DISCRETE DOMAIN REPRESENTATION FOR SHAPE CONCEPTUALIZATION

Zoltán Rusák, Imre Horváth, György Kuczogi,
Joris S.M. Vergeest, Johan Jansson

Department of Design Engineering
Delft University of Technology
Delft, The Netherlands

ABSTRACT

This paper presents a solution for discrete domain representations of 3D geometric models, and techniques for shape instance extraction from a distribution domain. The discrete domain representation captures modality, impreciseness and uncertainty. It facilitates both shape conceptualization and computer processing. We model the elements of the shape by particle clouds, which are generated from regular and dense enough point-set(s) obtained from specific input devices e.g. hand movement detector, 3D scanner. A particle cloud contains a finite number of particles. A particle is a weakly defined 3D point specified by its reference vector, metric occurrence, mass, and velocity. The metric occurrence of the particle represents the geometric uncertainty of the shape. It is defined as the range of distribution between the primary and secondary covering of the domain of variance. Technically, the metric occurrences are handled by the so-called connectivity bushes between primary and secondary covering of a particle cloud. Instance generation operators make it possible to obtain arbitrary number of shape instances of the same type. The paper also presents an application example.

KEYWORDS

Particle cloud, primary and secondary covering, connectivity bush, metric occurrence, shape instancing

1. Introduction

This paper presents a discrete domain modeling methodology for representing uncertainty and modality of design objects. In several industrial applications (e.g., in engineering tolerancing, representation of antropometric body measurements, modeling of stochastic physical phenomena, conceptual shape design), in order to represent the probabilistic geometry of models there is a need to describe a cluster of shapes, rather than single instances. The shape clusters circumscribe a domain of variance of the possible shapes and/or the possible sizes of the model. In some cases, for instance in engineering tolerancing only the size, while in the case of shape conceptualization, the shape of the model is uncertain. In the case of antropometric body modeling, both the size and the shape of the subjects have uncertain properties. There exist several modeling techniques, which are able to model geometric uncertainties and impreciseness. For instance, modeling with fuzzy-sets is one of the popular techniques. Yamaguchi [1] applied the membership function of a fuzzy set, to represent probabilistic property of the position of individual surfaces. He used the fuzzy-set theory for solid models. Blinn described the density of a molecular model using membership-function of a fuzzy-set [2]. Particle systems have already been used for representation of fuzzy objects by Reeves [3] to simulate natural phenomena such as fire, waterfall. However, in order to explicitly represent the uncertainty, we should apply vague representation, where the uncertain domains of geometric points, called metric occurrences, are not intersecting with each other. The application of vague representation assures that the shapes extracted from the vague domain are not self-intersecting.

2. Mathematical fundamentals of discrete domain modeling

In this section we describe the basic modeling entities applied in discrete domain modeling. Figure 1 summarizes the basic modeling entities. Our discrete domain modeling methodology has been built up in a *vector space*, where geometric modeling entities are represented by structures of geometric vectors, **v,** in a reference frame Γ. To define shapes, we follow the conventional definition of a vector. Let $\Re^n$ be a subspace of $\Re^N$, and x and y points so as $(x, y) \in \Re^N \times \Re^n$. We call $\mathbf{v}=(x, y) \in \Re^n$ geometric vector marking off from the x, and y is a position point of **v.** A collection of all vectors, $\mathbf{v} \in \Re^n$, at a point *x*, marked off form point x, is called *vector space*. A collection of *n* linearly independent vectors, $\mathbf{v} \in \Re^n$, is called *reference frame*, Γ, if they are at x and define an orthogonal basis, $\mathbf{e} \in \Re^n$.

Our basic notion is the vague vector, which is defined as follows: A *vague vector* is an ordered pair $\mathbf{v}^v=(\mathbf{v}, \mathrm{e})$, where $\mathbf{v}=(x, y) \in \Re^n$, e is a vector space, where $\mathrm{e}=(y, z) \in \Re^N \times \Re^m \neq \underline{\mathbf{0}}$, and $\|\mathrm{e}\| \neq \infty$. ◊ The collection of vectors, e, satisfying the above definition is called the *metric occurrence* of $\mathbf{v}^v$. The definition of the vague vector can be directly used to define a crisp vector: A *crisp vector* is an ordered pair $\mathbf{v}^c=(\mathbf{v}, \underline{\mathbf{0}})$, where $\mathbf{v}=(x, y) \in \Re^n$. ◊ A crisp vector equals to what is traditionally called geometric vector, i.e., $\mathbf{v}^c \equiv \mathbf{v}$. Therefore, whenever it does not lead to confusion, we refer to a crisp vector simply as *vector*, and denote it by **v.** A *crisp geometric shape* S_C is a non-empty subspace of $\Re^n$ in Γ so that $S_C = \cup\, \mathbf{v}^c$, $|\, \mathbf{v}^c \,| \neq \infty$, and for $\forall$ i,j, i≠j, $\exists \mathbf{v}^c_{\mathbf{i}}$, $\mathbf{v}^c_{\mathbf{j}}$,, $|\, \mathbf{v}^c_{\mathbf{i}} - \mathbf{v}^c_{\mathbf{j}} \,| \leq \rho$, and $\rho \rightarrow \underline{\mathbf{0}}$. ◊ A crisp geometric shape can be either continuous or discrete. The property of discreteness for a set of geometric vectors can be defined as follows: A set of crisp geometric vectors, V_{DC}, is said to be *discrete* if for any two vectors, $\mathbf{v}_i$, $\mathbf{v}_j \in V_{DC}$, i≠j, there exist arbitrary neighborhoods, ρ_i and ρ_j at y_i and y_j, respectively, so as $\mathbf{v}_j \notin \rho_j(\mathbf{v}_j)$ and $\mathbf{v}_j \notin \rho_i(\mathbf{v}_i)$. ◊ A *crisp discrete geometric shape* $S_{DC}(\delta, \beta)$ is a crisp discrete vector-set V_{DC}, for which (a) $\forall \mathbf{v}_i$, $\exists \mathbf{v}_j$, i≠j, $\mathbf{v}_i$, $\mathbf{v}_j \in S_{DC}$, $|\mathbf{v}_i - \mathbf{v}_j| < \delta$, where $\mathbf{v}_j$ are the closest neighbors of $\mathbf{v}_i$, and (b) for $\forall \mathbf{v}$, $\mathbf{v} \in S$, $\exists \mathbf{v}_i$, $|\mathbf{v} - \mathbf{v}_j| < \beta$. ◊ A set of vague geometric vectors, V_{DV}, is said to be *discrete*, if $V_{DV}=\{\mathbf{v}^v\}$, and V_{DV} is discrete, and $\mathrm{e}_i \cap \mathrm{e}_j = \underline{\mathbf{0}}$ ◊ A *vague discrete geometric shape* $S_{DV}(\delta, \beta)$ is a vague discrete vector-set V_{DV}, for which (a) $\forall\ \mathbf{v}^v_i$, $\exists \mathbf{v}^v_j$, i≠j, $\mathbf{v}^v_i$, $\mathbf{v}^v_j \in S_{DV}$, $|\mathbf{v}_i - \mathbf{v}_j| < \delta$, where $\mathbf{v}^{\mathbf{v}}_j$ are the close neighbors of $\mathbf{v}^v_i$, and (b) for $\forall \mathbf{v}$, $\mathbf{v} \in S$, $\exists \mathbf{v}^v_i$, $|\mathbf{v} - \mathbf{v}_j| < \beta$. ◊

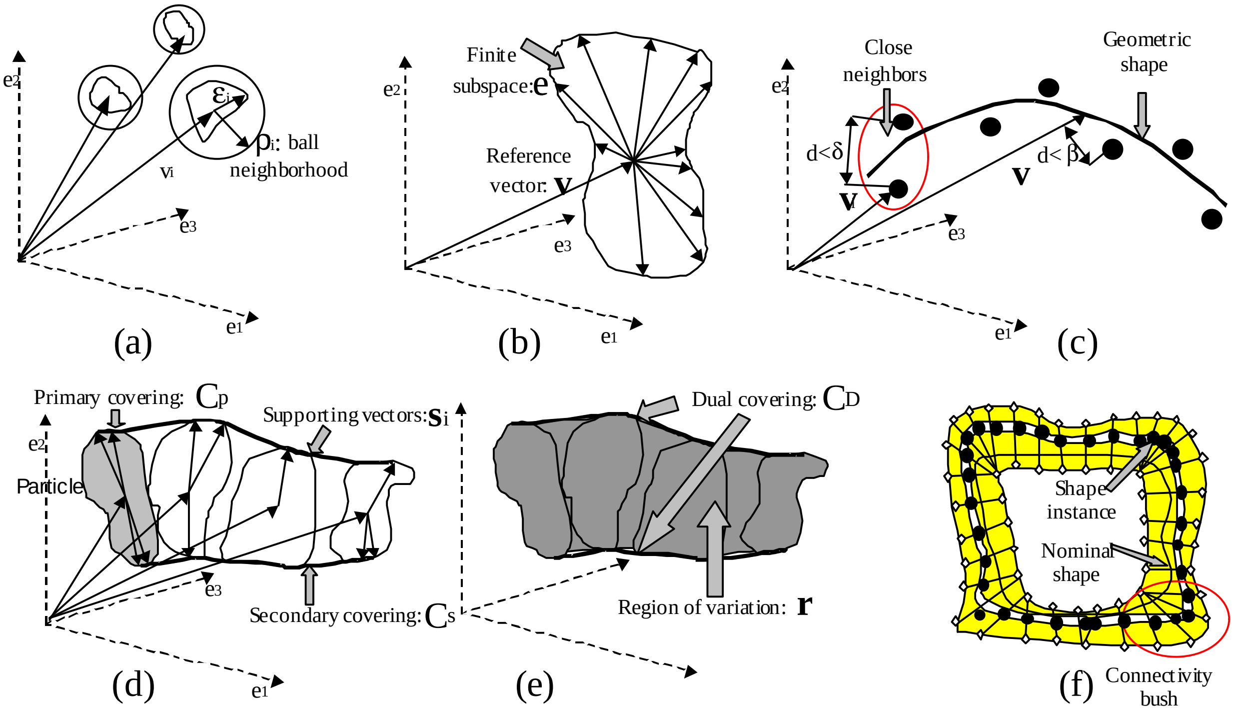

Figure 1: Geometric modelling entities for discrete domain modeling

Our aim is not only to represent the geometric properties of an object, but also to model the behavior of the object. Therefore, we have to assign materialistic properties to the vague vector, and adopt a notion, called particle, defined as follows: A *particle* π is a triplet $\pi = (\mathbf{v}^v, \mu, \mathbf{v}^e)$, where $\mathbf{v}^v$ vague vector specifies the geometry of the particle**,** μ is its mass, and $\mathbf{v}^e$ is its velocity. ◊

In order to describe a domain represented by a set of particles, the boundaries are defined by the interpolation of the particles with a surface(s). *Primary covering* C_P of a finite set of particles $\{\pi_1\ldots\pi_n\}$ is a fictitious surface ϕ_P, which, for all close neighbor particles $\{\pi_i\ldots\pi_j\}\subset\{\pi_1\ldots\pi_n\}$ interpolates the metric occurrences $\{\mathbf{e}_i\ldots\mathbf{e}_j\}$, through the positioning points z of the supporting vectors, $V_P\{\mathbf{v}_1..\mathbf{v}_n\}$, $V_P\subset\{\mathbf{e}_i\ldots\mathbf{e}_j\}$, in such a way that the vectors touch ϕ_P at one side only, and ϕ_P fulfills the condition of having locally minimum strain energy. ◊ Secondary covering of a set of particles is defined as follows: *Secondary covering* C_S of a finite set of particles $\{\pi_1\ldots\pi_n\}$ is a fictitious surface ϕ_S, which, for all close neighbor particles interpolates $\{\pi_i\ldots\pi_j\}\subset\{\pi_1\ldots\pi_n\}$ the metric occurrences $\{\mathbf{e}_i\ldots\mathbf{e}_j\}$, through the positioning points z of the supporting vectors $V_S\{\mathbf{v}_1..\mathbf{v}_n\}$, which are not contained by primary covering, $V_S\cap V_P=0, V_S\subset\{\mathbf{e}_i\ldots\mathbf{e}_j\}$, and ϕ_S fulfills the condition of locally minimum strain energy. ◊ The subset of vectors, whose positioning points are on the interpolating surface, are called supporting vectors. *Dual covering* C_D is the composition of exactly one primary and one secondary covering. ◊ The definition of the dual covering can be directly used to define the region of the variation of a vague shape: The *region of variation* r of a vague shape S_{DV} is the domain indicated by a dual covering. ◊ For a finite set of particles the notion of complete and incomplete coverings are defined as follows: A primary (secondary) covering of a finite set of particles C_P (C_S) is considered *complete*, and denoted by C_{PC} (C_{SC}), if C_{PC} (C_{SC}) is the closed boundary of a subspace; otherwise C_P (C_S) is called *incomplete*, and denoted by C_{PI} (C_{SI}). ◊

Our basic geometric entities called, particle cloud and particle system is defined using the notion of complete and incomplete coverings: A *particle cloud* Π is a finite set of particles, with incomplete the dual covering C_D, and in a specific way of the C_{SI} specified by materialization. ◊ A *particle system* Π_S is the union of a finite set of particle clouds $\{\Pi_1\ldots\Pi_n\}$, in such a way that the dual covering of the particle system is complete. ◊

The implementation of the discrete domain modeling necessitates the introduction of specific mathematical notions, for the reason that, in some of the applications the covering of the domain is known, and we look for the attributes of particles. In these cases, the discrete domain of a particle cloud is calculated as the superposition of a finite number of crisp discrete shapes, which describe the positioning point z of the supporting vectors for primary and secondary covering of the particle cloud. We specifies connectivity bushes among the positioning points between the primary and secondary coverings based on the smallest distance principle. The definition of the connectivity bush is the following:
Connectivity bush is a set of vectors $C_B\{\mathbf{v}_1...\mathbf{v}_n\}$, where for $\forall\mathbf{v}_i\in C_B$, $\mathbf{v}_i(w, z)$, y is a positioning point of a supporting vector for a primary (secondary) covering, z is a positioning point of a supporting vector for a secondary (primary) covering, and $w_i = w_j$, for $\forall\mathbf{v}_i$, $\mathbf{v}_j$. ◊ For each connectivity bushes an average vector can be calculated to indicate the orientation of the distribution. It is defined as follows: *Distribution orientation* of $\mathbf{e}$ of π is a unit vector $\mathbf{d}$ determined by the vector sum of all supporting vectors V of C_D. ◊

3. General methodology for discrete domain modelling

Below, we elaborate on the connections among the our basic modeling entities, and introduce the process of the discrete domain modeling. Figure 2 shows the general methodology for discrete domain modeling. It is assumed that the input of the process is one or more disordered 3D point-sets generated by various techniques e.g. hand movement detection and 3D scanning. First, crisp discrete shapes are generated from the point-sets in the course of pre-processing. Depending on the number of shapes, i.e. whether the input contains one or more, the process divides into two parts. In the case of multiple shapes, the discrete domain is circumscribed by the superimposed shapes. In the case of a single shape the scattering of the shape describes the domain. Both methods of vague domain generation result in a vague particle cloud, which can be manipulated either by domain modification operators or by Boolean operators. Furthermore, particle cloud instances can be extracted by using sampling operators. The domain modification operators are applied to increase or decrease the vague domain of the shape, while the Boolean operators to obtain compound modeling entities, called

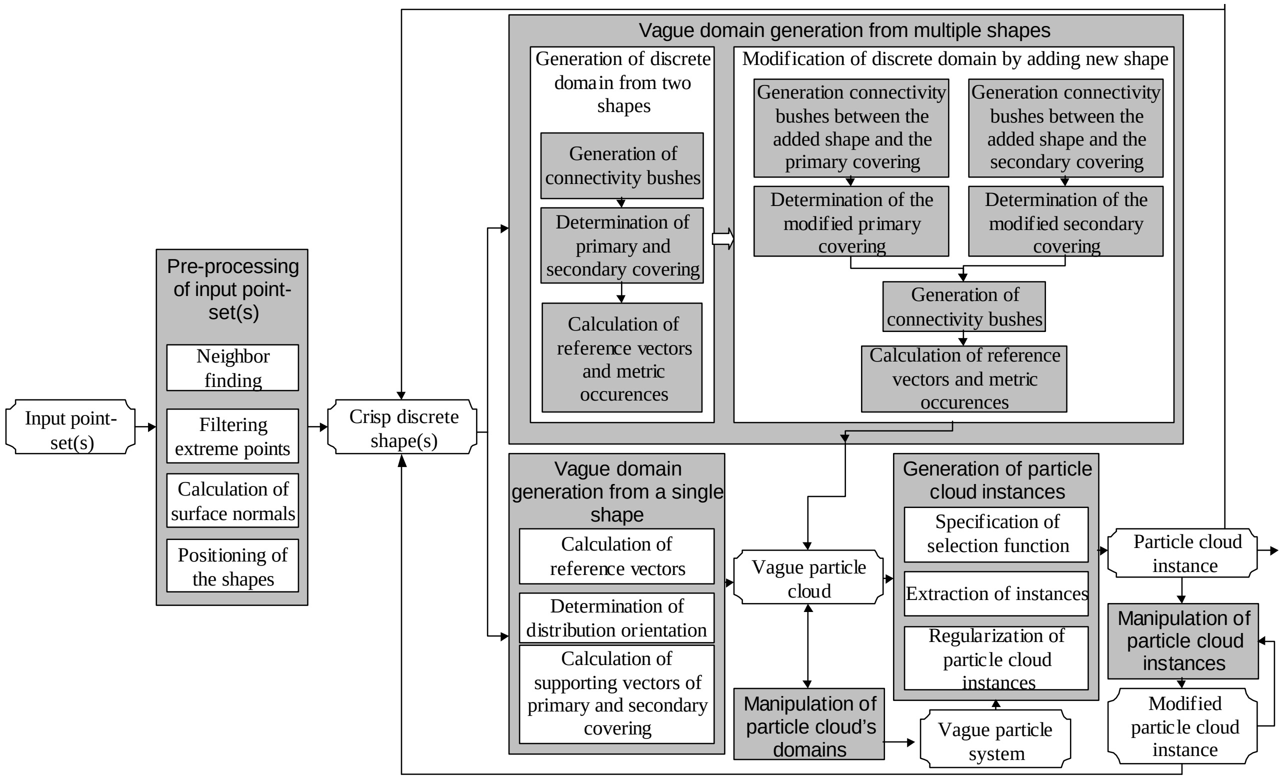

Figure 2: General methodology of discrete domain modeling

particle system. Due to the limited extend of the paper, we will not discuss these operators. Different type of shape extraction methods can be applied on vague particle cloud to specify the shape of an artifact more precisely. On the extracted instances, we can apply behavior simulation, physical deformation using shape modification operators. The output of these methods is a modified shape instance, which can be again used as input either for discrete vague domain generation or by downstream activities.

Preprocessing of input point-sets

The point-sets represent discrete geometric shapes. In the course of preprocessing, disordered point-sets are converted to a crisp discrete shape (from now on, it is called shape) through four main steps.

- The close neighbors are described using BSP (Binary Space Partitioning), as it was proposed by Rusak [4].
- The measurement errors are filtered out based on the change of the distance among the close neighbors.
- The surface normals can be calculated either by using triangulation methods or based on the relative location of the close neighbor particles.
- The point-sets are positioned in the same coordinate system in order to be able to generate the vague domain from them.

Discrete domain generation from multiple shapes

The discrete domain generation from multiple shapes consist of two main phases. First, it is generated from two shapes, afterwards it is modified by adding new shape(s) to the domain. Figure 3 shows the difference between the two methods. In the course of generation of discrete domain from two shapes, first, two shapes are selected having the largest and smallest bounding boxes, and connectivity bushes are specified among their vectors. In order to be able to generate connectivity bushes, we have to find the closest neighbor of each vector on the other shape. This method is O[$2\cdot n\cdot m$] algorithm, where n and m are the number of the vectors of the two shapes, unless, we apply BSP [1], which can reduce the algorithm to O[$n\cdot logm+m\cdot logn$]. The connectivity bushes are generated between the closest neighbors. However, in some cases meaningless connectivity can appear, when there is some intersection between the two shapes (see Figure 3).

The connectivity is called meaningless, if vectors of two primary- or two secondary coverings are connected. To find the *meaningless connectivity* we examine the following condition on all connectivity bushes: *sign* ($\mathbf{n_1}^x\mathbf{v_{12}}$) ≠ *sign* ($\mathbf{n_2}^x\mathbf{v_{12}}$). If this condition is fulfilled, the next closest point is assigned as connectivity bush to these problematic vectors. In order to assure the condition of discrete vague modeling that there are no intersections between the metric occurrences, only n:1, 1:1, and 1:n mapping are allowed in one connectivity bush. Therefore, we have to filter out the connected roots as it is showed in Figure 3.

The primary and secondary covering of the domain is described based on the connectivity bushes using the surface normal vectors ($\mathbf{n_1}\ldots\mathbf{n_m}$) in the given input vectors. The input vector $\mathbf{v_1}$ is on the primary covering if $\mathbf{v_{12}= v_2\text{-}v_1}$ and $\mathbf{n_1}^x\mathbf{v_1<0}$, else $\mathbf{v_1}$ is on the secondary covering and $\mathbf{v_2}$ is on the secondary and primary covering, respectively. The previous equations are interpreted as follows: (a) $\mathbf{v_{12}}$ is the connectivity bush (in vector form) between $\mathbf{v_1}$ and $\mathbf{v_2}$ vectors, where $\mathbf{v_1}$ and $\mathbf{v_2}$ appertain to different shapes, (b) $\mathbf{n_1}$ is surface normal vector at $\mathbf{v_1}$.

In the next step, the reference vectors and the metric occurrence of the particles are calculated based on the generated connectivity bushes. First, the distribution orientation is specified as the normalized vector-sum of the connectivity bush in order to describe the reference vector $\mathbf{v}$(x, y) of the particle π. The positioning point of the reference vector y, can be placed along the distribution orientation within the domain. In order to represent the set of reference vectors the nominal shape of the discrete domain, the reference vector is calculated as the weighted average of the vectors of the shapes determining the domain. Finally, the supporting vectors are specified as $\mathbf{s}_i$(y, z_i) for both the primary and secondary covering of the particle cloud.

In the second phase, the discrete domain is modified by adding new shape(s) to the domain. This method originates from the previous one in such a way that it generates connectivity bushes between the added shape and the primary covering and between the added shape and the secondary covering separately. In the course of determining the modified primary covering, it compares the original primary covering with the added shape. If an added vector appears on the regenerated primary covering, then the original supporting vector of the primary covering is deleted, otherwise the added vector is removed from the shape. This method is applied respectively for the secondary covering. The result of the method is two shapes, which describes the modified primary and secondary coverings of the discrete domain. In the next step, connectivity bushes are generated between the two shapes. Finally, the reference vectors and metric occurrences of the particles are calculated using the same method as it was applied in the case of discrete domain generation from two shapes.

Vague domain generation from a single shape

In the case of a single shape, the vagueness of the model is calculated from the scattering, which is caused by measurement errors. First, the reference point of the particles is specified using overturning average algorithm, as it was proposed in Horváth's paper [5]. He substituted n vectors {$\mathbf{i}_1\ldots\mathbf{i}_n$} of a shape by one particle. Then the distribution orientation of the particles is specified by the calculated surface normal on the triangulated model of the shape. The supporting vector for the primary covering is determined from those sub-set vectors {$\mathbf{i}_i\ldots\mathbf{i}_j$}⊆{$\mathbf{i}_1\ldots\mathbf{i}_n$}, for which $\forall\mathbf{i}_k\in${$\mathbf{i}_i\ldots\mathbf{i}_j$}, $\mathbf{i}_k\cdot$ $\mathbf{d}$>0 property is fulfilled. The supporting vector of the primary covering is

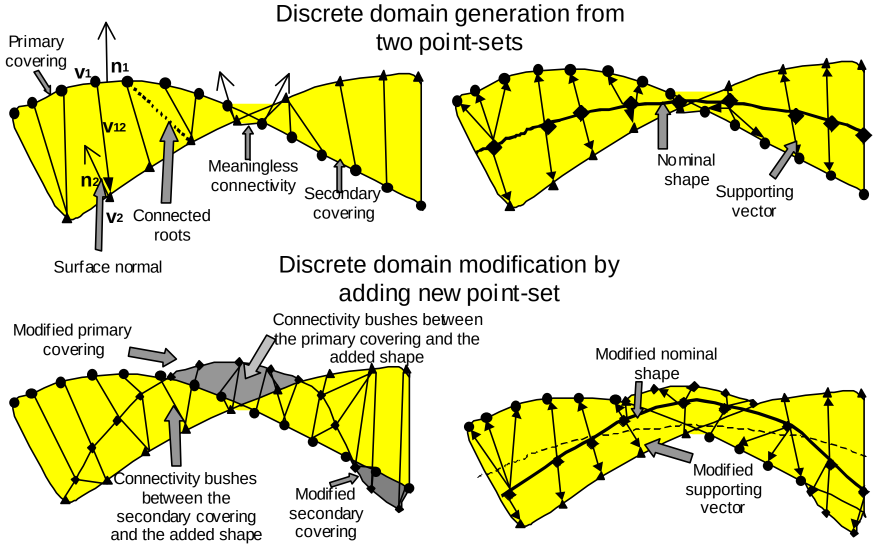

Figure 3: Vague domain generation from multiple point-sets

$\mathbf{s}_p = \mathbf{d} \cdot \max_{l=0}^{l=n}\left[\frac{|\mathbf{v} - \mathbf{i}_l| \cdot \mathbf{d}}{\mathbf{d}^2}\right]$. The supporting vector of the secondary covering can be calculated respectively using $-\mathbf{d}$ instead of $\mathbf{d}$.

Discrete shape generation from vague domains

We distinguish two types of shape extraction methods for specifying shapes from a discrete domain. The instancing operator specifies a real object, where the engineering tolerances are taken into account with ε_0 metric occurrence. The other operator, called sampling, describes the model of a nominal shape, reducing the metric occurrence to zero. Both methods have the same structure, only the shape extraction methods differ. First, the user specifies a selection function, based on which the shape samples are extracted from the vague domain. Depending on the goal of the user, sampling or instancing operator is used for shape extraction. Due to the fact that the more than one supporting vector can start from the same origin, the resulted shape instance can have irregularity nearby to the origin. Therefore, regularization is executed on all generated instances. At the end of the process, the user can select the most appropriate shape(s) for further elaboration of the model.

4. Application example

In this section, we present an application example for generating discrete vague domain from multiple shapes, and sample extraction from discrete domain. Figure 4 shows the result of discrete domain generation from two shapes, and three generated instances. In order to visualize the shape of the discrete domain model more expressively, a triangulated model of the primary and secondary covering are generated. Furthermore, half transparency is given to the primary covering (Figure 4a). Figure 4b-d show the generated instances based on the following selection functions: $\mathbf{p}_k = \mathbf{v}_k + a * \mathbf{s}_k$, where $\mathbf{p}_k$ is the local vector of the *k*-th point in the instance, $\mathbf{v}_k$ is the *k*-th is the reference vector of the particle, *a* is a scalar number, and $\mathbf{s}_k$ is the supporting vectors of the *k*-th point. Only the *a* parameter is changed in the course of sample generation, which has the *a= 0.9* in the first, *a= 0* second, *a= -0.9* third case as it is showed in figure b-d, respectively. Using this simple selection function we can see that quite a large variance of shape samples can be generated when the shape of the closure is quite different.

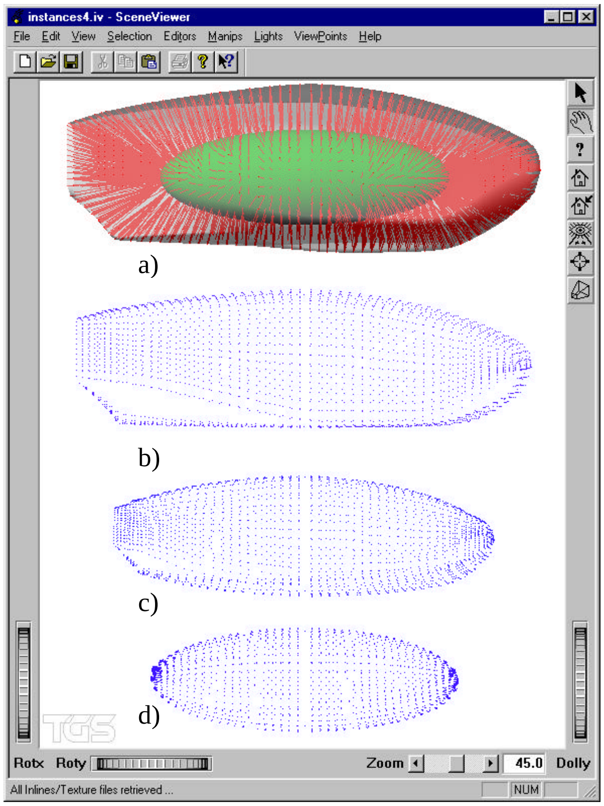

Figure 4: Instance generation from discrete vague domain

References

1. Yamaguchi, Y., Nakamura, H., Kimura, F., (1992), Probabilistic solid modeling: a new approach for handling uncertain shapes, Geometric Modeling for Product Realization, 95-108.
2. Blinn, J.F., (1982), A generalization of algebraic surface drawing, ACM Transactions on Graphics, 235-256
3. Reeves, W. T., (1983), Particle Systems-A technique for modeling a class of fuzzy objects, Computer Graphics (SIGGRAPH ‘83 Proceedings), 17, 359-376.
4. Rusák, Z., Horváth, I., Vergeest, J. S. M., Kuczogi, Gy., Jansson, J. (2000), First results of the development of particle system modeling for shape conceptualization, INTERNATIONAL DESIGN CONFERENCE - DESIGN 2000, 339-347
5. Horváth I., Rusák Z., Vergeest J. S. M., Kuczogi, G., (2000), Vague Modeling for Conceptual Design, TMCE 2000 conference, 131-144