International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:1, No:7, 2007

Geometry Design Supported by Minimizing and Visualizing Collision in Dynamic Packing

Johan Segeborn, Johan S. Carlson, Robert Bohlin, and Rikard Söderberg

***Abstract*—**This paper presents a method to support dynamic packing in cases when no collision-free path can be found. The method, which is primarily based on path planning and shrinking of geometries, suggests a minimal geometry design change that results in a collision-free assembly path. A supplementing approach to optimize geometry design change with respect to redesign cost is described. Supporting this dynamic packing method, a new method to shrink geometry based on vertex translation, interweaved with retriangulation, is suggested. The shrinking method requires neither tetrahedralization nor calculation of medial axis and it preserves the topology of the geometry, i.e. holes are neither lost nor introduced. The proposed methods are successfully applied on industrial geometries.

***Keywords*—**Dynamic packing, path planning, shrinking.

I. Introduction

AUTOMOTIVE analysis and verification gradually moves from a physical environment to a virtual one. As virtual models of products and processes are developed long before physical prototypes are built, analysis and verification are initiated at earlier stages of the car development projects. Since the costs of solving design related problems heavily increases as project time progresses, early identification and solving of problems are fundamental prerequisites for cost effectiveness.

Dynamic packing analysis deals with solving geometry conflicts, determining a collision-free assembly path for each product sub assembly. Traditionally done by time consuming manual analysis, using various general CAE-tools, dynamic packing is currently often carried out using automatic path planning. Fig. 1 shows a typical application of rigid-body path planning at Volvo Car Corporation.

Manuscript received August 31, 2007. This work was supported by VINNOVA (the Swedish Governmental Agency for Innovation Systems) through the MERA program (Manufacturing Engineering Research Area).

Johan Segeborn is with the department of Manufacturing Methods and Tools at Volvo Car Corporation (e-mail: jsegebor@volvocars.com).

Johan S. Carlson is with the department of Quality engineering and motion planning at Fraunhofer Chalmers research center (e-mail: johan.carlson@fcc.chalmers.se).

Robert Bohlin is with the department of Quality engineering and motion planning at Fraunhofer Chalmers research center (e-mail: robert.bohlin@fcc.chalmers.se).

Rikard Söderberg is with the department of Product and production development at Chalmers University of technology (e-mail: rikard.soderberg@me.chalmers.se).

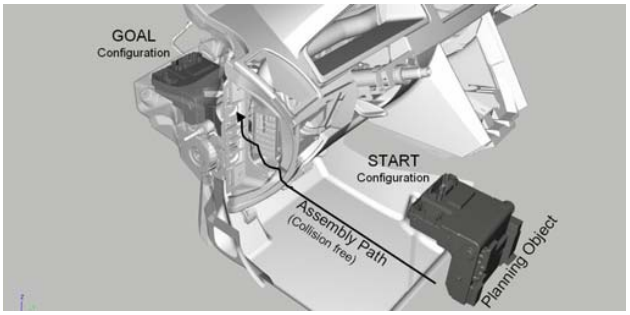

Fig. 1 A typical application of rigid-body path planning at Volvo Car Corporation. Engineer input consists of choosing planning object along with setting its start and goal configurations. A collision-free path, if in existence, is automatically determined

However, at early project stages, product and process design are of an approximate nature and are subject to frequent design changes and restriction volume conflicts. This impedes the use of automatic path planning, as planning algorithms require the existence of a collision-free start and goal configuration as well as the existence a collision-free path. Moreover, experience shows that design changes due to dynamic packing conflicts generally are governed by experience and often even trialed by error, rather than by any design change cost minimization or assembly path optimization.

The use of automatic path planning enables path optimization and substantial reduction of planning times, compared to manual path planning. Thus, there is an apparent productivity gain in extending the applicability of automatic path planning to include also the earlier, in terms of the level of geometrical detail, less defined, automotive project phases.

A. Previous Work

Path planning has been a field of research since the seventies. Many different approaches have been suggested. For overviews see, [1] and [2]. In practice, there are two approaches that are especially widespread and used, due to their conceptual simplicity and implementability. The Probabilistic Roadmap Method, PRM was first described by [3], [4] and the Rapidly-Exploring Random Trees Approach; RRT was first described by [5]. Path planning approaches are however primarily aimed at finding a collision-free solution. The quality of the path, in terms of clearance to obstacles, smoothness and path length can be addressed in a post

processing step. Methods of such are described in [6].

Various approaches to reduce running times of PRM planners have been suggested. For instance, the Lazy PRM planner suggested by [7], minimizes the number of collision checks made during planning. Dilation of the collision-free sub set of a robot's configuration space has been suggested to increase visibility in conjunction with narrow space sampling. Dilation of free space by permitting some penetration is treated by [8]. Penetration depth computation are however increasingly difficult with the complexity of geometries, which limits industrial relevance. Penetration depth computation are treated by [9] and [10]. Dilation of free space by shrinking of geometries are suggested in [11] and [12]. Path planning, aided by shrinking of geometries are previously also treated by [13] and [14].

The shrinking method in [12] is based on medial axis computation. The geometry is shrunk, around its medial axis. However, as computation of medial axis is difficult, because of its instability and algebraic complexity [15], industrial applicability is limited. The shrinking method in [11] utilizes a tetrahedralization of the model. The tetrahedralization is used to ensure that the shrunken model is contained within the periphery of the initial model. Shrinking of the model is carried out by translation of its vertices. The distance that the model can be shrunk depends on its triangulation and tetrahedrilization.

B. Proposed Method

This paper suggests a method to automatically support a minimal geometry design change that results in a collision-free assembly path. The method supports dynamic packing cases when no collision-free path can be found. The overall method is described in more detail in section 2.

The principal idea is to shrink geometries, until a general path planner can find a collision-free path. Path clearance is then increased, using local path optimization techniques similar to [6]. This can be supplemented by a sub part categorization with respect to sub part redesign costs, according to which, a variable clearance is sought during optimization. As this path is run, using the original geometry, all colliding triangles can be noted and presented to the designer, supporting design changes.

A new method to shrink geometry, based on vertex translation interweaved with retriangulation, is presented in section 4.

II. Overall Method Work Flow

In this section, the overall work flow of the proposed dynamic packing method is elaborated, chronologically positioning the methods and algorithms presented in this paper. Each subsection covers a chronological step of the method.

1. Find collision-free start and goal configurations

In rigid-body dynamic packing, the task at hand is to obtain collision-free paths for all subassemblies to their respective assembled configurations. Prior logistic and material handling operations, bringing the subassembly to the assembly cell, generally does not require detailed geometry simulation. Hence, either the planning start or goal configuration can be chosen just outside the minimum bounding box of the obstacle. Let us define this configuration as the start configuration. In the assembled configuration, which thus is defined as the goal configuration, the subassembly often is in collision with, or is locked in by, its fasteners.

Path planning algorithms require the existence of a collision-free start and goal configuration to plan any path. Traditionally, using path planning, a collision-free assembly position or goal configuration has been obtained by hand by excluding geometry in a trial and error process. Two ways to automatically obtain a collision-free goal configuration are described in this paper.

The first method shrinks until clearance is obtained. The algorithm to shrink geometry, presented in section 4, is primarily used in conjunction with the path planning step, but can also be applied for this purpose. This approach is however not feasible in cases when geometry of the planning object is locked in by the geometry of the obstacle. In such cases the planning object could be translated and rotated to an alternative configuration of clearance, and the planning case is divided into two parts. This approach is further described in section 3.

2. Plan Path

If there exists no collision-free path between given start and goal configuration, then to obtain any path, either the collision detection need to permit a certain penetration depth or the planning object M and/or its obstacle geometries need to be modified such that the modified planning object M' is contained within the periphery of the original planning object, i.e. $M' \subseteq M$. To permit penetration, calculations of penetration depth need to be integrated with the collision detection, whereas the modification of geometry can be executed as a pre step to the actual Path Planning. One way to modify the geometry such that $M' \subseteq M$, without making far-reaching design assumptions is to shrink it, since the clearance provided by shrinking is relatively uniformly distributed. Shrinking can be defined as the process where all geometry of an object M, is normally translated away from its surroundings into its interior. The geometry is shrunk to such an extent that path planning can obtain a collision-free path. A new method to shrink geometry is presented in section 4.

3. Optimize clearance along path

Having obtained a collision-free path in the previous step, path clearance is then increased using local path optimization techniques similar to [6], producing a path, along which, the clearance between the planning object and the obstacle are maximized and evenly distributed.

In an industrial case, it is however the minimal design change of M, that results in a collision-free assembly path that is sought. It is therefore primarily the cost of redesign that

should be minimized. Hence, the optimization can be supplemented by a sub part categorization with respect to sub part redesign costs, according to which, clearance to prioritized geometries are added and clearance to less important geometries are subtracted. This categorization is further discussed in section 5.

4. Identify resulting design change

In the previous three steps a collision-free path, optimized with respect to clearance, and valid for a shrunken object M' have been obtained. Thus, as the original model M is run along this path, the minimal design change, in terms of all colliding geometry can be noted and separately presented to the designer.

In section 6, the overall dynamic packing method, outlined in this section, is applied on an industrial geometry.

III. Auto Escape Algorithm

In this section an algorithm to automatically find a collision-free goal configuration, is suggested.

Whereas shrinking can be used in some cases, it is however not feasible in cases when the geometry of the planning object is locked in by the geometry of the obstacle, as illustrated in Fig. 2. Such cases can be addressed by the algorithm presented in this section.

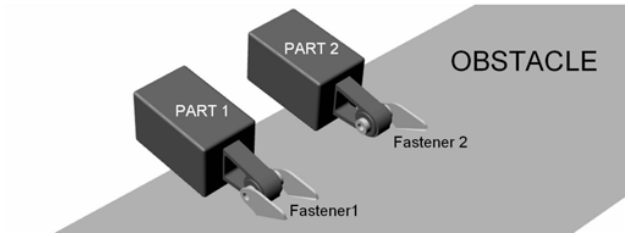

Fig. 2 Shrinking of geometries does in many cases produce a collision-free goal configuration. PART 2 can be shrunk to clearance. PART 1 is however locked in by its fastener, assuming rigid body geometries. The algorithm presented in this section can be used in such cases

The algorithm searches for an alternative configuration of clearance. The available configurations, in a first search, is constituted by translations to all nodes of a Cartesian grid of resolution ρ, contained within a sphere of radius r, centered at a frame f, which is located at the circumcentre of a planning object and, in industrial cases, aligned with the orientation of the overall product coordinate system. R is the circum radius of the planning object. The configuration space is initially limited to translations, to, if possible, make the new, collision-free, goal configuration as similar to the initial goal configuration as possible. If no collision-free configuration is found within the first search, a second search is conducted, where a number of $3*(2*r)/\rho$ primary rotations of an increment ρ/R around f are tested for collision. The increment is set to ρ/R in order to limit maximum displacement to ρ. Both searches are illustrated in Fig. 3.

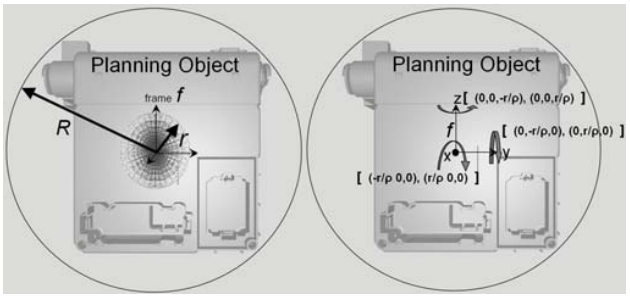

Fig. 3 The Auto escape algorithm searches for an alternative collision-free configuration. This is done in two steps

Upon finding a collision-free goal configuration; there are three configurations of interest, as illustrated in Fig. 4. The initial goal configuration G, the new collision-free goal configuration G' and the start configuration S. The overall task is to find a path between S and G. Whereas path planning between S and G' is conducted using any path planner the feasibility of the path between G' and G, which can not be determined by automatic path planning, is left for the engineer to manually estimate.

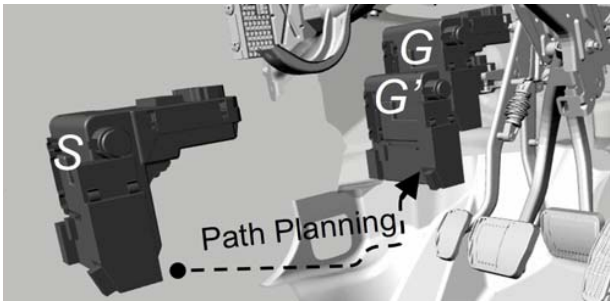

Fig. 4 Whereas path planning is conducted between S and G', the feasibility between G' and G is left for the engineer to manually estimate. The distance between G and G' is heavily exaggerated for graphical purposes

To limit running times, on using this algorithm, the user is presented with an input dialogue, where maximum translation distance r and grid resolution ρ are requested. Upon entering r and ρ, an estimation of the resulting running time and memory usage are instantly calculated and presented to the user, supporting an informed decision regarding r and ρ. A high grid resolution ρ and a large maximum translation distance r increase the precision of the search at the cost of longer running times and higher memory usage.

In most industrial design, part geometries are aligned with the overall product coordinate system. Therefore, as a first guess, searches are conducted in the overall product coordinate system. However if information regarding the master locating system is available, the search can be conducted with further precision. As a colliding start configuration, in an automotive case, as previously mentioned, often is caused by fasteners, the master locating system of the planning object provides with useful information regarding

the kinematical interaction between the planning object and its fasteners. As fasteners and mating surfaces of the obstacle are the geometrical realization of the master locating system, to search within a coordinate system given by the master locating system makes it possible to increase the precision of examined configurations.

IV. Shrinking Algorithm

In this section a new algorithm to shrink geometry is proposed. In section 6 an implementation of the algorithm is applied on a couple of geometries. Compared to previous work this method contributes in the following respects:

- The algorithm requires neither tetrahedralization nor calculation of medial axis.
- The output, i.e. the shrunk geometry does not depend on vertex translation order, as all vertices can be shrunk simultaneously.
- Instead of shrinking with a uniform vertex translation, this algorithm shrinks with a uniform surface translation, which preserves geometric features, which for instance means that the perpendicularity of a corner is preserved as it is shrunk. This is important in an industrial application, as surfaces may function as interfaces to other geometries.
- The topology of the model is preserved, i.e. holes are neither lost nor introduced.
- Dependency on triangulation is reduced as a result of continuous retriangulation. Models can thus be shrunk beyond the distance where triangles otherwise would degenerate, if shrunken in a single step.

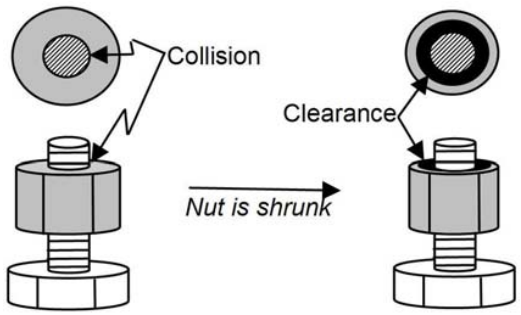

Fig. 5 Shrinking must not be confused with scaling. For instance, while a nut can not be scaled down to obtain clearance from its screw, shrinking provides with the necessary clearance also on inner peripheries

The method presented in this paper takes triangulated, i.e. meshed, geometries as input data. Thus, shrinking is carried out via vertex translation inwards, normally from the periphery, as described in Fig. 6. In formulating an algorithm to shrink geometry there are three things that need to be determined for each vertex: *i)* vertex normal, i.e. translation direction, *ii)* desired translation distance and *iii)* constraints on translation distance.

To ensure that $M' \subseteq M$, the inside of the model M need to be distinguished from its outside. In standardized industrial geometrical product representation, surface normal vectors denotes inside respective outside of the geometries. CAD models are however manually designed, why compliance to modeling standards can not be absolute. If a surface normal points in the wrong direction, then the model will locally grow upon shrinking, possibly aggravating any collision instead of providing with clearance. Moreover, the sub CAD model structure of any geometry is usually lost in the triangulation process. Therefore any triangulated input data is initially to be considered a triangle soup.

Reference [16] suggests methods to classify the triangle soup into either surfaces or solids, and for the latter also into inside and outside. Assuming that such a classification is fully comprehensive, either by the application of such methods or by initial CAD model design, then a triangulated model M can be shrunk.

A. The Proposed Shrinking Algorithm

The model M is shrunk by translation of its vertices. If the surface shall be translated a distance of ε normally inwards into the model, then the resulting translation of the vertices, preserving geometric features, need to be determined, i.e. the vertex normals are sought. We calculate a vertex normal based on the normals of adjacent triangles and edges. A triangle is given a surface normal if its direction can be determined, i.e. if it has been classified as part of a solid. An edge is given a normal in the plane of its triangle, if and only if it is located at a mesh boundary. Let the vertex normal direction be given by the vector c-p between the specific vertex p and the circumcentre, c, of the smallest sphere, which circumscribes all normals of adjacent triangles and edges of p. This is the direction that minimizes the maximum distance between the vertex normal and any of the triangle normals and edge normals, n_i, adjacent to the vertex, $\min_c \max_i |n_i - (c - p)|$.

Having determined the direction of the vertex translation, uniformity gives the distance, $\varepsilon/|c-p|$ as illustrated in Fig. 6. The model, M, is thus shrunk by translating each surface vertex p a distance of $-\varepsilon/|c-p|$ along the $(c\text{-}p)$ vector, Hence, the new vertex p' can be calculated by

$$p' = p - \varepsilon \frac{(c-p)}{|c-p|^2}. \quad (1)$$

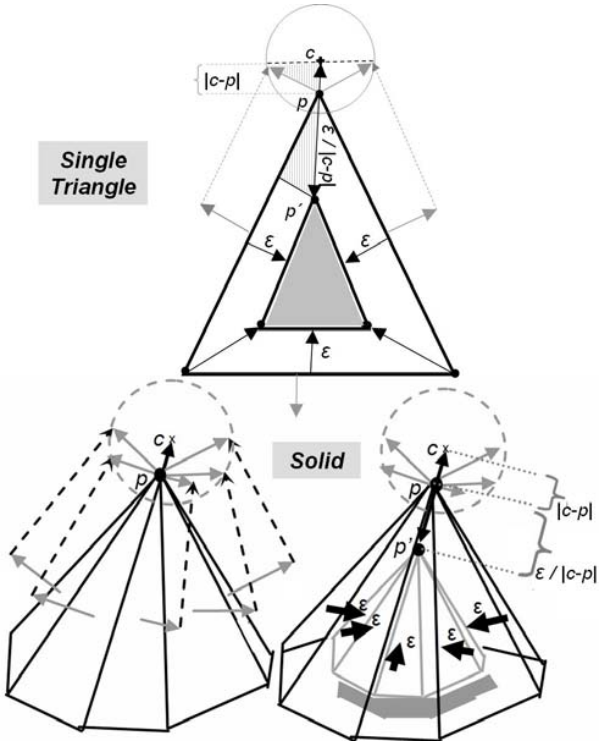

Fig. 6 Translation of vertex p using (1), applied on single triangle and a solid

Using (1), the vertices are translated such that a *uniform surface translation* is obtained. This preserves the geometric features of the overall geometry, but is sensitive to cases when $1/|c-p|$ is large. An alternative basic approach is to instead apply a *uniform vertex translation,* which is obtained by moving all vertices $-\varepsilon$ along their normals. Using this approach, however, the resulting shrunk geometry is dependent on triangulation, as illustrated in Fig. 7.

The advantages of these basic approaches could be combined by moving each vertex - $min(\varepsilon /|c-p|, \varepsilon \sqrt{3})$, which would preserve perpendicular corners but limit the shrinking of sharper corners.

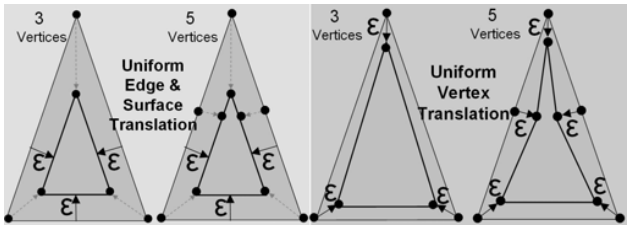

Fig. 7 Uniform surface translation and Uniform vertex translation

Furthermore, ideally, upon shrinking, the number of vertices needed for representation, generally increases as the shrinking process progresses, as shown in Fig. 8. This method does however not add vertices for that reason. Nonetheless, in conjunction with moderate shrinking distances, the difference in accuracy does not justify the heavier representation of the rapidly growing amount of vertices.

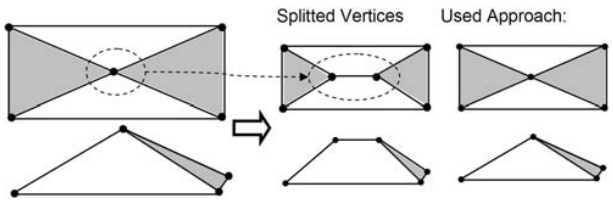

Fig. 8 The number of vertices needed for representation generally increases as the shrinking process progresses. Picture a cheese-slicer that is uniformly applied on the triangles. The approach taken in this paper is however not to split vertices

Having determined the desired new position p', given by (1) below we study the two main constraints on translation distance. These constraints are then addressed, widening the applicability of (1).

B. Triangle Degeneration

The amount of shrinking that can be applied in a single step depends on the triangulation of the model. At some shrinking distance, triangles with non parallel vertex normals, may degenerate, in terms of at least one vertex crossing its opposite edge, resulting in corrupt triangles and folded surfaces as shown in Figs. 9 and 10.

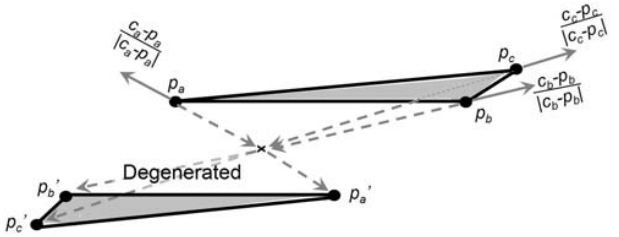

Fig. 9 Triangle degeneration. A triangle that is part of a mesh is presented. The case when all three vertices cross their opposite edges simultaneously is illustrated

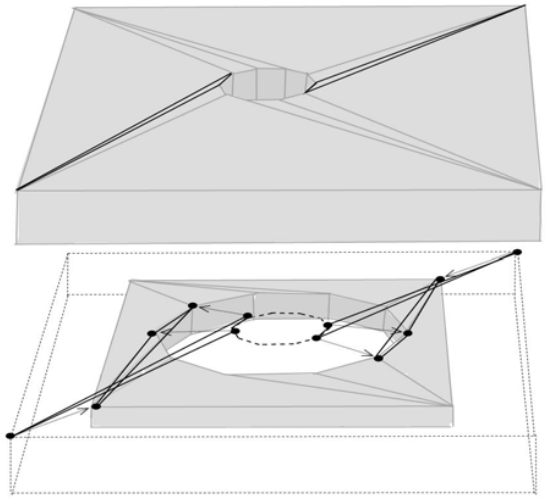

Fig. 10 The figure shows a practical example, where the triangles are narrow with acute angles. Furthermore the vertex normals differ heavily. As a result the triangles degenerate almost instantly upon shrinking, covering a larger portion of the hole as shrinking progresses

A simple worst case analysis of a single triangle *T*, depicted in Fig. 11 shows that it degenerates if its vertices are moved $d_{min}(T)/2$, where $d_{min}(T)$ is the minimum height of *T*.

Thus, if each vertex of M is moved less than half the minimum height of adjacent triangles, then no triangle degenerate. We say that *such a shrinking step is feasible*.

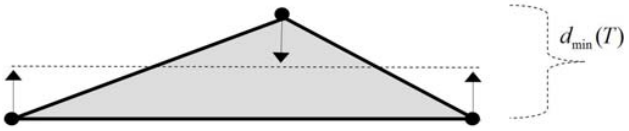

Fig. 11 A simple worst case analysis of a single triangle T shows that it degenerates if its vertices are moved $d_{min}(T)/2$, where $d_{min}(T)$ is the minimum height of T

The approach we take to avoid triangle degeneration is to emulate a continuous shrinking process, in which retriangulation occurs at critical times as needed in order to avoid triangle degeneration. This idea is implemented as follows. We choose a number *N*, say about 100, and divide the desired shrinking distance *E* into *N* steps, each of size $\varepsilon = E/N$. Before shrinking by ε, the current model *M′* is retriangulated using edge flipping, edge removal and mesh simplification techniques, similar to [17]. The model *M′* is retriangulated until a *feasible shrinking step by ε can be made*. It should be noted that the gradual mesh simplification either results in finding a feasible triangulation or ends up with a single tetrahedron.

The limitations triangulation poses on shrinking can thus be addressed. There is however an absolute limit posed by the geometry itself, described in the next sub section.

C. Minimal Thickness

A solid can not be shrunk a further distance than half its minimal thickness, as described in Fig. 12, i.e. shrinking need to be halted at the centre of the object. If vertices are translated beyond the centre of the object, then the topology changes or the peripheries of the solid shift sides, turning the solid inside out. The centre of an object can be defined by its medial axis. However, as computation of medial axis is difficult, because of its instability and algebraic complexity [15], an alternative approach is suggested. We introduce a distance function *f* that measures the minimum distance between the vertices of the shrunken object *M′* and the geometry of the initial object *M*, to prevent from translating surfaces beyond the centre of the model.

When moving a vertex *p* towards its desired new position *p′* given by (1), we make sure *f* is always increasing. If not, *p* is halted and this gives its new position.

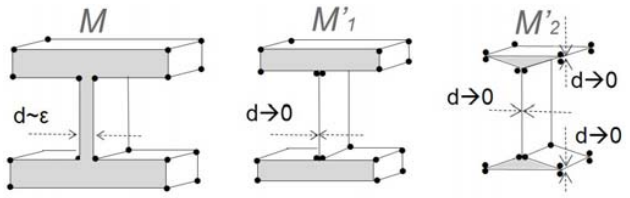

Fig. 12 A model M with a narrow section is shrunk using (1) in steps, utilizing retriangulation. The vertices of the narrow section is halted before the first local maximum of their distance functions f, producing M_1'. Further shrinking, given this triangulation, eventually produces M_2'

D. Shrinking Algorithm Summarized

The Shrinking Algorithm can now be summarized as follows:

1. Introduction of distance function

A distance function *f* is introduced, measuring the minimum distance from a point to the original model *M*, to prevent from translating surfaces beyond the centre of the model.

2. Shrinking a single step

To shrink ε, each vertex p_i is translated, in direction c_i-p_i. The translation distance is $-\varepsilon/|c_i-p_i|$, but if passing a local max of *f*, translation is halted. If each vertex p_i is translated strictly less than half the minimum height of adjacent triangles, then no triangles are degenerated and *we say this step is feasible*.

3. Multiple step shrinking

For a given total shrinking distance *E* and discretization *N*, divide *E* into *N* steps of shrinkage $\varepsilon = E/N$. To ensure that no degeneration arises; in each iteration, perform mesh simplification until "*Shrinking a single step"* is feasible.

V. Redesign Cost Weight Assignment

In this section an approach to minimize redesign cost is suggested. As shrinking provides with a relatively uniform clearance, the original object *M* will, run along a path which is barely collision-free for the shrunken object *M′*, also display a relatively uniform collision pattern.

In an industrial case, it is however the minimal design change of *M*, that results in a collision-free assembly path, that is sought. It is therefore primarily the cost of redesign that should be minimized.

There is hence an interest in controlling which parts of the overall model that will be subject for this redesign. This can be done, by seeking a variable clearance based on redesign cost during path optimization which repels prioritized geometry from potential collisions and crashes the ad hoc solutions instead. Reference [18] describes how path optimization can be carried out with a variable clearance, in that case based on geometrical variation. An alternative approach is to apply more shrinking the less prioritized the specific part geometries are. Both approaches require some

categorization of the geometrical elements making up the overall product.

As the surfaces of industrial CAD models often are categorized with respect to significance regarding industrial design, function and part interfaces, each set of surfaces can be assigned a weight representing the cost to redesign the geometry contained within the specific set. Existing industrial categorization is a good basis for priorities that could be extended, either as further attributes in the CAD files or as separate entries in the overall PDM system.

VI. Implementation

In this section methods and algorithms presented in this paper are applied on geometries.

A. Shrinking Algorithm

An implementation of the shrinking algorithm presented in section 4 has been applied on two geometries. The first geometry is a densely triangulated cube with rounded edges and a throughout hole. The second geometry is a design revision of an industrial model from Volvo Car Corporation. This part, called the SIPS unit, which is short for Side Impact Protection System, is assembled in the rear doors.

The cube is shrunk in a single step using (1) whereupon the narrow triangles along the rounded edges and the hole degenerates producing the characteristic geometry displayed in Fig. 13b. If, however, the shrinking, still using (1), is applied in steps utilizing retriangulation as described in section 4, the adequate result displayed in Fig. 13c is obtained.

The SIPS unit is also shrunk using (1). Fig. 14 depicts, analogue to Fig. 13, shrinking in a single step and shrinking divided in several steps.

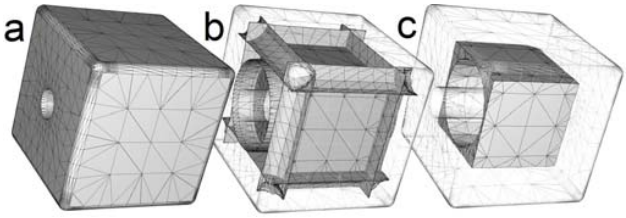

Fig. 13 a) A densely triangulated cube b) The cube, shrunk in a single step. c) The cube, shrunk in steps, utilizing retriangulation

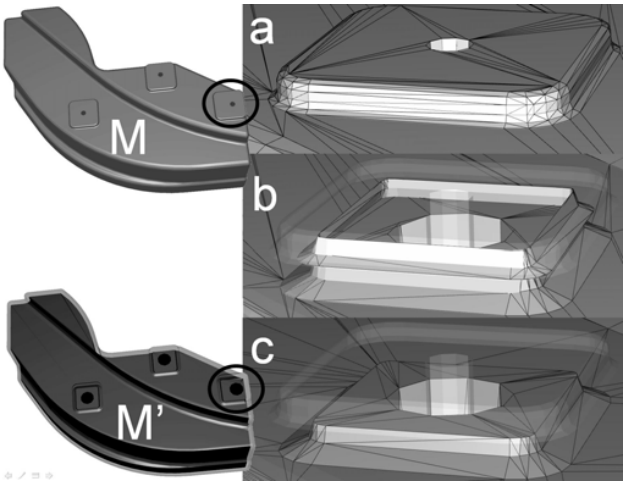

Fig. 14 An industrial model, in the picture denoted *M*, from Volvo Car Corporation is shrunk, using the proposed method. In *a),* a close up on the original model is depicted. In *b)* shrinking is applied in a single step with triangle degeneration as a result. In c) shrinking is applied in steps utilizing retriangulation, thus avoiding triangle degeneration, producing *M′*

B. Overall Dynamic Packing Method

In section 2, the overall work flow of the proposed dynamic packing method was outlined. Below it will be applied on the SIPS unit, introduced in the previous sub section. First we need to make sure that there is a collision-free start and goal configuration. Methods for that have been suggested in section 3 and 4. However in this particular case, the assembly position of the SIPS unit, which coincides with the planning goal position, is collision-free to begin with. Therefore we move on to the actual path planning. No collision-free path is found for this revision of the SIPS unit. To increase free space, the SIPS unit is shrunk, which is illustrated in Fig. 14. Path planning, now instead carried out for the shrunken SIPS unit, in the figures denoted M', returns a collision-free path, as depicted in Fig. 15. The clearance of this path is then increased, using local path optimization techniques similar to [6]. We now have a collision-free path, valid for a shrunken SIPS unit, with a clearance that is maximized and evenly distributed.

As the original SIPS unit is run along this path, the resulting collisions along the path are noted by accumulation of all intersections between SIPS unit- and obstacle surfaces along the path. Fig. 16 presents the SIPS unit along with the accumulated collisions along the path. The accumulated geometry, i.e. the black areas in figure 16, constitutes a minimized design change that results in a collision-free assembly path.

The resulting collisions are evenly distributed, in terms of penetration depth. Let us however assume that we, for any reason, would find the collision, in Fig. 16 denoted number 1, especially undesirable, as it is located on a mating surface. Seeking a variable clearance during path optimization, as suggested in section 5, the clearance to this feature could be

increased at the cost of the other features of the model, thus avoiding this particular collision.

Fig. 15 Path Planning is carried out for the shrunk SIPS unit

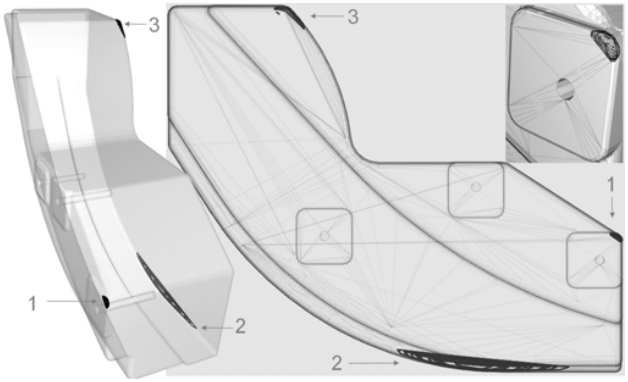

Fig. 16 The figure shows the SIPS unit with the accumulated collisions along the path. The accumulated geometry, i.e. the black areas in the figure, constitutes a minimized design change that results in a collision-free assembly path

REFERENCES

[1] H. Choset, K. *et al*., "Principles of Robot Motion: Theory, Algorithms, and Implementations," MIT Press, Boston, 2005.
[2] S. M. LaValle, "Planning Algorithms," Cambridge University Press, 2006.
[3] L. E. Kavraki and J.-C. Latombe, "Randomized Preprocessing of Configuration Space for Fast Path Planning," In *Proc. the International Conference on Robotics and Automation,* IEEE Press, San Diego, CA, 1994.
[4] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, 1996.
[5] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," In *Proc. IEEE International Conference on Robotics and Automation*, 1999.
[6] R. Geraerts and M. H. Overmars, "Creating High-quality Paths for Motion Planning," The International Journal of Robotics Research, 2007.
[7] R. Bohlin, L. E. Kavraki, "Path Planning using Lazy PRM," In *Proc. the 2000 IEEE lntemational Conference on Robotics & Automation San Francisco,* 2000.
[8] D. Hsu, L. Kavraki, J-C. Latombe, R. Motwani, and S. Sorkin. "On finding narrow passages with probabilistic roadmap planners," In Proc. Int. Workshop on Algorithmic Foundations of Robotics, 1998.
[9] L. Zhang, Y. J. Kim, G. Varadhan, D. Manocha, "Generalized Penetration Depth Computation," Computer-Aided Design, Volume 39, Issue 8, 2007.
[10] S. Redon and M. C. Lin, "A Fast Method for Local Penetration Depth Computation," In Journal of Graphics Tools, Vol. 11, No. 2: 37-50, 2006.
[11] D. Hsu, G. Sánchez-Ante, Ho-lun Cheng and J.-C. Latombe, "Multi-Level Free-Space Dilation for Sampling Narrow Passages in PRM Planning," In *Proc. IEEE International Conference on Robotics and Automation*, 2006.
[12] M. Saha and J.-C. Latombe, "Finding narrow passages with probabilistic roadmaps: The small step retraction method," In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2005.
[13] B. Baginski, "Local motion planning for manipulators based on shrinking and growing geometry models," In *Proc. IEEE Int. Conf. on Robotics & Automation*, 1996.
[14] P. Ferbach and J. Barraquand, "A method of Progressive Constraints for Manipulation Planning," IEEE, Tr. Rob. Abd Aut., 13(4):473-485, 1997.
[15] M. Foskey, M. Lin, and D. Manocha, "Efficient computation of a simplified medial axis," In ACM Symp. on Solid Modeling & Applications, 2003.
[16] J. Havner and O. Karlsson, "Automatic correction of surface normals on 3D models," M.S. thesis, Dept. Mathematical Sciences, Chalmers University of Technology, Gothenburg, Sweden, 2005.
[17] M. Garland, P. S. Heckbert, "Surface Simplification Using Quadric Error Metrics," *IEEE Visualization 98*, 1998.
[18] J. S. Carlson, R. Söderberg, R. Bohlin, L. Lindkvist and T. Herrmansson, "Non-nominal path planning of assembly processes," In *Proc. the 2005 ASME International Mechanical Engineering Congress and exposition November 5-11, Orlando, FL*, 2005.