Proceedings of the 5th IEEE
International Symposium on Assembly and Task Planning
Besançon, France • July 10-11, 2003

Efficient Assembly Sequence Planning Using Stereographical Projections of C-Space Obstacles

Ulrike Thomas, Mark Barrenscheen and Friedrich M. Wahl

Institute for Robotics and Process Control
Technical University of Braunschweig
Braunschweig, Germany, 38106

Abstract

This paper presents a new assembly planner, which is based on stereographical projections of C-Space obstacles. The assembly planner applies the well-known assembly-by-disassembly *strategy. The complete geometric feasible AND/OR-graph is built. For testing separability a new very fast approach has been implemented with the advantage of direct hardware support. The entire AND/OR-graph is evaluated with respect to two criteria: separability and parallelism. We have generated example assembly plans for realistic industrial assemblies, which are modelled with more than two thousand triangles. Despite the fact of high model complexity, we can achieve reasonable computation times*

1 Introduction

Assembly planning for industrial products is very important for many purposes. On one hand it provides feedback to the designer, indicating him how well his product is suited to be assembled by robots. On the other hand, assembly planning is the basis for generating robot programs automatically. By this, it helps to reduce costs in the manufacturing process. Requirements for the design of new types of robots also can be attained from assembly planning, e.g. number of DOF, ranges of joint angles, etc.

In the past decades, many assembly planning systems have been developed. Early assembly planning systems used symbolic spatial relations and generated robot programs for position control, e.g. R-APT [1]. AUTOPASS and LAMA provided a high level robot programming language [2, 3]. All these assembly planning systems had and still have to cope with the burden of the inherent combinatorial complexity. Some sophisticated approaches have been presented with the STAAT-System and the Archimdes-System [4, 5, 6]. These systems use *non-directional-blocking-graphs* NDBGs to gain assembly sequences. The separability has been computed based on contact analysis. These systems are able to compute monotone assembly sequences for assemblies consisting of hundreds of parts. For computing mating directions, a discrete approach has been chosen. In [7] a new framework, the motion space approach, is outlined theoretically. In this paper, we present a new assembly planner based on the computation of C-space obstacles and the well-known *assembly-by-disassembly* strategy. We apply stereographical projections to the C-space obstacles in order to reduce the complexity of computing separability; our technique is quite simular to Halperin's motion space approach. The usage of stereographical projections of the union of Minkowski sums also has been mentioned in [8]. A configuration space approach for assembly planning has been presented in [9]: The authors have applied a linear optimization algorithm to compute the separability. Due to this approach the computation times heavily increase with the number of facets used to model the assembly group, but they successfully solve the m-handed assembly sequence problem for small assemblies. In contrast to most assembly

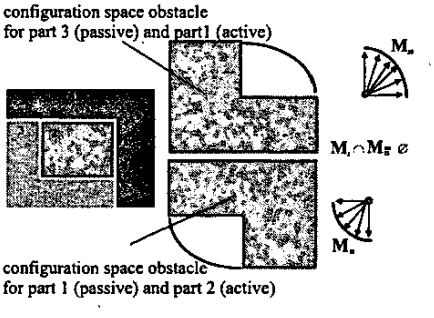

Figure 1: Principle of our separability approach in 2D

planners, e.g. [10], we have applied our approach to realistic industrial products consisting of thousands of triangles. Due to the configuration space approach, our system is able

to deal with geometrical part tolerances. The computational burden of a high number of triangles in the configuration space is solved by stereographical projections. Fig. 1 illustrates our algorithm for the two dimensional case. The implementation within our assembly planner has been realized for 3D $\mathcal{C}$-spaces. The processing examples given in this paper are also of 3D nature.

2 Configuration Space Approach for Separability

The separability of an assembly P into two subassemblies P_i and P_j is possible, if a vector of motion $\vec{m} = (\alpha, \beta, z)^T$ exists, such that applying $\vec{m}$ to all parts $p_i \in P_i$ does not cause collision with parts $p_j \in P$ of the assembly, or vice versa. For testing the separability, we compute configuration space obstacles of all pairs (p_i, p_j) with $p_i \in P$ and $p_j \in P$. With the configuration space approach, described in the following the separability can be computed precisely. In addition, it can deal with geometrical tolerances of parts.

2.1 The Calculation of the Configuration Space Obstacles

The geometry of objects is assumed to be available in some kind of boundary-representation. For accurate separability computation, $\mathcal{C}$-space obstacles should be available in boundary-representations as well, instead of discrete representations like, octrees, bintrees etc. From [11] it is well-known in theory, how to construct configuration space obstacles in $\mathbb{R}^n$. We employ this technique to compute $\mathcal{C}$-space obstacles in 3D. For each object $p_i \in P_1$ and $p_j \in P_2$ we compute pairwise $\mathcal{C}$-space obstacles, hence we declare p_i as active part and p_j as passive part. Assuming the boundary-representations of p_i and p_j are given by all vertices $v_i \in p_i$ and $v_j \in p_j$, and by all facets $f_i \in p_i$ and $f_j \in p_j$ belonging to p_i and p_j respectively. Fig. 2 shows an exploded view of a typical assembly group of a headlight consisting of approximately 200 000 triangles altogether. If we want to apply our assembly planner to such complex assemblies, we need an efficient implementation of the computation of $\mathcal{C}$-space obstacles.

First, we triangulate object surfaces by applying the plane sweep algorithm suggested in [12]. Assuming, objects p_i and p_j consist of f surfaces each containing k vertices on average and r concave vertices, triangulation of all f surfaces takes $O(f(k + r\log(r)))$ time. As it is known, for generating the Minkowski difference $p_i \ominus p_j = \{v_j - v_i \mid v_i \in p_i, v_j \in p_j\}$, all non-convex polyhedrons in p_i and p_j need to be decomposed into convex polyhedrons.

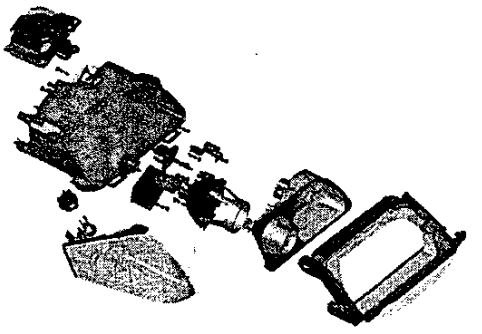

Figure 2: The headlight with indicator light assembly

The complete surface decomposition takes $O(n + r\log(r))$ time, where n denotes the number of edges and r represents the number of non-convex edges. We know, that an optimal and complete decomposition into a minimal number of convex polyhedrons is NP-hard. As explained in [13], we start decomposition of surfaces by creating one convex patch and add adjacent triangles as long as (a) adding the next triangle results in non-convex edges, or (b) the front face of the next triangle is not visible by any front face of the currently constructed patch, or (c) a generated convex hull triangle does not intersect any triangle of the original polyhedron nor does it intersect free space either. We have improved the approach implemented in [14], i.e. we have accelerated the intersection test by using a voxel space of triangles instead of intersecting each generated convex hull triangle with all triangles of the original polyhedron. As decomposition result, we obtain two sets D_i and D_j of convex parts. Computing the Minkowski point set of all convex parts, takes $O(v_i v_j)$ time, where v_i represent the vertices of D_i and v_j those of D_j respectively. In order to obtain a complete boundary-representation of $\mathcal{C}$-space obstacles, we have to generate a convex hull for each Minkowski point set. For determining convex hulls, we use the algorithm proposed in [15], which takes time $O(n\log(r))$, where n denotes the number of vertices and r represents the number of vertices lying on the boundary of the convex hull. All together, the complete construction of $\mathcal{C}$-space obstacles in 3D takes $O(v_i v_j \log(r))$ time, where r denotes the vertices representing the boundary of the $\mathcal{C}$-space obstacles. Fig. 3 shows the result of our algorithm after each processing step, when applied to two complex parts of the headlight assembly. For the lamp, which has to be separated from the optical unit, the complete time of constructing the corresponding $\mathcal{C}$-space obstacle takes 4.86 s. The two objects consist of 4472 triangles and the constructed $\mathcal{C}$-space obstacle contains 800 032 triangles. In

# Triang.	Triang.	Decomp.	Minkowski	C.-Hull	Ray-Casting	Total	# Triang. $\mathcal{C}$-Space O.
1 500	< 0.01 s	0.07 s	0.23 s	0.42 s	0.46 s	1.19 s	22 376
4 472	< 0.01 s	0.09 s	0.85 s	3.91 s	8.54 s	13.4 s	800 032
17 386	0.02 s	0.31 s	18.95 s	37.48 s	88.52 s	145.28 s	7 184 836

Figure 4: Measured computation times on an SGI Octane 2 for the $\mathcal{C}$-space obstacles and the stereographical projection generated by ray-casting

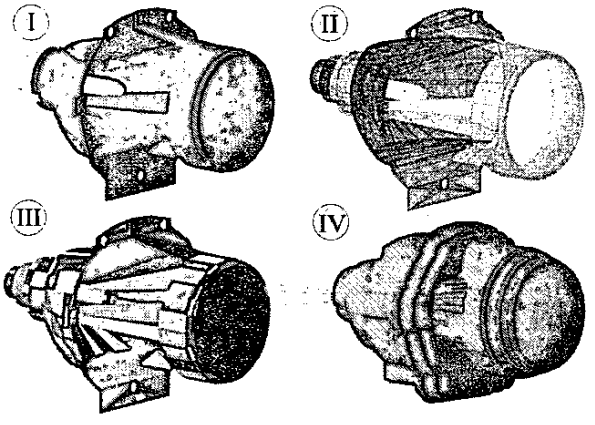

Figure 3: I: High beam lamp as active part(light) and the optical unit as passive part (dark); II: Triangulated surfaces; III: Result of convex decomposition; IV: $\mathcal{C}$-space obstacle with calculated mating directions

Fig. 4 computation times of three examples are given as well as the number of triangles representing the $\mathcal{C}$-space obstacles in comparison to the number of triangles representing the original parts.

One drawback of the configuration space approach is the dramatically increasing number of triangles in the configuration space. This motivated us to develop a new approach capable to deal with such complex $\mathcal{C}$-space objects and at the same time yielding the possibility to handle parts with geometrical tolerances.

2.2 Generating and Evaluating Separability Directions

For two objects p_i and p_j we have shown how to construct the $\mathcal{C}$-space obstacle. In order to find suitable vectors of motion $\vec{m}_j = (\alpha, \beta, d)^T$, which represent possible separation directions (α, β) of the active part p_i from the passive part p_j with translational distance d, we use the special unitary group $SU(2)$. To establish the group for separability, we use the Moebius transformation of the Riemanns sphere. The unit sphere $S^2 = \{(\xi, \eta, \varsigma) \in I\!R^3 \mid \xi^2 + \eta^2 + \varsigma^2 = 1\}$ is mapped onto the extended complex plane $\boldsymbol{C} \cup \{\infty\}$ by stereographical projection. The point $z = x + iy$ in the complex plane corresponds to the point $\vec{P} = (\xi, \eta, \zeta)^T$ on the sphere with coordinates

$$\xi = \frac{2x}{\mid z^2 \mid +1} \quad \eta = \frac{2y}{\mid z^2 \mid +1} \quad \zeta = \frac{\mid z^2 \mid -1}{\mid z^2 \mid +1} \qquad (1)$$

The conformal map $SU(2)$ of the Riemanns sphere onto itself is described by the Moebius transformation. The Moebius transformation is a group homomorphism. In particular, isometric transformations of the sphere, i.e. rotations, are conformal. This property provides the advantage to investigate and evaluate, which direction of motion is most suitable in 2D (α, β)-space. The goal is, to obtain motion directions with maximum distance to obstacle regions in $\mathcal{C}$-space. The projection of a $\mathcal{C}$-space object results in the image $SP(2)$. In order to avoid singularities, we use two stereographical projections, one applied to the north half of the sphere denoted by $SP_n(2)$ and the other applied to the south half of the sphere represented by $SP_s(2)$ respectively (Fig. 5).

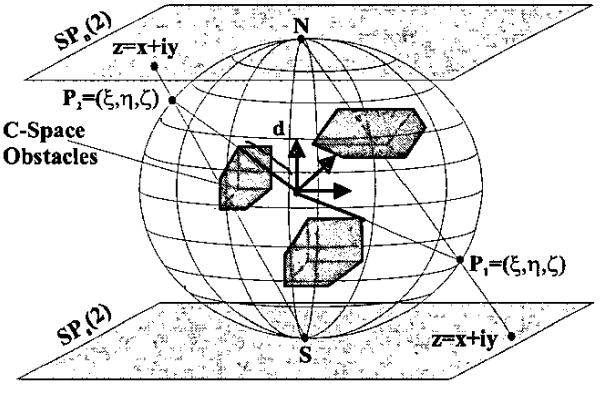

Figure 5: Stereographical projection of $\mathcal{C}$- space obstacles; projectioned the north half of the sphere is mapped to one plane $SP_n(2)$ and the south half to plane $SP_s(2)$.

We extend these images by a third dimension, so that each entry $x + iy$ in image $SP_n(3)$ represents a vector of motion $\vec{m}$ with length d, which can be obtained with respect to equation (1) by $\vec{m} = d \cdot (\xi, \eta, \zeta)^T$ To obtain the corresponding vectors $\vec{m}$ in the southern half of the sphere,

we apply the same equation to entries $x + iy$ of the image $SP_s(3)$, but we need a further multiplication of $\vec{m}$ by $(0, 0, -1)^T$. Fig. 6 shows a gray-scaled image of the stereographical projection for the example shown on the left side of this figure. The gray-value at point (x, y) represents the possible maximal distance d of motion in direction of the corresponding vector $\vec{m} = (\alpha, \beta, d)$ without collision. These two images represent all possible mating directions and corresponding distances for motion, where no collision with any other part occurs. Of course, the most suitable di-

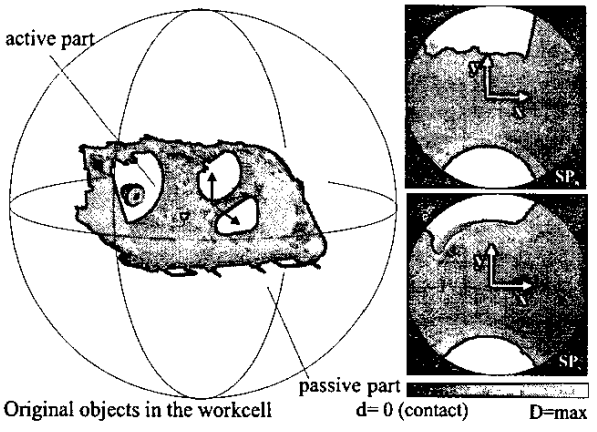

Figure 6: Images of the stereographical projections of the C-Space obstacles for the high beam lamp (active) and the headlight case (passive).The gray-values give information about distances between the objects.

rections for motion are those with longest distances. Thus, we search for all vectors $\vec{m}$ with

$$\vec{m}_z = \max\{SP_n(3), SP_s(3)\} \qquad (2)$$

If $\vec{m}$ is not unique we obtain various areas in the projected planes. We construct the binary mappings of the extended stereographical projected fields $SP_{n/s}(3)$ to $BSP_{n/s}(2)$ by

$$BSP_{n/s}(2)|_{x,y} = \begin{cases} 1\ if\ SP_{n/s}(3)|_{x,y} = \max\{SP_{n/s}(3)\} \\ 0\ else \end{cases} \qquad (3)$$

The most suitable vector of motion $\vec{m}$ is that, which yields the largest distance between the active and passive parts, while mating or separating parts. Thus, we may generate a Voronoi diagram of the binary mapped stereographical projections. Another possibility is to generate potential fields for all binary mapped stereographical projections.

By this, we determine the most suitable direction $\vec{m}$ for a separation or mating process by searching all global maxima of the potential fields. The corresponding vectors are given by applying equation (1). With this approach, we are able to generate and evaluate the most suitable vectors for motion.

For detecting global separability, it suffices to find one vector $\vec{m}$ with length d. The part p_i is geometrically separable from part p_j, if an entry (x, y) with $SP_{n/s}|_{x,y}(3) = R$ exists in the projected field, where R denotes the radius of the sphere. This is the case, if an entry (x, y) exists in the binary mapped stereographical projection with $BSP_{n/s}|_{x,y}(2) = 1$.

2.3 Handling Parts with Geometrical Tolerances

A long term aim, known to be an open problem, is the consideration of production tolerances in the assembly planning process [7]. For solving this problem, the C-space approach is very attractive. It is possible to augment each part description with tolerances of the shapes analytically. If no tolerances for the objects are modelled, we apply the following procedure, so that the computation of geometrical feasibility also becomes possible: After decomposing all non-convex polyhedrons into a set of convex polyhedrons, we shrink each convex subpolyhedron, which belongs to the active part, about a tolerance factor μ, which might be within the production tolerance. We then repeat the previously explained steps to compute the C-Space obstacles. The resulting stereographical projections contain higher values due to the shrink procedure. To avoid incorrect computation, we apply a filter to the stereographical projections subtracting each entry according to the given tolerance factor μ. The global separation vectors can be obtained by applying equation (1) in the same way. With this approach a far more accurate computation of motion vectors $\vec{m}$ for nearly every kind of object is possible, even in the case of parts with geometrical tolerances.

3 Generating and Evaluating Assembly Sequences

For sequence generation, we apply the *assembly-by-disassembly* strategy and store all geometrical feasible assembly sequences in an AND/OR-graph introduced in [16]. For the specification of assemblies, we use symbolic-spatial relations in a CAD-environment [1]. Our new assembly planner is based on the stereographical projections of C-space obstacles as described before. The resulting AND/OR-graph can be evaluated considering certain criteria like parallelism of assembly, necessary reorientations, geometric feasibility etc. Fig. 7 gives an overview of the entire system. After the designer has specified the assembly product consisting of parts $P = \{p_1, \ldots, p_n\}$ the system computes in a preprocessing step

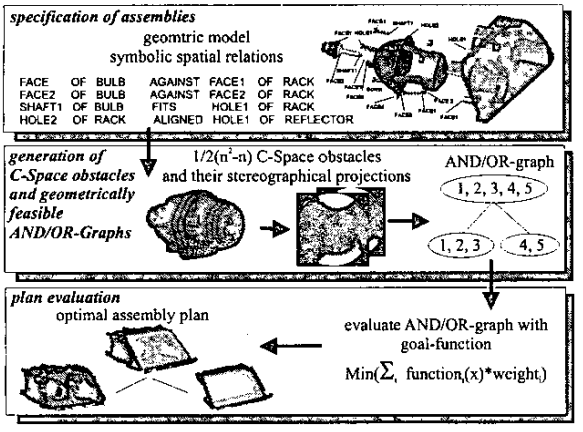

Figure 7: An overview of the assembly planner

the stereographical projections for the $\frac{1}{2}(n^2 - n)$ pairs of parts of the assembly. Thereby, for each pair (p_i, p_j) with p_i and $p_j \in P$ the projections $SP_{n/s}(3)$ are stored. The images for the inverse pairs are obtained by reflection according to the point $(0, 0)$, in the x, y-plane

$$SP_{n/s}(3)|_{x,y}(p_i, p_j) = SP_{s/n}(3)|_{-x,-y}(p_j, p_i) \quad (4)$$

This takes $O(r^2)$ time, where r represents the size in x, y-direction of the images. This information is stored; the computation of the geometrical feasible AND/OR-graph is based on these images only.

3.1 Generating an Accurate AND/OR-Graph for Geometric Feasibility

It is well-known, that the complete computation of the AND/OR-graph is NP-hard. Even for partitioning an assembly group $P = \{p_i, \ldots, p_n\}$ into two different groups P_i, P_j with $P_i \cup P_j = P$ and $P_i \cap P_j = \emptyset$, $\frac{1}{2}\sum_{i=1}^{n-1}\binom{n}{i} = 2^{n-1} - 1$ possibilities exist. In order to reduce the complexity, we consider two-handed assemblies only. We make the restriction, that each subassembly must not be disconnected. This implicates, that only two subassemblies are mated within one operation. If we consider weakly connected assemblies, the number of assembly sequences can be reduced to $\sum_{i=1}^{n-1} i(n-i) = \frac{(n+1)n(n-1)}{6}$; for monotone assemblies it can be reduced to $\sum_{i=1}^{n-1}(n-i) = \frac{n(n-1)}{2}$. If the assembly group is heavily connected the number of sequences grows to $\sum_{i=2}^{n}\binom{n}{i}\cdot(2^{i-1}-1) = \frac{3^n+1}{2} - 2^n$.

The algorithms for generating a geometrical feasible AND/OR-graph works as follows: First we build up the connectivity graph $G = (E, N)$ for the entire assembly. Each part of the assembly is represented by a node. For each edge we look up the stereographical projections. If we can find an entry smaller than a certain threshold value in one of the corresponding fields $SP_{s/n}(3)$ of (p_i, p_j), we insert an edge into the graph. For this procedure, we have to consider all n^2 combinations, hence the computation is of order $O(n^2)$. Using the connectivity graph, we generate all possible partitions of G into two-connected subassemblies. For each part p_i , we generate all subassemblies containing m parts $P_1 = \{p_1, \ldots, p_i, \ldots, p_m\}$ by depth first search. If the rest of the assembly $P_2 = \{p_{m+1}, \ldots, p_n\}$ is connected, we test the geometrical feasibility by an AND-conjunction of all binary mapped stereographical projections $BSP_{n/s}(2)$. If we find an entry in the resulting image equal to 1, the subassemblies are geometrical feasible.

$$\{\exists\, x, y \in \mathbb{R} | \bigcap_{i=1}^{m} \bigcap_{j=m+1}^{n} BSP_{n/s}|_{x,y}(2)(p_i, p_j) = 1\} \quad (5)$$

Fig. 8 illustrates the algorithm for a single partitioning problem. With our approach, the complexity of partitioning is $O(n^2k^2)$, where n is the number of parts and k is the resolution of the stereographical projection. The advantage of this approach is obvious; it can be computed very fast e.g. by graphics hardware support.

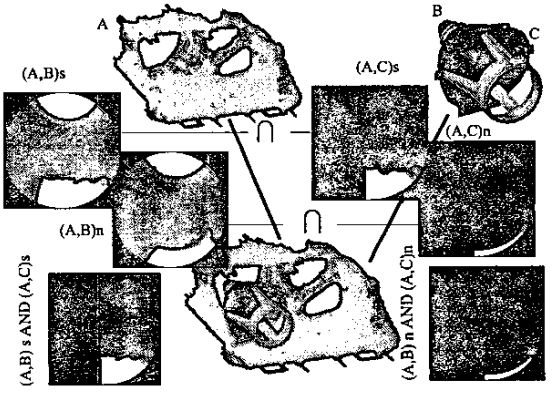

Figure 8: The separability test for three parts of the headlight assembly: Just the AND-conjunction of the images need to be computed

For each generated subassembly we recursively call the separation procedure, until either the size of the subassembly is smaller than one, or until no more partitions can be found. By this, we obtain the complete geometrical feasible AND/OR-graph. As mentioned above, the complexity of building the complete AND/OR-graph depends on the connectivity of the assembly group. If we do not make restrictions concerning the connectivity, the complexity is not polynomial. Nevertheless, due to our fast partitioning algorithm, we get acceptable running times.

3.2 Evaluating the AND/OR-Graph

For the evaluation of subassemblies, we have applied an accurate search algorithm. So far we investigate the geometrical feasible AND/OR-graph considering two criteria: parallelism $\kappa_1 = 1 - \frac{\min(|P'|,|P''|)-1}{\max(|P'|,|P''|)}$ and separability $\kappa_2 = 1 - \frac{4\pi Area}{Perimeter^2}$, with $\kappa_i \in [0 \ldots 1]$. The first criterion can be computed very fast. The second criterion is computed by counting all values in the binary mapped projected images, which are equal to 1. This means the size of the area for all possible translational movements. The perimeter is obtained by counting all boundary points in the projected images. By this, we get an evaluation criterion for the compactness of all possible motion directions.

For the evaluation of an assembly sequence, we start with the root of the AND/OR-graph and select the cheapest hyper-arc. The evaluation of hyper-arcs is done by weighting the above mentioned evaluation criteria and by sorting them with respect to the computed costs $C = \omega_1\kappa_1 + \omega_2\kappa_2$. Of course, we evaluate the hyper-arcs of both subassemblies by selecting again the cheapest hyper-arc. If we can not find a suitable hyper-arc, we go back one level and search for a new hyper-arc meeting the criteria. By this, we are able to generate good assembly plans in reasonable time. Of course, in the worst case the computation time is not polynomial, because we have to consider all nodes of the complete AND/OR-graph.

4 Assembly Planning Results

All computations have been done on an *SGI Octane2* with 3.5 GB memory and a *R*12000 processor. The used compiler is the *SGI MipsPro C/C++* Compiler. We have evaluated our assembly planner with three assemblies, a toy-aircraft kit containing 32 parts, a simple cube-assembly, and the headlight/turnlight assembly consisting of 33 parts modelled with 69 700 polygons and 200 031 triangles. Fig. 9 shows the computation times of these benchmarks. The preprocessing step, i.e. the generation of all $2\frac{1}{2}$D stereographical projections of the headlight assembly takes about 70 minutes; the major part of the computation time is due to the ray-casting. The advantage of the method, proposed in this paper, is the possibility to obtain all distances in any direction. Therewith, we will be able to plan combined translational and rotational movements, in the near future. One computed assembly plan is depicted in Fig. 10. The weights for evaluation criteria have been set to 0.2 for parallelism and 0.8 for separability. The complete computation time took 82 minutes.

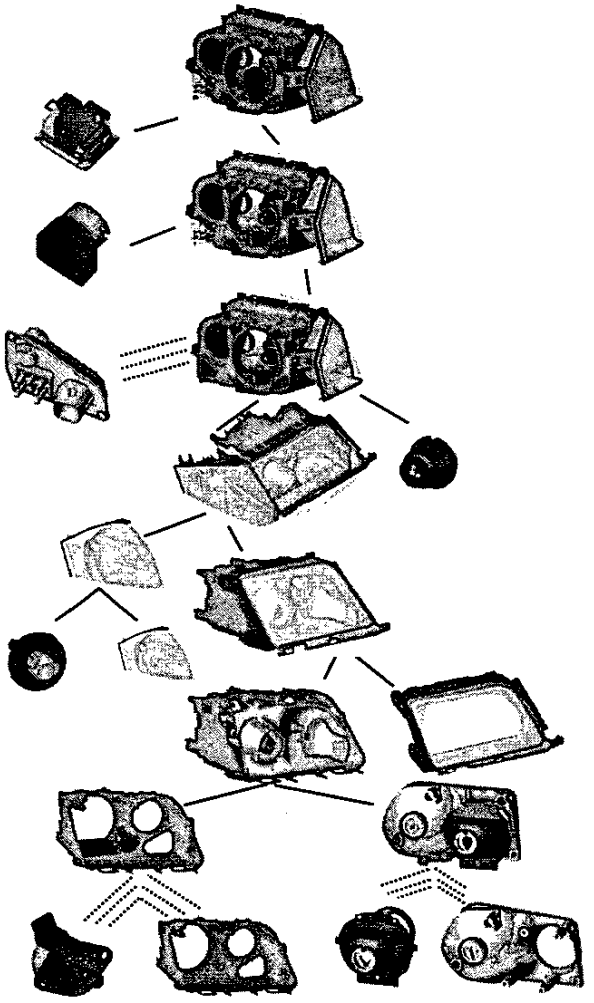

Figure 10: An excerpt of the computed assembly plan for the headlight assembly

5 Conclusion and Future Work

In this paper we presented a new assembly planning approach yielding a very precise computation of $\mathcal{C}$-space obstacles in 3D. In order to reduce the burden of complexity for considering all combinations of objects, we apply stereographical projections of $\mathcal{C}$-space obstacles. This approach is very related to the motion space approach suggested in [7]. For computing translational mating movements, the approach presented here generates binary images of stereographical projections. This gives the advantage, that the generation and evaluation of assembly sequences can be accomplished without considering the vertices contained in the assembly group. The computation time for partitioning only depends on the used resolution

Assembly	# Parts	# Polygons	# Triangles	Resolution	Generation	Evaluation	Total
Cube	10	500	1 751	16	18 s	10 s	28 s
Toy-aircaft	32	3 848	47211	16	432 s	214 s	646 s
Headlight	33	69 700	200 031	256	67 min.	15 min.	82 min.

Figure 9: Computation times for benchmarks

of the stereographical projections. Thus, it can be computed in $O(n^2r^2)$ time, which even can be accelerated by graphics hardware. Our experiments with industrial products consisting of thousands of triangles have shown, that our approach is applicable for practical assembly planning problems. Nevertheless, possibilities of improvement of evaluation of the AND/OR-graphs will be investigated in near future. We also plan to include constraints induced by the robot kinematics, the number of DOFs of the used robots and constraints due to grasping, etc. In the future, this assembly planner constitutes the basis for automatically generating robot programs for handling and assembly in manufacturing applications.

References

[1] A. P. Ambler and R. J. Popplestone. Inferring the position of bodies from specified spatial relationships. Technical Report 117, Department of Artificial Intelligence, 1975.

[2] L. I. Liebermann and M. A. Wesley. AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly. *IBM J. Res. Devl.*, 21(4):321–333, 1977.

[3] T. Lozano-Perez and P. H. Winston. LAMA: A Language for Automatic Mechanical Assembly. In *International Joint Conferences on Artificial Intelligence*, pages 710–716, 1977.

[4] R. H. Wilson and J. C. Latombe. Geometric Reasoning about Mechanical Assembly. *Artificial Intelligence*, 71:371–396, 1994.

[5] B. Romney, C. Godard, M. Goldwasser, and G. Ramkumar. An Efficient System for Geometric Assembly Sequence Generation and Evaluation. In *ASME Intellegent Computers in Engineering Conference*, pages 699–712, 1995.

[6] S. G. Kaufmann, R. H. Wilson, R. E. Jones, and T. L. Calton. The Archimedes 2 Mechanical Assembly PLanning System. In *IEEE International Conference on Robotics and Automation*, pages 3361–3368, 1996.

[7] D. Halperin, J. C. Latombe, and R. H. Wilson. A General Framework for Assembly Planning: The Motion Space Approach. *Algorithmica*, 26:577–601, 2000.

[8] P. K. Agarwal and M. Sharir. Pipes, Cigars, and Kreplach: The Union of Minkowski Sums in Three Dimensions. In *ACM-SIAM Symposium on Discret Algorithms*, pages 00–00, 1996.

[9] F. Schwarzer and A. Schweikard. C-space Cell Defragmentation for General Translational Assembly Planning. In *IEEE International Conference on Robotics and Automation*, pages 1537–1543, May 2001.

[10] S. Sundaram, I. Remmler, and N. M. Amato. Disassembly Sequencing Using Motion Planning Approach. In *IEEE International Conference on Robotics and Automation*, pages 699–712, 2001.

[11] D. Lozano-Perez. Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers*, 32(2):108–120, February 1983.

[12] K. Mehlhorn. *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*. Springer-Verlag, 1998.

[13] B. Charzelle and L. Palios. Decomposing the Boundary of a Nonconvex Polyhedron. *Algorithmica*, 17:245–265, 1997.

[14] S. A. Ehmann and M. C. Lin. Accurate and Fast Proximity Queries Between Polyhedra Using Convex Surface Decomposition. In *The Europe Conference on Computer Graphics*, pages 00–00, 2001.

[15] M. B. Barber, D. P. Dobkin, and H. Huhadanpaa. The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathmatical Software*, 22(4):469–483, December 1996.

[16] L. S. De Mello and S. Lee. *Computer-Aided Mechanical Assembly Planning*. Kluwer Academic Publisher, 1991.