Research Article

Development of the Korean Spine Database and Automatic Surface Mesh Intersection Algorithm for Constructing *e*-Spine Simulator

Dongmin Seo, Hanmin Jung, Won-Kyung Sung, and Dukyun Nam

Korea Institute of Science and Technology Information, Daejeon 305-806, Republic of Korea

Correspondence should be addressed to Dukyun Nam; dynam@kisti.re.kr

By 2026, Korea is expected to surpass the UN's definition of an aged society and reach the level of a superaged society. With an aging population come increased disorders involving the spine. To prevent unnecessary spinal surgery and support scientific diagnosis of spinal disease and systematic prediction of treatment outcomes, we have been developing *e*-Spine, which is a computer simulation model of the human spine. In this paper, we present the Korean spine database and automatic surface mesh intersection algorithm to construct *e*-Spine. To date, the Korean spine database has collected spine data from 77 cadavers and 298 patients. The spine data consists of 2D images from CT, MRI, or X-ray, 3D shapes, geometry data, and property data. The volume and quality of the Korean spine database are now the world's highest ones. In addition, our triangular surface mesh intersection algorithm automatically remeshes the spine-implant intersection model to make it valid for finite element analysis (FEA). This makes it possible to run the FEA using the spine-implant mesh model without any manual effort. Our database and surface mesh intersection algorithm will offer great value and utility in the diagnosis, treatment, and rehabilitation of patients suffering from spinal diseases.

1. Introduction

e-Health technologies supporting high efficiency and low-cost medical service based on IT-BT convergence technology have grown in importance because medical expense is increasing and many people are asking for customer-driven medical service in today's aging society. In particular, chronic diseases such as degenerative spinal diseases, high blood pressure, diabetes, and cancer cannot be treated adequately and require steady self-care. Therefore, *e*-Health technologies are being widely used to treat chronic diseases widely. Nowadays, next generation chronic disease management technologies are being extensively researched based on mobile service, cloud computing service, social network service, big data analysis service, genome sequencing service, and computer simulation service [1–3]. By 2026, Korea is expected to surpass the UN's definition of an aged society and reach the level of a "superaged society." As a result, degenerative spinal diseases and related surgical procedures will increase exponentially. As of 2007, medical expenses incurred due to

spinal surgery in Korea totaled 178.6 billion won/year, and the treatment duration reached 1.82 million days/year as shown in Figure 1. The resulting medical burden and economic loss are increasing at a rapid rate. Spinal diseases make everyday life of people impossible and impede economic activities, resulting in a compromised quality of life. Between 2002 and 2004, spinal surgery increased at a particularly high rate among the older demographic, by 68.2% among ages 60–69 and by 94.6% among those aged 70 and older. Among the leading causes of hospitalization for ages 65 and older, spinal diseases ranked number 2 with over 65,000 instances [4].

To prevent unnecessary spinal surgery resulting from overtreatment, systematic prediction of treatment outcomes is necessary, including scientific diagnosis, scientific effect analysis, and analysis of spinal rehabilitation exercises. Computer simulations have been utilized in biomechanical research for the past three decades. Today, advances in computer hardware and software are bringing continually increasing simulation accuracy. We have been developing *e*-Spine which is composed of a spine database (DB) and
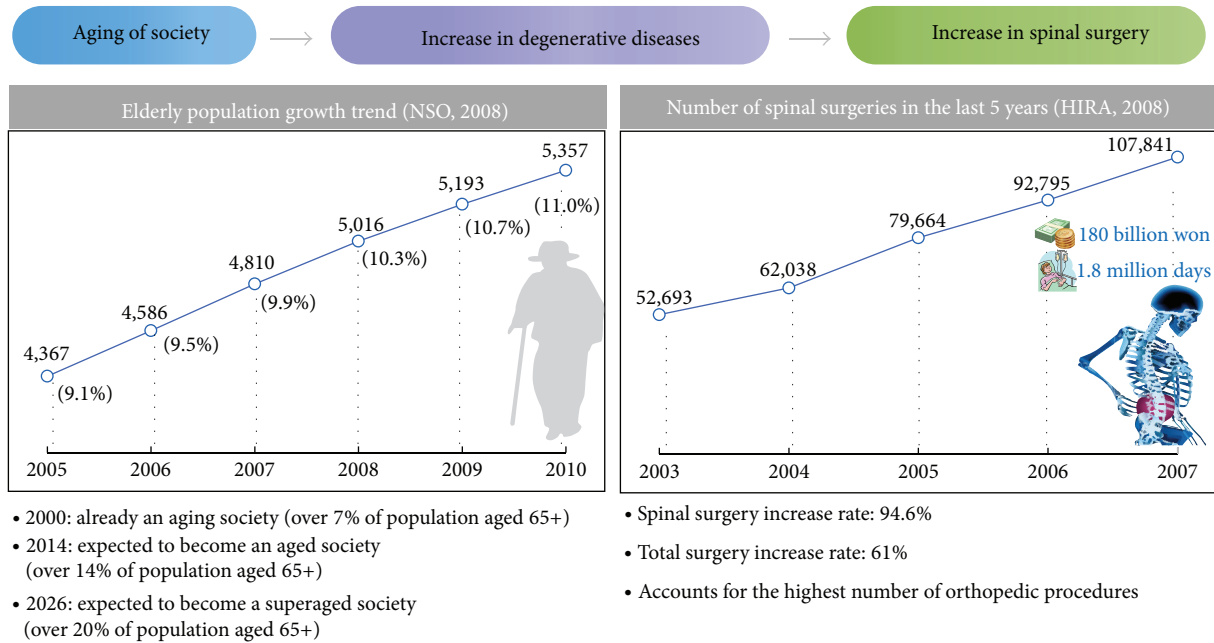
Figure 1: Status of an aged society and spinal surgeries in Korea.

a simulation environment for running a computer simulation model of human spines created by mathematically calculating images, geometries, and properties of human spines and will allow virtual testing without using a real human spine. For this, the high quality spine data and a surface mesh intersection algorithm are essential for realizing *e*-Spine (The proposed intersection algorithm is applicable to a triangular surface mesh. In this paper, the surface mesh means the triangular surface mesh). In virtual surgery with an implant, surgeons or biomechanics researchers select the desired model from the spine DB and try to merge the spine model with an implant model as a virtual case. At this point, the two models of the spine and implant are automatically merged by means of the surface mesh merging functionality. With the merged model, they run the computer simulation and analyze the simulation result. To build the spine database (DB) for *e*-Spine, we have produced and collected many images, geometry, and property data of spines from the Korean cadavers and patients with normal spine or degenerative spinal diseases [5, 6]. For automatic mesh merging functionality, we implement the surface mesh intersection algorithm used in the procedure of remeshing the spine-implant intersection model for finite element analysis (FEA). This makes it possible to run the FEA using the spine-implant mesh model without any manual effort.

In this paper, we present our spine database and surface mesh intersection algorithm in detail. The rest of the paper is organized as follows. Section 2 explores *e*-Spine. Section 3 explains the Korean spine database obtained from cadavers and patients with degenerative spinal diseases. Section 4 describes our surface mesh intersection algorithm on the Korean spine data. Section 5 shows our experimental results. Finally, Section 6 presents the conclusion.

2. *e*-Spine

e-Spine is a computer-run simulation model created by mathematically modeling collected human spinal image data, which allows virtual testing without using a real spine. Figure 2 shows a comparison with a vehicle's navigation system. For optimal and safe driving, a navigation system collects map information, models a map, and predicts a route. Similarly, for optimal treatment, *e*-Spine collects spine images, models 3D spine models, and predicts virtual testing and results. The expected effect of *e*-Spine is as follows:

(i) acquisition of reliable, economical, advanced IT-based medical support technologies that can be used in the diagnosis and treatment of degenerative spinal diseases;

(ii) strengthened market competitiveness for Korea's medical equipment industry through the utilization of *e*-Spine;

(iii) reduction of medical expenses and improving the quality of life during old age by making available reliable, affordable IT-based medical technologies.

3. Korean Spine Database

3.1. Spine Sample Selection. There are distinct differences, such as facial features, skin color, and hair, between the human races. Differences in the organs of the body are also evident. Generally, spine data obtained from hospitals are only CT, MRI, or X-ray, which focus on particular parts of the spine with degenerative diseases. The data are only available for diagnosing spinal diseases. Therefore, to construct *e*-Spine and support the Korean spine research, we constructed the
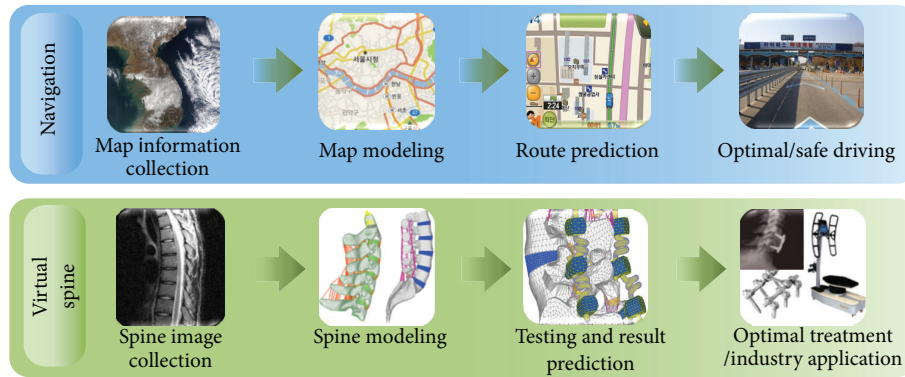
FIGURE 2: Comparison between a vehicle's navigation system and *e*-Spine.
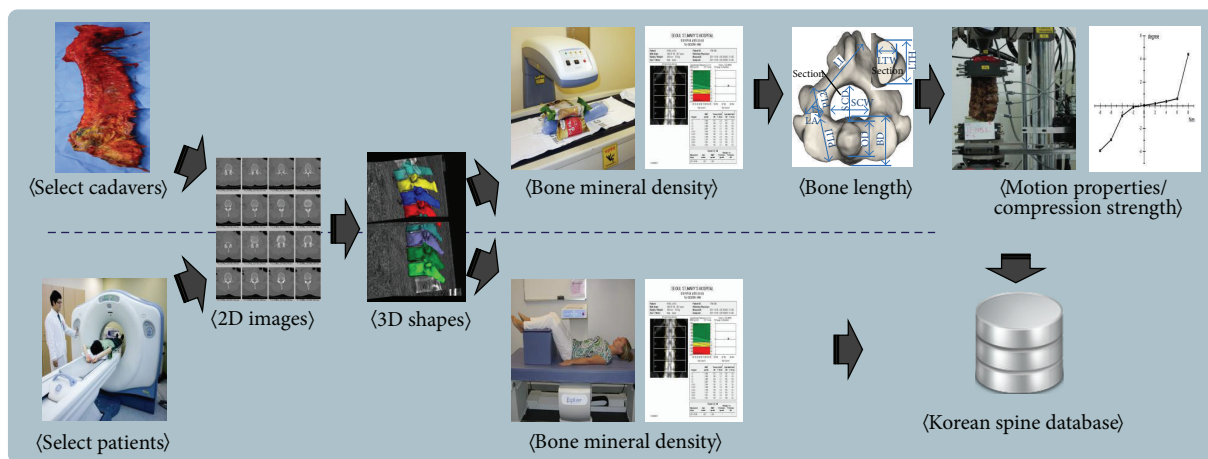

FIGURE 3: Process of the Korean spine database construction.

Korean spine data for degenerative spinal diseases. Figure 3 shows the process of the Korean spine database construction. The metadata and schema of our Korean spine database are accepted as a standard from Telecommunications Technology Association and are discussed in detail in [7]. To date, we have collected spine data from 77 cadavers and 298 patients. Among the leading causes of hospitalization for ages 65 and older, spinal diseases ranked number 2 next to cataracts. Therefore, most of the data were obtained from aged cadavers and patients over 50 but some were gathered from younger patients because even young people occasionally suffer from degenerative diseases of their vertebras.

3.2. Spine 2D Image and 3D Shape Data. Our spine 2D images consist of various images from X-ray, CT, MRI, and BMD because different types of images are useful for diagnosing different types of diseases. These images are stored in the Digital Imaging and Communication in Medicine (DICOM) file format, which is a standard format for storing medical data. Some images like those from CT often comprise a series of images produced with small intervals and these images need to be stored and managed as a group in our database for efficient search and management. To accomplish this, we add series numbers to the end of each image file

name while sharing the prefix of each file name and manage the file name prefix with start and end of series number as additional metadata. From a series of CT images, we can make a 3D shape model of the spine body by piling up image series in order and filling up small triangles to correct intervals between the images. This process was done with Maya, 3D animation software. Therefore, users can analyze the correlation between cross-sectional CT images and 3D shape models.

3.3. Spine Geometry and Property Data. Spine geometry data are lengths and angles of key elements for representing shapes and features of the spine. We selected 481 spine geometry data items in [8–10] and measured those of our cadavers. The selected spine geometry data items are accepted as a standard from Telecommunications Technology Association and are discussed in detail in [11]. We did not measure the geometry data of a patient because we did not harvest the spine from a patient. The selected spine geometry data are utilized to analyze characteristics of the Korean spine. Figure 4 shows the selected elements in the spine geometry data related to cervical vertebras.

Figure 5 shows the selected elements in the spine geometry data related to thoracic vertebras.

(a) Cervical vertebra 2
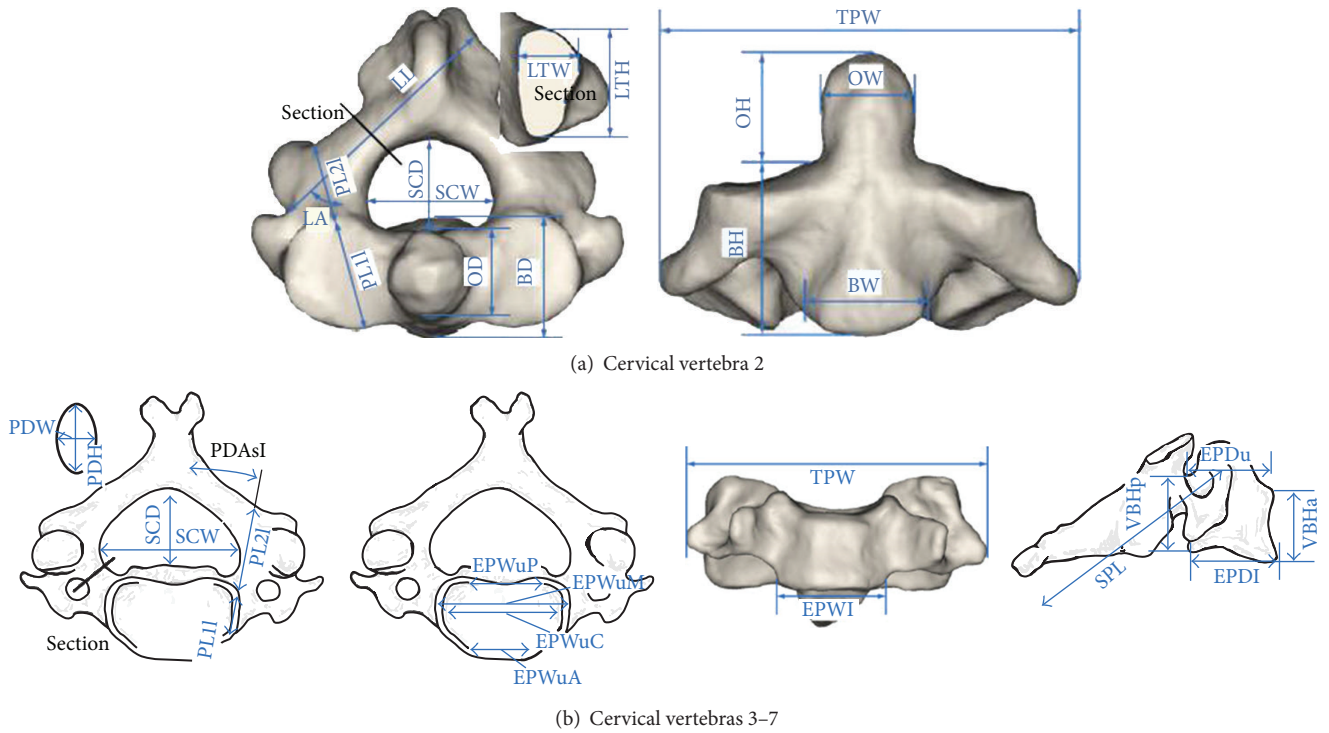

(b) Cervical vertebras 3–7

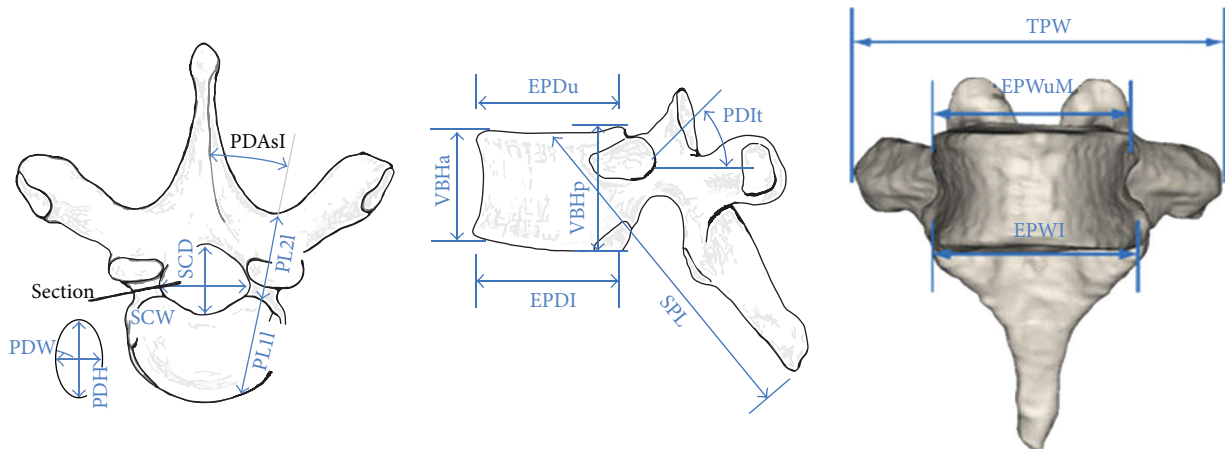FIGURE 4: Geometry data related to cervical vertebras.

FIGURE 5: Geometry data related to thoracic vertebras.

Figure 6 shows the selected elements in the spine geometry data related to lumbar vertebras.

To obtain spine property data, we distribute a spine to cervical vertebras (C3–C7), thoracic vertebras (T1–T6, T7–T12), and lumbar vertebras (L1-S1) and measured flexion-extension, lateral bending, twist, and disc compression between bones. Table 1 shows the test conditions and Figure 7 shows the testing devices consisting of spine simulator of MTS and Liberty of Polhemus for measuring spine property data.

3.4. Degenerative Diseases on Spine. The human spine is divided into three parts: cervical, thoracic, and lumbar

TABLE 1: Test conditions for measuring property data.

Region	Maximum moment	Load step	Holding time
Cervical	1-2 Nm	4	30 sec
Thoracic	4–6 Nm	4	30 sec
Lumbar	8 Nm	4	30 sec

vertebras, starting from head. As humans age, they often suffer from various degenerative diseases of the cervical and lumbar vertebras or intervertebral discs, while they rarely suffer from diseases of the thoracic vertebras. Therefore, we decided to target the following degenerative diseases
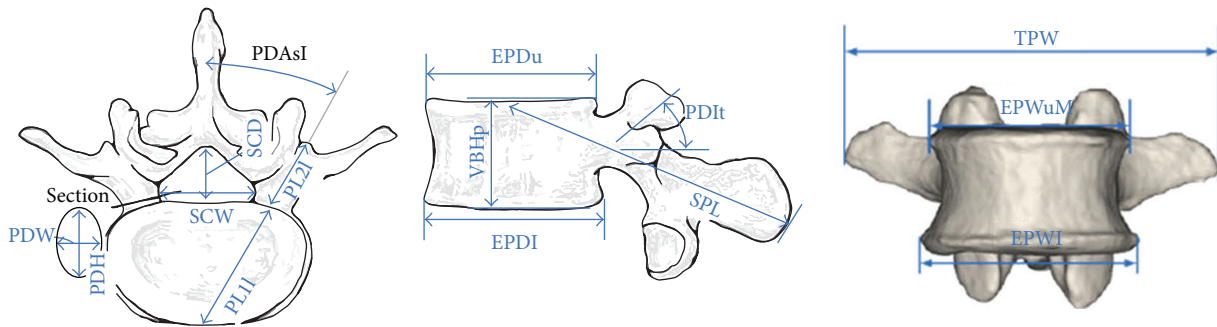
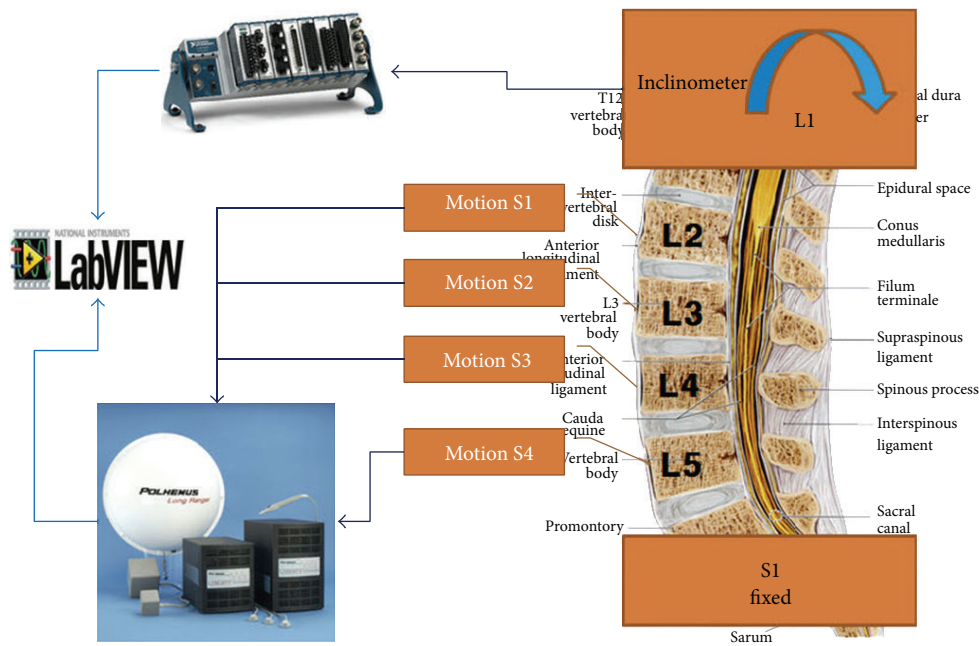Figure 6: Geometry data related to lumbar vertebras.

Figure 7: Testing devices for measuring property data.

frequently occurring at the cervical or lumbar vertebras of older people and further decided to classify these diseases into four or five grades (i.e., grade 0 to grade 3 or grade 4) through consulting with several specialists in the human spine. The larger the grade number is, the more severe the disease is. Grade 0 indicates normal and grades 1, 2, 3, and 4 correspond to *mild*, *moderate*, *severe*, and *very severe* grade of disease, respectively. Below is a brief description of the selected degenerative diseases on the spine.

(i) Osteophyte. This occurs at both cervical and lumbar vertebras and can be further classified into anterior osteophyte and posterior osteophyte. This can be generally diagnosed by X-ray and CT.

(ii) Ligament calcification (or ossification). There are three kinds of ligaments around cervical vertebras: anterior and posterior longitudinal ligaments and ligamentum nuchae. Thus, this disease can be further divided into three subdiseases: ossification of the anterior longitudinal ligament (OALL), ossification

of the posterior longitudinal ligament (OPLL), and ossification of ligamentum nuchae (OLN). This can generally be diagnosed by X-ray and CT.

(iii) Endplate sclerosis. This occurs at both cervical and lumbar endplates and can generally be diagnosed by X-ray, CT, and MRI.

(iv) Disc height reduction. This occurs at both cervical and lumbar intervertebral discs and can generally be diagnosed by X-ray and CT.

(v) Disc herniation. This occurs at both cervical and lumbar intervertebral discs and can generally be diagnosed by X-ray, CT, and MRI. This is usually classified into five grades: normal, bulging, protrusion, extrusion, and sequestration.

(vi) Disc degeneration. This occurs at both cervical and lumbar intervertebral discs and can generally be diagnosed by X-ray, CT, and MRI. This is also classified into five grades.
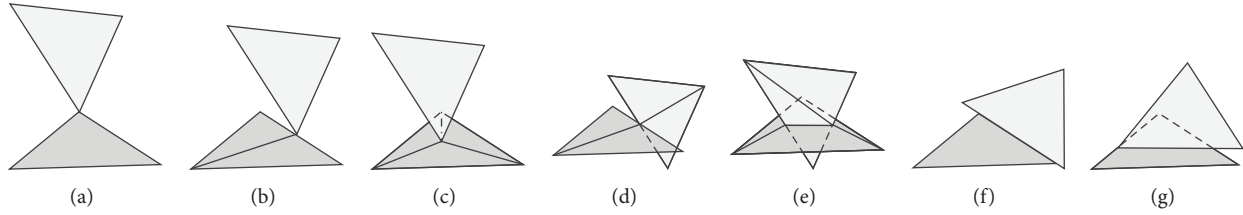
FIGURE 8: Searching for intersection regions. (a) Coincident nodes intersection. (b) Nodes along edge intersection. (c) Nodes onto face intersection. (d) Edges cutting other edge intersections. (e) Edges cutting face intersection. (f) Edges overlapping other edge intersections. (g) Edges overlapping face intersection.

(vii) Facet joint degeneration. This occurs at both cervical and lumbar facet joints and can generally be diagnosed by X-ray and CT.

(viii) Spondylolisthesis. This occurs at both cervical and lumbar vertebras, but most commonly at lumbar vertebras, and can generally be diagnosed by X-ray.

(ix) Osteoporosis. This occurs at any vertebras and can be generally diagnosed by X-ray and BMD (bone mineral density).

3.5. Statistics about the Korean Spine Data. To date, we have collected the Korean spine data from 77 cadavers and 298 patients with normal spine or degenerative spinal diseases. The detailed statistics on the collected data is shown in Table 2.

4. Automatic Surface Mesh Intersection Algorithm

To support automatic mesh merging functionality, we implement the mesh intersection algorithm. We applied a tracking algorithm in order to rapidly and accurately explore an intersection. The tracking algorithm finds the intersection regions along the intersection curve from a valid tracking point [12, 13]. After finding an initial intersection point, the algorithm starts at the initial intersection point and creates an intersection curve along the direction to new intersection points. While this happens, the algorithm also searches for intersection regions. We use the existing data structure in [12] and add one factor which can distinguish between spines and implant meshes.

4.1. Finding Intersection Points

4.1.1. Searching for Intersection Regions. A plane equation is derived from the outer product of the points of mesh elements. An intersection point is calculated using the topological relation between two intersection planes. After that, we derive the angle from the inner product of three points of a triangle element. If the sum of the derived angles is 2π, the intersection point is inside the triangle element. The intersection of planes on three-dimensional space has various cases. When the tracking algorithm searches for an intersection, we determine the intersection from an interrelation of a line and

TABLE 2: The number of cadavers or patients according to degenerative diseases.

Type	Vertebra	Disease	Number
Cadavers		Normal spine	22
	Lumbar	Osteophyte formation	50
Patients	Cervical	Normal spine	23
		Disc degeneration	4
		Disc height reduction	6
		Disc herniation	6
		Endplate sclerosis	6
		OPLL	23
		Ossification	16
		Osteophyte	10
	Thoracic	Compression fracture	16
	Lumbar	Compression fracture	32
		Disc degeneration	20
		Disc height reduction	15
		Disc herniation	17
		Endplate sclerosis	11
		Facet joint degenerative	12
		LDK	20
		Osteophyte	25
		Osteoporosis	20
		Spondylolisthesis	16

a side and there are cases of intersection between intersected triangles in Figure 8.

4.1.2. Tolerance. The intersection cannot be mathematically determined because the numerical calculation of a computer does not work on consecutive space. We have to define a tolerance to calculate an intersection. Suppose that the gap has the difference d as in Figure 9. If $d \geq tolerance$, two triangles are not intersected. Otherwise, they are intersected. When two meshes intersect, the shape of intersected meshes changes. This is because the intersection points are derived from the tolerance by moving existing shapes. The shape of an implant in a spine-implant insertion model should not change using the tolerance. Algorithm 1 defines an intersection generation procedure with the tolerance. In case of an intersection point between edge E and face F, the point's position depends on which one is an implant mesh between

```
Input: E (edge), F (surface mesh), tolerance
Output: P (intersection point), p and q (points of E), a, b, and c (points of F)
IF E ∈ M_implant, F ∈ M_spine
    IF Distance: p to abc < tolerance
        RETURN P = p;
ELSE IF Distance: q to abc < tolerance
    RETURN P = q;
ELSE IF E ∈ M_spine, F ∈ M_implant
    IF Distance: p to abc < tolerance
        RETURN P = Create new point (qp, F);
ELSE IF Distance: q to abc < tolerance
    RETURN P = Create new point (pq, F);
```
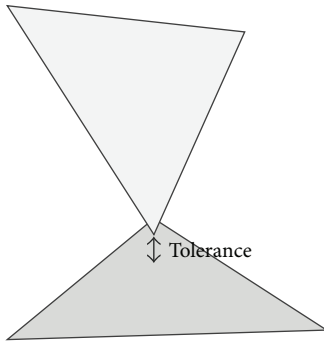
ALGORITHM 1: Tolerance.

FIGURE 9: Node-face intersection with tolerance.

edge E and face F. If E is an element of an implant mesh, the intersection point is the start or end point of an edge E. On the other hand, if F is an element of an implant mesh, the intersection point is generated by intersecting F and the extension of E. New mesh points are located on the edge or face of implant mesh because our unique constraint is that implant mesh does not change as much as possible. The tolerance is automatically set up according to the size of a model or by user's configuration.

4.2. Tracing Algorithm. In the model for the volume finite element, an intersection curve is always closed, so tracking an intersection can start from an initial intersection point P_0 to the intersection progress direction. After finding an initial intersection point, the intersection points continuously search along the intersection progressing direction [12–14]. An intersection curve is continuously generated by inputting the continuously generated intersection points to an intersection curve C. Using the intersection curve data, we retrieve intersection regions and generate new mesh of the intersection region.

4.3. Meshing. To generate a spine-implant finite element intersection model, the generated mesh on an intersection region should satisfy three conditions.

(i) The shape of the spine model can be changed but the shape of the implant cannot be changed. This is because this spine-implant structure analysis simulates the process of inserting a strong durable implant into a relatively weak durable spine. If we allow the shape change of the implant, nonintended stress concentration can occur so the implant shape does not allow the change.

(ii) We do not generate more meshes than the user needs. The size and the shape of a mesh work are important factors. A small size mesh is densely formed in a complicated shape, but if the number of meshes is blindly large, we waste more analysis resources than necessary.

(iii) A mesh should have the right quality for the finite element analysis. The intersection point, which can only be created by an intersection search, has an unbalanced gap. A mesh with the intersection points is not enough for finite element analysis, so we need to add fixed points along the intersection curve to create a mesh for finite element analysis.

When generating the intersected mesh model, we have to satisfy the above three requirements. First of all, the points of the intersection curves should be reorganized. Algorithm 2 reorganizes the points of the intersection curve. Figure 10(b) shows the points to define the shape of an implant among the points of an intersection curve. $d_{\min}$ is defined as the minimum gap of all other points. The unequally distributed points P_n on the cross-curve are reorganized equally based on the $d_{\min}$ standard.

The intersection points created on the points or the lines of an implant element are presented as follows:

$$T\left(P, F_{\text{implant}}\right) = \left(F_{\text{implant}}, r, n\right) \quad (r = 0, 1 \mid n = 0, 1, 2). \quad (1)$$

The algorithm automatically creates a new element network to keep the shape of the original model by using Delaunay triangulation in Figure 10(d). The element created by Delaunay triangulation may be not suitable for finite element analysis. Therefore, the algorithm reorganizes the triangulation element in a spine model through a remesh process. In addition, we remesh the intersection element with neighboring elements to prevent a sharp form. The tracking

```
Input: C (Intersection curve), P (Intersection point)
Output: E_hard (Hard edges), V_hard (Hard points)
FIND d_min (minimum distance between P_k and P_{k+1},
       P_k and P_{k+1} are points on the vertexes or edges of F_implant);
LET P_0 = get first vertex in C;
FOR i = 0; i < end of index P
   IF T(P_i, F) = (F_implant, r, n) (where = 0, 1 | n = 0, 1, 2)
       LET P_1 = P_i;
       IF distance between P_0 and P_1 > d_max × k (user variable)
           FOR j = 0; distance between P_{0−j} and P_1 > d_min × k; j++
               LET P_{0−j+1} = Create new point onto C;
               LET S_new = Create new segment (P_{0−j}, P_{0−j+1});
               INSERT P_{0−j+1} into V_hard;
               INSERT S_new into E_hard;
           END
       LET P_0 = P_1;
   END
```

ALGORITHM 2: Node generation algorithm on the intersection curve to prevent the change of implant model.
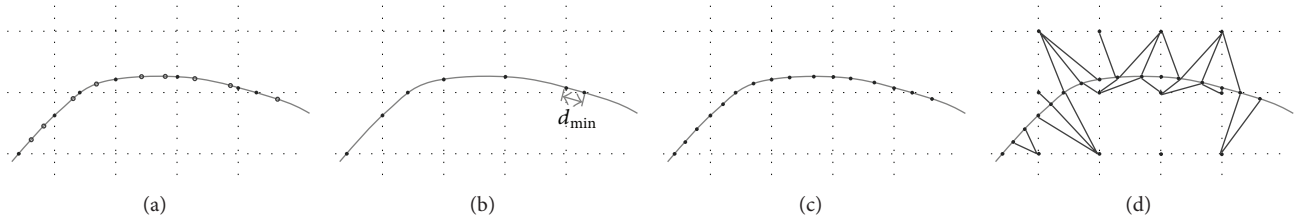

| (a) | (b) | (c) | (d) |

FIGURE 10: Meshing algorithm. (a) Original intersection curve and intersection points (black and white points are the intersection points on implant and spine models, resp.). (b) Intersection points on implant models. (c) Intersection points reconstruction. (d) Triangulation.

algorithm searches for the continuous two points P_1 and P_2 to satisfy this condition. In Figure 11, we generate the points $P_{1−n}$ with a uniformed gap. The algorithm stores the points on the new generated curve and constructs the line S_{new}, between newly generated points. The new organized line is sequentially stored as a hard edge, E_{hard}. The hard edge is fixed and a base form on mesh or remesh processes.

4.4. Inserting Implant. After processing triangulation and remesh, we remove the existing elements of intersection to insert the reorganized elements. The algorithm intersects two finite element models by creating the inserted part inside a target element model. We search for the intersection part with the outer product of surface elements along the intersection curve. In Figure 12, the algorithm removes the region where an implant is inserted in the spine model after searching for the part of the implant inserted in the spine model. It finishes an automatic intersection processing of an implant and a spine by inserting the implant according to the direction to the spine model.

5. Performance Evaluation

Figure 13 shows an accurate validation of motion properties with the extension and flexion on L5-S1 on a spine.

The line with the x symbol is the values of motion properties obtained by [15]. The solid line is obtained by [16]. The line with the squared symbol is obtained by our database. The motion properties from [15] and our database are constructed through experiments. The motion properties from [16] are constructed through computational models. In conclusion, we can confirm that our spine data is accurate because the shape of each line is similar.

A mesh is closely related to an analysis result and the intersection processing in the intersection regions influences the existing shape. Therefore, in most cases, a mesh is manually created to guarantee the mesh quality. We show that the proposed algorithm creates an appropriate mesh for a structural mesh. The algorithm creates a mesh by automatic intersecting of spine and implant models. As the spine model, we use a three-layered spine model consisting of a vertebral arch, an outer vertebral body, and an inner vertebral body. To evaluate the usefulness of the automatic intersection algorithm, we prepare three automatic intersection models with different sizes of implants, as mesh sizes 1.0, 1.5, and 2.0 in Figure 14.

In Figure 15, we set up the analysis conditions: 10,000 load to the $−Z$ direction on top of the spine with fixed 6 degrees of freedom (DOF). We performed the analysis using an ABAQUS solver.
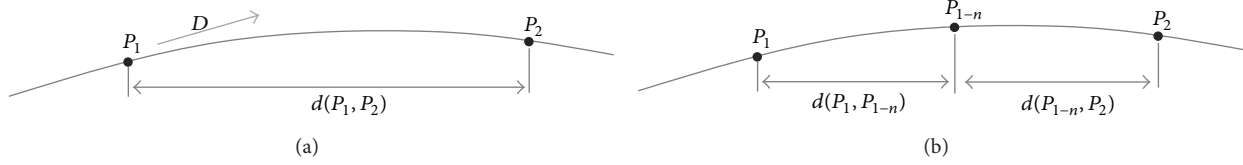
Figure 11: Remeshing algorithm. (a) P_1 and P_2 with distance d on the intersection curve. (b) New generated point P_{n-1} between P_1 and P_2.
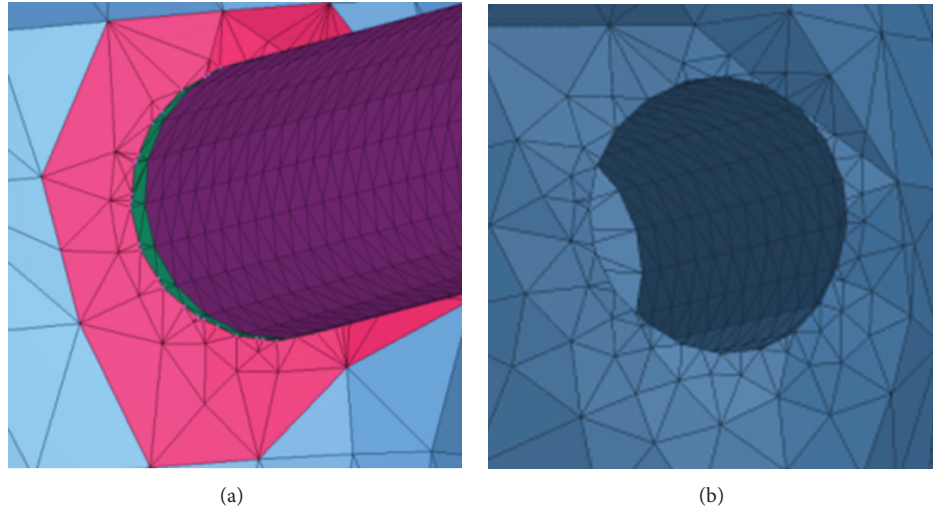

(a)

(b)

Figure 12: Inserting implant. (a) Remeshed intersection model according to the intersection curve. (b) Hole model based on implant shape for automatic intersection.
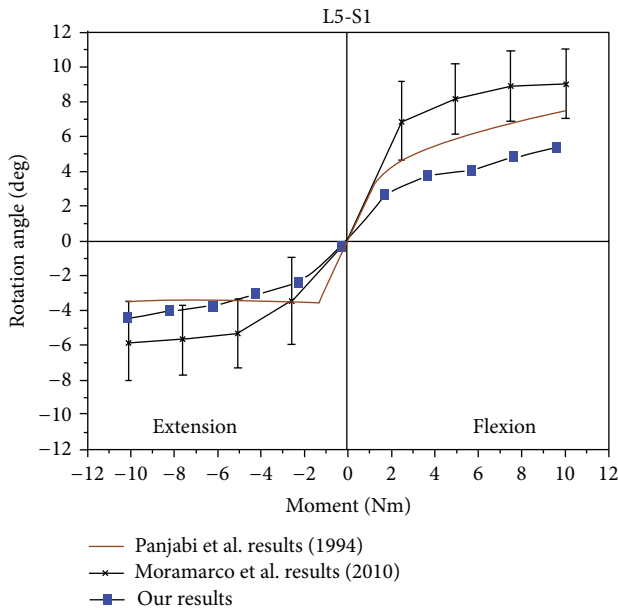

Figure 13: Accurate validation of our spine data.

Figure 16 shows the common patterns even though there is little derivation according to the element size. We cannot find the abnormal results due to intersection regions. Therefore, we expect that the proposed automatic intersection algorithm can be used without human intervention.

6. Conclusion

We collected various Korean spine data items from 77 cadavers and 298 patients with normal spine or degenerative spinal diseases to provide a wealth of information related to spine to medical students, physicians, and biomedical engineers. We also propose the automatic surface mesh intersection algorithm for spine and implant models. Our algorithm automatically remeshes the spine-implant intersection model to make it valid for finite element analysis (FEA). A spine-implant intersection model is manually created so far. The automatic intersection procedure using the proposed intersection algorithm reduces the manual labor time for spine-implant model. Therefore, this makes it possible to run the FEA using the spine-implant mesh model without any manual effort. We show the validation of intersection mesh quality in the simulation. In the near future, we plan to define criteria to check the quality of a spine-implant intersection mesh and perform the numerical analysis based on the defined criteria.

We will offer our spine data and surface mesh intersection algorithm to many researchers and doctors to foster spine research development. In conclusion, our spine data and surface mesh intersection algorithm will be used to realize reliable, economical, and advanced IT-based medical support technologies that can be used in the diagnosis and treatment of degenerative spinal diseases. Furthermore, our technical skills will be used to vitalize the related research fields and
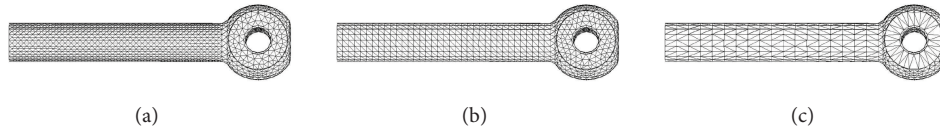
FIGURE 14: Three different implant models. (a) Mesh size 1.0. (b) Size 1.5. (c) Size 2.0.
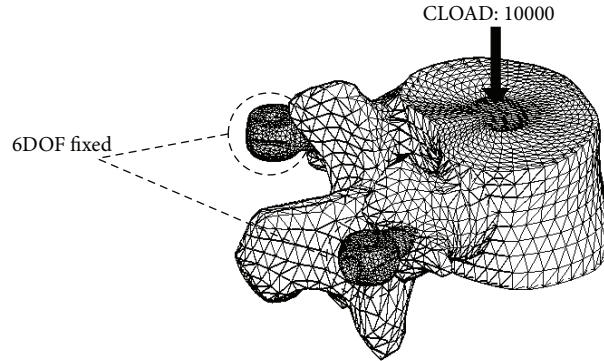

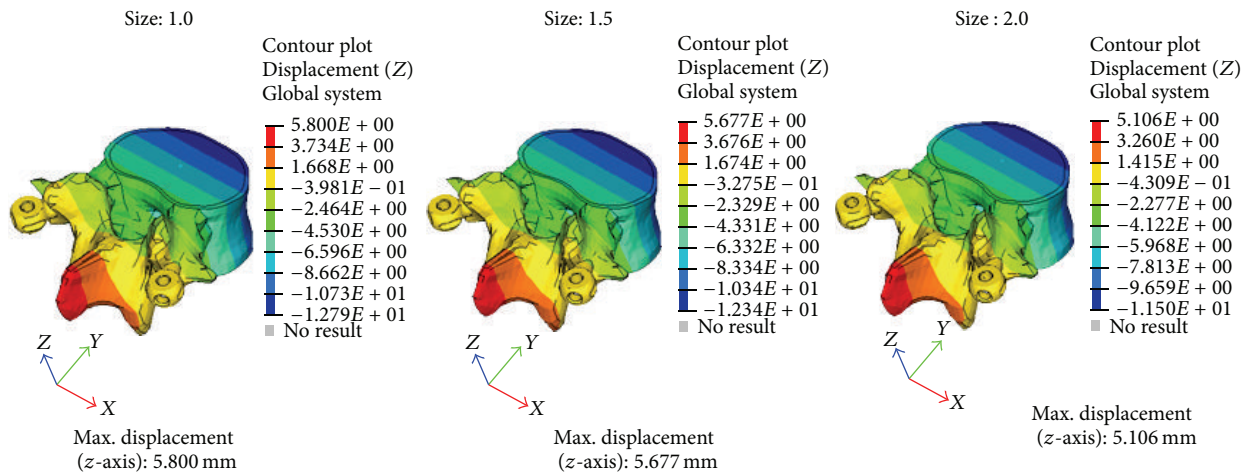FIGURE 15: Boundary and load condition for evaluation.

FIGURE 16: Analysis results.

industries by providing the developed human spinal information database, model, and virtual simulator to relevant researchers.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

References

[1] N. Howard and E. Cambria, "Intention awareness: improving upon situation awareness in human-centric environments," *Human-Centric Computing and Information Sciences*, vol. 3, no. 9, pp. 2–17, 2013.

[2] P. Gargiulo, T. Helgason, P. Ingvarsson, W. Mayr, H. Kern, and U. Carraro, "Medical image analysis and 3-d modeling to quantify changes and functional restoration in denervated muscle undergoing electrical stimulation treatment," *Human-Centric Computing and Information Sciences*, vol. 2, no. 10, pp. 2–11, 2012.

[3] M. Brahami, B. Atmani, and N. Matta, "Dynamic knowledge mapping guided by data mining: application on Healthcare," *Journal of Information Processing Systems*, vol. 9, no. 1, pp. 1–30, 2013.

[4] National Health Insurance, "2009 Major surgery statistics," *Press Release*, vol. 12, no. 7, pp. 1–15, 2010.

[5] "*e*-Spine project," http://www.espine.kr/.

[6] D. M. Seo, S. W. Lee, S. B. Lee, S. H. Lee, H. M. Jung, and W. K. Sung, "Implementation of spine database system on Korean patient with degenerative spinal diseases," in *Proceedings of International Smart Media Application (SMS '12)*, pp. 1–3, 2012.

[7] S. B. Lee and S. H. Lee, "Metadata for managing the human body data," Telecommunications Technology Association, TTAStandard (TTAK.KO-10.0723), 2013.

[8] M. M. Panjabi, J. Duranceau, V. Goel, T. Oxland, and K. Takata, "Cervical human vertebrae: quantitative three-dimensional anatomy of the middle and lower regions," *Spine*, vol. 16, no. 8, pp. 861–869, 1991.

[9] M. M. Panjabi, K. Takata, V. Goel et al., "Thoracic human vertebrae: quantitative three-dimensional anatomy," *Spine*, vol. 16, no. 8, pp. 888–901, 1991.

[10] M. M. Panjabi, V. Goel, T. Oxland et al., "Human lumbar vertebrae: quantitative three-dimensional anatomy," *Spine*, vol. 17, no. 3, pp. 299–306, 1992.

[11] S. B. Lee and S. H. Lee, "The measuring elements for representing the human spine shape," Telecommunications Technology Association, TTAStandard (TTAK.KO-10.0724), 2013.

[12] A. H. Elsheikh and M. Elsheikh, "A reliable triangular mesh intersection algorithm and its application in geological modelling," *Engineering with Computers*, vol. 30, no. 1, pp. 143–157, 2014.

[13] S. H. Lo and W. X. Wang, "A fast robust algorithm for the intersection of triangulated surfaces," *Engineering with Computers*, vol. 20, no. 1, pp. 11–21, 2004.

[14] S. H. Lo, "Automatic mesh generation over intersecting surfaces," *Numerical Methods in Engineering*, vol. 38, no. 6, pp. 943–954, 1995.

[15] M. M. Panjabi, T. R. Oxland, I. Yamamoto, and J. J. Crisco, "Mechanical behavior of the human lumbar and lumbosacral spine as shown by three-dimensional load-displacement curves," *Journal of Bone and Joint Surgery*, vol. 76, no. 3, pp. 413–424, 1994.

[16] V. Moramarco, A. P. del Palomar, C. Pappalettere, and M. Doblaré, "An accurate validation of a computational model of a human lumbosacral segment," *Journal of Biomechanics*, vol. 43, no. 2, pp. 334–342, 2010.