GENETIC ALGORITHMS FOR GIELIS SURFACE RECOVERY FROM 3D DATA SETS

Y. Bokhabrine, Y. D. Fougerolle, S. Foufou, and F. Truchetet

Le2i Lab. UMR CNRS 5158, Université de Bourgogne, FRANCE

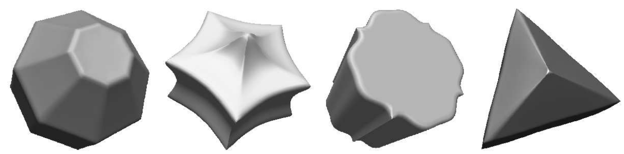
ABSTRACT

In this paper, we apply genetic algorithms to reconstruct Gielis surfaces from 3D data sets. The Levenberg-Marquardt method has been used as a standard for superquadrics recovery and has recently been extended to Gielis surfaces. Unfortunately, the non homogeneity of the Gielis surface parameters requires additional heuristic to determine discrete parameters such as the number of symmetries. Genetic algorithms overcome this issue and provide a more general framework for Gielis surface reconstruction.

Index Terms— Geometric modeling, surface reconstruction, Gielis surfaces, superquadrics, supershapes, genetic algorithms

1. INTRODUCTION

Several efficient deterministic methods for superquadrics recovery have been proposed in the past two decades [1–4]. A recent extension with rational and irrational degrees of symmetry, namely the supershapes or Gielis surfaces, has been proposed in [5]. Compared to previous work in [6], where we extended deterministic superquadric reconstruction methods to Gielis surfaces, the focus of this paper is a more detailed study of reconstruction techniques for individual supershapes using genetic algorithms. The work presented in this paper belongs to a larger framework of reconstruction of complex surfaces represented as Boolean operations between globally deformed supershapes. Such objects can be represented by potential functions that are built using R-functions [7]. The minimization of such functions using deterministic methods raises several difficulties, such as segmentation, symmetry detection, and heuristic choices. The original contributions of this paper are an extension of existing supershapes recovery techniques using genetic algorithms that handle the detection of the symmetry number similarly to the other parameters in a unified framework. The structure of the paper is as follows: Section 2 presents Gielis surfaces and an associated cost function to be minimized, Section 3 is dedicated to the Levenberg-Marquardt algorithm and genetic algorithms. We present and discuss our results in Section 4. Section 5 presents our conclusions and some ideas for future extensions of this work.



(a) (b) (c) (d)

Fig. 1. Supershapes with various shape coefficients and symmetry numbers.

2. GIELIS SURFACES AND COST FUNCTION

Gielis defines the parametric radius of a superpolygon as:

$$r(\theta) = \frac{1}{\sqrt[n_1]{\left|\frac{1}{a}\cos\left(\frac{m\theta}{4}\right)\right|^{n_2} + \left|\frac{1}{b}\sin\left(\frac{m\theta}{4}\right)\right|^{n_3}}}, \qquad (1)$$

with a, b, and $n_i \in \mathbb{R}^+$. Figure 1 presents examples of supershapes for various shape coefficients.

For a unit supershape ($a = b = 1$) and m and M natural numbers, we have defined a radial implicit function as:

$$f(x, y, z) = 1 - \frac{1}{r_2(\phi)} \sqrt{\frac{x^2 + y^2 + z^2}{\cos^2\phi\,(r_1^2(\theta) - 1) + 1}}, \qquad (2)$$

The problem of 3D supershape recovery can be seen as a minimization problem, *i.e.* the determination of the parameters set, noted $\Lambda = \{t, r, s, n_i, m, N_i, M\}$, $i = 1, 2, 3$, which minimizes an error of fit noted $EOF(\Lambda)$ for a given data set. Parameters t, r, and s correspond respectively to the translation, the rotation, and the scale parameters. Parameters n_i, N_i, $i = 1, 2, 3$, m, and M correspond to the parameters of the recovered supershape. Following the conclusions presented in [1–4], we defined in [6] a cost function to measure the fitting error:

$$EOF(\Lambda) = s_x s_y s_z \sum_{i=1}^{n} (F(P_i))^2 . \qquad (3)$$

The function $F(P)$ corresponds to the supershape function proposed in equation 4 in the general case of a translated, rotated, and scaled supershape:

$$F(x,y,z) = f\left(\left(T \circ R \circ S\right)^{-1}(x,y,z)\right) , \qquad (4)$$

with T, R, and S being the transformation matrices that correspond to the translation t, the rotation r and the scale s.

3. RECONSTRUCTION ALGORITHMS

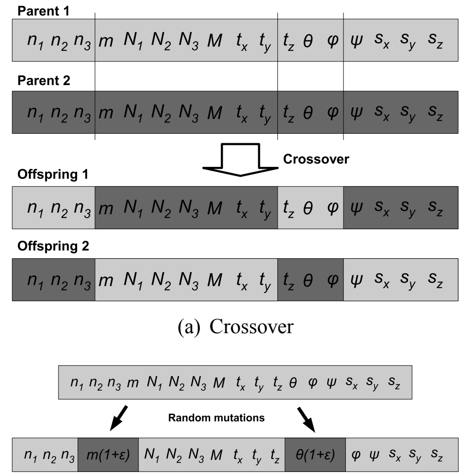
3.1. Levenberg-Marquardt algorithm

Levenberg-Marquardt (LM) method has been used as a standard for superquadrics recovery [1–4], please refer to [8] for explanations concerning its implementation. During an iterative process, the recovered supershape evolves to obtain a minimal value of the suitable cost function. At each iteration, a new step in the parameter space is computed and requires the computation of all the partial derivatives of the cost function $EOF(\Lambda)$, which raises three main issues. First, Gielis surfaces are not differentiable everywhere because of the absolute values in the parametric definition of the radius. Fortunately, this problem arises only when points have specific angular coordinates in the supershape referential, which rarely happens in practice. For such points, the partial derivatives of the cost function cannot be computed and are set to zero, which is equivalent to ignore these points in the reconstruction process. Second, the equation 2 requires the symmetry parameters m and M to be natural numbers. Therefore, equation 3 is not differentiable regarding parameters m and M, which means symmetries have to be handled separately from the other parameters. In previous work, we used a heuristic based on sweeping around the supershape inertia axes to efficiently determine symmetry numbers. When symmetries are not correctly detected, symmetries are set to 4 and our algorithm may converge to a local minimum with an important error of fit. The third issue concerns the non unicity of the parameters for a same shape, which leads to difficulties to assert the direction of each step of the algorithm and its convergency to numerous local minima.

3.2. Genetic Algorithms

Genetic algorithms (GAs) are stochastic search methods that have been successfully applied in many optimization problems. The fundamental principles of GAs were first presented by Holland [9]. A GA randomly generates an initial population. Elements of the population are coded as a string of symbols, known as genes. A chromosome is composed of genes, and represents a solution to the optimization problem.

At each iteration of the algorithm, called a generation, the chromosomes in the current population are evaluated using the measure of fitness that corresponds in our case to the function $EOF(\Lambda)$ defined in equation 3. Genetic operators control the evolution of successive generations. At



(a) Crossover

(b) Mutation

Fig. 2. Illustration of the genetic operators Crossover and Mutation.

each generation, the best solution corresponds to the chromosome with the smallest error of fit. The two basic genetic operators are crossover and mutation: the crossover operates on two individuals at a time and generates two offsprings by combining random genes from both parents. The mutation is a background operator, which produces spontaneous random changes in various individuals. Figure 2 illustrates the behaviour of both genetic operators. A genetic algorithm proceeds as follows:

1. Create initial random generation
2. For each individual, evaluate its fitness applying Eq. 3
3. Create next generation using crossover and mutation
4. Loop until a convergency test is not satisfied

4. RESULTS AND DISCUSSION

Results presented in this paper on figures 3, 4, and 5 have been generated using a population of 500 chromosomes. In most cases, the algorithm converges after 200 generations, which takes between 5 to 10 minutes and requires a memory of 25Mb.

To verify that the GA converges to the optimal solution, we applied it to several types of supershapes, similarly to the LM method presented in [6]. We verify its convergency for synthetic unit supershapes, as illustrated in Figure 3. We then extended the dimension of the parameters space with scales, translations and rotations, as illustrated in Figure 4. Results for incomplete data are presented in Figure 5.

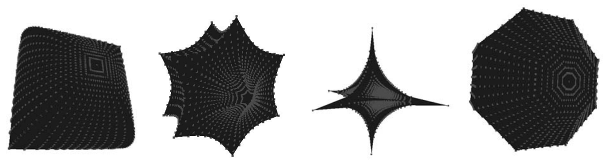
We verify that GAs can be used to recover Gielis surface parameters. Naturally, their convergency is slower than Levenberg Marquardt algorithm and they requires more memory

figure	n_1	n_2	n_3	N_1	N_2	N_3	$EOF(\Lambda)$
3(a)	50/1895	50/1845	50/1886	2/1918	2/3	2/16	0.005
3(b)	100/76.2	100/76.2	100/76.2	100/76.2	100/76.2	100/76.2	0.0007
3(c)	0.5/0.508	0.5/0.508	0.5/0.508	0.5/0.508	0.5/0.508	0.5/0.508	2.017
3(d)	1000/1086	250/268	250/288	1000/360	250/90	250/90	0.0008

Table 1. Recovered shape parameters for objects in Figure 3. Notation: Initial / recovered values.

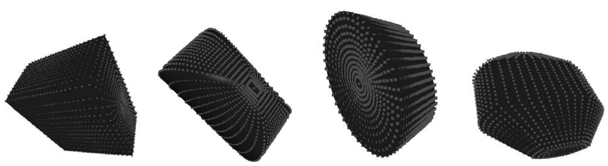
figure	n_1	n_2	n_3	N_1	N_2	N_3	$EOF(\Lambda)$
4(a)	50/56.6	50/55	50/57.6	50/56.6	50/56.6	50/55	0.2
4(b)	50/170	50/162	50/171	2/1679	2/1868	2/3.6	0.08
4(c)	1000/1397	250/366	250/284	1000/638	250/134	250/132	0.68
4(d)	2/1215	2/7	2/34	50/143	50/143	50/141	0.26

Table 2. Recovered shape parameters for objects in Figure 4. Notation: Initial / recovered values.



(a) (b) (c) (d)

Fig. 3. Reconstruction of canonical supershapes

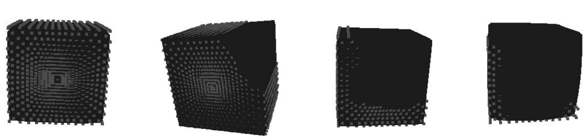


(a) (b) (c) (d)

Fig. 4. Reconstruction of translated, rotated and scaled supershapes

to handle the population of several supershapes. Nevertheless, using GA avoids the fundamental problem of the symmetry detection and allows to handle all the parameters in a same manner.

Compared to the work proposed in [6], we observe that the results obtained by GA and LM algorithms share some similar properties. Some results illustrate that a supershape can be represented by completely different parameters sets, as illustrated by the results of Figure 5(a). Orientation parameters can also be different. Actually, it corresponds to several combinations of rotations of π or $\pi/2$, which leads to important changes in the shape parameters. It can also be seen as a permutation between the two generating superpolygons, which is equivalent to axes permutation. Numerical values of the recovered parameters corresponding to the results shown on Figures 3 and 4 are summed up in tables 1, 2, and 3, respectively. The data in Figure 5 have been generated for a unit cube and have been degraded to represent an incomplete cube. Once again, we observe that the cube is correctly reconstructed and the computed shape coefficients are different from original ones as mentioned in table 4. In terms of computational complexity, our GA is slower than the LM approach used in [6]. It is due to the very nature of this algorithm, because LM is a deterministic algorithm while GA is stochastic. At each iteration, the LM method requires one evaluation of the cost function and its gradient, whereas GA requires as many evaluations as the total number of chromosomes in the considered population. Considering a population of n chromosomes is therefore approximately n times slower than using LM method. Nevertheless GA can handle cases the LM method cannot such as variations of the symmetries.



(a) 100% of the data (b) 80% (c) 40% (d) 25%

Fig. 5. Reconstruction of incomplete data

5. CONCLUSIONS AND FUTURE WORK

Exploration of the parameters space for the reconstruction of Gielis surfaces is still a challenging and open problem. Using deterministic algorithm can be an efficient solution if symmetries can easily be determined. Nevertheless, using GAs has a major advantage over LM method since all the parameters, such as the shape, the position, and the symmetries, are handled in a unified framework.

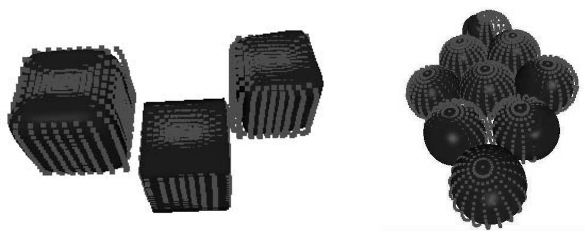
It is also important to keep in mind that real world objects may not be exactly representable by Gielis surfaces, which leads to consider a larger representation model using Boolean operations and global deformations. In such a global model,

figure	t_x	t_y	t_z	ϕ	θ	ψ	s_x	s_y	s_z	$EOF(\Lambda)$
4(a)	1/1	4/4	-1/-0.9	1/1	-1/2	-3/3	6/6	3/3	5/5	0.2
4(b)	1/1	4/4	-1/-1	1/1	1/2	-3/0.1	6/6	3/3	5/5	0.08
4(c)	1/1	4/4	-1/-0.9	1/1	-1/2	-3/6	6/6	3/3	5/5	0.68
4(d)	1/1	4/4	-1/-0.9	1/1	-1/2	-3/3	6/6	3/3	5/5	0.26

Table 3. Recovered translation, rotation, and scale parameters for objects in Figure 4. Notation: Initial / recovered values.

figure	n_1	n_2	n_3	N_1	N_2	N_2	$EOF(\Lambda)$
5(a)	1000/33.95	1000/33.95	1000/33.95	1000/473.37	1000/851.17	1000/383.47	0.23
5(b)	1000/1433.95	1000/707.6	1000/1468.6	1000/800	1000/900	1000/800	0.04
5(c)	1000/41.48	1000/41.48	1000/41.48	1000/800	1000/900	1000/789.42	0.16
5(d)	1000/1557	1000/1755.09	1000/1083	1000/1596.96	1000/1407.49	1000/1521	0.19

Table 4. Recovered supershape parameters for the incomplete data of Figure 5. Notation: Initial / recovered values.



(a) Reconstruction of 3 cubes (b) Reconstruction of 9 spheres

Fig. 6. Reconstruction of several supershapes using genetic algorithm

the dimension of the parameters space becomes very important and leaves very few hope for the development of a direct, deterministic, and general reconstruction algorithm. Furthermore, the deterministic method proposed in [6] strongly relies on a correct segmentation. On the other hand, using GAs can be an alternative to obtain a acceptable initial result that could be used as an initial guess for a generalized LM method.

The extension of this work to the recovery of several Gielis surfaces is currently under consideration. R-functions are used to combine supershapes through Boolean operations. The genetic algorithm is slightly modified to handle the global optimization of a R-function of Gielis surfaces (extension of the chromosome size to represent the parameters of several supershapes, as well as the number of supershapes and the Boolean operations). Some preliminary results are shown on Figure 6.

This work belongs to the research dedicated to individual Gielis surfaces reconstruction using genetic algorithm. Several research directions are currently investigated and include the simplification of the parametric radius in equation 1 into a rational fraction to simplify and reduce the number of computations, the development of more suitable cost functions for a better discrimination between Gielis surfaces, and more generally, the adaptation and the application of the numerous optimization techniques (deterministic and stochastic) for the recovery of large, complex, noisy 3D data sets.

6. REFERENCES

[1] F. Solina and R. Bajcsy, "Recovery of parametric models from range images: The case for superquadrics with global deformations," *The Visual Computer*, vol. 12, no. 2, pp. 131–1476, 1990.

[2] A. Jaklič, A. Leonardis, and F. Solina, *Segmentation and recovery of superquadrics*, Kluwer Academic Publisher, Dordrecht, 2000.

[3] A. Gross and T. Boult, "Error of fit measures for recovering parametric solids," in *Proc. Internat. Conf. Computer Vision*, 1988, pp. 690–694.

[4] Y. Zhang, "Experimental comparison of superquadric fitting objective functions," *Pattern Recognition Letter*, vol. 24, pp. 2185–2193, 2003.

[5] J. Gielis, B. Beirinckx, and E. Bastiaens, "Superquadrics with rational and irrational symmetry," in *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications 2003*, Seattle, Washington, USA, June 2003, pp. 262–265.

[6] Y. D. Fougerolle, A. Gribok, S. Foufou, F. Truchetet, and M. A. Abidi, "Supershape recovery from 3D data sets," in *Proc. of IEEE Int. Conf. on Image Processing (ICIP)*, Atlanta, GA, USA, October 2006, pp. 2193–2196.

[7] Y. D. Fougerolle, A. Gribok, S. Foufou, F. Truchetet, and M. A. Abidi, "Boolean operations with implicit and parametric representation of primitives using R-functions," *IEEE Transactions on Visulisation and Computer Graphics*, vol. 11, no. 5, pp. 529–539, 2005.

[8] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992.

[9] John H. Holland, *Adaptation in natural and artificial systems*, MIT Press, Cambridge, MA, USA, 1992.