Extraction of Sliding Collision Area of Knee-Form for Automobile Safety Inspections

Masatomo Inui
Dept. of Intelligent Systems Engr.
Ibaraki Univ.
Nakanarusawa 4-12-1
316-8511, Hitachi, JAPAN
masatomo.inui.az@vc.ibaraki.ac.jp

Kenta Gunji
Dept. of Intelligent Systems Engr.
Ibaraki Univ.
Nakanarusawa 4-12-1
316-8511, Hitachi, JAPAN
17nm914h@vc.ibaraki.ac.jp

Nobuyuki Umezu
Dept. of Intelligent Systems Engr.
Ibaraki Univ.
Nakanarusawa 4-12-1
316-8511, Hitachi, JAPAN
nobuyuki.umezu.cs@vc.ibaraki.ac.jp

ABSTRACT
The United Nations Economic Commission for Europe defines a safety regulation based on the possible collision between the driver's knee and an automobile's interior parts. The "knee-form" apparatus is used to evaluate compliance with this regulation. Current software for analyzing possible collisions of the knee-form is not applicable to the part whose surface is vertical and near parallel to the knee-form approaching direction. In this paper, we propose a novel algorithm named "push-and-slide" for extracting the knee-form colliding area on the console part and door panel. In the first step of the algorithm, the target surface of the part is transformed to grid-like points in a high resolution. The knee-form models in various positions and orientations are prepared in the second step. Each knee-form model is pushed to the grid-like points. The model is then moved along the part surface to detect possible collisions between the knee-form and the grid-like points. An experimental system is developed and some computational experiments are performed.

Keywords
UNECE inspection, automobile safety, collision detection, knee-form, CAD.

1. INTRODUCTION
In driving an automobile, speed control is basically realized by the driver's leg and foot motion. If interior parts locating around the driver, such as instrument panel, console part and door panel, hinder the smooth motion of the leg, 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 interior parts [GAR]. Similar regulations are defined in other countries including Japan and China.

Figure 1 explains the UNECE regulation No. 21 concerning the knee collision. In the automobile design, CAD model of the automobile body, which is an assembly of exterior and interior part models, is positioned so that its front part faces the negative direction of the x-axis of the object's coordinate frame. An apparatus imitating the knee shape during driving is defined in the regulation. This "knee-form"

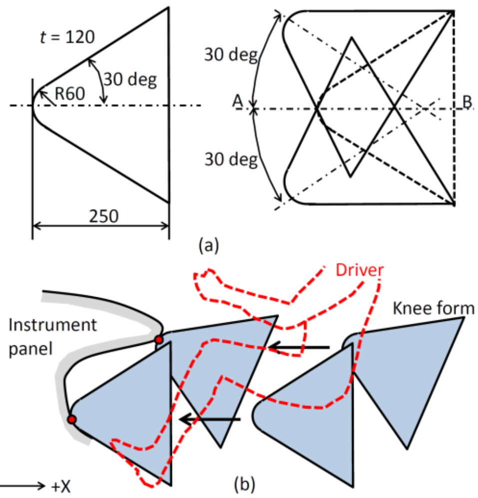

Figure 1. Knee-form collision analysis as defined in the UNECE regulation No. 21.

has an equilateral triangle-like shape and a thickness of 120 mm. It has a rounded corner that can be represented by a cylinder with a radius of 60 mm as shown in Figure 1(a). To inspect for compliance with the regulation, the knee-form is positioned in the driver's space as its round corner faces towards the front part of the automobile. It is moved linearly in the negative direction of the x-axis, where the front surface of the knee-form eventually collides to an interior part, for example instrument panel. This inspection is repeated by changing the initial position of the knee-form. In this way, as shown in Figure 1(b), surface areas on the interior parts are detected where the knee can make collision. Before starting the movement, the apparatus can be rotated above and below the horizontal by up to 30 degrees to represent different orientations. In the actual driving, seat position, position and orientation of the driver on the seat, and his/her height affect the knee collision on the interior part. The UNECE regulation, however, do not consider such parameters in the collision detection.

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

2. OUTLINE OF CONTRIBUTION

Although UNECE 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 methods.

Some commercial systems are known for verifying the UNECE regulations, such as the CAVA system [CAVA]. The Toyota Motor Corporation submitted some patents relevant to the automatic inspection of UNECE regulations 17, 21, and 25, concerning collisions of a large sphere representing an infant's head [JP06]. These systems detect only the sphere contacting area on automobile parts and do not support the regulations concerning the knee-form.

Yamazaki et al. proposed a UNECE regulation inspection system based on detecting the intersection between a CAD model and spheres placed on the model surface [Yam11]. We developed an improved method for the rapid detection of the sphere-contacting shape for evaluating the UNECE regulations [Inu15]. These systems use the parallel processing capability of a Graphics Processing Unit (GPU) for accelerating the computation [CUDA]. They do not support the collision detection with the knee-forms.

We proposed a method for automatically detecting the contacting area of the knee-form on the instrument panel [Inu17]. The "knee-form" apparatus can be modeled as a Minkowski sum [Li11] 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. With the parallel computation capability of a GPU, our system can detect and output the knee-form contacting area in a practical time period.

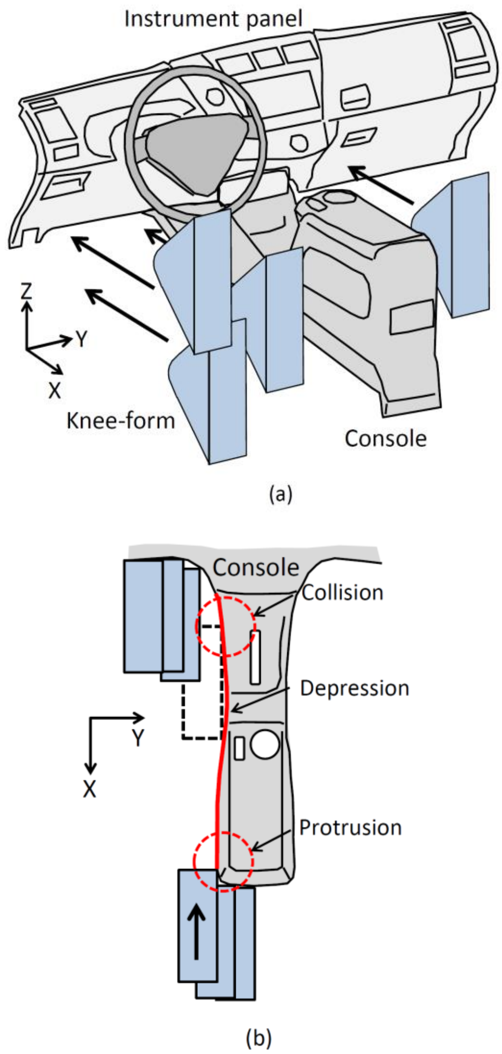

Figure 2. Limitation of our prior method [Inu17].

In this method, it is assumed that the knee-form approaches the parts from a sufficiently far position from the parts. This assumption is acceptable for detecting the knee-form contacting area on the instrument panel, however it is not appropriate for detecting the area on the console part and door panel. Figure 2 illustrates this problem. Most surface of the instrument panel faces towards the positive direction of the x-axis. Since there is no obstacles between the knee-form in a far position and the instrument panel, the approaching knee-form can directly collide any part of the instrument panel as shown in Figure 2(a).

On the other hand, most surface area of the console part and door panel is vertical and near parallel to the approaching direction of the knee-form. The knee-form can make a collision at a shallow depression in the side surface of the console part as shown in Figure 2(b). Such collision is not detected in our prior method because a knee-form approaching from a far position make a collision at a protrusion and it cannot reach the depression locating behind the protrusion.

One possible method of extracting the knee-form colliding area may be the application of 3D collision detection technology; established technologies in this field are described in [Mol99], and [Fau08] 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. 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.

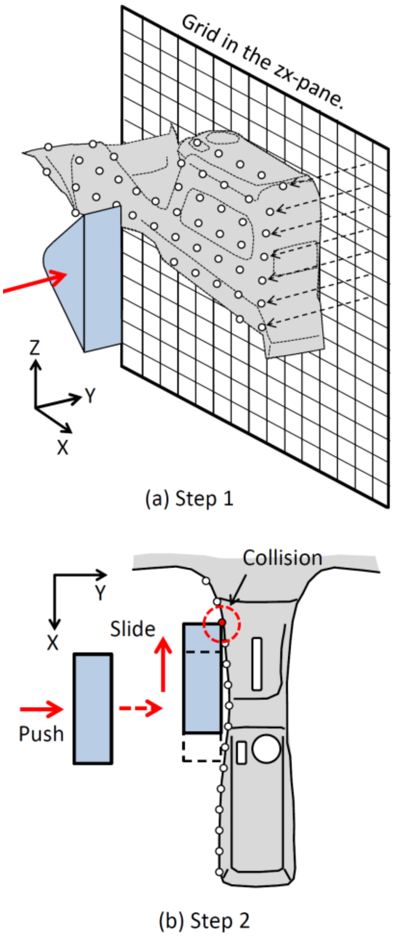

Figure 3. Outline of push-and-slide algorithm.

In this paper, we propose a novel algorithm named "push-and-slide" specialized for detecting the knee-form colliding area in the surface of the interior part which is vertical and near parallel to the knee-form approaching direction. The input data for the method consists of a polyhedral model that approximates the interior part shape. Most commercial CAD systems provide a function to output the model data as a group of triangular polygons, such as in the STL format.

Figure 3 illustrates the outline of the algorithm. In the first step, the target surface area of the part is transformed to grid-like points in a high resolution (see Figure 3(a)). The knee-form models in various positions and orientations are prepared in the second step. Each knee-form model is moved in the y-axis direction and pushed to the grid-like points. The knee-form model is then moved in the negative x-axis direction to detect possible collision with the grid-like points (see Figure 3(b)). The system finally visualizes the knee-form colliding area on the display by converting the detected collision points to a set of small polygons.

Our method has similarities to the collision detection based method, however it can realize better performance by properly using the geometric characteristic of the grid structure of the points on the part surface. In the following sections, details of our knee-form colliding area extraction algorithm are explained. In section 4, techniques for improving the performance are given. Experimental extraction result from a CAD model of a console part is given in Section 5, and we summarize our conclusions in Section 6.

3. BASIC ALGORITHM

In this section, details of the basic processing flow of our push-and-slide algorithm is explained. In the following discussion, we consider a collision analysis of the knee-form to the left side surface of the console part as an example. The same method is applicable for detecting knee-form colliding area in the other side of the console part or in the surface of the door panel by modifying some conditions, for example pushing direction of the knee-form to the part surface.

Step 1 Sampling of Grid-Like Points

The first step of the algorithm is the sampling of grid-like points on the near vertical surface of the console part. The console part is oriented parallel to the zx-plane and the left side surface of the part faces to the negative direction of the y-axis. An axis-aligned regular grid with sufficient width and resolution is prepared in the zx-plane. From each grid point, a ray is extended along the negative direction of the y-axis, and the intersection points between the ray and the left side surface of the console part are computed as shown in Figure 3(a). The intersection point with the smallest y coordinate is selected and recorded as the surface point corresponding to the grid point. This operation is repeated for all grid points, and an approximated representation of the left side surface of the console part is obtained as grid-like points.

Such surface points are obtained by rendering a hidden-surface eliminated picture of the console part using the orthogonal projection as illustrated in Figure 4 [Mol99]. In a preparation, a local coordinate frame for the orthogonal projection is placed so that its x-axis and y-axis are aligned with respect to the x-axis and z-axis of the object's coordinate frame. The negative z-axis direction of the projection's coordinate frame corresponds to the viewing direction. Six clipping planes (*left*, *right*, *top*, *bottom*, *near*, *far* in the figure) defining the visible regions in the object's coordinate frame are given such that they confine the console part with a sufficient margin around the part so that any knee-forms contacting the part surface can be contained within.

After the preparation, component polygons of the console model are rendered into the frame buffer in an arbitrary order. During the rendering, the color and the depth value of points on a polygon corresponding to each pixel of the frame buffer is updated, and the result is the pixels with the colors of points on the visible surfaces of the console part and the entries of the points in the depth buffer.

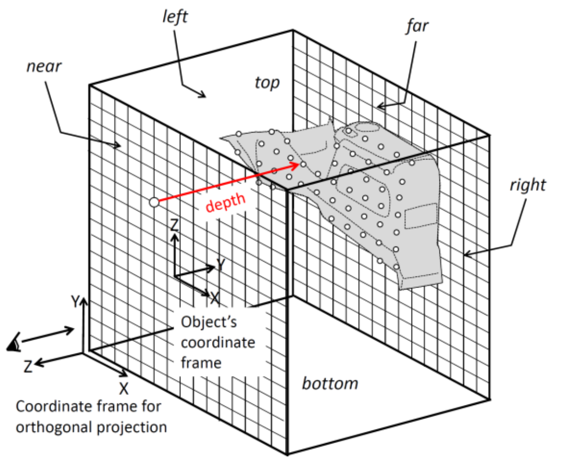

Figure 4. Depth buffer-based method for sampling grid-like points on the part surface.

Data stored in the depth buffer can be transferred to a 2D array in a C language program using a buffer manipulation function, for example glReadPixels( ) in OpenGL [Kes16]. Based on the clipping plane definition and the pixel position in the display, the x and z coordinates of the visible point at the pixel in the object's coordinate frame can be computed. The depth value at the pixel corresponds to the y coordinate of the point. The left side surface of the console part is thus derived as a set of grid-like points aligned with respect to the pixel grid of the display.

Step 2 Push-and-Slide Operation

In the definition of UNECE No.21, the knee-form can be rotated above and below the horizontal by up to 30 degrees. In our current implementation, the knee-form can be rotated in every one degree around the y-axis in the range. The rotated knee-form model is set to a certain initial position. We consider that the center point of the round corner part of the knee-form corresponds to the reference point of the knee-form. Points in the axis-aligned regular grid in the zx-plane are traced in row by row order. For each grid point, the rotated knee-form is placed so that its reference point (green point in Figure 5(a)) has the same x and z coordinates of the grid point and a sufficiently small y coordinate. The knee-form in Figure 5(a) is placed at (i, j) grid point.

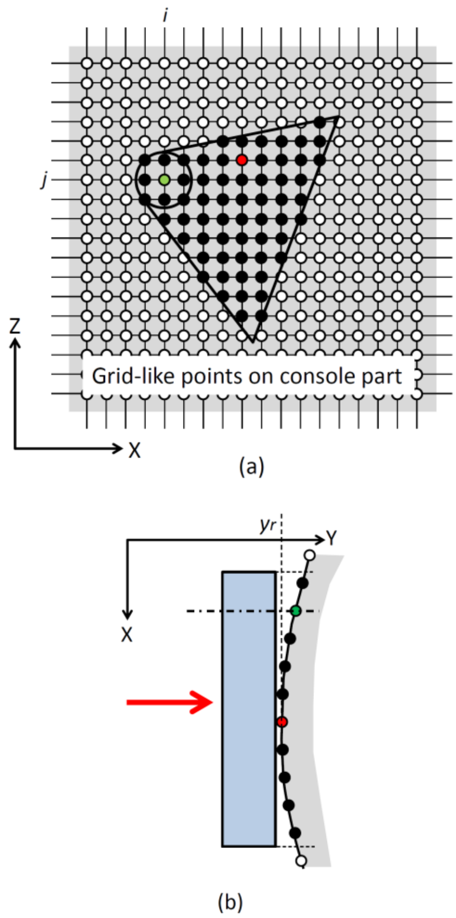

Figure 5. Determination of the knee-form position by a pushing operation.

3.1.1 Pushing Operation

The knee-form is then pushed linearly in the positive direction of the y-axis until it realizes a contact at a grid point in the left side surface of the console model. The y coordinate of the knee-form model in the contacting condition is determined by projecting the knee-form shape to the left side surface of the console model and by checking the y coordinates of the grid-like surface points contained within the projection. In Figure 5(a), the black points represent such points within the projection. In these points, a point with the smallest y coordinate y_r (red point in Figure 5(b)) corresponds to a point where the knee-form realizes a contact after a pushing operation.

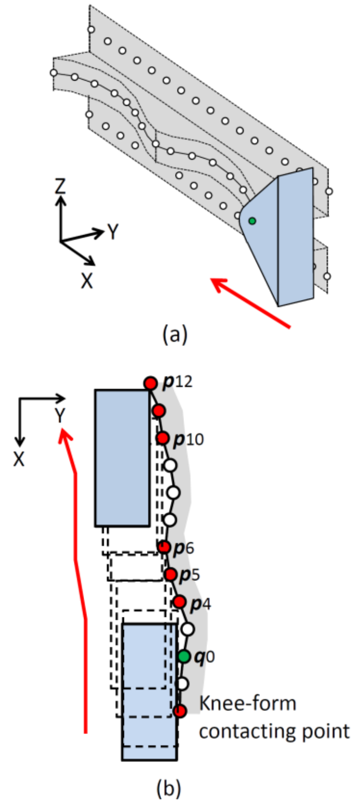

Figure 6. Sliding operation for detecting colliding points in a single row.

3.1.2 Sliding Operation

After the placement of the knee-form model on the console surface, the model is moved linearly in the negative x-axis direction until it collides to the console surface on its front surface. Since we represent the surface of the console part as a set of grid-like points, the collision between the knee-form in the sliding motion and the grid-like points is actually checked.

The knee-form continues the sliding operation after detecting the first collision for finding other colliding points in the part surface. Figure 6 illustrates our strategy. In this figure, a simple narrow and wave-like protrusion is given as the side surface of the console part. A single row of points is placed on the narrow "passage" on the protrusion. Other surface points are placed on the bottom part of the surface as shown in Figure 6(a). The knee-form model is already placed on the surface of the protrusion as its reference point locates above a grid point $\boldsymbol{q}_0$. Because of this placement, the knee-form can collide only points in the row on the protrusion in the sliding motion.

The knee-form starts sliding in the negative x-axis direction to detect a first colliding point $\boldsymbol{p}_4$ as shown in Figure 6(b). After the detection, the knee-form continues the sliding operation along the leftward curve to detect new colliding point $\boldsymbol{p}_5$ on the same row of the grid-like points. After colliding $\boldsymbol{p}_6$ at the peak point of the curve, the knee-form stops the curve-following because motion along the rightward curve causes gauges between the knee-form and the part surface. Otherwise, it starts the straight motion along the negative direction of the x-axis from $\boldsymbol{p}_6$ to detect new colliding point $\boldsymbol{p}_{10}$ after getting over a depression between $\boldsymbol{p}_6$ and $\boldsymbol{p}_{10}$. The knee-form starts the leftward sliding operation again from $\boldsymbol{p}_{10}$ until it reaches the edge of the console part.

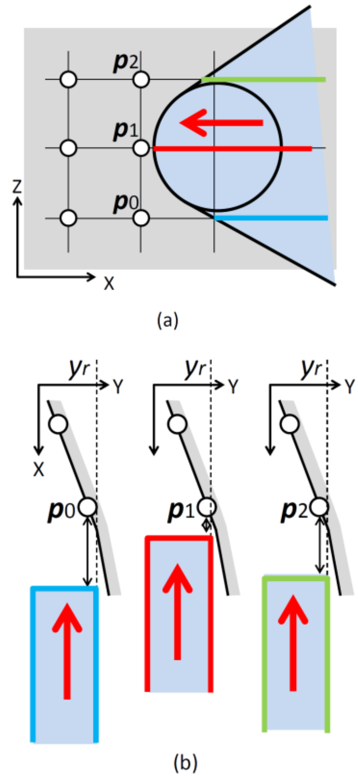

Figure 7. Detection of the colliding points in multiple rows.

Side surface of the actual console part generally has much complex curved shape. Consider rows of grid-like points locating in front of the knee-form. In Figure 7(a), three such rows are illustrated. These rows are easily determined based on the range of knee-form shape in the z-axis direction. Figure 7(b) illustrates three sections of the knee-form and the part surface obtained by horizontally slicing them along blue, red, and green lines given in Figure 7(a).

Y coordinates of the points on the rows and distances between the points and the front surface of the knee-form in the x-axis direction are measured. The first colliding point in the sliding motion of the knee-form is a point whose y coordinate is smaller than the y coordinate y_r of the side surface of the knee-form contacting to the console part, and the distance between the point and the front surface of the knee-form is the smallest. As shown in Figure 7(b), y coordinates of points $\boldsymbol{p}_0$, $\boldsymbol{p}_1$, and $\boldsymbol{p}_2$ locating in front of the knee-form are all smaller than y_r. In these points, the knee-form collides to $\boldsymbol{p}_1$ because it is the nearest point to the front surface of the knee-form in the x-axis direction. The knee-form then collides to $\boldsymbol{p}_2$ and $\boldsymbol{p}_0$ in its following sliding motion.

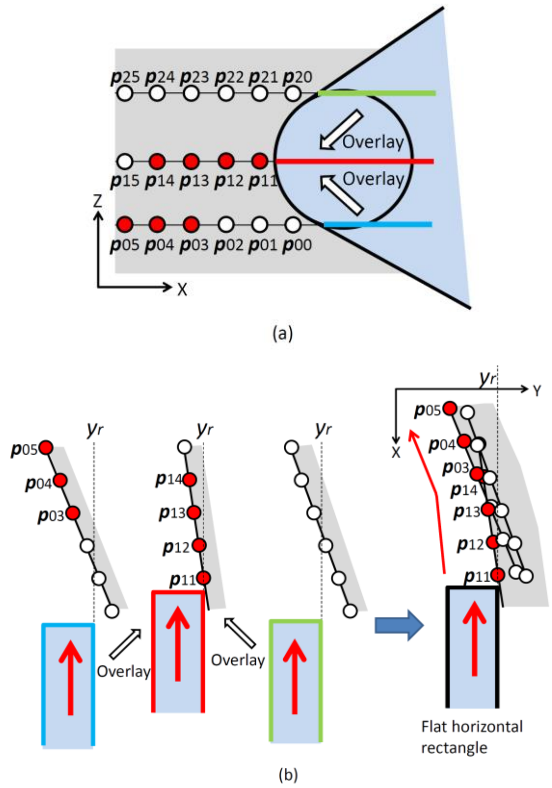

Figure 8. Sliding operation using sorted points.

To detect all colliding points in multiple rows in the sliding motion, we realize an algorithm based on the sorting operations of grid-like points. Figure 8(a) illustrates 3 rows of points existing in front of the knee-form. The knee-form realizes collisions at $\boldsymbol{p}_{11}$, $\boldsymbol{p}_{12}$, $\boldsymbol{p}_{13}$, $\boldsymbol{p}_{14}$, $\boldsymbol{p}_{03}$, $\boldsymbol{p}_{04}$ and $\boldsymbol{p}_{05}$ in its sliding motion in the negative x-axis direction.

After placing the knee-form model on the console surface by a pushing operation, the grid-like points locating in front of the knee-form model are collected. For each point, the distance between the point and the front surface of the knee-form in the x-axis direction is measured, then the points are sorted according to their distances. The sorted points correspond to the result of overlaying the rows of points in front of the knee-form so that the 3 sections of the knee-form are superimposed to be a flat horizontal rectangle as shown in Figure 8(b).

The sorted points are scanned to find out a point whose y coordinate is smaller than y_r (see point $\boldsymbol{p}_{11}$ in Figure 8(b)). This point corresponds to the first colliding point of the knee-form to the surface. The sliding operation is executed by using the sorted points to collect the knee-form colliding points. As shown in Figure 8(b), the sorted points are traced along the leftward curve to collect colliding points $\boldsymbol{p}_{12}$, $\boldsymbol{p}_{13}$, $\boldsymbol{p}_{14}$, $\boldsymbol{p}_{03}$, $\boldsymbol{p}_{04}$ and so on.

The pushing and sliding operations mentioned above are repeated for all the knee-form placements above the grid-points in the zx-plane to determine the colliding points for the knee-form in a specific orientation. The final result is obtained by repeating the operation for each rotation angle in the range between -30 degree to 30 degree. The obtained colliding points are properly connected based on the adjacency information of the grid, and converted to an equivalent polyhedral surface representing the knee-form colliding area on the part. This result is used for displaying and for outputting to the data file in STL format.

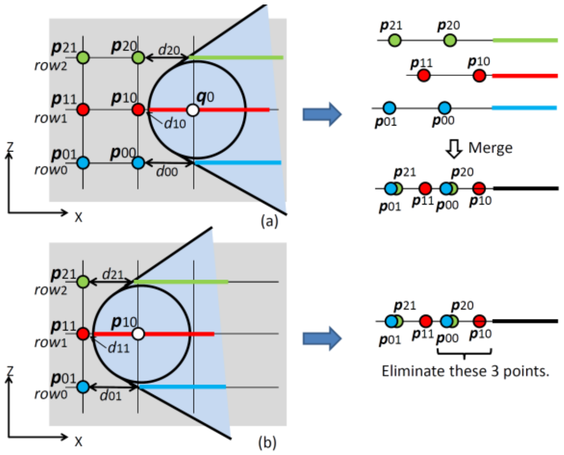

Figure 9. Merging of 3 rows of points for obtaining the sorting result.

4. PERFORMANCE IMPROVEMENT

In our method, sorting operation in the sliding operation is the most time consuming task. Since points on the part surface is defined based on the regular square grid in the zx-plane, horizontal distance between two adjacent points in a row is equal for all rows. Sorting is thus realized by a simple merging operation of rows of points. Consider 3 rows of points as shown in Figure 9(a). For each row, select a point closest to the front surface of the knee-form. They are $\boldsymbol{p}_{00}$ for row_0, $\boldsymbol{p}_{10}$ for row_1, and $\boldsymbol{p}_{20}$ for row_2. Their distances to the knee-form in the x-axis direction are d_{00}, d_{10}, and d_{20} respectively where $d_{10} < d_{20} < d_{00}$. In this case, sorting result of points in the 3 rows is obtained by merging 3 rows of points as $\{\boldsymbol{p}_{10}, \boldsymbol{p}_{20}, \boldsymbol{p}_{00}, \boldsymbol{p}_{11}, \boldsymbol{p}_{21}, \boldsymbol{p}_{01}, \ldots, \boldsymbol{p}_{1i}, \boldsymbol{p}_{2i}, \boldsymbol{p}_{0i}, \ldots\}$.

In our knee-form collision analysis, the knee-form model in a same orientation is positioned for each grid point of the regular square grid in the zx-plane in row by row order. In each row, the knee-form is placed on a grid point in the row from right to left order. In this case, the cost for obtaining the sorted points can be further reduced by reusing the sorting result obtained for the prior collision detection. Figure 9(a) illustrates the sorted points obtained for a knee-form at grid point $\boldsymbol{q}_0$. After collecting the knee-form colliding points by sliding the knee-form from this point, the initial position of the knee-form is shifted to the left side neighbor point of $\boldsymbol{q}_0$ (which is $\boldsymbol{p}_{10}$ as shown in Figure 9(b)). In this case, the sorted points computed for the knee-form at $\boldsymbol{q}_0$ can be reused. Because of the shifting operation from $\boldsymbol{q}_0$ to $\boldsymbol{p}_{10}$, first 3 points $\boldsymbol{p}_{10}$, $\boldsymbol{p}_{20}$, $\boldsymbol{p}_{00}$ in the sorted list are now useless, therefore they are eliminated from the list. Other points in the sorted list is applicable for detecting colliding points with the knee-form in the sliding motion from $\boldsymbol{p}_{10}$.

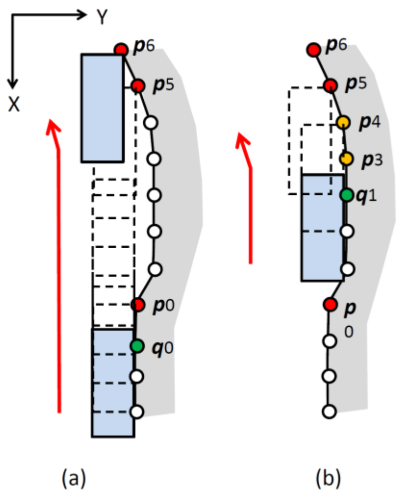

Figure 10. Cancellation of sliding operations.

In our algorithm, some sliding operation can be canceled. Figure 10(a) illustrates a detection process of knee-form colliding points using the sorted points. The knee-form is firstly placed at point $\boldsymbol{q}_0$, then red points are detected as the knee-form colliding points by tracing the sorted points. In the following operation, the initial position of the knee-form is shifted to $\boldsymbol{q}_1$ as shown in Figure 10(b). From this position, the knee-from starts the sliding operation and new colliding points which are yellow points in the figure are detected. The knee-form then reaches point $\boldsymbol{p}_5$ which is a colliding point already detected in the prior sliding motion as shown in Figure 10(a). At this point, the following sliding operations can be canceled because no new colliding points can be found in the operation.

5. COMPUTATIONAL EXPERIMENTS

The proposed knee-form colliding area computation system was implemented using Visual C++ and OpenGL, and a series of computational experiments were performed using an Intel Core i7 Processor (2.6 GHz) with 16 GB memory and an NVIDIA GeForce GTX-960M GPU. We applied the system to several console part models provided by a software company and an automobile company, and the proposed system was found to successfully compute the knee-form colliding area for all parts in 40 to 50 minutes, with the time varying according to the resolution of the grid used in the sampling of the surface points.

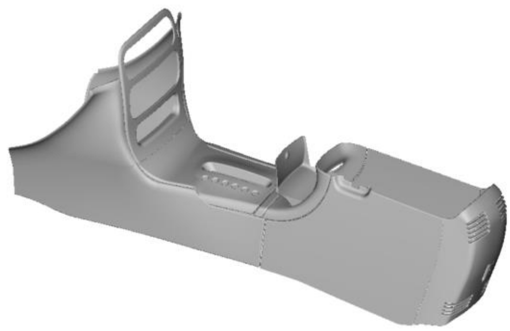

Figure 11. Sample part.

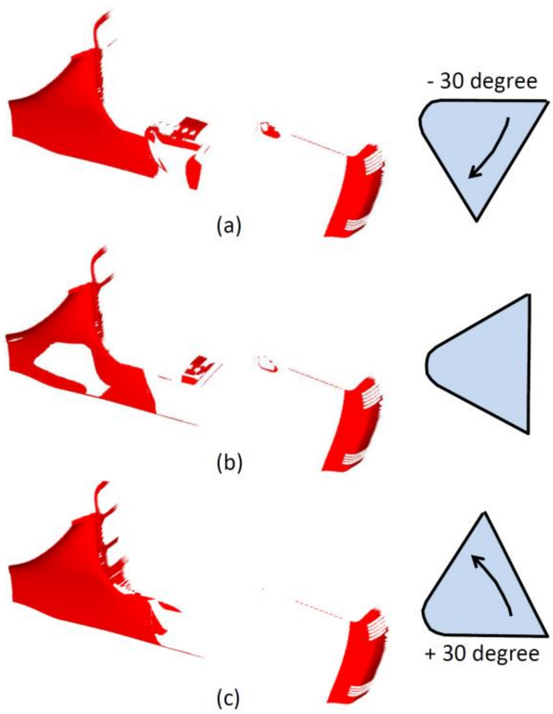

Figure 12. Extraction result using a knee-form in different orientation.

For the purpose of maintaining confidentiality, the computation result for only one sample is shown here. Figure 11 illustrates the sample console model with 369,408 polygons. The left side surface of the part is converted to a grid-like points in Step 1. In this computation, a depth buffer with a grid resolution of 1538 x 801 was used. The grid size, which corresponds to the computation accuracy of our method, was 1.24 mm. In Step 2, the knee-form colliding area is extracted by repeating the push-and-slide operation with the knee-form model with different rotation angle between -30 degree to 30 degree for every one degree interval. 61 extraction results with different knee-form orientations are obtained. Figure 12(a) – (c) show three results of the knee-form colliding area on the part for knee-forms with rotation angle -30 degree, 0 degree, and 30 degree, respectively. Figure 13 illustrates the composition of the 61 results.

Our proposed system was found to require 2519.92 seconds for obtaining the result given in Figure 13. The system can output the polygons of the detected area in STL format, and the automobile designer can verify compliance with the UNECE regulation by superimposing the knee-form colliding area on the CAD model of the console part.

Figure 13. Composition of 61 extraction results.

6. CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a novel algorithm named “push-and-slide” for detecting the knee-form colliding area in the surface of the interior part which is vertical and near parallel to the knee-form approaching direction. This algorithm is developed for automating the UNECE safety regulation inspections of the automobile interior parts.

In the first step of the algorithm, the target surface area of the part is transformed to grid-like points in a high resolution. The knee-form models in various positions and orientations are prepared in the second step. Each knee-form model is pushed to the grid-like points. The model is then moved in the negative x-axis direction to detect possible collisions with the grid-like points on the part surface. In the sliding operation, sorting of the grid-like points locating in front of the knee-form is necessary. By using the grid structure of the points, the costly sorting operation is replaced to a simple merging operation of points.

Our algorithm relies on the grid-based representation of the surface shape of the interior part. We are planning to realize much accurate computation by using a higher resolution grid, which inevitably demand more computation cost in the collision detection. To reduce the cost, use of the parallel processing capability of GPU is considered. 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.

7. REFERENCES

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

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

[Fau08] 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. dx.doi.org/10.2312/SCA/SCA08/155-162

[GAR] Global Auto Regulations, https://globalautoregs.com/unece, United Nations Economic Commission for Europe.

[Inu15] 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. dx.doi.org/10.1016/j.jcde.2014.11.006

[Inu17] Inui M.; Nakano S.; Umezu N.; Asano T.: GPU-based visualization of knee-form contact area for safety inspections, Computer-Aided Design and Applications, 14(3), 2017, 356-365, dx.doi.org/10.1080/16864360.2016.1240456

[JP06] Japanese Patent [P2006-277304A], http://www.j-tokkyo.com/2006/G06F/JP2006-277304.shtml, Toyota Motor Corporation, in Japanese.

[Kes16] Kessenich J.; Sellers G.; Shreiner D.: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5, Addison-Wesley Professional, USA, 2016.

[Li11] Li W.; McMains S.: Voxelized Minkowski sum computation on the GPU with robust culling, Computer-Aided Design, 43(10), 2011, 1270-1283. dx.doi.org/10.1016/j.cad.2011.06.022

[Mol99] Moller T.; Haines E.: Real-Time Rendering. A. K. Peters Ltd., Natick MA, USA, 1999.

[Yam11] 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. dx.doi.org/10.1109/icma.2011.5986280