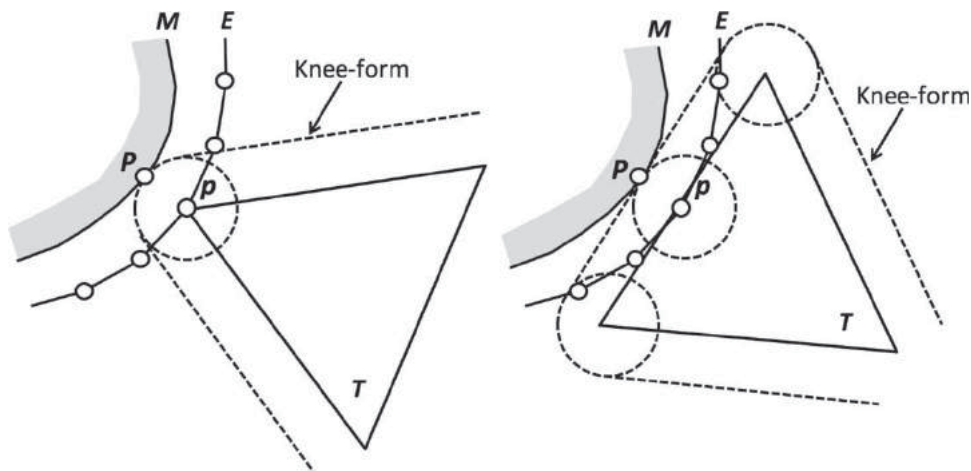

Figure 8. Relationship of contact point p between equilateral triangle T and expanded panel model E, and actual contact point P between the knee-form and the instrument panel model M.

into small groups according to their proximity. This classification proceeds according to the hierarchical structure of the Axis-Aligned Bounding Box (AABB) [15]. 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 triangular polygons of the model. Polygons in the AABB are sorted and are classified to two groups. For each polygon group, a smaller AABB is formed and registered as a descendant of the original AABB. The process of defining descendant AABBs is iterated, and a binary AABB tree is obtained.

Using the AABB tree, the cost for computing a distance between a small triangle of the Minkowski sum shape and the original instrument panel model can be reduced. For each small triangle *t* of the Minkowski sum shape, a small AABB tightly holding the triangle is defined; this AABB is named *box*. Traversing the AABB tree from the root node to the leaf nodes in the depth-first manner, at each node of the AABB tree, the distance between *box* and the AABB associated with the node is compared. If the distance is greater than ε, then the traversal of its descendant nodes is canceled; otherwise, the tree traversal is continued. After the tree traversal, some AABBs at the leaf nodes of the tree are obtained that hold polygons sufficiently close to the given triangle *t*. The distance computation between *t* and the instrument panel model is reduced to the distance computation between *t* and polygons within these leaf AABBs, and the GPU's parallel processing technology is applied to these distance computations to further reduce the computation time.

4. Computational experiments

The proposed knee-form contacting area computation system was implemented using Visual C++, CUDA 6.5, and OpenGL, and a series of computational experiments was performed using an Intel Core i7 Processor (3.4 GHz) with 32 GB memory and an NVIDIA GeForce GTX-980 GPU. We applied the system to three instrument panel models provided by a software company and an automobile company, and the proposed system was found to successfully compute the knee-form contact area for all parts in 5 to 15 minutes, with the time varying according to the number of polygons of the instrument panel model and the resolution of the depth buffer used in the computation.

For the purpose of maintaining confidentiality, the computation result for only one sample is shown here. Fig. 9 illustrates the sample model with 597,455 polygons. The surface of the Minkowski sum shape of the sample part and a horizontal cylinder of 60 mm radius

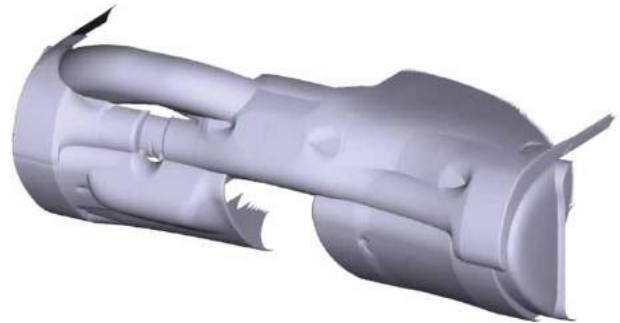**Figure 9.** Sample instrument panel model with 597,455 polygons.

Figure 10. Surface region of the Minkowski sum shape of the instrument panel model and a horizontal cylinder visible from the negative direction of the x-axis.

and 120 mm length is obtained in Step 1, and the resulting surface is shown in Fig. 10. In this figure, the shaded image of the Minkowski sum shape was generated by painting points on the shape with the normal vector information at the points. In Step 2, some points are eliminated because the equilateral triangle cannot contact them. Fig. 11 shows the opposite side of the surface of the Minkowski sum shape of the points marked in Step 2 and the horizontal cylinder. After checking the intersection between the shape shown in Fig. 11 and the original model shown in Fig. 9, the knee-form contacting area is extracted as shown in Fig. 12. Because the IP level is not
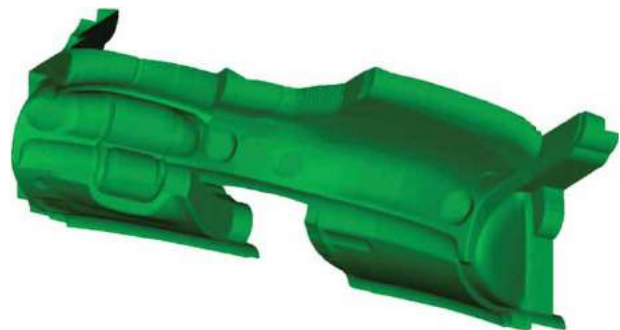

Figure 11. Surface region of the Minkowski sum shape of the points marked in Step 2 and a horizontal cylinder.

Figure 12. Knee-form contacting area on the instrument panel surface.

evaluated in the current system, some knee-form contact areas are detected on the top side of the panel where the knee of the driver never collides. In the current design practice, the IP level is given as a curve on the instrument panel surface. One possible method for incorporating the IP level in the system is to superimpose the curve on the extracted area so that the designer can recognize the effective region of the extracted area in the display. Automatic recognition of the effective region is considered as our future research subject. Incorporation of the steering wheel shape in the detection must be considered in the future also.

Our proposed system was found to require 40.16 seconds for Step 1, 125.43 seconds for Step 2, and 79.94 seconds for Step 3, for a total of 5 minutes required to analyze the knee-form contacting area for this instrument panel model. In this computation, a virtual depth buffer with a grid resolution of $13{,}236 \times 5{,}069$ was used, a much higher resolution than is allowed by a single display. The grid size, which corresponds to the computation accuracy of our method, was 0.128 mm. The proposed system can output the polygons of the detected area in STL format, and the automobile designer can verify compliance with the UN regulation by superimposing the knee-form contact area on the CAD model of the instrument panel.

5. Conclusions

In this paper, we have proposed a novel method for assisting UN safety regulation inspections of instrument panels. The driver's knee-form can be recognized as a Minkowski sum shape of a vertical equilateral triangle and a horizontal cylinder of radius 60 mm and thickness 120 mm. The knee form contacting condition is geometrically equivalent to the contacting condition of the equilateral triangle to a Minkowski sum shape of the instrument panel model and the horizontal cylinder. Based on this concept, our algorithm realizes the extraction of the knee-form contacting area.

Minkowski sum computations and the analysis of the contact between the vertical equilateral triangle and the object are the most time consuming processes in our algorithm. However, with the parallel computation capability of a GPU, our system can extract the knee-form contacting area in less than 15 minutes. The proposed system is still in an experimental stage, and the UN regulation describes some additional conditions concerning the steering wheel and the IP level; however, these conditions are not considered in this study. Incorporation of the conditions in the detection process is an important topic in the next phase of this research. We are preparing a field test of the system in an actual automobile design process, and further improvements based on comments and requests from the designers will be reflected in our future work.

ORCID

Masatomo Inui http://orcid.org/0000-0002-1496-7680
Shunsuke Nakano http://orcid.org/0000-0002-0573-2580
Nobuyuki Umezu http://orcid.org/0000-0002-7873-7833
Tetsuya Asano http://orcid.org/0000-0002-4988-6928

References

[1] Benouamer M. O.; Michelucci D.: Bridging the gap between CSG and Brep via a triple ray representation, Proceedings of ACM Symposium on Solid Modeling and Applications, 1997, 68–79. http://dx.doi.org/10.1145/267734.267755

[2] CAVA Systems, http://www.edstechnologies.com/index.php/component/content/article/86-solution/cad-quality-check/135-cava.html, EDS Technologies.

[3] CUDA Compute Unified Device Architecture Programming Guide, http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf, NVIDIA.

[4] Faure F.; Barbier S.; Allard J.; Falipou F.: Image-based collision detection and response between arbitrary volume objects, Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2008, 155–162. http://dx.doi.org/10.2312/SCA/SCA08/155-162

[5] Forsyth M.: Shelling and offsetting bodies, Proceedings of Third Symposium on Solid Modeling and Applications, 1995, 373–381. http://dx.doi.org/10.1145/218013.218088

[6] Global Automotive Regulations, https://globalautoregs.com/unece, United Nations.

[7] Inui M.: Fast inverse offset computation using polygon rendering hardware, Computer-Aided Design, 35, 2003, 191–201. http://dx.doi.org/10.1016/S0010-4485(02)00052-0

[8] Inui M.; Ohta A.: Using GPU to accelerate die and mold fabrication, IEEE Computer Graphics and Applications, 27(1), 2007, 82–88. http://dx.doi.org/10.1109/MCG.2007.23

[9] 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://dx.doi.org/10.1299/jamdsm.7.818

[10] 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

[11] Li W.; McMains S.: A GPU-based voxelization Approach to 3D Minkowski sum computation, Proceedings of the 14th ACM Symposium on Solid and Physical Modeling, 2010, 31–40. http://dx.doi.org/10.1145/1839778.1839783

[12] Li W.; McMains S.: Voxelized Minkowski sum computation on the GPU with robust culling, Computer-Aided Design, 43(10), 2011, 1270–1283. http://dx.doi.org/10.1016/j.cad.2011.06.022

[13] Lien J. M.: Covering Minkowski sum boundary using points with applications, Computer Aided Geometric Design, 25, 2008, 652–666. http://dx.doi.org/10.1016/j.cagd.2008.06.006

[14] Menon J. P.; Marisa R. J.; Zagajac J.: More powerful solid modeling through ray representations, IEEE Computer Graphics and Applications, 14(3), 1994, 22–35. http://dx.doi.org/10.1109/38.279039

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

[16] Japanese Patent [P2006-277304A], http://www.j-tokkyo.com/2006/G06F/JP2006-277304.shtml, Toyota Motor Corporation, in Japanese.

[17] Rossignac J. R.; Requicha A. A. G.: Offsetting operations in solid modeling, Computer Aided Geometric Design, 3, 1986, 129–148. http://dx.doi.org/10.1016/0167-8396(86)90017-8

[18] Satoh T.; Chiyokura H.: Boolean operations on sets using surface data, Proceedings of ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications, 1991, 119–127. http://dx.doi.org/10.1145/112515.112536

[19] Shade J.; Gortler S.; He L. W.; Szeliski R.: Layered depth image, Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIG-GRAPH, 1998, 231–242. http://dx.doi.org/10.1145/280814.280882

[20] Van Hook T.: Real-time shaded NC milling display, Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, 20(4), 1986, 15–20. http://dx.doi.org/10.1145/15922.15887

[21] 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. http://dx.doi.org/10.1016/j.compind.2011.02.004

[22] Wang C. C. L.; Manocha D.: GPU-based offset surface computation using point samples, Computer-Aided Design, 45(2), 2013, 321–330. http://dx.doi.org/10.1016/j.cad.2012.10.015

[23] 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://dx.doi.org/10.1109/icma.2011.5986280

[24] Zhao H.; Wang C. C. L.: Parallel and efficient Boolean on polygonal solids, The Visual Computer, 27(68), 2011, 507–517. http://dx.doi.org/10.1007/s00371-011-0571-1