INTERNATIONAL DESIGN CONFERENCE - DESIGN 2004
Dubrovnik, May 18 - 21, 2004.

DESIGN 2004

A NEW CONCEPT OF SUPPORTING COMPLEX MODEL BUILDING IN VAGUE DISCRETE INTERVAL MODELLING

Z. Rusák and I. Horváth

Keywords: vague discrete interval modeling, particle cloud, particle system, positioning operator, constrained instantiation operator, vague positional relationship

1. Introduction

In addition to generating and manipulating shape variants, shape conceptualization also involves definition, modification and maintenance of positional relationships between geometric entities. To resolve this issue, constraint based geometric design is applied in conventional CAD systems, which refers to specification of shape by means of geometrical constraints relating shape features to shape parameters [Boothroyd 1992]. Constraint based geometric design has become useful since it permits both flexibility in geometric specification and conveying design intent. Nonetheless, designing based on geometric parameters somewhat limits the application of such a system to industrial design, where free form shapes are dominating. To be able to specify free form shapes by geometric parameters designers have to understand the mathematical representation of the model. This destroys the intuitiveness of shape conceptualization.

To better support creation of complex free form products in the conceptual phase a novel shape modelling technique, called vague discrete interval modelling (VDIM), has been developed [Rusák 2003]. VDIM facilitates the designer to (a) define a cluster of free-form shapes in one model without parameterization, (b) build complex models by applying Boolean operations and by defining positional relation between vague shape components, and (c) select large number of shape instances based on linguistic rules and at the same time keeping the geometric constraints valid. In this paper, we only address parts of the last two issues, namely, (a) mathematical fundamentals of positional relations between interval models, (b) development low level system functions, called position operators, to define, maintain and manipulate positional relations of vague and nominal shapes, (c) development of high level system functions, called constrained instantiation operator, which applies position operator during instantiation of shape components, to maintain and modify existing positional and geometric constraints between shape variants, and (d) application of constrained instantiation in shape conceptualization of an industrial product. Constrained instantiation operator is a part of rule based instantiation, which facilitates instantiation of shape alternatives based on shape formation principles and linguistic expressions. For more details, interested readers are referred to [Rusák 2004]. In this paper, constrained instantiation is used to embed positioning operators into the instantiation process to embed positioning operators into the instantiation process. In the current implementation, constraints are handled pair-wised, and network of constraints is not treated due to the limited extent of the paper. However, we note that constraints solution methods e.g. [Suzuki 1985, Kimura 1987, and Juster 1992] in the geometric context can be also be applied to VDIM.

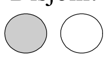
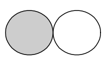
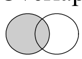
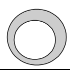
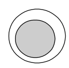
2. Interpretation and formalization of positional relations

In VDIM, positional relations can be defined between either two nominal shapes or two vague shapes. The former one is called nominal positional relation and the latter is vague positional relation. We identified five types of positional relations between two nominal shapes, S_A and S_B:

1. S_A is disjoint from S_B, if all points of S_A are outside the boundary of S_B.
2. S_A contains S_B, if all points of S_B are inside the boundary of S_A.
3. S_A is contained by S_B, if all points of S_A are inside the boundary of S_B.
4. S_A touches S_B, if there exists at least one point on the boundary of S_A coincides with one point of the boundary of S_B, and the rest of S_B's points are either outside or contained in the boundary of S_A.
5. S_A overlaps S_B, if S_A is divided by S_B into three non-empty subsets, $S_{AS,}$ which contains the points of S_A outside S_B, S_{AT}, which contains points of S_A touching S_B, and S_{AC}, which contains points of S_A contained by S_B.

These five relations are called nominal positional relations, since they exist between nominal shapes.
A vague interval model can be represented by a particle cloud i.e. the geometric entity of a component shape or by a particle system i.e. geometric entity of an assembly or subassembly. Since the boundary of a particle cloud/particle system is an interval, it is possible to define two borders that constraints this subspace. We call these borders minimal and maximal overlaying surfaces as they are in contact with the material of the product or with the environment, respectively. Positional relations between vague shapes can be derived by specifying positional relations between the minimal and maximal overlaying surfaces. Positional relations between particle clouds, Π_1 and Π_2, can be defined by bringing their overlaying surfaces into interaction in the following way: (a) $C_{\Pi 1max}$ with $C_{\Pi 2max}$, (b) $C_{\Pi 1min}$ with $C_{\Pi 2min}$, (c) $C_{\Pi 1max}$ with $C_{\Pi 2min}$, and (d) $C_{\Pi 1min}$ with $C_{\Pi 2max}$.
Since we can select different types of nominal positional relations for each pair of the overlaying surfaces, the number of possibilities can be described as N^4, where N the number of possible positional relations. Thus, in principle we can derive 5^4=625 vague relations. However, in practice, some of the relations cannot be realized due to the fact that the maximal overlaying surface always contains the minimal overlaying surface. This reduces the possible number of relations to 62. For instance, having a disjoint or touching relation between the maximal overlaying surfaces implies that the relations between the other 3 pairs of closures must be of disjoint. If the relation between $C_{\Pi 1max}$ and $C_{\Pi 2max}$ is contain or contained by, it implies a contain or contained by relation between $C_{\Pi 1max}$ and $C_{\Pi 2min}$, respectively. Table 1 shows a specific example.
For vague shapes, we can define robust relations, by which a positional relation between two particle clouds is defined in such a way that every pair of selected instance shapes has the same relation. In this case, we assume that the position of the local reference frame of each instance shape is fixed relative to the local reference frame of the vague shape models. In order to define a robust vague relation, each pair of overlaying surfaces has to be in the same relation: for instance, each of them has to be disjoint. This means that only five robust vague relations can be defined: (a) disjoint, (b) touching, (c) overlapping, (d) containing, and (e) contained by.

Table 1. Overlaying surfaces-relation table between two particle clouds

Overlaying surface / relations	Disjoint 	Touch 	Overlap 	Contain 	Contained by 
$C_{\Pi 1max}$- $C_{\Pi 2max}$		X			
$C_{\Pi 1min}$- $C_{\Pi 2min}$	X				
$C_{\Pi 1max}$- $C_{\Pi 2min}$	X				
$C_{\Pi 1min}$- $C_{\Pi 2max}$	X				

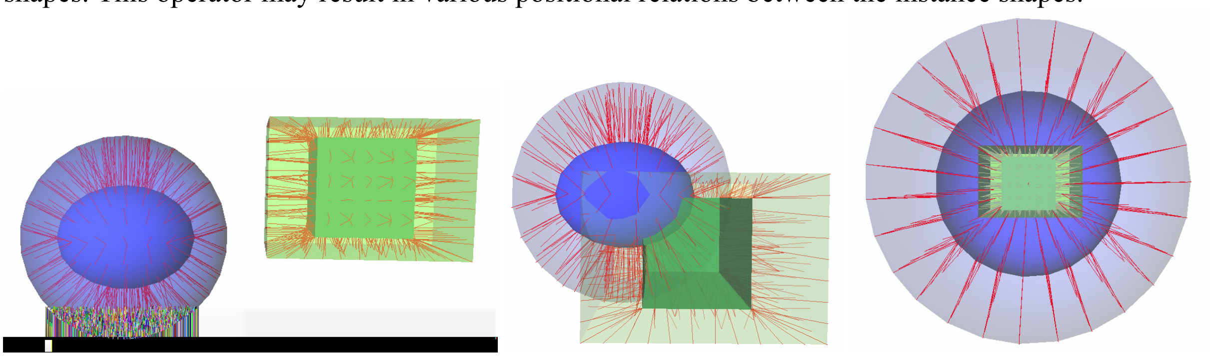
However, it is not possible to define a touching relation between each pair of overlaying surfaces, unless $C_{\Pi1min}$ coincides with $C_{\Pi1max}$ and $C_{\Pi2min}$ coincides with $C_{\Pi2max}$. Thus, the following robust positional relations can be defined for two particle clouds Π_1 and Π_2:

1. Π_1 is disjoint from Π_2 if $C_{\Pi1max}$ is disjoint from $C_{\Pi2max}$.
2. Π_1 overlaps Π_2, if $C_{\Pi1max}$ overlaps $C_{\Pi2max}$, $C_{\Pi1min}$ overlaps $C_{\Pi2min}$, $C_{\Pi1max}$ overlaps $C_{\Pi2max}$, and $C_{\Pi1min}$ overlaps $C_{\Pi2max}$.
3. Π_1 contains Π_2, if $C_{\Pi1min}$ contains $C_{\Pi2max}$.
4. Π_1 is contained by Π_2, if $C_{\Pi1max}$ is contained by $C_{\Pi2min}$.

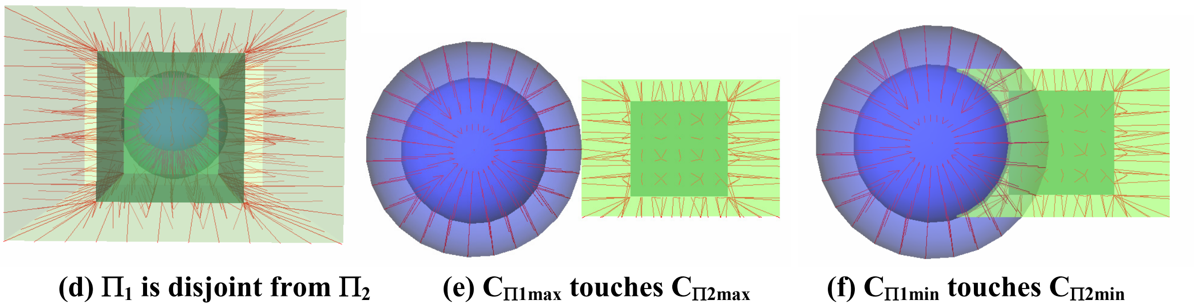
To be able to define a touch relation between any instance shapes of the vague model, the position of the vague shapes has to be manipulated. Since a touch relation of the components occurs most frequently in assembly modelling, we have to provide a solution which allows the designer to define touch relations between vague interval models. To address this issue the concept of position operator has been developed.

3. Operator oriented management of positional relations

The role of position operators is to define, change, or maintain positional relation between vague or nominal shapes by transforming the local reference frame of one of the input particle clouds or particle systems relative to another input shape. Two kinds of position operators can be defined based on positional relations. One kind of operator keeps the position of two interacting particle clouds (i.e., the position of the reference frames of the particle clouds/systems) fixed relative to each other, and detects the changes in positional relations. The other kind of operator changes in the position of the two particle clouds in order to maintain their positional relation. The first kind of operator will be referred to as a permanent position operator, and the second kind as a flexible position operator. A permanent position operator, Ψ^P, (a) defines positional relations between two particle clouds, Π_1 and Π_2, in which their local reference frames, $\Gamma_{\Pi1}$ and $\Gamma_{\Pi2}$, are fixed so as ${}^{\Gamma\Pi1}\Gamma_{\Pi2} = {}^{\Gamma\Pi1}\Gamma \cdot {}^{\Gamma}\Gamma_{\Pi2} \doteq$ constant, where Γ is the global reference frame, ${}^{\Gamma\Pi1}\Gamma_{\Pi2}$ is $\Gamma_{\Pi1}$ defined in $\Gamma_{\Pi2}$, ${}^{\Gamma\Pi1}\Gamma$ is $\Gamma_{\Pi1}$ defined in Γ, and ${}^{\Gamma}\Gamma_{\Pi2}$ is Γ defined in $\Gamma_{\Pi12}$ (b) computes the type of relation of Π_1 and Π_2, (c) maintains the relation when one of the particle clouds is repositioned in Γ, and (d) calculates the positional relation between each pair of instance shapes. This operator may result in various positional relations between the instance shapes.



(a) Π_1 is disjoint from Π_2 **(b) Π_1 overlaps from Π_2** **(c) Π_1 contains Π_2**



(d) Π_1 is disjoint from Π_2 **(e) $C_{\Pi1max}$ touches $C_{\Pi2max}$** **(f) $C_{\Pi1min}$ touches $C_{\Pi2min}$**

Figure 1. Some positional relations of vague shapes

When permanent position operator is applied robust vague positional relations can be specified between two particle clouds as it is shown in Figure 1. a-d. Figure 1. e-f present examples of the weak 'touch' positional relations of vague shapes, in which either the maximal or the minimal overlaying surfaces are in contact. To be able to specify a robust touch relation between vague shapes, the position of the related shapes has to be manipulated.

The flexible position operator facilitates repositioning of the particle clouds in the course of instantiation in order to maintain the requested positional relation between them. A flexible position operator, Ψ^F, (a) defines positional relations between particle clouds, Π_1 and Π_2, so that their local reference frames, $\Gamma_{\Pi 1}$ and $\Gamma_{\Pi 2}$, respectively, are placed in such a way that the Hausdorff distance of $I_{\Pi 2}$ and $I_{\Pi 2}$ in R_1 and R_2, $I'_{\Pi_2} = I_{\Pi_2}(Z)$ is minimized as follows: $Z_{\arg} = \arg(\min_Z H(I_{\Pi 2}(Z), I_{\Pi 1}))$, where Z are the location and orientation parameters of $\Gamma_{\Pi 2}$ and a given translation parameter defined by the user (b) applies a user defined translation and rotation $\Gamma_{\Pi 2}$, where the matrices are specified in $\Gamma_{\Pi 1}$, (c) computes the type of relation of Π_1 and Π_2, and (d) maintains the relation between each pair of the derived instance shapes, or when one of the particle clouds is repositioned in Γ. For further details on the computation method of Hausdorff distance can be found in (Vergeest et al., 2003). R_1 and R_2 are the influenced regions specified on Π_1 and Π_2. They can be defined either by the user, or by computing the volumetric operation. Based on the above definition of flexible position operator, it is possible to apply a robust touching relation of two particle clouds.

4. Revisiting the issue of constrained instantiation of shapes

Constrained instantiation operator applies geometric constraints to the instantiated shapes of the vague models of assembled components when the geometries of the instances are dependent on each other. Typically, some sort of positional relation is also defined between these components, which needs to be considered in the process of instantiation. The positional relations of particle clouds are defined by position operators and are maintained or redefined by an instantiation operator. Two constrained instantiation operators have been defined, one, called instantiation with offsetting, applies permanent positioning operator, and the other, called instantiation with repositioning, uses flexible position operator. In case of the first one, to define the geometric constraints of two instance shapes, $I_{\Pi 1}$ and $I_{\Pi 2}$, we introduce parameter *z* to express the distance of $I_{\Pi 1}$ and $I_{\Pi 2}$ in a user selected regions, R_1 and R_2, of $I_{\Pi 1}$ and $I_{\Pi 2}$. Depending on the value of *z*, $I_{\Pi 1}$ can be separate, overlap, touch, contained or contained by $I_{\Pi 2}$. The instantiation operator with offsetting, Ξ (a) identifies two particle clouds, Π_1 and Π_2, positioned by $\Psi^P{}_{12}$, (b) selects instance shapes of Π_1 and Π_2 based on shape formation principles (see [Rusák 2004]), (c) identifies a region, R_1, on $I_{\Pi 1}$ selected by the user (d) determines a region, R_2, on $I_{\Pi 2}$, as the intersection of R_1 and $I_{\Pi 2}$, and (e) offsets $I_{\Pi 2}$ in the region R_2 by the value z. Instantiation with offsetting facilitates the generation of a partial copy of a given instance shape to another instance shape. This technique is useful to instantiate contact surfaces between product components.

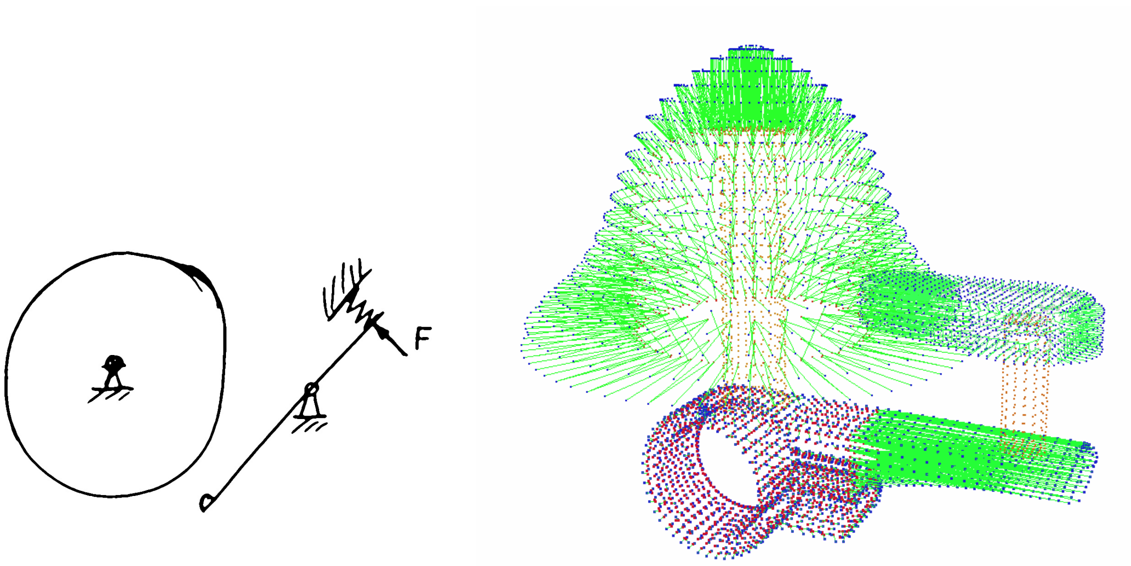
To maintain positional relations of particle clouds, which are positioned relative to each other by flexible positioning, instantiation operator with repositioning has been introduced. Instantiation operator with repositioning, Λ (a) identifies particle clouds, Π_1 and Π_2, positioned by Ψ_{12}, (b) selects instance shapes of Π_1 and Π_2 based on shape formation principles, and (c) computes Ψ_{12}.This instantiation operator facilitates the generation of two independent instance shapes, which are repositioned in the modeling space in such a way that a positional relation between them is maintained.

5. Application of constrained instantiation

This section presents an example for the application of constrained instantiation of vague discrete interval models. In this example, alternatives for a bicycle bell are generated. In the first step of the design, a functional concept is created, which operates the bike bell from the side of the bell by rotational movement as it is shown in Figure 2a. First, the vague shapes of the components of the bike bells are generated. The designer modeled the outside and inside shapes, and combined them into one vague model. Each component is generated in the same way and put together using position operators that specify relations between regions of the components. Figure 2b shows the vague model of the bike bell. The positional relation between the cloche and the holder is defined by the nominal shape of a pin. Thus, these two vague particle clouds are indirectly positioned relative to the pin. This indirect positioning can be realized by using two specialized direct position operators. The first one is applied to the pin and the cloche as a flexible position operator, which establishes a touch relation. In the case of the second one, a nominal positional relation has to be established between the regions of the objects, since this part of the holder has been represented by a nominal shape. Thus, a permanent position operator is applied to establish a touch relation.

For each position operator, the reference object is the pin, and the positioned objects are the cloche and the holder. When instance shapes of the cloche are generated, they are repositioned to the nominal shape of the cylinder. To build the side clapper, a vertical pin represented by a nominal shape has been used to define the position of the side clapper relative to the holder. Due to the fact that the pin is represented as a nominal shape, the vertical position of the clapper is fixed. The position operator is applied again to define a permanent positional relation between the clapper and the pin. The relative positions of the particle system of the clapper and the pin are defined by establishing a touch relation between the cloche and the clapper with the position operator. The touch relation is specified only between selected regions of the cloche and the clapper, in such a way that the cloche is the reference object and the clapper is the positioned object. The relation between the holder and the vertical pin is defined by a flexible position operator. This means that instantiation of the holder is influenced by the position of the pin.

The last step of the design process is the instance generation. Since the shape of the cloche is the most determinant feature of the bell, the biggest shape variances were introduced here. Nevertheless, the shape of the cloche influences the position of the clapper, which affects the shape of the holder. Various instantiation functions have been applied to the cloche, to select a variety of free form shapes. In the next step, the instance shape of the side clapper is generated. To match the instance shapes of the cloche and the clapper, the position of the clapper is recalculated by the position operator. The position of the vertical pin is fixed relative to the clapper, and it is repositioned together with it. The instance shape of the holder is determined by the geometry and the current position of the pin that holds the clapper. Here, constrained instantiation operator is applied with zero offsetting. Four



(a) (b)

Figure 2. Functional model and vague model of a bike bell

instance products that are shown in Figure 3a-d differs in the shape of the cloche, the size of the clapper, and the size/shape of the holder.

6. Conclusions

In this paper, we presented a novel approach to model complex shapes by VDIM. This approach facilitates definition of positional relations between interval shapes, and maintaining or manipulating positional and geometric relations in the course of instantiation of vague interval models. Managing positional and geometric relations in VDIM, provides an easy way to generate large number of product alternatives. The proposed operator oriented solution is unique from the aspect that the positional relations are determined by the geometry of the shape and not by parameters. However, this means that the time spent to compute the network of constraints is increasing. Future research focuses on extending this approach to handling other types of relations, e.g. kinematic, kinetic, deformation, field, which facilitates simulation of the behavior of products.

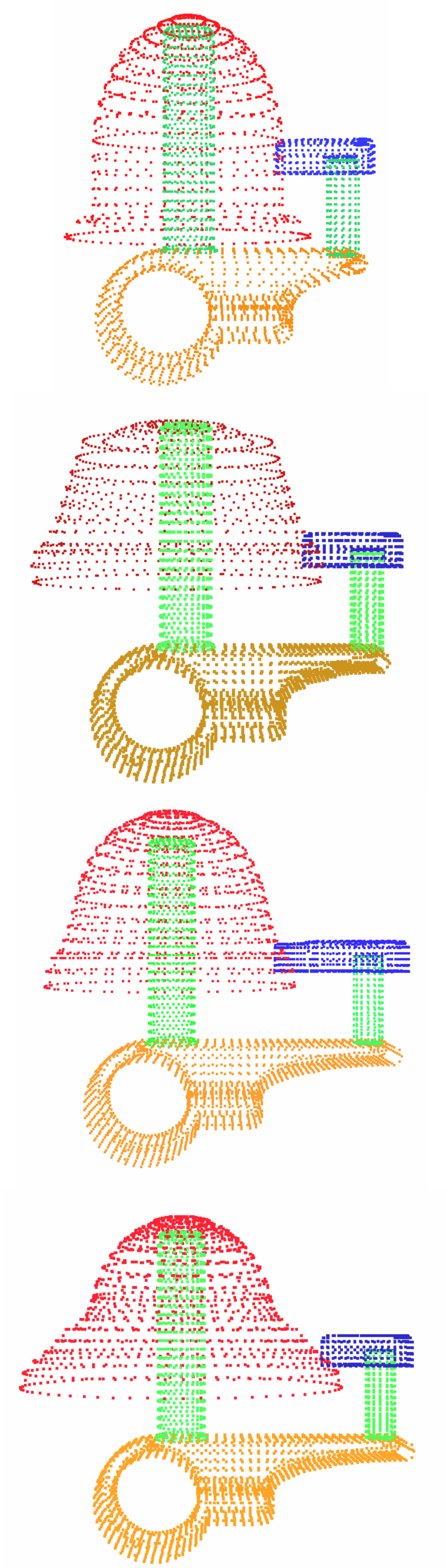


Figure 3. Instance shapes of the bike bells

References

Boothroyd, G., "Assembly Automation and Product Design", Marcel Dekker, New York, 1992.

Juster, N. P., "Modeling and representation of dimensions and tolerances: a survey", Computer Aided Design, Vol. 24, No. 1, 1992, pp. 3-17.

Kimura, F., Suzuki, H., Ando, H., Sato, T., Kinosada, A., "Variational geometry based on logical constraints and its applications to product modeling", Annals of the CIRP 36/1, 1987, pp. 75-78.

Rusák, Z., "Vague Discrete Interval Modeling for Product Conceptualization in Collaborative Virtual Design Environments", Ph.D. thesis, Millpress, Rotterdam, 2003.

Rusák, Z., "Deriving product variances by Rule based instantiation of Vague Discrete Interval Models", to appear in Proccedings of TMCE 2004 Symposium, 2004.

Suzuki, H., Kimura, F., Sata, T., "Treatment of dimensions on product modeling concept", Design and Synthesis, North Holland, 1985, pp. 491-496.

Vergeest, J. S. M., Spanjaard, S., Wang, C., Song, Y., "Complex 3D Feature Registration Using a Marching Template", Journal of WSCG, Vol. 11, No. 1, 2003, pp. 488-495.

Dr. Zoltán Rusák
Faculty of Design, Engineering and Production
Delft University of Technology
Landbergstraat 15, 2628 CE, Delft The Netherlands
Telephone: +31 15 278 3788, Telefax: +31 15 278 1839
E-mail: z.rusak@io.tudelft.nl