Fast Detection of Head Colliding Shapes on Automobile Parts

Masatomo INUI* and Nobuyuki UMEZU**

*Department of Intelligent Systems Engineering, Ibaraki University
4–12–1 Nakanarusawa, Hitachi-shi, Ibaraki 316–8511, Japan
E-mail: inui@mx.ibaraki.ac.jp

Abstract
In the ECE regulation, any surface of the interior part of automobiles must have sufficient roundness where a sphere of diameter 165mm, which is equivalent to the average head size of infants, may collide. A system for automatically detecting sphere contacting shape on the automobile part is developed. In the current practice, some Japanese automobile manufactures detect the shape by using the virtual milling method. In this method, a milling simulation with a ball end cutter of diameter 165mm is executed on the part to detect the sphere contacting shape. The method generally needs a lot of computation time for the detection. In this paper, the authors propose an improved virtual milling method for fast detecting the sphere contacting shape. Our algorithm initially generates points sufficiently covering the visible surface of the model. For each point, the contact condition of the cutter is evaluated by using the inverted offset surface of the part. An inverted offsetting method accelerated with the depth buffer mechanism of GPU is introduced. An experimental system is implemented and some computational experiments are performed. Our system can detect the sphere contacting shape on the part surface with approximately 1 million polygons in a few seconds.

Key words: Safety Verification, Automobile Design, CAD, Collision Detection, Inverted Offsetting, GPU

1. Introduction
Safety is always a primal concern in designing automobiles. Various regulations are specified on the interior shape of the automobile so that it does not hurt the driver and passengers when the automobile crashes. ECE (Economic Commission for Europe) defines several regulations on the interior shape to reduce impact on the infant head in the car crash. They are referred to ECE-17, -21 and -25[1]. ECE-17 defines shape conditions on the automobile seat. ECE-21 and ECE-25 define the roundness condition of the corner shape of the interior part and the headrest, respectively. Similar regulations are defined in other countries including Japan[2].

Figure 1 illustrates the basic idea of ECE-21 and ECE-25. The corner shape of any interior parts and headrests of the automobile must have roundness larger than R3.2 if a sphere of 165mm diameter may collide with it. This sphere size is equivalent to the head size of the infant. In ECE-17, similar shape condition is demanded on the car seat. Any component parts of a seat assembly must have roundness larger than R3.2 if the sphere of 165mm diameter can touch. Seat designers thus often put wire thicker than R3.2 under the seat as shown in Fig. 2, so that the head cannot directly collide to the sharp corner of the seat part.

In the current practice, these regulations are checked by specialists in the final design stage of the automobile part. They use many 2D section drawings of the part for manually checking the colliding condition of a circle of diameter 165mm on them. Some Japanese automobile manufactures detect the sphere contacting shape on the part by using the virtual milling method. In this method, a milling simulation is executed on the part with a ball end cutter.
cutter of diameter 165mm to detect the sphere contacting shape on the part. The existence of the sharp corners on the detected sphere contacting shape is then reported to the designer, and such corners are resolved by re-designing the part. This method generally needs a lot of computation time for the detection. Additional cost and time for the detection and re-designing the part is a serious problem in many automobile manufacturers.

![Figure 1 Geometric meaning of safety regulation ECE-21 and ECE-25.](image1)

![Figure 2 Geometric meaning of safety regulation ECE-17.](image2)

In this paper, the authors propose an improved virtual milling method for fast detecting the sphere contacting shape. Our algorithm initially generates sufficient points covering the visible surface of the model. For each point, the contact condition of the ball end cutter is evaluated by using the inverted offset surface of the part. An inverted offsetting method accelerated with the depth buffer mechanism of GPU is introduced to realize the fast evaluation. An experimental system is implemented and computational experiments are performed. Our system can detect the sphere contacting shape on the part with approximately 1 million polygons in a few seconds.

The organization of the paper is as follows. In the next section, some related studies are briefly reviewed. The basic processing flow of our sphere contacting shape detection algorithm is explained in the third section. The details of the algorithm are then given in its following section. In the fifth section, experimental detection results with complex CAD models of automobile interior parts are given. Limitation of the algorithm and our future work are also discussed.

2. Related Studies

Although verification of the safety regulation is an important topic for any automobile manufacturers, technology for automating the verification process has not been well studied. Many manufacturers still use manual based methods as mentioned above.

Toyota Motor Corporation submitted some patents about the automatic verification of ECE-17, -21, and -25 in 2005(3). Toyota’s method uses the offset surface of the automobile
part CAD model with the radius of the infant head (≈ 82.5mm). By using the offset surface, possible head colliding surface is detected on the CAD model and the roundness of the surface is checked. Offset computation of the complex 3D shape is generally difficult, unstable, and time-consuming. Other automobile companies may have some in-house technologies for the verification that are not published.

The authors proposed a safety verification system based on the intersection detection between a sphere and a CAD model\(^4\). A new parallel intersection detection algorithm combined with a hierarchical geometric data management was introduced to accelerate the computation. By using GPU technology for parallel intersection detection and OBB tree for hierarchical data management, the verification result can be visualized in a few seconds. This method is, however, not applicable to the interior parts with sharp corners because proper placement of spheres on them is difficult.

3. Algorithm Outline

3.1. Input and Output

Our detection system requires CAD models of automobile seats, interior parts, and headrests as input data. The authors assume that the model shape is represented as a set of triangular polygons. Most CAD systems provide a function to output the model data as a group of polygons, for example in the STL format. After the computation, the detected surface portions on the CAD models where the infant head may collide are painted with red color in the display.

Fig. 3 Virtual milling method for detecting the infants head colliding shapes. (a) An interior part CAD model, (b) cutter path for milling the part with a ball end cutter of 165mm diameter, and (c) an overlaid figure of the milling simulation result and CAD model.

3.2. Improved Virtual Milling Method

Some Japanese automobile manufacturers detect the infants head contacting shape on the interior part by using the virtual milling method. Figure 3 illustrates its idea. By using the CAM system, a cutter path for milling interior part with a ball end cutter of 165mm diameter is generated. Fig. 3 (a) shows an interior part CAD model and (b) shows a scan type cutter path for milling this shape with the ball end cutter. Milling simulation system is then invoked to obtain a result shape after the milling operation with the path and the cutter. The resulting shape and the interior part model are overlaid as shown in (c) to visualize the surface portion where the interior part model is exactly machined (blue shapes in the figure). Such a surface corresponds to the shape where the infant head may collide.

The authors’ group in Ibaraki University has developed a GPU accelerated CAM system with the cutter path generation and milling simulation functions\(^5\). By using this system, the virtual milling operation mentioned above can be completed in a few minutes. This computation speed is much faster than the CAM systems currently used in the companies for the virtual milling, it is, however, still too slow for the designer to use the system in an interactive manner.

Most of the computation time in the virtual milling is consumed in the milling simulation to distinguish the shape where the sphere of 165mm diameter may collide. Basing on
Improved virtual milling method based on the inverted offsetting the part surface. 

**Input:** CAD data (STL) of part model.

**Step 1:** Point set definition on the part surface.

**Step 2:** Inverted offsetting the part surface with the ball end cutter.

**Step 3:** Projection of the point on the inverted offset surface to the part surface.

**Output:** Coloring the points where infant’s head can collide.

This analysis, the authors propose an improved method without using the milling simulation. Figure 4 illustrates the basic idea of our method. As shown in the figure, the cutter path for a ball end cutter of 165mm diameter is on the inverted offset surface of the interior part $S$. Instead of generating a cutter path and executing milling simulation with the path, our method simply projects the point on the inverted offset surface to the part surface $S$ in the reversed direction of the normal vector at the point. For example, a point $p$ on the inverted offset surface is projected to the part surface in its reversed direction of the normal vector at $p$ as shown in the figure. Point $q$ obtained by the projection corresponds to a contacting point of a cutter of 165mm diameter locating at $p$. This projecting operation is executed to all points on the inverted offset surface, and the infants head contacting shape on the part (red points on the surface) is detected.

### 3.3. Processing Flow

Figure 5 shows the input, output, and the processing flow of the algorithm. 

- **Step 1:** Points enough to cover the visible surface of the input CAD model are distributed on the model surface.

- **Step 2:** Consider a ball end cutter of diameter 165mm, and generate the inverted offset surface of the interior part with the cutter. In the surface generation, the cutter is set so that its axis is aligned in the viewing direction of the part.
Step 3: Consider dense points covering the inverted offset surface. For each point \( p \), compute the normal vector of the inverted offset surface at the point, and project \( p \) to the part surface in the reverse direction of the normal vector. After the projection, detect the closest point on the visible surface to the projected point. The detected point is painted with red to show the head contacting position on the part surface.

4. Details of the Algorithm

In this section, details of the processing steps of our algorithm are explained. In the following explanation, the viewing direction of the object is assumed to be in the opposite direction of the \( z \)-axis of the object coordinate frame. The pixel grid of the frame buffer is also assumed to be aligned with respect to the \( x \)- and \( y \)-axes of the object. The same method is applicable to any viewing direction and the frame buffer orientation. In such a case, the object and the viewer must be transformed so that the viewing direction and the pixel grids are aligned to the object coordinate frame before the processing. After detecting the head colliding shape, the computation results are transformed back to the original position and orientation by applying the inverse transformation.

4.1. Step 1: Point Set Definition on the Part Surface

Our system notifies possible unsafe shape to the designer by painting red color to surface portions on the part model where infants head may collide. Coloring operation is realized by generating sufficient number of points on the visible surface of the model in step 1, and assigning red color to the points selected in step 3. The authors adopted a point generation method based on the hidden surface elimination with the depth buffer\(^6\). Figure 6 illustrates our method for generating point on a visible surface.

A depth buffer with the same number of entries as the frame buffer is prepared. All the entries of the depth buffer are initialized to a sufficiently small value. After the preparation, polygons of the objects are rendered into the frame buffer in an arbitrary order. During the rendering, the color and the \( z \)-value of a point on a polygon corresponding to each pixel \([x, y]\) of the frame buffer is checked. If the \( z \)-value is larger than the value currently in the depth buffer, then the color and \( z \)-value of the point replace the old ones in the buffers. This process is repeated for all polygons, and the result is the pixels with the color of points on the visible surfaces and the entries of the depth buffer with the \( z \)-values of the points.

For each pixel, our algorithm places a point \([x, y, z]\) on the object surface where \( x \) and \( y \) correspond to the pixel coordinates and \( z \) means the \( z \)-value stored in the depth buffer at \([x, y]\). Since the points obtained in this method completely cover the visible surface of the object, any surface portion can be properly opaque by simply painting its corresponding set of points.

![Fig. 6 Point generation method on the part surface.](image-url)
4.2. Step 2: Inverted Offsetting of Part Surface

The inverted offset surface of the object and a ball end cutter of 165mm diameter is obtained by sweeping the inverted cutter on the part surface while maintaining its center point on the surface. The axis of the cutter is set to be parallel to the viewing direction of the object, which is the z-axis of the object coordinate frame. The top surface of the swept volume corresponds to the inverted offset surface.

![Fig. 7 Component spheres, cylindrical pins, and slabs of the inverted offset shape.](image)

Since the orientation of the cutter is vertically fixed, the top surface is generated by the spherical part of the inverted cutter. This shape is equivalent to the top surface of a Boolean union shape of spheres, cylinders and thick slabs being placed on the part surface as follows (see Fig. 7):

1. Spheres of 165mm diameter are placed on all vertices of the surface.
2. On each edge $e$ of the surface, a cylindrical pin shape of 165mm diameter is placed so that its center axis and $e$ become coincident.
3. On each polygonal face $f$, a slab shape of the same area and thickness of 165mm is placed so that the center plane of the plate and $f$ become coincident.

Exact computation of the Boolean union shape of many spheres, cylinders and slabs is time-consuming and unstable, therefore Z-map based approximation algorithm for the top surface generation is adopted. In this algorithm, the top surface is represented as a set of points covering the surface. The points are computed on an axis-aligned regular grid in the $xy$-plane. From each grid point, an upward ray is extended along the $z$-axis direction, and the highest intersection point between the ray and the top surface of the swept volume of the inverted cutter (i.e. the Boolean union shape of spheres, cylindrical pins, and slabs) is computed. This operation is repeated for all grid points, and a set of points densely covering the top surface is obtained.

Because of the similarity to the point generation method used in step 1, the authors applied the same depth buffer based algorithm in our system. Before the computation, all spheres, cylindrical pins, and slabs are finely tessellated and all the entries of the depth buffer are initialized to a sufficiently small value. All polygons of component objects of the swept volume of the inverted cutter are then rendered into the frame buffer. For each grid point in the $xy$-plane, the depth buffer mechanism automatically samples the highest point, and the result is the set of points densely covering the top surface of the swept and the pixels with the color of the points.

In our implementation, each sphere, cylindrical pin, or slab shape is defined to have a
unique color. Before the rendering operation, the geometric properties of the spheres, pins, and slabs are recorded in the database with their corresponding color information as their keys. After the rendering, the color of each point represents its original component shape (sphere, pin, or slab) of the swept volume. Figure 8(a) illustrates the inverted offset surface with a ball end cutter of 165mm diameter. This surface is generated for a part model given in Fig. 3 (a). The points on the inverted offset surface actually have colors as shown in Fig. 8 (b). Each color has a unique correspondence to the original component sphere, cylindrical pin, or slab shape of the swept volume.

![Fig. 8 (a) Inverted offset surface of the part shown in figure 3(a). (b) The points on the inverted offset surface with different colors.](image)

**4.3. Step3: Projection of Offset Point to Part Surface**

For each point \( p \) covering the inverted offset surface, the system computes the normal vector \( n \) of the inverted offset surface at the point, and project \( p \) to the surface of the original part model in the reverse direction of the normal vector. Since point \( p \) is painted with a unique color corresponding to its original shape (sphere, cylindrical pin, or slab), the geometric property of the original shape is retrieved from the database by using the color information as the key. The normal vector at the point is easily and precisely computed based on the retrieved geometric information. As shown in Fig. 9 (a), if the original shape is a sphere, a vector from the center point of the sphere to \( p \) becomes the normal vector at \( p \). The normal vectors for the cylindrical pin and the slab shape case can be computed in a similar manner. See Fig. 9 (b) and (c).

By using the reversed vector of the normal vector at \( p \), the point is projected back to the original surface. The projected point \( q \) is obtained by \( q = p - r \cdot n \) where \( n \) means the unit normal vector at \( p \) and \( r \) means the radius of the ball end cutter, which is 82.5mm in the case. After projection, the closest point on the part surface to the projected point is detected. Since this point corresponds to a possible head colliding location, it is painted red to show the un-safety.

**5. Computational Experiments**

By using the technology mentioned above, the head colliding shape detection system is implemented using Visual C++ and OpenGL, and computational experiments are performed. A PC with Intel Core i7 Processor (3.4GHz), 8GB memory and NVIDIA GeForce GTX-580 GPU is used in the experiments. A window for displaying the CAD model is set to be 1920 × 1080 pixel size.

Table 1 shows the required computation time for detecting the head colliding shape on 5 interior parts of an automobile. The table includes number of polygons representing the part shape, number of points distributed on the model surface, and required computation time for coloring the head colliding points on the model surface. As shown in the table, un-safety shape with possible infant’s head collision can be detected in a few seconds even for a part.
with approximately 1 million polygons. Figure 10 illustrates a detection result of a sample part A. The red regions in the figure mean the un-safe shape on the part with possible infant’s head collision.

The virtual milling method has one limitation in the head colliding shape detection. This method considers the sphere contact problem as a sphere accessibility problem when the sphere approaching along the viewing direction. It cannot detect some un-safe shapes where a sphere can contact while the sphere cannot access there in a linear motion along the viewing direction. Figure 11(a) shows such a result with our system based on the virtual milling method, and (b) shows the result with our prior system\(^{(4)}\). As shown in the figure, the virtual milling based system failed to detect an un-safe shape appearing inside of the part.
because the sphere cannot access there along the viewing direction. Since the detection of the inside of the part is usually not requested in the safety verification, our system still has practical usefulness for the designer.

6. Conclusions

The authors proposed a method for assisting the safety verification of ECE-17, -21, and -25 specified on the interior part of the automobile. In the regulations, the surface of the interior part is requested to have sufficient roundness where a sphere of diameter 165mm may collide. Our system initially generates points sufficiently covering the visible surface of the model. For each point, the contact condition of the cutter is evaluated by using the inverted offset surface of the part.

An inverted offsetting method accelerated with the depth buffer mechanism of GPU is introduced to perform the fast evaluation. An experimental system is implemented and computational experiments are performed. Our system can detect the head colliding points on the part surface in a few seconds with a complex CAD models. By using this software, the interior part designer with less knowledge on the safety regulations can efficiently detect and modify the CAD model in a very short time period.

Our system is still in an experimental stage. The authors are now preparing the field test of the system in the actual automobile design process. Further improvement of the algorithm based on the comments and requests from the designer is included in our future work.

References

(1) Crash Network, European Regulations related to Crash Testing (online), available from <http://www.crash-network.com/Regulations/ECE_Regulations/ece_regulations.html>, (accessed 2013-04-15).
(2) Japan Automobile Standards Internationalization Center (online), available from <http://www.jasic.org/e/index_e.htm>, (accessed 2013-04-15).
(3) Kogane, H., Methods for supporting design processes(online, in Japanese), available from <http://www.j-tokkyo.com/2006/G06F/JP2006-277304.shtml>, (accessed 2013-04-15).
(4) Yamazaki, S., Baba, T., Umezu, N., and Inui, M., Fast Safety Verification of Interior Parts of Automobiles, Proceedings of IEEE ICMA(2011-08), pp.1957–1962.
(5) Inui, M. and Ohta, A., Using GPU to accelerate die and mold fabrication, IEEE Computer Graphics & Applications Magazine, Vol.27, No.1(2007), pp.82–88.
(6) Moller, T. and Haines., E., Real-time rendering(1999), A K Peters.
(7) Inui, M., Fast inverse offset computation using polygon rendering hardware, Computer-Aided Design, Vol.35(2003), pp.191–201.