ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2005, Vol. 34, No. 4

AN ALGORITHM SPECIALISED FOR MAPPING OF ELECTRONIC COMPONENTS ONTO 3D OBJECTS

Giedrius Liutkus, Jonas Matickas

Department of Multimedia Engineering, Kaunas University of Technology
Studentų St. 50, LT–51368 Kaunas, Lithuania

Abstract. In this paper we introduce novel algorithm for mapping of electronic components onto 3D objects. The algoritm, called Algorithm Specialised for Electronic Components, is compared with other algorithms suitable to perform this task. Exhaustive experimental results are provided and discussed as well. Algorithm Specialised for Electronic Components is recognized as best suitable for 3D-MID technology, but it is not limited to it.

1. Indroduction

As more and more miscellaneous products contain integrated parts and mechanisms, knowledge from various fields is required in order to manufacture them. The arrangement of electronic components (interconnect and packaged components) within a mechanical structure is an example of such integration.

Because the size requirements for many devices equipped with electronics are getting stricter – the electronic piece parts should be as small as possible. While placing electronic components within mechanical structures, unused space must be minimized.

In various papers it is proved that considering many aspects (mechanical constraints applied to electronic circuitry, thermal, pressure analysis, etc.) in early design stages saves manufacturing time and money [2, 3, 15]. Appearance of commercial software confirms the importance of this topic [4, 16]. Companies like Boeing, Rockwell Collins, NASA, Lockheed Martin, Airbus, Renault participating in international consortiums PDES Inc. [13] and ProStep [14] are producing multi discipline products and aim to solve integrated tasks.

During the analysis, 3D-MID (Moulded Interconnect Device) technology [6] was recognized as suitable for implementing the mapping of electronic components onto 3D objects. Scientific work is only emerging in this field [1], though the application area is very broad [9, 10, 11], 12]. Computer aided designing aspects of electronic components mapping onto 3D objects using 3D-MID technology are not enough researched yet. Nevertheless, the need to have more automatic ways to map electronic component onto 3D objects, which are suitable for 3D-MID technology, is increasing now. The algorithm proposed here is targeting to fill exactly this gap.

The task of mapping electronic components onto 3D objects is solved successfully by using algorithms based on orthogonal projection and parametric equations of surfaces [8]. The following disadvantages of those algorithms are recognized after their implementation and testing:

1. Avoidance of overlap of electronic component is not guaranteed,
2. Preservation of the original shapes of packaged components, the width of tracks is not guaranteed,
3. The orthogonal projection-based algorithm can cause big distortions.

2. Algorithm Specialised for Electronic Components

A new algorithm (called „Algorithm Specialised for Electronic Components“) is dedicated to one task only – to map PCB and PCA components onto 3D object. The algorithm based on the orthogonal projection is used in the areas, where distortions are not important, partially for the mapping of traces.

3D objects and surfaces they consist of are approximated using triangles. The projection-based algorithm is used as triangulation algorithm here [7].

Some steps in Algorithm Specialised for Electronic Components are used for all kind of electronic components:

1. Triangles approximating 3D surfaces are transformed by the following matrix.

   Here $a = sin\theta \times \sin\varphi \times \cos\psi - \cos\theta \times \sin\psi$,
   $b = \sin\theta \times \sin\varphi \times \sin\psi + \cos\theta \times \cos\psi$,
   $c = \cos\theta \times \sin\varphi \times \cos\psi + \sin\theta \times \sin\psi$,
   $d = \cos\theta \times \sin\varphi \times \sin\psi - \sin\theta \times \cos\psi$.

$$T = \begin{pmatrix} \cos\phi * \cos\psi & \cos\phi * \sin\psi & -\sin\psi & x \\ a & b & \sin\theta * \sin\phi & y \\ c & d & \cos\theta * \cos\phi & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

2. Triangles are skipped depending on their normal vectors.
3. Triangles which are completely outside the outline of PCB are skipped.
4. Modified *Z* buffer is applied [5, 668-672 pp.].
5. Triangles are transformed in such a way that their normal vector becomes equal to (0, 0, 1).
6. Adjacent triangles are determined.

In the first step of algorithm all triangles are transformed by matrix *T* (1), in order to minimize distortions of 2D shapes of electronic components. In this matrix θ, ψ and φ are rotation angles around *X*, *Y* and *Z* axes, respectively. The major goal here is to get smallest possible angle between the normal vector of the surface containing the biggest number of electronic components mapped to it and vector (0, 0, 1), in order to minimize distortions.

The normal vectors of all the triangles are analyzed in the second step. If the normal vector is pointing outside the end-user (*Z* component of the vector is negative), the triangle is skipped and not processed further.

In the third step, the remaining triangles are processed using a modified *Z* buffer algorithm. The algorithm used here differs from the regular Z buffer algorithm so that one triangle out of two, containing the biggest *Z* coordinate, is skipped only if the triangle is fully covered by other triangles with a smaller Z coordinate. The term 'cover' can be explained using CSG operations – if the result of the subtraction of the area of currently processed triangle (with biggest Z coordinate) from the common area of all other triangles is empty – this triangle is skipped as it is not visible to the end-user.

One more operation to reduce the number of triangles to be processed is performed in Step 4. Each triangle is evaluated according to the outline of PCB. If a triangle is fully outside the outline, the triangle is skipped.

In Step 5, the remaining triangles are transformed so that their normal vectors become equal to (0, 0, 1). This step allows reducing the distortions if traces are mapped using the orthogonal projection algorithm. Some of the triangles may overlap after transforming them so that their normal vectors become equal to (0, 0, 1). After this transformation the triangles become 2D – all Z coordinates of three triangle vertices become equal to 0. This is exactly the goal of this step. The triangles, which are adjacent in 3D, are no longer adjacent and/or overlap after the transformation performed in this step. After further transformations, the triangles are made adjacent again. This is done not for all triangles, but only within the area of one mapped track.

In the last step all adjacent triangles are determined. The simplest way to do this is to take each triangle and search for triangles sharing two vertices with the current triangle. Each triangle of 3D object has 3 adjacent triangles if 3D object is closed and correctly triangulated.

After the analysis, the following conclusion is made – the algorithm described here gives good results if adjacent triangles are treated those sharing one common edge. Gaps between triangles, overlapping of triangles and other cases breaking adjacency of triangles are not allowed.

3. Mapping of packaged components and footprints

The main requirement for the mapping of packaged components and footprints – is to minimize the distortions of components after they are mapped. The shape of a packaged component is not distorted if the algorithm described in this article is used. If the packaged component is mapped to the surface of other type than plane – the component is touching the surface in one or few places, but not completely. So packaged components can't be mapped to surfaces whose normal vector within the region where the packaged component is mapped varies more than some predefined threshold.

The steps of the algorithm for mapping of the packaged components are as follows:

1. Calculate the geometrical centre point of the packaged component outline.
2. Determine the point on 3D surface where the geometrical centre point calculated in previous step is mapped.
3. Find the triangle containing the point determined in the previous step.
4. Calculate the normal vector of the triangle.
5. The packaged component is mapped so that its surface would be as close as possible to the triangle determined in Step 3.

A graphical representation of packaged component mapping steps is provided in Figure 1, a – e, while additional steps for footprints – in f and g.

In order to simplify calculations – a bounding box of a packaged component is used instead of an exact outline, while calculating geometrical centre point (Figure 1, a).

In the second step of the algorithm (Figure 1, b), all surfaces of 3D object are analyzed. Surfaces containing the point with the same (x, y) coordinates as the geometrical centre point of the packaged component are determined. If a few such surfaces are found, they are compared and the surface with the biggest Z coordinate is chosen.

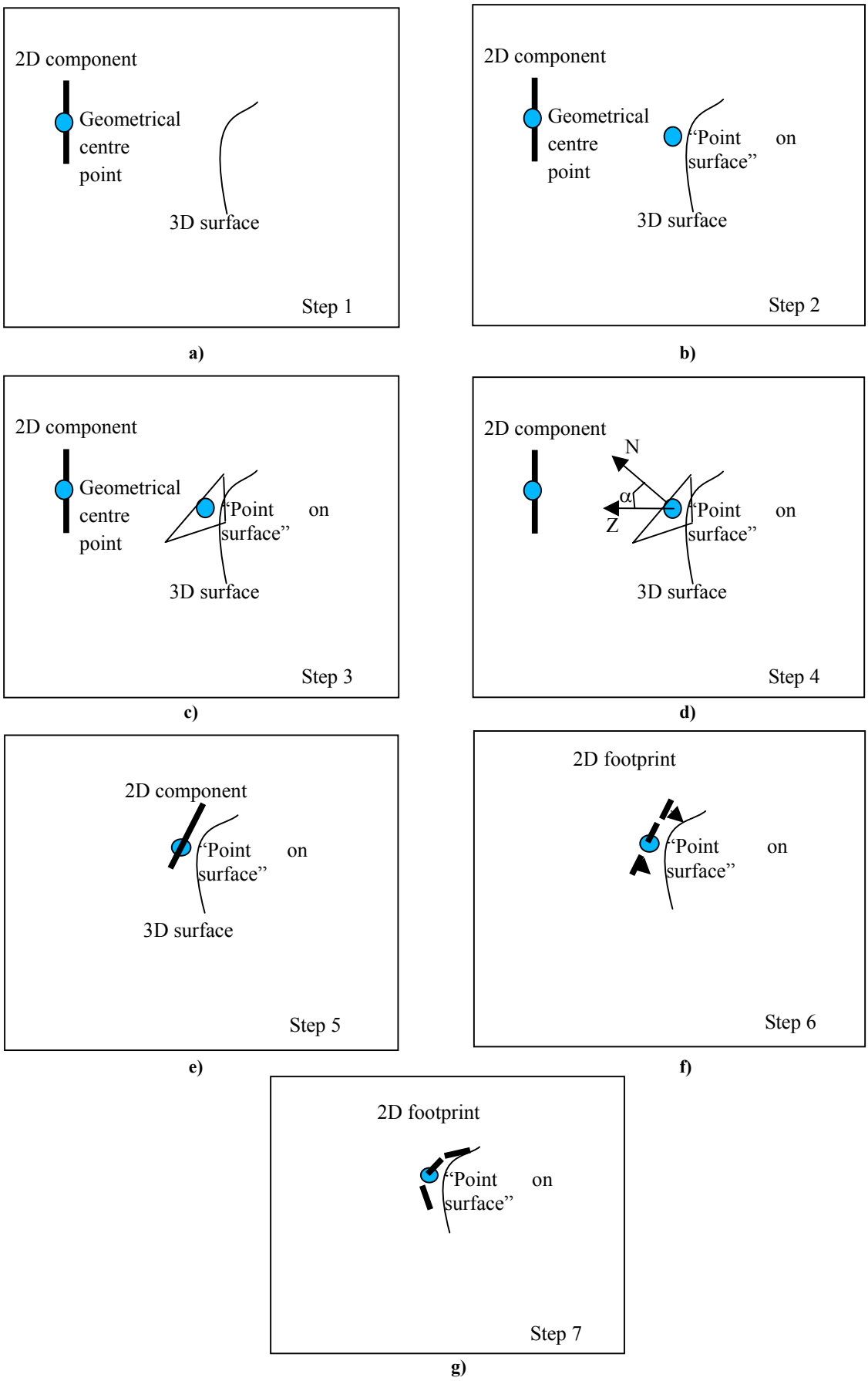

Figure 1. Mapping of 2D packaged component and its footprint onto 3D object

The triangle, approximating the surface determined in Step 2 and containing the geometrical centre point of the packaged component to be mapped, is determined in Step 3 (Figure 1, c).

In Step 4 (Figure 1, d), the normal vector of the triangle is calculated (as multiplication of 2 vectors derived from the edges of the triangle).

In Step 5 (Figure 1, e), 2D shape of the packaged component is transformed in such a way that its Z vector of the plane (which is initially equal to (0, 0, 1)) would coincide with the normal vector of the surface at the determined point. First of all, 2D shape of the packaged component is translated so that its geometrical centre point would coincide with the fiducial point. The 2D shape of the packaged component is rotated by angle α around the calculated vector $S = N * Z$. Finally, the geometrical centre point of 2D shape of the packaged component and the point on the surface are transformed so that they match. The composition matrix for this step is as follows:

$$L = A \times B \times A^{-1},$$

where

$$A = \begin{pmatrix} 1 & 0 & 0 & -TP_X \\ 0 & 1 & 0 & -TP_Y \\ 0 & 0 & 1 & -TP_Z \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

TP_X, TP_Y and TP_Z are the coordinates of the point on the 3D surface and B is the rotation matrix about the vector S by angle α.

An automatic mapping of packaged components onto 3D surfaces may map them on non-planar surfaces or cause minor overlapping. This can be fixed interactively by changing the position of some packaged components.

Mapping of footprints is similar to mapping of packaged components in Algorithm Specialised for Electronic Components. The geometrical centre point of the outline of footprint is used while mapping footprints instead of the geometrical centre point of packaged components. Two additional steps are performed during the mapping of footprints onto 3D object.

// MAPPING OF PACKAGED COMPONENTS

// 1) The geometrical centre point of outline of packaged component is calculated.

$C_X = (\min(x^{2D}) + \max(x^{2D}))/2$; $C_Y = (\min(y^{2D})\min(L)_Y + \max(y^{2D}))/2$;

$\max Z = -\infty$; int $ind = -1$;

// 2) The point on 3D surface matching the geometrical centre point of a packaged component is calculated (f – the function, which calculates the point P on 3D surface, T – the transformation matrix of 3D object, TR – the set of triangles)

```
for (int i = 0; size(TR); i++) {
     P = f(TRi, CX, CY, T);
if (P == null) continue;
```

// 3) Determine the triangle of surface, containing the geometrical centre point of a packaged component mapped onto 3D surface.

// 4) Calculate the normal vector at the point P within the triangle.

```
N = f2(TRi, P, T);
if (N == null) continue;
NI = T * N;
if (PZ > maxZ){ maxZ = PZ; ind = i;}}
```

// 5) The packaged component is mapped so that its surface is as close as possible to the triangle TR_{ind}

$$M1 = \begin{pmatrix} 1 & 0 & 0 & -P_X \\ 0 & 1 & 0 & -P_Y \\ 0 & 0 & 1 & -P_Z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$V^I = (0, 0, 1)$; $\alpha = f(N^I, V^I)$; $S = N^I \times V^I$;

// $M2$ – the rotation matrix around the vector S by angle α.

$M1^{I} = M1 \times M2$;

$$M3 = \begin{pmatrix} 1 & 0 & 0 & P_X \\ 0 & 1 & 0 & P_Y \\ 0 & 0 & 1 & P_Z \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad M4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & P_Z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$M1^{II} = M1^{I} \times M3$;

$M4^{I} = M4 \times M1^{II}$;

// The transformation of the geometrical centre point of the packaged component.

$C^{I} = C \times M4^{I}$;

In Step 6 (Figure 1, f), lands contributing to footprint are used instead of the outline of the footprint. The Lands remain in the same plane as the outline of footprint transformed in Step 5.

In Step 7 (Figure 1, g), each land is processed separately – it is mapped using an orthogonal projection. The main difference between mapping of footprints and packaged components is obvious in this step. Footprints are mapped onto 3D surface completely. Therefore packaged components are usually touching the surface at a few places only. They are completely on the surface only if this surface is plane.

4. Analysis of Algorithm Specialised for Electronic Components

During the implementation of Algorithm Specialised for Electronic Components it was determined that traces and lands, which were connected initially, are not connected after they are mapped onto 3D object (Figure 3). So the task of preserving the connectivity between traces and lands has to be solved. The traces are mapped using a different algorithm than lands – the algorithm based on the orthogonal projection. Lands and traces would connect after they are mapped onto 3D surface if lands would be mapped using the algorithm based on the orthogonal projection. This is shown in Figure 3 as a distorted land. But the land mapped using Algorithm Specialised for Electronic Components is not distorted (Figure 3). Possible solutions for this task are the following:

1. Transform the vertex of the track so that it would connect with the land.
2. Transform the land so that it would connect with the track.
3. Add an additional linear segment connecting vertex of the track with the land.

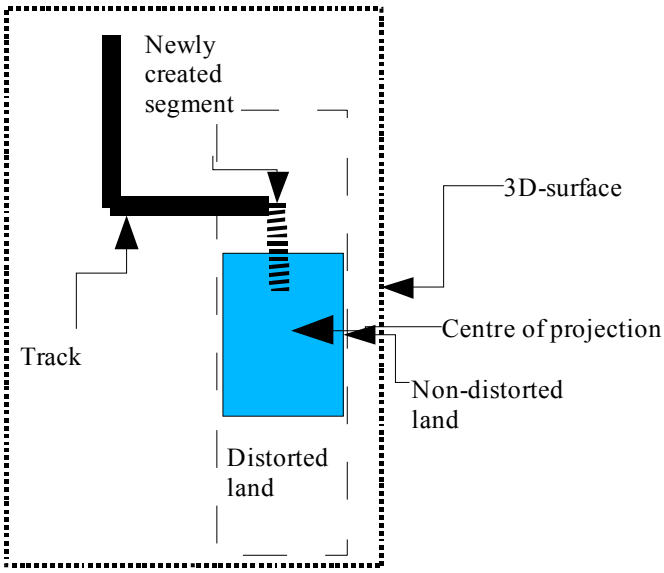

Figure 3. The loss and reconstruction of connectivity between the track and land after the projection onto 3D object

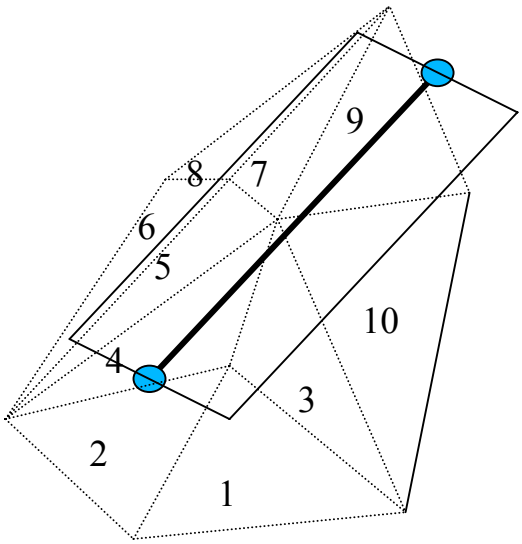

Figure 4.

After the analysis of those three solutions, the following drawbacks are noticed:

a) A track may intersect with other traces and/or lands, which it was not intersecting with in the initial design after transforming its vertex. This would violate the topology of the initial design.
b) Individual transformation of the land is usually not acceptable as the position of the land is dependent on the position of the terminal of the packaged component it is designed to be connected with. The transformation of the land may result in a loss of connectivity between the land and terminal of the packaged component.
c) Addition of a linear segment changes the initial design.

The third solution is recognized as causing less bad consequences than other solutions. So it is chosen here.

After the analysis of Specialized Algorithm for Electronic Components it was discovered that CSG intersection between the rectangle of the track segment and triangle might be not empty, even if the centreline of the track is not intersecting with particular triangle. This can happen when width of the track is bigger than any edge of the triangle (Figure 4). The centreline of the track is drawn thicker, while the rectangle of the track segment – as thin solid lines. Triangles are numbered and drawn as dashed lines. In the example from Figure 4, the centreline of the track is intersecting with triangles numbered 4th, 3rd, 10th and 9th only. But the rectangle of the track segment is intersecting with all triangles drawn in this figure. The sequence in which triangles are processed depends on

the numbering of edges in triangles. The algorithm provided here process triangles against the rectangle of the track segment starting at any vertex of the linear segment. In this example the sequence is as following: 4th (the vertex of the track is in this triangle), 2nd (adjacent to the 4th triangle), 1st (adjacent to 2nd), 3rd (adjacent to 1st), 10th, 9th, 7th, 8th, 6th and 5th. A complete tree of triangles processed in this example is provided in Figure 5.

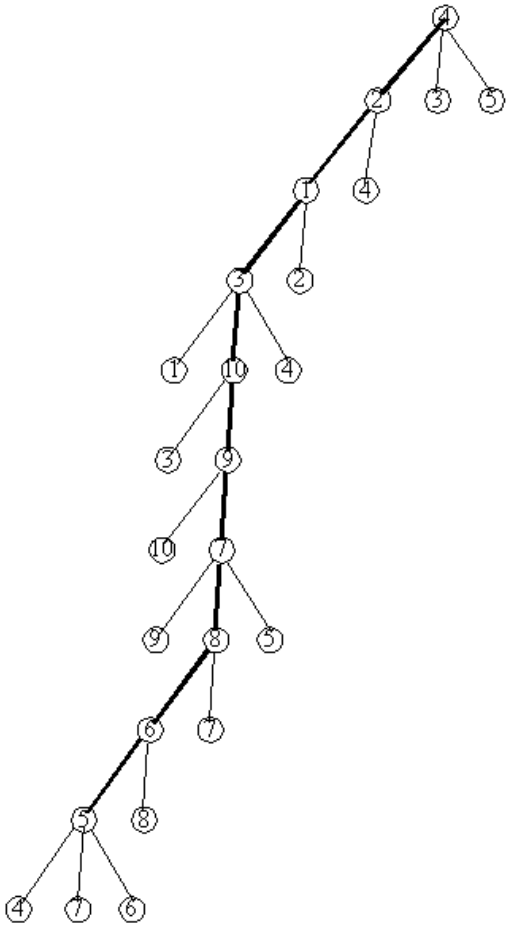

Figure 5. Example of the tree reflecting the sequence of processed triangles

The sequence described above is the minimal path and is drawn as bold lines in Figure 5. Multiple processing of the same triangle is avoided by storing the processed triangles in a special list. In the example provided in Figure 5, 10 triangles are processed in 25 iterations. The algorithm described here in the worst case can determine and process n triangles needed for one-track segment in 3n iterations. Extra (redundant) 2n iterations are very fast as they are just logical comparisons based on the special list mentioned above.

Calculation time dependence on the number of triangles for Algorithm Specialised for Electronic Components using two examples named *Trial* and *Sensor*, respectively. The calculation time is compared with the curve reflecting $O(n \times \ln(n))$ complexity for those examples (Figure 6).

It can be clearly seen that the calculation time is worse than $O(n \times \ln(n))$, but better than $O(n^2)$. In spite of the fact that there are 17 times more packaged components in the *Sensor* example rather than in *Trial*, the calculation time for those examples is not much different. Most of the algorithm calculation time is spent for analysing triangles and determining adjacent triangles. The latter action does not depend on the number of packaged components in the initial 2D design. The calculation time for determining adjacent triangles using the same 3D object depends on the transformation applied to the 3D object. This transformation is indirectly provided by the end-user and it is partially random, so the curve reflecting the calculation time dependence on the number of triangles is not smooth.

We also compared the calculation time needed for the orthogonal projection based algorithm and Algorithm Specialised for Electronic Components (Figure 7). Four different examples approximated by a different number of triangles are used here. After analysing diagrams of the calculation time for *Trial* PCB example provided in Figure 7 we can make a conclusion: though Algorithm Specialised for Electronic Components is more complex than the algorithm based on the orthogonal projection, the calculation time differs for bigger examples only (4th 3D object in Figure 7).

The difference between the calculation time needed for those two algorithms gets even smaller if more complex PCB designs are used (Figure 8). The reason is already mentioned earlier: a larger part of the calculation time is spent for analysis of triangles, which is independent from 2D design. The order of the complexity of Algorithm Specialised for Electronic Components is the same as the order of the orthogonal projection algorithm (worse than $O(n \times \ln(n))$ and better than $O(n^2)$). Though theoretical complexity of Algorithm Specialised for Electronic Components is bigger, due to the semantic richness of data, this algorithm is processing less graphical primitives than the algorithm based on the orthogonal projection. In the example provided in Figure 7, the first algorithm processes 16 components, while the second – 1990 graphical primitives. In the case of the example from Figure 8, those numbers are 277 and 16657, respectively.

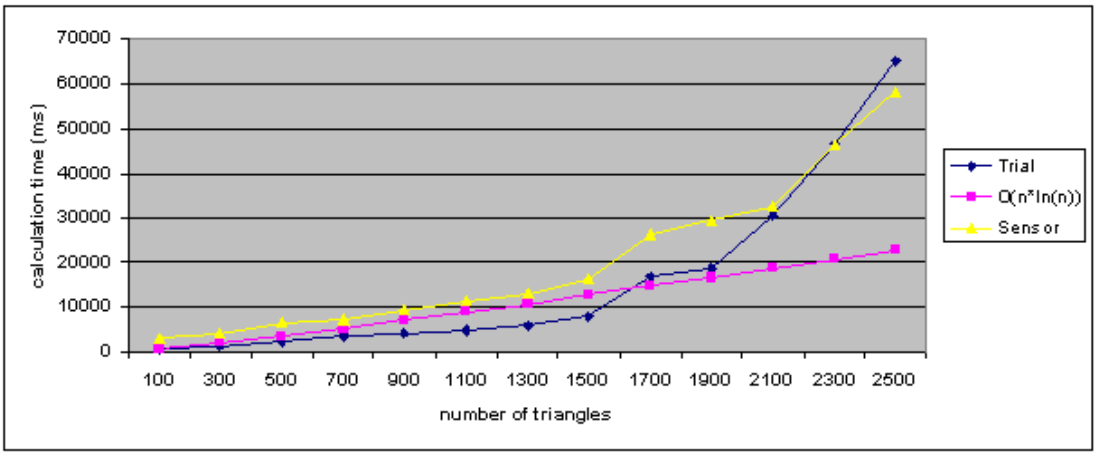

Figure 6. Calculation time dependence on the number of triangles for Algorithm Specialised for Electronic Components

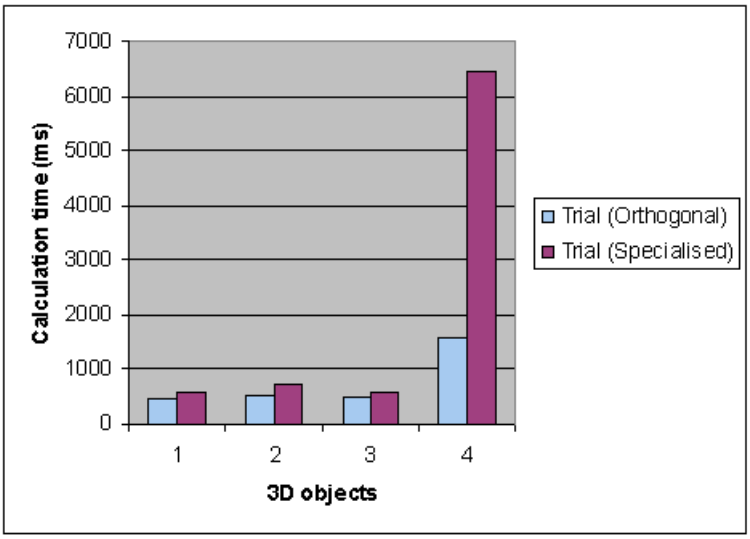

Figure 7. The comparison of orthogonal projection-based algorithm and Algorithm Specialised for Electronic Components using Trial example

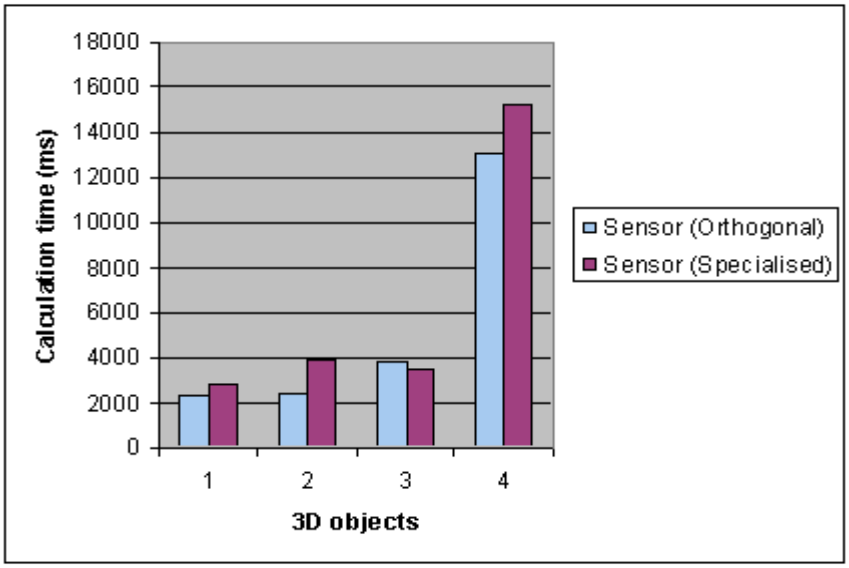

Figure 8. The comparison of orthogonal projection-based algorithm and Algorithm Specialised for Electronic Components using *Sensor* example

5. Conclusions

1. The advantages of Algorithm Specialised for Electronic Components are the following:
   a) geometrical models of electronic components (packaged components and track width) remain the same, or change a little only (footprints and padstacks),
   b) the topology of electronic components is not altered,
   c) the overlapping of electronic components is not caused.
2. Algorithm Specialised for Electronic Components described in this article is suitable for single sided boards and surface mount devices only.
3. The calculation time of Algorithm Specialised for Electronic Components is of the same order as the orthogonal projection-based algorithm (longer than $O(n \times \ln(n))$, but shorter, than $O(n^2)$). In spite of the fact, that, theoretically, the first algorithm is more complex than the second one, it has to process less geometrical objects due to the semantic richness of data it is using.
4. The relevance of the problems solved in this work is proved by a successful deployment of software implementations of the algorithms, formulated in this work. Software is deployed in companies like LPKF, Boeing, NASA JPL and Rockwell Collins. In the latter one, the results of this work are used in real manufacturing processes.

References

[1] **A. Carpenter.** 3D exploration of Printed Circuit Boards. UK Design Forum 2001. *Conference proceedings*, *Manchester, UK*, 2001, *March* 25-26.

[2] Construction History and Parametrics: Improving affordability through intelligent CAD data exchange. *CHAPS Program Final Report. Advanced Technology Institute*, 2004.

[3] Economic Impact Assessment of the International Standard for the Exchange of Product Model Data (STEP) in Transportation Equipment Industries. *Final Report. National Institute of Standards and Technology*, 2002.

[4] EM Designer. Zuken. *http://www.zuken.com/ electromechanical/ em_designer.asp*.

[5] **J.D. Foley et al.** Computer Graphics: Principles and Practice in C (2nd Edition). *Addison-Wesley Professional*, 1995. ISBN 0201848406.

[6] **A. Housden, J. Gould.** Moulded Interconnect Devices. *PRIME Faraday Partnership*, 2002. ISBN 1-84402-008-8.

[7] **G. Liutkus, J. Budreckytė.** Enhancement of algorithms for virtual exposure. *Information technology and control* (ISSN 1392-124X), 2003, *T*.28, *No*.3, 67-75.

[8] **G. Liutkus, J. Matickas.** Virtual modeling of 3D-MID device functionality. *Information technology and control* (ISSN 1392-124X), 1999, *No*.4, 28-37.

[9] **P. Mapleston.** 3-D interconnect offer sizeable market opportunity. *Plastiscope*. 2000, March.

[10] Mechatronic Project. Bmb+f Forschungsverbundprojekt. *http://www.mechatronic-project.com/*.

[11] Molded Interconnect Device Moulded Interconnect Device LDS Technology. *LPKF Laser & Electronics AG. http://www.lpkf.com/applications/3d-mid/index.htm.*

[12] Moulded interconnect devices switch mobile signals. *Tyco electronics http://www.electronicstalk.com/news/tya/tya145.html*.

[13] PDES. *Inc. Accelerating the Development and Implementation of ISO 10303, STEP. http://pdesinc.aticorp.org/.*

[14] ProSTEP. *Integrate the Future. www.prostep.de*.

[15] Rockwell Collins. *Inc. http://www.rockwellcollins.com/.*

[16] Solid Edge for electro-mechanical design. EDS. *http://www.solidedge.com/industry/files/SolidEdgeElectro-MechanicalBrochure102902.pdf.*