Techniques for generating shape instances from domain distributed vague models

Zoltán Rusák
Imre Horváth
György Kuczogi
Joris S.M. Vergeest
Faculty of Design, Engineering and Production
Delft University of Technology
Jaffalaan 9, 2628 BX, Delft,
The Netherlands
TELEPHONE: +31 15 278 3788
FAX: +31 15 278 1839
E-MAIL: z.rusak@io.tudelft.nl

ABSTRACT

We are developing and implementing a novel method, called vague shape modeling, which is able to model geometric uncertainty by describing distribution domain of the shapes. In order to obtain shape instances, we have implemented specific instance generation operators that extract a set of boundary particles from the distribution domain based on predefined selection functions. In this paper, we present three instance generation methods: (a) monolithic instancing by a single selection function, which is applied to a single particle cloud (b) monolithic instancing by multiple selection functions, which applies different selection functions on different parts of a single particle cloud, and (c) constrained instancing by multiple selection functions, which uses various selection functions on each particle cloud contained by a particle system. In the first case, the application of single selection function naturally assures the required order of discrete continuity. In the second case, a fuzzy technique is applied on the selection functions to achieve the required continuity condition. Finally, for the third case, we developed a technique, called constrained instancing, which uses correction functions in a transition strip specified in between the selected instances.

1. INTRODUCTION

Vague shape modeling has been proposed to support computer internal representation of a cluster or family of shapes without conventional geometric parameterization. This approach enables us to represent the distribution domain of vague shapes between a maximal and a minimal set of discrete entities, called particles. A particle is a weakly defined 3D point specified by its reference vector, and metric occurrence. The reference vector defines the position of the particle in a vector space, and the metric occurrence, which is a finite vector space, represents the geometric uncertainty of this position. A finite set of particles, called particle cloud, forms a discrete, vague shape. The particle system, which is combined from a finite set of particle cloud, is the highest ordered modeling entity in our geometric modeling method. In the space of the metric occurrence, so-called supporting vectors are identified that stretch a minimal and maximal boundary for the vague particle cloud/system. The directions, in which the significance distributions of from a design point of view is described, are represented by the distribution trajectories of the particles. Since each particle is contained by only one vector space, the vague shape modeling technique shows a significant difference with fuzzy modeling, which is discussed in specific papers [1, 2].

We have developed a specific shape generation method, called instancing, to extract parametric shapes from domain distribution. This technique can overcome a shortcoming of the conventional parameterization techniques, since it does not require a strict system of parameters and provides knowledge intensive support of manipulation of geometric entities. Instancing applies knowledge-driven selection functions to specify the position of particles along the distribution trajectories based on characteristic shape attributes. Application of a single selection function on a single cloud, called *monolithic instancing* by single selection function, can naturally assure the required order of discrete continuity defined in [3]. We interpret the discrete continuity as the continuity of the best matching continuos curve in the considered finite region.

In order to be able to apply multiple functions on a single particle cloud, we have developed a method, called *compound instancing*, which uses fuzzy technique in specification of multiple selection functions to maintain the required discrete continuity condition. The third method, called *constrained instancing*, applies multiple selection functions on a particle system, which can result in disconnected components of instance shapes. To obtain the required continuity: (a) a transition strip is specified between the shapes, (b) extra particles are generated in the gaps, and (c) the particles of the transition strip are manipulated. The further parts of the paper discuss the newly introduced instance generation methods in detail.

2. METHODS FOR INSTANCE GENERATION

2.1. Monolithic instance generation using single selection functions

In the simplest case, an instance shape can be generated in the distribution domain of the archetype by one selection function. The mathematical form of the selection function is derived based on shape formation rules of a particular application (e.g. aesthetics, ergonomics, functionality). The selection function considers the boundaries of the distribution domain and specifies the set of particles along the distribution trajectories. Figure 1 illustrates the variables used in the instance generation.

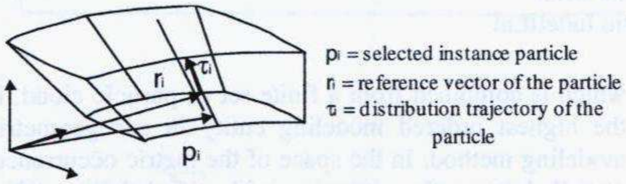

Figure 1: Variables in instance selection

To be able to calculate the selection functions, we parameterize the distribution trajectories of each particle in an interval α[−1; 1]. In this manner, all the selection function returns with α, and the position $\mathbf{p}_i$ of the particle is computed as follows:

$$\mathbf{p}_i = \mathbf{r}_i + \alpha \cdot \boldsymbol{\tau}_i \qquad (1)$$

The condition is that the instance function has to be continuos in third order at least. It means, sufficiently smooth functions can only be considered in this case. However, in certain cases, lower level of discrete continuity can also be accepted.

2.2. Monolithic instance generation using multiple selection functions

Monolithic instance generation by multiple selection functions is a more sophisticated method, which considers region specifications of a particle cloud. It is able to apply various selection functions in the different regions. This technique, called compound instancing, interfaces the results of the individual instancing functions by applying corrective functions and achieves lower order discrete continuity between the regions this way.

To specify regions R_i on a particle cloud Π, we implemented a method, by which the user can interactively manipulate the particles of a region with user-defined, free-from virtual tools. The particles are assigned to a specific region by moving the tool in the modeling space, and checking the reference vectors of the particles, whether the tool contains them. Afterwards, particles are identified that are not contained by any regions, and assigned to the closest region.

The compound instance generation applies compensated selection functions in the regions of a particle cloud.

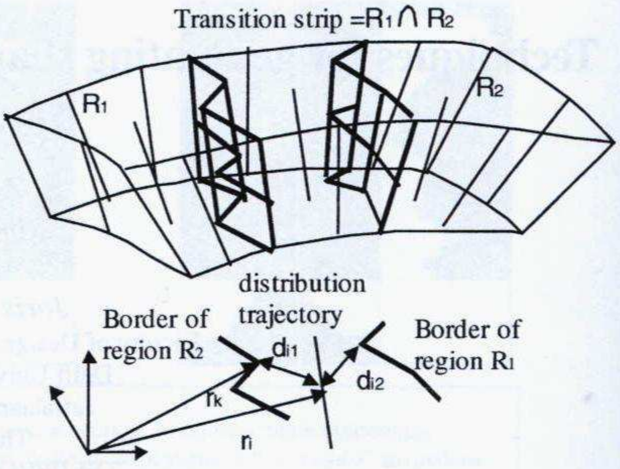

Figure 2: Specification of wieghts of the selection functions

First, the algorithm specifies a transition strip from the overlap regions described by the user. It searches for the borders of the overlap regions, and displaces them to the distance of m-th neighbor particles from each other, keeping the overlap condition. The value of m is selcted to be 8 for C^1 and 32 for C^2 Then for all particles π_i in the transition strip, distances d_{ik} are computed from the borders of the transition strip. Since we assume linearly changing fuzzy membership functions in the transition strip, the proportion of the member values w_{ik} of R_i can be computed based on d_{ik} as follows:

$$w_{ik} = \frac{d_{ik}}{\sum_{k=1}^{N} d_{ik}} \qquad (2)$$

Figure 2 shows the example of specification of the member values in the transition strip. Afterwards, the position $\mathbf{p}_i$ of the instance particles can be calculated as follows:

$$\mathbf{p}_i = \mathbf{r}_i + \boldsymbol{\tau}_i \frac{\sum_{k=1}^{N} w_{ik}\alpha_k}{\sum_{k=1}^{N} w_{ik}} \qquad (3)$$

The result of this method can be further elaborated either by increasing the size of the transition strip or by changing the linear membership function.

2.3. Constrained instance generation on a particle system

The constrained instance generation method applies various selection functions on different particle clouds of a particle system. It can result in a set of disconnected surface patches. Therefore, the concerned instancing functions have to be appended by additional correction functions. The correction functions modify the position of the particles in the transition strip, which is defined based on the required continuity conditions. Figure 3 illustrates a pair of disconnected surface patches and the transition strip, whose geometric smoothness needs improvement.

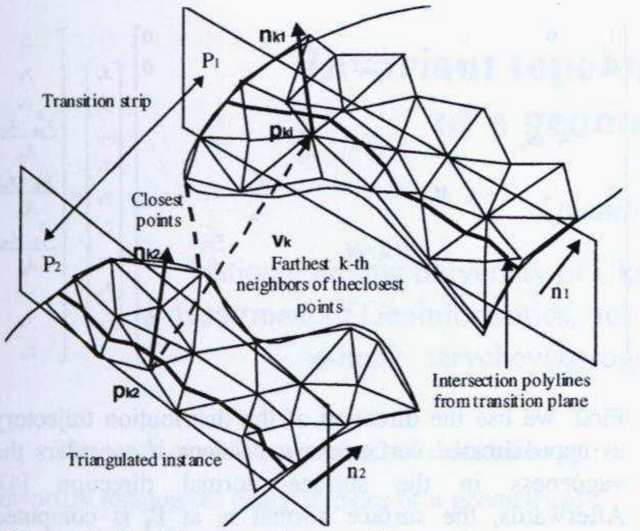

Figure 3: Specification of a transition strip

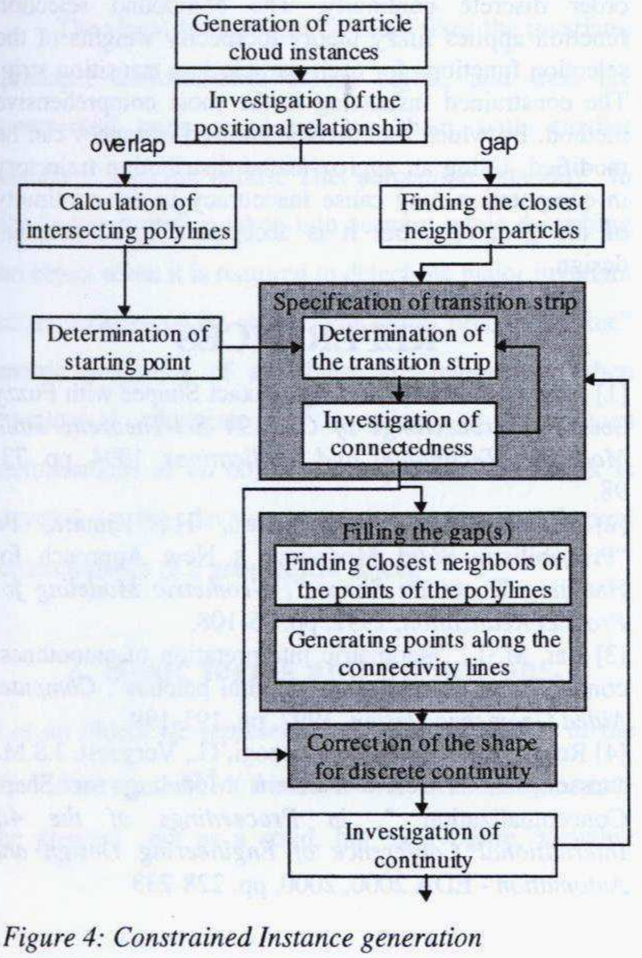

Figure 4: Constrained Instance generation

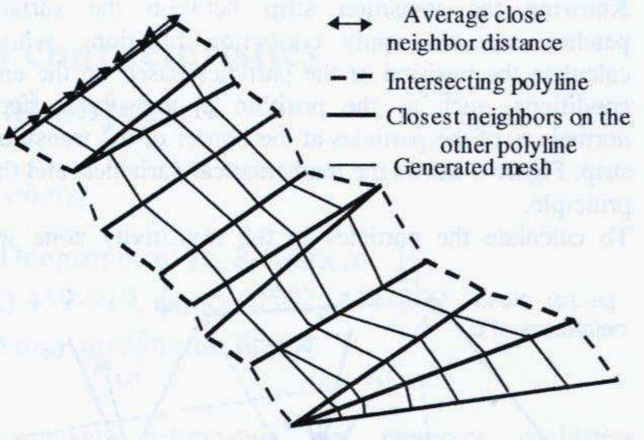

Figure 5: Particle generation in the transition strip

To generate a smooth transition between disconnected or intersecting shape instances, we propose a method that we call constrained instancing. Figure 4 presents the process of constrained instancing. In the first step, selection functions are applied to the particle clouds that extract shape instances from the domain distribution. Next, the relationship between the particle clouds is investigated based on intersection detection between the triangulated models of the instances. When the shapes are intersecting, the intersecting polylines are calculated, and the extra particles are removed from the model. Afterwards, the starting particle of a transition strip is determined. When there is no intersection, the closest neighbor particles are determined as starting particles. To specify the transition strip, two planes P_1, and P_2 are determined, as the border of the transition strips, and the intersection polylines between the planes and the instances are calculated. Figure 3 illustrates the specification of a transition strip. The transition strip of two instances is described by planes P_1, and P_2 placed into the k-th neighbor of the starting particle(s), where the value of k is chosen to be 4 for C^1 and 16 for C^2. From all the k-d neighbors on the two instances, the farthest ones are selected as the border particles of the transition strip. The normal vectors of P_1, and P_2 are defined by the positions ($\mathbf{p}_{k1}$, $\mathbf{p}_{k2}$) of farthest k-th neighbors of the starting particle(s) as follows: $\mathbf{n}_1 = \mathbf{n}_{k1} \times \mathbf{v}_{kp}$ and $\mathbf{n}_2 = \mathbf{n}_{k2} \times \mathbf{v}_{kp}$, where $\mathbf{v}_{kp}$ is the perpendicular component of the $\mathbf{v}_k = \mathbf{p}_{k1} - \mathbf{p}_{k2}$, and $\mathbf{n}_{k1}$ and $\mathbf{n}_{k2}$ are the normal vectors at $\mathbf{p}_{k1}$ and $\mathbf{p}_{k2}$ respectively. The transition strip is investigated based on the type of the intersecting polylines. The border of the transition strip is extended, until the intersection results in one closed or one open polyline.

Since our method manipulates the position of particles included in the strip to achieve smooth connection between the shapes, it requires quasi-regular particle-set in the transition strip. Therefore, new particles are generated in the gap(s) of the transition strip in two steps. First, the closest neighbors of the particles are identified on the other polyline, and connected by so-called connectivity lines. Then, for each connecting line, the average distance of close neighbors is calculated with respect to the neighbors of the end particles of the connectivity line. Second, particles are generated along the connectivity lines dividing them into line-segments with quasi-regular length. The regular length means an approximation of the average distance between the neighboring particles. Finally, approximated surface normals are calculated in each generated particle based on the triangulated mesh. Figure 5 illustrates application of the algorithm.

Afterwards, the generated particles in the transition strip are manipulated based on the correction function. Finally, the second order discrete continuity condition of the constrained instancing is evaluated based on the Lai's method [3], and the transition strip can be partially modified at the insufficient places.

Knowing the transition strip between the surface patches, we can apply correction functions, which calculate the position of the particles based on the end conditions, such as the position $\mathbf{p}_i$ and the surface normals $\mathbf{n}_i$ of the particles at the border of the transition strip. Figure 6 shows the mathematical variables, and the principle.

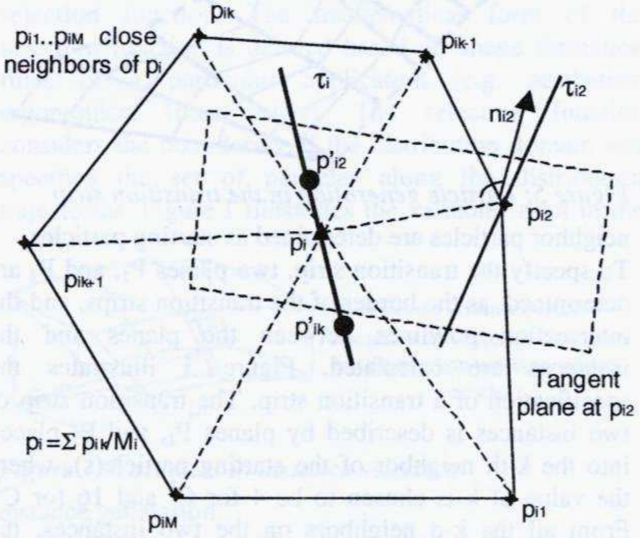

Figure 6: Principle of the correction function

To calculate the particles in the transitivity zone we applied an analytical method. The position $\mathbf{p}_i$ of particle π_i is described by the distribution trajectory τ_i at π_i, and the positions $\mathbf{p}_1..\mathbf{p}_{Mi}$ together with the normal vectors $\mathbf{n}_1..\mathbf{n}_{Mi}$ of all the neighbor particles ($\pi_1..\pi_{Mi}$) of π_i. In the first calculation, the distribution trajectory τ_i is approximated by the surface normal vector $\mathbf{n}_i$ at P_i, so as $\tau_i(\mathbf{p}_i, \mathbf{n}_i)$. From a neighbor particle $\pi_k \in (\pi_1..\pi_{Mi})$ of π_i, we compute the intersection-points $\mathbf{p'}_{ik}$ of the tangent plane at P_k and the approximated distribution trajectory τ_i of the π_i as follows:

$$\mathbf{p'}_{ik} = \frac{(\mathbf{p}_k - \mathbf{v}_i)\cdot \mathbf{n}_k}{\mathbf{n}_k^2 \cdot \left(\mathbf{n}_k \cdot \frac{\boldsymbol{\tau}_i}{|\boldsymbol{\tau}_i|} \right)} \cdot \mathbf{n}_k = \frac{(\mathbf{p}_k - \mathbf{v}_i)\cdot \mathbf{n}_k}{A_{ik}} \cdot \mathbf{n}_k \qquad (4)$$

The position $\mathbf{p}_i$ of the concerned particle is the average of all intersection-point $\mathbf{p'}_{ik}$ specified by the neighbors of a particle:

$$\mathbf{p}_i = \frac{1}{M_i} \sum_{k=1}^{k=M_i} \mathbf{p'}_{ik} \qquad (5)$$

The above equations can be applied for all the particles in the transition strip, and the result can be organized in a linear system of equations. The coefficient matrix of this linear system is a sparse matrix, which can be solved by a frontal method. The complexity of the matrix is linearly increasing with the number of elements.

$$\begin{bmatrix} 1 & 0 & \cdots & & & & & 0 \\ 0 & 1 & 0 & \cdots & & & & 0 \\ \cdots & & & & & & & \\ & -x_i \cdot M_i & & \cdots & \frac{x_{nk}^2}{A_{ik}} & & \cdots & \\ 0 & & -y_i \cdot M_i & & & \frac{y_{nk}^2}{A_{ik}} & \cdots & \\ & & & -z_i \cdot M_i & & & \frac{z_{nk}^2}{A_{ik}} & \\ & & & & & & & 0 \\ & & & & & & & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ \cdots \\ x_k \\ y_k \\ z_k \\ \cdots \\ y_n \\ z_n \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \cdots \\ \frac{x_{nk}^2 \cdot x_{vi}}{A_{ik}} \\ \frac{y_{nk}^2 \cdot y_{vi}}{A_{ik}} \\ \frac{z_{nk}^2 \cdot z_{vi}}{A_{ik}} \\ \cdots \\ y_n \\ z_n \end{bmatrix}$$

First, we use the direction of the distribution trajectory as approximated surface normal, since it considers the vagueness in the surface normal direction [4]. Afterwards, the surface normal $\mathbf{n}_i$ at P_i is computed based on triangulation over the set of the generated instance particles. The calculation of correction function is repeated with the new surface normals, until the result fulfils the required discrete continuity conditions.

3. CONCLUSIONS

In this paper, we presented three methods of shape instance generation techniques. The techniques have been developed for selection of instances from domain distributed discrete models. Since the proposed techniques working by analytic manner, they provide real time calculation of shape instances. The monolithic instance generation by a single selection function naturally generates particle cloud instances with second order discrete continuity. The compound selection function applies fuzzy theory to specify weights of the selection functions for each particle in a transition strip. The constrained instancing is the most comprehensive method, by which the already selected instances can be modified. Using an approximated distribution trajectory in computation may cause inaccuracy in the continuity of the geometry, but it is acceptable for conceptual design.

REFERENCES

[1] Martin, R. R., “Modeling Inexact Shapes with Fuzzy Sets”, *in Proceedings of CSG 94 Set-Theoretic Solid Modeling: Techniques and Applications*, 1994, pp. 73-98.

[2] Yamaguchi, Y., Nakamura, H., Kimura, F., “Probabilistic Solid Modeling: a New Approach for Handling Uncertain Shapes”, *Geometric Modeling for Product Realization*, 1992, pp. 95-108.

[3] Lai, M. J., “Geometric interpretation of smoothness conditions of triangular polynomial patches”, *Computer Aided Geometric Design*, 1997, pp. 191-199

[4] Rusák, Z., Horváth, I., Kuczogi, G., Vergeest, J.S.M., Jansson, J.," Discrete Domain Modeling for Shape Conceptualization ", *in Proceedings of the 4th International Conference of Engineering Design and Automation* - EDA 2000, 2000, pp. 228-233