Journal of
Mechanical
Science and
Technology

Journal of Mechanical Science and Technology 21 (2007) 2133~2140

Interference verification of injection molds on the web-based CAD viewer

In-Ho Song[1] and Sung-Chong Chung[2,*]

[1]*Department of Mechanical Design and Production Engineering, Hanyang University, Seoul 133-791, Korea*
[2]*School of Mechanical Engineering, Hanyang University, Seoul 133-791, Korea*

(Manuscript Received November 20, 2006; Revised June 22, 2007; Accepted July 1, 2007)

Abstract

As the interference between mold parts occurs in a small region, it is difficult to find the interference at the design stage. It is usually found at the assembly stage of injection molds, which delays both design and manufacturing lead-times. Commercial CAD systems with interference inspection functions have been used so far. However, they are very expensive for the mold industry, and inadequate for performing collaborative work over the Internet. An interference examination system executable on a lower cost CAD viewer is required before releasing final drawings in the mold industry.

In this paper, an interference examination algorithm for the Web-based interference verification system applicable to a single-level assembly is studied for injection mold design processes. To design a collaborative system executable on the Internet, a lightweight CAD file converted from a CAD data is used as a native file of the CAD viewer with interference verification capability. The validity of the developed system is confirmed through case studies.

Keywords: CAD viewer; Collaboration; Injection mold; Interference; Single-level assembly; Web

1. Introduction

Plastic products are used widely and their significance is increasing. They are manufactured through injection molding. Their life cycle is becoming shorter and shorter. Injection mold companies reduce lead-time by using 3D CAD systems. However, investment cost and maintenance expenditure of CAD systems are high, injection mold companies want to apply collaborative CAD viewers instead of the CAD systems for reduction of the lead-time. Several commercial web-based viewing tools, such as Spinfire of Actify Inc. [1] and AutoView of Cimmetry Inc. [2] have been developed. However, the commercial viewing tools offer only simple measurement functions and do not have accurate interference examination functions applicable to injection mold design processes [3]. In addition, since they are general viewers, it is difficult to apply the viewers to the design process of injection molds [4].

Ye et al. [5] proposed an automatic assembly algorithm using definitions of hierarchical relationships and geometrical constraints of injection mold parts. Chin and Wong [6] studied a knowledge-based evaluation system in the conceptual design stage of injection molds. Shin and Lee [7] proposed an interference examination algorithm to check surfaces of injection molds. They described a search algorithm to eject the mold product by using side cores. Interferences at side cores are modified automatically. However, application of this paper was limited to certain forms of injection molds.

In order to apply CAD viewers to the design verification process of injection molds, a draft examination system [4] was developed through the incorporation of the draft inspection algorithm into the previously developed Web-viewer [3]. An optimized lightweight

*Corresponding author. Tel.: +82 2 2220 0444, Fax.: +82 2 2298 4634
E-mail address: schung@hanyang.ac.kr

CAD file [3] produced from a commercial CAD file is used to develop the dimension examination, markup and draft inspection modules. However, interference verification of injection molds has not been studied yet.

Interference of injection molds generates fatal problems, such as impossible assembly, damage of injection molds, etc. Since interferences of injection mold parts occur in small regions, it is difficult to find interferences at the design stage. They are usually found at the assembly stage of injection molds. It is required to inspect and compensate for the interference before releasing final drawings. Interference between an angle pin and an evasion hole of the angle pin directly affects the fitting of a slide. Considerable time and cost are required to compensate for the interferences in the manufacturing process of injection molds.

In this paper, an AABB (axis-aligned bounding boxes) tree [8] and a data structure of the lightweight CAD file [9] are used to develop an interference examination algorithm for a single-level assembly composed of one group of solids. By combining this interference inspection algorithm with the previously developed Web-viewer [3], a Web-based interference verification system for injection molds is developed in this paper. Validity of the developed system is confirmed through case studies.

2. Interference parts

Interferences of injection mold primarily occur by slide cores, angle pins, ejector pins, bolts, etc. Interference of a slide core occurs at the curved surface between the slide and the locking block. In this case, the slide core and the locking block are not assembled due to the interference. Fig. 1(a) shows an example of the interference between the slide core and the locking block.

An angle pin is a locking unit that fixes a slide core preventing interference between the product and the injection mold. It changes the length and the inclination angle of the angle pin according to working distance of the slide core. Interference between an angle pin and an angle pin evasion hole occurs owing to length and inclination angle of the angle pin. This damages slide cores. Fig. 1(b) shows this example.

Interferences of ejector pins or bolts mainly occur owing to positioning error between the ejector pin and the ejector pin hole. Fig. 2(a) shows an example of

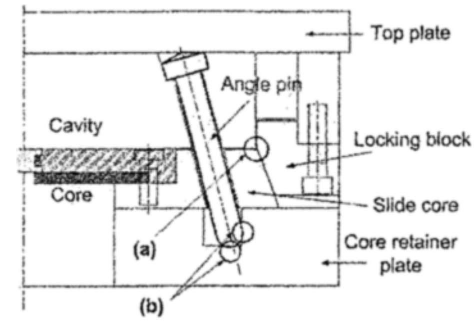

Fig. 1. Interference between two parts: (a) slide core and locking block, (b) angle pin and angle pin hole.

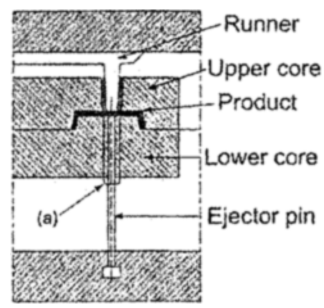

Fig. 2. Interference between the ejector pin and the core.

the interference.

3. Related researches

Interference of objects in virtual reality should include an interaction among objects, such as interference occurring in actual reality. However, in order to express interference objects in virtual reality, a large amount of time and cost is required. It needs hierarchical object representation methods to realize the effective interference examination in virtual reality. BV(bounding volume) and space division methods have been used for hierarchical object representation [10, 8]. OBB (oriented bounding box) tree and Spheretree methods use BV to check interference. Gottshalk proposed an interference inspection algorithm using the OBBtree [10]. He introduced a fast and accurate separating axis theorem. However, the efficiency of the system in the case of non-convex hull and parts of an injection mold is poor. Hubbard [11] proposed an interference verification method using the Spheretree and Space-Time bounds. However, model cubes used as injection mold parts need many spheres, and empty spaces created by the spheres degrade the performance of the interference examination.

Interference inspection methods using the space division technique are Octree, BSP (binary space partitioning) tree etc. Vemuri et al. [12] proposed an algorithm to verify the interference in the flow process of small parts with curved surfaces through the Octree. They proposed a fast interference inspection method of objects including convex hulls or non-convex hulls. Ar et al. [13] introduced the concept of self-customized data structures, and investigated it in the case of BSPtree for interference verification. However, comparing the space division method with the BV method, the space division is inefficient due to extensive computational amount. In addition, those interference examination algorithms are difficult to implement the interference verification of injection molds composed of axis-aligned cubes and/or cylinder elements.

In this paper, an interference examination method composed of AABBtree [8] and the hierarchical data structure of lightweight CAD files [9] is proposed for a single-level assembly. The AABBtree algorithm verifies interference through creation of aligned BVs with respect to coordinate planes of an object. Using this method as well as maximum and minimum vertexes of the BVs, interference verification between bounding boxes is conducted rapidly and simply. It is possible to maximize the performance of AABBtree to inspect part interferences of injection molds composed of axis-aligned parts. If BVs have interferences, BVs of the subordinate hierarchy are inspected whether the subordinate BVs have interferences. Geometric information of mold parts is included in the lightweight CAD file. The structure of the lightweight CAD file and the detail interference examination method are described in the next chapter.

4. Interference verification of a single-level assembly

4.1 Data structure

In this paper, an efficient interference examination algorithm is devised by using the hierarchical data structure of the previously developed lightweight CAD file [9]. Data structure shown in Fig. 3 is used to propose an interference examination algorithm adopting AABBtree and the structure of the lightweight CAD file. Solid means a solid entity of a CAD file. Face contains surface information of a solid. Face is composed of triangular meshes and edges. The triangular meshes are applied for the visualization of CAD files. Using this data structure, solids, faces and triangle meshes are able to be successively searched and inspected for interference examination. In order to reduce the interference inspection time between solids and faces, the AABBtree is applied to exclude non-touching entities during the inspection process.

4.2 Interference examination algorithm

Fig. 4 shows the interference examination process

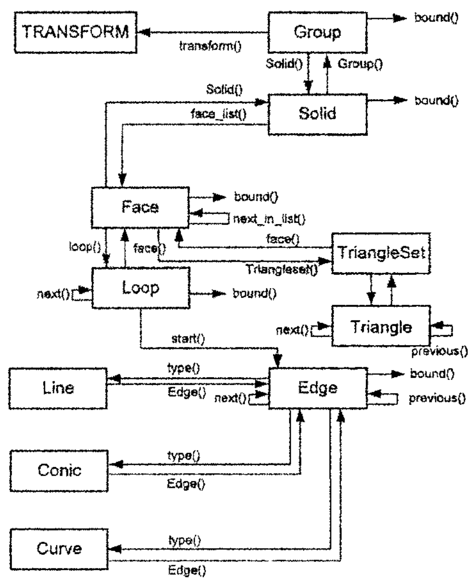

Fig. 3. Structure of the lightweight CAD file.

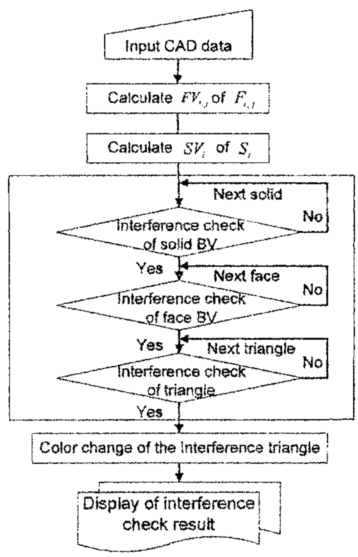

Fig. 4. Algorithm for the interference examination.

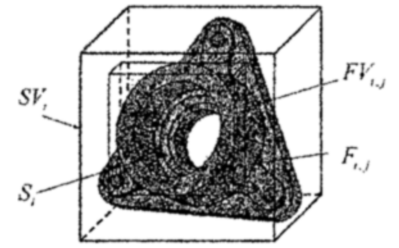

Fig. 5. BVs of a solid and faces.

of a single-level assembly composed of one group of solids. Following 5 steps are used for the interference inspection:

(1) First step is the computation of a BV of a face. Maximum and minimum values of vertexes included in the face are extracted from the vertex information. In Fig. 5, BV of a face is computed by the maximum and minimum values ($FV_{i,j}$) that are stored at the class of face ($F_{i,j}$).Where i is the number of solids and j the number of faces. This process is repeated for other faces.

(2) Second step is the computation of a BV of a solid. Comparing maximum/minimum values of $FV_{i,j}$ included in S_i, SV_i is obtained. In Fig. 5, BV of the i th solid (S_i) is given by SV_i.

(3) Third step is the interference examination of solids. The interference inspection of SV_i , BV of the i th solid, and BV of other solids is performed repeatedly. If no interference is found, interference examination of SV_{i+1}, BV of the next solid, and BV of remaining solids are executed. These processes are iterated to complete the interference verification of BVs of all solids. A more detailed method of interference examination of BV of solids is suggested by Ye et al. [5]. Details are described in section 4.3

(4) Fourth step is the interference verification of faces. $F_{i,j}$ is one of the faces included in the solid S_i. If interference is detected in BVs of two solids, interference examination of BVs, $F_{i,j}$ and $F_{i+1,k}$, is performed. This process is repeated to execute interference inspection of BVs of all faces contained in two solids.

(5) Fifth step is the interference verification of triangular meshes. If interference between BVs of two faces occurs, the first triangular mesh $T_{i,j,l}$ of $F_{i,j}$ is selected. And inspect interference with triangular meshes of $F_{i+1,k}$. If no interference is found, inspect the interference between $T_{i,j,l+1}$ of $F_{i,j}$ and $T_{i+1,k,m}$ of $F_{i+1,k}$. In this case, l and m are a sequence of triangular meshes. The process is iterated to inspect interferences of all triangular meshes included in $F_{i,j}$ and $F_{i+1,k}$. In order to visualize results of the interference examination, the color of the meshes with interference is modified during the inspection process. Using this procedure, it is possible to find interference spots in the manufacturing process of injection molds.

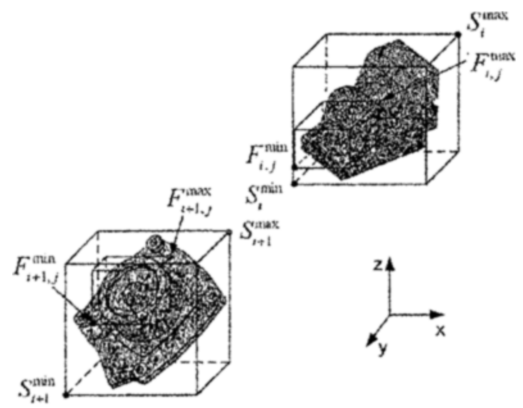

Fig. 6. Maximum and minimum points of BVs.

4.3 Interference between solids

The maximum and minimum value of vertexes in a solid is computed by the vertex information. BV of a solid is constructed through the maximum and minimum vertexes. Interference between BVs of the solids is recognized by the following interference condition [14].

$$if(S_i^{min} > S_{i+1}^{max} \ or \ S_{i+1}^{min} > S_i^{max}) \quad (1)$$

If the condition, Eq. (1) is satisfied, BVs of two solids do not intersect. In the other case, BVs of the two solids have interference.

4.4 Interference between faces

Interference examination between faces uses BVs of faces. If Eq. (2) is satisfied in each axis, BVs of two faces do not have interference. In the other case, the faces intersect each other.

$$if(F_{i,j}^{min} > F_{i+1,j}^{max} \ or \ F_{i+1,j}^{min} > F_{i,j}^{max}) \quad (2)$$

4.5 Interference between triangle meshes

The interference verification between triangle meshes consists of three steps. First, compute a plane equation of a triangle. Secondly, calculate the intersection point between the plane and an edge of the other triangle. Thirdly, inspect interference between the two triangles.

In Fig. 7, a, b and c are three points of a triangle element, and p, q and r are three points of the other triangle element. Vectors V_a and V_b are computed from the three points a, b and c of the first triangle element. Using the cross product of V_a and V_b, vector components of the result are obtained as C_x, C_y and C_z. The plane equation of the triangle abc is computed at the plane point R as

$$D=-(C_x x_b+C_y y_b+C_z z_b) \tag{3}$$

Using the parametric representation for $0 \le t \le 1$, an intersection point f between line segment pq of the second triangle and the plane D is obtained from the following equations [15]:

$$p(x_p,y_p,z_p)t=q(x_q,y_q,z_q)(1-t) \tag{4}$$

$$t=-\frac{((C_x x_q)+(C_y y_q)+(C_z z_q)+D)}{(C_x(x_p-x_q)+C_y(y_p-y_q)+C_z(z_p-z_q))} \tag{5}$$

$$f(x,y,z)=p(x_p,y_p,z_p)t+q(x_q,y_q,z_q)(1-t) \tag{6}$$

where, $p\in x_p,y_p,z_p$, $q\in x_q,y_q,z_q$

Using the three angular points of the first triangle and the intersection point f, three angles between f and opposite two angular points are obtained. If the summation of the three angles is 2π, it is concluded that the two triangles are intersecting each other.

Computation of an angle between the intersection point f and the opposite two angular points is conducted as follows: In Fig. 8 vectors V_1 and V_2 are computed from differences of position vectors b and f, and a and f, respectively. Using the dot product of two vectors V_1 and V_2, the angle θ is given by

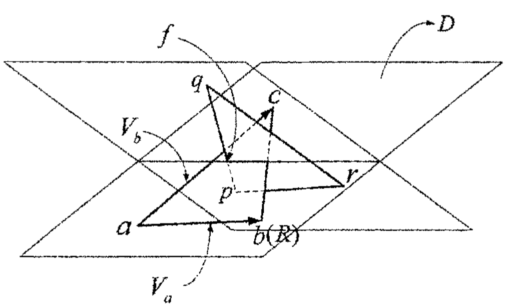

Fig. 7. Triangles consist of planes.

$$\theta=\cos^{-1}\frac{(V_1\bullet V_2)}{(|V_1||V_2|)} \tag{7}$$

Sum of three angles between three angular points of the first triangle and the intersecting point f is given by

$$\theta_{Total}=\theta_1+\theta_2+\theta_3 \tag{8}$$

Where θ_1, θ_2 and θ_3 are three angles between the intersection point f and the opposite two angular points of the first triangle. If θ_{Total} is 2π, the two triangles intersect each other as shown in Fig. 9(a). If θ_{Total} is smaller than 2π, the two triangles do not have interference as shown in Fig. 9(b).

4.6 System performance

In order to verify the efficiency of the proposed algorithm, the total number of calculations to inspect the interference of two solids is compared. The two solids include faces, and numbers of faces of the two solids are denoted by F_l (10) and F_m (10), respectively. Number of pairs of the faces with interference is F_n. Let the numbers of triangular meshes included in each of the faces be T_l (100) and T_m (100), respectively.

In an extreme case, where interference inspection of two solids does not have a hierarchical structure like VRML, the total number of calculations to inspect interference is calculated according to the whole search method [10] of triangular meshes as given by

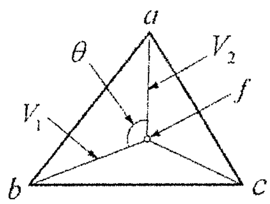

Fig. 8. Angle between V_1 and V_2.

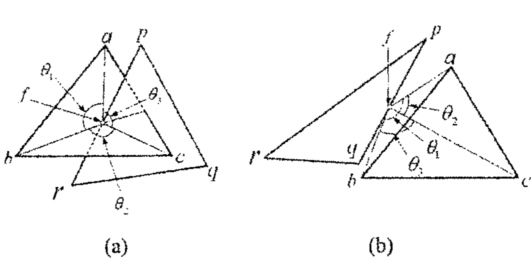

Fig. 9. Cases of intersection between two triangles.

Table 1. Comparisons of interference between the worst and proposed cases.

	A	*B*	*Worst case*	*Proposed case*
Solid	1	1	-	1
Face	10(F_l)	10(F_m)	-	100
Triangle	100(T_l)	100(T_m)	1000000	10000
Total	111	111	1000000	10101

Eq. (9). However, the number of calculations of the proposed interference inspection is calculated according to Eq. (10). As Eq. (9) includes many multiplications and Eq. (10) is expressed in smaller amount of multiplication and addition, the proposed method, Eq. (10), is faster and more efficient. In addition, if two solids interfere at only one pair of faces, when F_n is 1, $N_{proposed}$ decreases markedly.

$$N_{worst} = F_l F_m T_l T_m = 10\times10\times100\times100 = 1{,}000{,}000 \tag{9}$$

$$\begin{aligned} N_{proposed} &= F_l F_m + F_n T_l T_m \\ &= 10\times10 + 1\times100\times100 = 10{,}100 \end{aligned} \tag{10}$$

Table 1 shows the comparison of interference between the worst and proposed cases. The fourth column shows the calculation number of the worst case. The fifth column is the calculation number of the proposed case.

5. System architecture

Fig. 10 shows the overall framework of the developed system. The system consists of designer clients, the integrated server and Web clients. In order to register, search, upload and download CAD data, the integrated server exchanges messages with design or Web clients for file transmission. Message exchange is realized by using socket communication.

Principal functions of the developed system are as follows: The Web-server notifies Web pages and grants users certifications. In addition, a designer is able to register CAD data constructed on the commercial CAD system through the Webpage. Translation server translates the registered commercial CAD file into the lightweight CAD file [9] for the Web-based verification. InterOp modules and kernels of Spatial Inc. are used for conversion of various commercial CAD files into the lightweight CAD files.

Design clients are able to search and verify not only

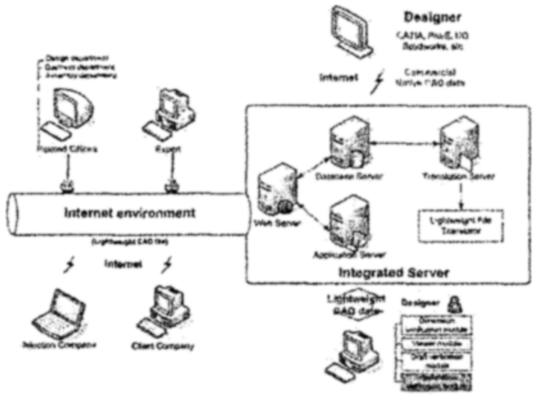

Fig. 10. Framework for the proposed system.

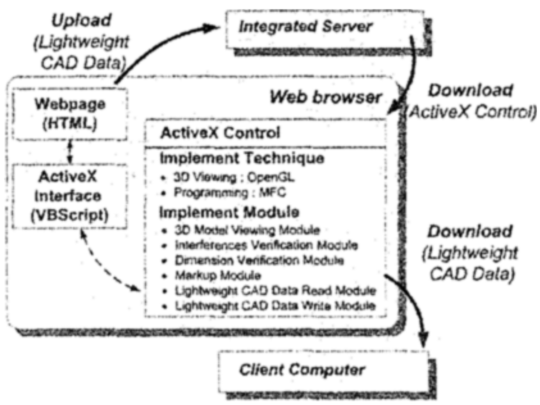

Fig. 11. Structure of the interference verification module.

design results but also the interference of the design results on the Internet. Searched lightweight CAD files corresponding to the commercial CAD data are transmitted from the server to the clients. After interference examination of the design result on the Web, the examination result is saved as a lightweight CAD file. It is transferred to the integrated server. Then, the interference verification result is able to be shared by many clients at the same time.

The interference examination and design verification modules consist of the lightweight CAD file translator, the 3D viewing, interference examination, markup and markup read/write modules as shown in Fig. 11. Each module is realized by using ActiveX technology of Microsoft and programmed by MFC. OpenGL is used for 3D graphics in the ActiveX control. HTML is used to notify ActiveX controls on the web. In addition, VBScript is utilized as interfaces for plug-ins. In this architecture, ActiveX is automatically installed to the client's PC when a client is connected to the server. After the first installation of

ActiveX on the client's computer, versions of the distributed ActiveX controls are checked at every login process. And then the updated ActiveX controls are automatically distributed to the client. In addition, interference examination process is performed on the client's PC, which reduce network dependence and server overload in case of 3D visualization or draft examination of injection molds [3].

6. Case studies

Efficiency of the proposed interference examination system is verified through actual CAD files for injection mold processes. The injection mold is designed on commercial CAD systems, and the CAD data are registered through the Web page of the developed system. The registered CAD files are converted into lightweight CAD files through the translation server [9]. When an injection mold CAD data is searched on the web and is requested to be verified, the viewer is automatically plugged-in on the user's web-browser, and the searched lightweight CAD file is downloaded and visualized.

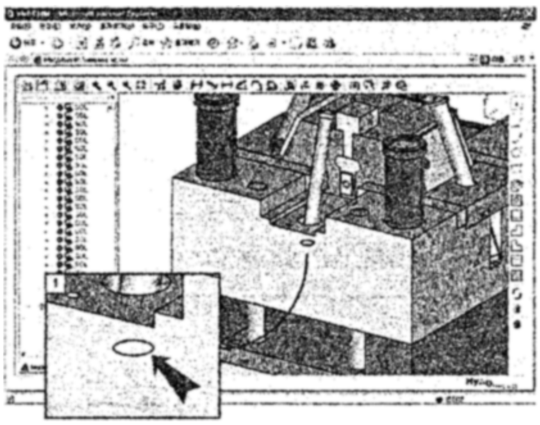

Fig. 12. Interference verification between the angle pin and hall.

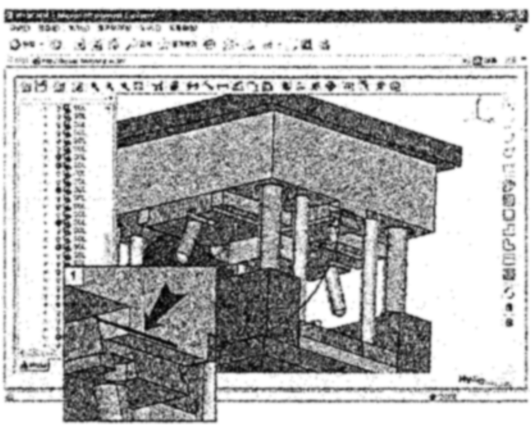

Fig. 13. Interference verification between the slide and lock.

Fig. 12 shows the result of interference examination of the angle pin and the evasion hole. Interference comes from contact between the angle pin and the evasion hole due to the longer length of the pin. The result of the interference verification is displayed on the screen in black as shown in $\boxed{1}$ of Fig. 12. Based on the result, designers are able to modify the length of the angle pin or size of the evasion hole, and apply changes to the design of the injection mold. Fig. 13 shows the interference between the slide and the locking block. In this case, interference occurs at the corner where the slide and the locking block contact together. The reason is that the radius of curvature of the slide is smaller than that of the locking block. The interference region is made distinguishable through color change and is shown in $\boxed{1}$ of Fig. 13.

7. Conclusions

An interference examination algorithm for a Web-based interference verification system applicable to a single-level assembly has been developed for injection mold design processes. Following conclusions have been obtained:

1) To design a collaborative system executable on the Internet, a lightweight CAD file converted from a CAD data is used as a native file of the CAD viewer with interference verification capability.

2) Using the developed interference verification system, commercial CAD systems are not required for assembly examination at the design stage of injection mold design.

3) The developed interference examination algorithm has been implemented in the mold design process, and is able to detect all intersects between complex geometries in a single-level assembly.

4) The effectiveness of the proposed system is verified through a case study.

References

[1] Actify Inc. Available online at :www.actfiy.com (accessed 30-9-2006).

[2] Cimmetry Inc. Available online at: www.cimmetry.com (accessed 30-9-2006).

[3] S. C. Chung and I. H. Song, A Collaborative design and dimension verification system on the internet, *Transactions of NAMRI/SME.* 33 (2005) 493-500.

[4] S. C. Chung, K. H. Yeon and I. H. Song, Web-based draft verification system for injection mold design, *Transactions of the KSME A.* 29 (10) (2005) 1353-1360.

[5] X. G. Ye, J. Y. H. Fuh and K. S. Lee, Automated assembly modelling for plastic injection molds, *Internal Journal of Advanced Manufacturing Technology.* 16 (10) (2000) 739-747.

[6] K. S. Chin and T. N. Wong, Knowledge based evaluation for the conceptual design development of injection molding part, *Engineering Applications of Artificial Intelligence.* 9 (4) (1996) 359-376.

[7] K. H. Shin and K. W. Lee, Design of side cores of injection mold from automatic detection of interference faces, *ASME.* 59 (1993) 27-34.

[8] G. Bergen, Efficient collision detection of complex deformable models using AABB trees, *Journal of Graphics Tools.* 2 (4) (1997) 1-13.

[9] S. C. Chung and I. H. Song, Design of lightweight CAD files with dimensional verification capability for web-based collaboration, *Transactions of the KSME A.* 30 (7) (2006) 488-495.

[10] S. Gottschalk, M. C. Lin and D. Manocha, OBB tree : a hierarchical structure for rapid interference detection, Computer Graphics Proceeding (SIGGRAPH) 30 (1996) 171-180.

[11] P. M. Hubbard, Interactive collision detection, in IEEE Symposium on Research Frontiers in VR, (1993) 24-31.

[12] B. C. Vemuri, Y. Cao and L. Chen, Fast collision detection algorithms with applications to particle flow, *Computer Graphics Forum : Journal of the European Association for Computer Graphics.* 17 (2) (1998) 121-134.

[13] S. Ar, B. Chazelle and A. Tal, Self-customized BSP trees for collision detection, *Computational Geometry: Theory and Applications.* 15 (1) (2000) 91-102.

[14] T. Moller and E. Haines, Real-Time Rendering, A K Peters, Ltd:, (1999).

[15] T. Moller, A fast triangle - triangle intersection test, *Journal of Graphics Tools.* 2 (2) (1997) 25-30.