Animation Based in Dynamic Simulation Involving Irregular Objects with Non-Homogeneous Rugosities

Luis A. Rivera
Laboratorio de Matemática, UENF,
Av. Alberto Lamego 2000,
28015-620, Campos, Rio de Janeiro, RJ, Brasil

Departamento de Informática, PUC-Rio,
Marquês de São Vicente, 255,
22453-900, Rio de Janeiro, RJ, Brasil
rivera@inf.puc-rio.br

Paulo C. P. Carvalho
Luiz Velho
IMPA–Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110,
22460 Rio de Janeiro, RJ, Brasil
{pcezar,lvelho}@visgraf.impa.br

Abstract

We propose a new technique that provides a unified framework for modeling the collision dynamics of objects having arbitrary shape and rugosity. The technique assumes that boundary of the object is expressed by elementary segments with some random distribution of perturbation. Collision detection and treatment is done considering this conceptual representation. In order to efficiently detect collision between irregular objects, we introduce the aumented oriented bounded box tree.

1. Introduction

In order to be realistic, a simulation of the dynamic behavior of a set of rigid objects should consider all possible characteristics of the objects and their environment. One of the characteristics to be considered is the geometry of the objects. Much of the work in the area concentrates on objects defined by poligonal aproximation, or on piecewise smooth objects described by either implicit or parametric surfaces, but does not consider objects of arbitrary shape with perturbation or rugosity.

Several techniques have been proposed to model objects having arbitrary shapes, such as particle based modeling [28], variational methods [26, 27], direct manipulation [2, 3, 21], and free form deformation [4, 12], among others. All those techniques allow making objects of a desired form starting from a initial one. In this way we could represent a model that is an approximation to a real object. Usually, this approximation, however, is good only up to a certain level of detail, since the surface of real world objects is not completely smooth. Rather, it presents some irregular degree of rugosity, which can affect the dynamic behavior of the objects in animation.

Most simulation techniques treat collitions events using *friction coefficients* , that sumarize the effects of surface details and material properties in contacts [11], considering this as being constant on all object surface; these consideration is not all true, because in the real word the surface details are not allways homogeneous, for that the friction coefficient must vary in each point of the object surface, so affecting dynamic behavour of contacts.

In this work, we propose a new technique that provides a unified framework for modeling animation based in dynamic simulation with objects having arbitrary shape and rugosity. The technique consider the characteristic of surface details to treat the collision and then to produce real behavour of the objects in animation.

In Section 2 we discuss the modeling of objects having arbitrary shape and rugosity, using a multiresolution representation based on wavelets and statistical estimation techniques to deal with the finer levels of detail; in Section 3 we explain how to detect collisions between objects with rugosities; in Section 4 we formalize the dynamic aspects of such a collision. Finally, in Section 5 we conclude and discuss possible extensions of this approach.

2. Complex objects

In this paper a *complex object* is an object of arbitrary shape with irregular distribution of detail representing its rugosity. The details could be expressed by a multiresolution representation that, conceptually, considers all levels of

detail of the object, in such a way that it captures its rugosity.

We say *conceptually* because it is not practical to represent explicitly the finer levels of detail. Instead, explicit representation is given up to a certain level of detail. Beyond that level, detail is seen as a perturbation applied to the surface, according to certain statistical laws.

2.1. Surface representation

We consider only two-dimensional objects, bounded by a simple closed curve. In particular, we use periodic cubic B-spline curves [23] with a certain number of control points distributed in a way to define a shape of the object. The arbitrary density of the control points could simulate da complex shape of the real object defining their boundary equivalent to surface in 3D.

The B-spline can be always reparametrized to the unit domain $[0, 1)$. Thus, the parametric equation of a segment of curve f_j with four control points can be written as:

$$f_j(t) = B(t).C_j, \quad \text{for} \quad 0 \leq t < 1, \tag{1}$$

where the 4×4 matrix $B(t)$ is given by

$$B = \frac{1}{6}[t^3\ t^2\ t\ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}, \tag{2}$$

and the vector C_j of the corresponding control points is

$$C_j = [c_{j-1}\ c_j\ c_{j+1}\ c_{j+2}]^T.$$

All control points defining the object are given by vector $C = \{c_0, ..., c_{n-1}\}$, which generates n segment curves, $(f_j)_{j=1,...,n}$.

Starting with the control point vector $C^0 = C$, we could obtain the curve in multiresolution representation by recursively application of a wavelet transformation [8, 22, 25, 7]. This produces a sequence of representations having decreasing resolution, characterized by control point vectors $C^0, C^{-1}, \ldots, C^{-N}$.

2.2. Rugosity

In order to represent the rugosity of the object boundary in resolution corresponding to C^0 we could, in principle, resort to higher levels of resolution, given by control points arrays $C^1, C^2, \ldots, C^J$, up to the necessary level of detail. However, explicitly representing these levels of detail is not practical, due to the large storage requirements and to the fact that this information is only used locally, since a contact involves only a few points of each boundary.

Instead, we think about rugosity in terms of *tolerance intervals* over the object boundary. To each curve segment f_j of the boundary we associate a tolerance value λ_j that will be used in the collision detection and treatment procedures, as explained in Sections 3 and 4.

The value of the tolerance λ_j can be interpreted in statistical terms, by considering that the additional control points that determine the rugosity have displacements relative to f_j that are independently normally distributed, with mean 0 and variance σ_j^2, as noises that, in multiresolution, could represent the higher levels of details. A direct form to relate σ and λ is by using tolerance interval [18]: $\lambda = q\sigma$, where $0 < q \leq 1$; in particular we consider $q = 0.9$.

Onother way to relate σ and λ could be by considering that λ is the universal thresholding ([5]) corresponding to σ given by $\lambda = \sigma\frac{\sqrt{2\log m}}{\sqrt{m}}$, where σ is the variance of the highest level wavelet coefficients of a multiresolution representation and m is the number of such coefficients. We use the first form, by associating a tolerance interval λ to each segment of boundary.

3. Collision detection

Detecting collisions of complex geometric objects is the most time-consuming part of a dynamic simulation. Most of the existing approaches are oriented for objects described by a polygonal approximation [1, 17, 10, 14, 13] without considering their boundary details; it seems hard to extend these approaches to perform collision analysis for objects with non-homogeneous rugosity. We propose an extension of the OBBtree technique [9] to efficiently detect collisions of complex objects with such characteristics.

3.1. Hierarchical structures of oriented boxes

As in the OBBtree method, efficient collision detection is achieved by establishing a hierarchical structure of oriented boxes. The first step for generating our structure consists in creating elementary oriented boxes for each segment f_j of the boundary including their rugosities. With these elementary boxes we defined other boxes of the hierarchical structure.

The elementary box is created in two step; first we compute the fitted oriented box from segment f_j, after we include their rugosities expressed by the corresponding tolerance value λ_j (Figure 1). The principal axis of the fitted oriented box is determined by the eigen vector of the covariance matrix of a number of regularly sampling points on f_j, and the dimensions of box are computed by projecting the segment f_j over each principal axis. The addition of dimension of the box in λ_j corresponds to "fattening" the fitted oriented box (see [20] for details).

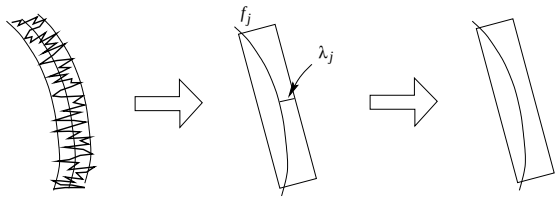


Figure 1. Oriented bounding box conserving rugosity.

For each pair of adjacent elementary boxes we construct another oriented box, called a *first level box* in the hierarchical structure. Each of these new boxes include two elementary boxes. After that, we construct the *second level boxes*, that include completely the elementary boxes corresponding to two adjacent first level boxes. This process is continued until a tree is built, where the highest level box includes all elementary boxes of the object. Note that, at each level, the orientation of a box is determined by the corresponding curve segment in the representation that corresponds to that level of resolution.

In all levels, the boxes built by the process above constitute augmented bounded boxes of the corresponding curve segment, in such a way as to include the perturbations determined by the surface rugosity. The construction of the augmented oriented bounded boxes tree (AOBBtree) is done in a bottom-up process, which differs from the approach used in the OBBtree method.

3.2. Contact detection

Since each object in the simulation environment has a AOBBtree, the test for intersection between two objects is done by testing the respective boxes in a hierarchical way, as described in [9], until elementary boxes level. Two boxes do not intersect if and only if there is a line such that the orthogonal projections of the boxes onto that line do not intersect. Actually, it suffices to test the lines which are parallel to a side of one of the boxes, which takes constant time.

When two elementary boxes intersect, the respective surface segments could be in three states: intersection, contact with tolerance, or separation. These states are detected by expressing, as in [24], the intersection of two segments as the solutions of the equation

$$H(t_f, t_g) = f_j(t_f) - g_k(t_g) = 0, \tag{3}$$

and the pair of nearest points in two segments as the solutions of equation

$$\begin{aligned} h_1(t_f, t_g) &= (f_j(t_f) - g_k(t_g)).f_j'(t_f) = 0, \\ h_2(t_f, t_g) &= (f_j(t_f) - g_k(t_g)).g_j'(t_g) = 0, \end{aligned} \tag{4}$$

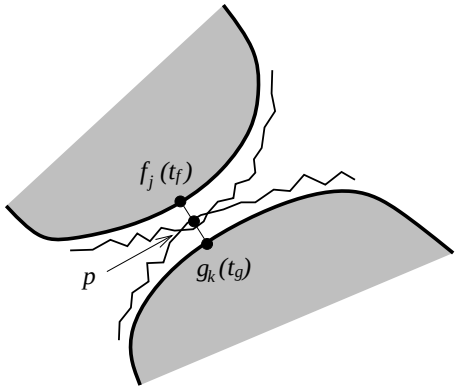


Figure 2. Tolerated contact between two piece of boundaries.

where the tangent vector of $f_j(t_f)$ is denoted by $f_j'(t_f)$. Both systems of equations have two unknows, t_f and t_g, and can be solved by the iterative Newton's method [19], which requires good starting solutions. To find them, we use the interval method [6], combined with a heuristic that consists in testing first those pairs of intervals for which the curve points corresponding to their centers are nearest.

The test for the state of the two segments, f_j and g_k, is done by first checking whether they intersect, using (3). If they do intersect, the process is aborted, and control is returned to the dynamic module for another loop with a smaller step size. If there is no intersection, we find the nearest pair of points on those segments using (4) and compare their distance with the combined tolerance of the segments. That is, if there exists a solution (t_f, t_g) for (4), we check whether

$$||d(t_f, t_g)|| \leq \lambda_f + \lambda_g,$$

where $d(t_f, t_g) = f_j(t_f) - g_k(t_g)$. Contact (with tolerance) occurs when this condition is satisfied.

The contact point between f_j and g_k is considered as belonging to the rugosity area, common to both segments, and is given by $p = g_k(t_f) + 0.5d(t_f, t_g)$, as shown in Figure 2.

4. Dynamic of contact

In all case of contacts, except in separation, we use dynamic based in impulses, an approach introduced by Mirtich [16] and Hahn [10]. The impulses at each contact point depend on the local friction coefficients associate to boundary, whose computation is explained below.

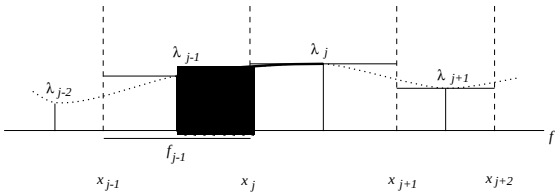
4.1. Rugosity approximation

For each contact point we compute the tolerance relative to $f_j(t_f)$, similar case is for $g_k(t_g)$, by interpolating m adjacent tolerances λ_j, where $m-1$ is the degree of polynomial interpolation (we addopt cubic interpolation, $m = 4$).

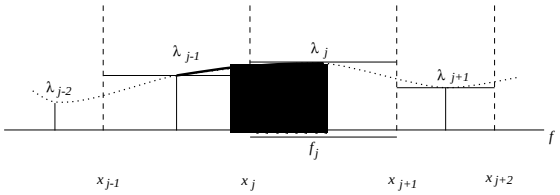
The level of rugosity over the boundary is aproximated by a continuous piecewise polynomial curve γ. Point $\gamma_j(t_f)$ of the rugosity curve corresponds to curve point $f_j(t_f)$, and the distance $\lambda_j(t_f) = \gamma_j(t_\lambda) - f_j(t_f)$ is the tolerance at that point.

The m adjacent tolerances $\{\lambda_{r-1}, \lambda_r, \lambda_{r+1}, \lambda_{r+2}\}$ for interpolation depend of $t_f \in [0, 1)$, see Figure 3. So, if $t_f < 0.5$ then r is $j-1$, and $t_\lambda = t_f + 1.5$; else, r is j, and $t_\lambda = t_f + 1.0$.

The coefficients of cubic polynomial $\gamma(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$ are obtained by evaluating $\lambda(t)$ for $t = 0, 1, 2, 3$, that correspond to the middle points of the adjacent intervals.



(contact point for $t_f \geq 0.5$)



(contact point for $t_f < 0.5$)

Figure 3. Tolerance function.

4.2. Friction coefficient adopted

The friction coefficient relative one object boundary, for example for f_j, at the contact point is modelled as a function $\mu(\lambda_j)$ of the corresponding tolerance value $\lambda_j(t_f)$. We adopt the following:

$$\mu(\lambda_j) = \begin{cases} 0 & \text{if } \lambda_j \leq \lambda_{min} \\ \lambda_j - \lambda_{min} & \text{otherwise.} \end{cases} \quad (5)$$

That is, we consider that surfaces having tolerance values smaller than a certain value (λ_{min}) have zero friction (see Figure 4).

In collision, both boundaries f_j and g_k, have different friction coefficients, $\mu_j(t_f)$ and $\mu_k(t_g)$ respectively, at their contact point p. Thus, the common friction coefficient for collision is considered as $\mu = \text{MAX}\{\mu_j(t_f), \mu_k(t_g)\}$.

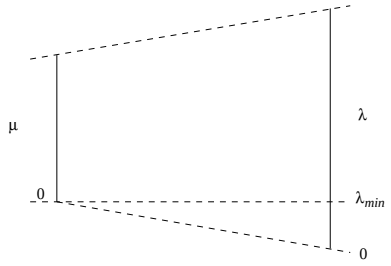


Figure 4. Relation between a tolerance and friction coefficient.

4.3. Impulse

If $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ represent point p for objects A and B, respectively, their relative velocity is given by

$$\boldsymbol{v}_{rel}(t) = \dot{\boldsymbol{p}}_a(t) - \dot{\boldsymbol{p}}_b(t).$$

Let $\hat{\boldsymbol{n}}$ be the unitary normal vector of object B at p, so the normal component of $\boldsymbol{v}_{rel}$ is $v_{rel\_n} = \boldsymbol{v}_{rel}.\hat{\boldsymbol{n}}$. In this way, when $v_{rel\_n} < 0$, the contact is a collision, whereas $v_{rel\_n} = 0$ implies continuous contact, and $v_{rel\_n} > 0$ implies separation.

The collision is treated by considering conservation of momentum [15]; for that, let $\boldsymbol{P}_a$ and $\boldsymbol{L}_a$ be the linear momentum and the angular momentum of A, respectively. The collision has a infinitesimal time of duration, in which occurrs the variation of momentum that corresponds to the impulse $\boldsymbol{J}$. The momentum after collision is given by

$$\boldsymbol{P}_a^+ = \boldsymbol{P}_a + \boldsymbol{J} \text{ and } \boldsymbol{L}_a^+ = \boldsymbol{L}_a - \boldsymbol{r}_a \times \boldsymbol{J}, \quad (6)$$

where $\boldsymbol{r}_a$ is the vetor from the center of mass of A to the contact point p.

The velocity of the contact point with respect to A, after collision, is expressed as

$$\begin{aligned} \dot{\boldsymbol{p}}_a^+ &= \dot{\boldsymbol{p}}_a + \frac{1}{m_a}\boldsymbol{J} + \left[\boldsymbol{I}_a^{-1}.(\boldsymbol{r}_a \times \boldsymbol{J})\right] \times \boldsymbol{r}_a \\ &= \dot{\boldsymbol{p}}_a + \left[\frac{1}{m_a}\mathbf{1} - (\tilde{\boldsymbol{r}}_a \boldsymbol{I}_a^{-1} \tilde{\boldsymbol{r}}_a)\right]\boldsymbol{J}, \end{aligned} \quad (7)$$

where $\mathbf{1}$ is the 3×3 identity matrix , $\boldsymbol{I}_a^{-1}$ is the inverse of the matrix of inertia of A, and $\tilde{\boldsymbol{r}}_a$ is the dual antisimetric matrix of $\boldsymbol{r}_a$. The velocity of p with respect to B, similarly to the case above, is

$$\dot{\boldsymbol{p}}_b^+ = \dot{\boldsymbol{p}}_b - \left[\frac{1}{m_b}\mathbf{1} - (\tilde{\boldsymbol{r}}_b \boldsymbol{I}_b^{-1} \tilde{\boldsymbol{r}}_b)\right]\boldsymbol{J}.$$

Therefore, the relative velocity after collision is given by

$$\boldsymbol{v}_{rel}^{+} = \boldsymbol{v}_{rel} + \boldsymbol{K}.\boldsymbol{J}, \tag{8}$$

where $\boldsymbol{K} = \left(\frac{1}{m_a} + \frac{1}{m_b}\right)\mathbf{1} - \left(\tilde{\boldsymbol{r}}_a \boldsymbol{I}_a^{-1} \tilde{\boldsymbol{r}}_a + \tilde{\boldsymbol{r}}_b \boldsymbol{I}_b^{-1} \tilde{\boldsymbol{r}}_b\right)$ is the collision matrix, which can be easily verified to be positive semi-definite. This implies that $\boldsymbol{J}$ can be computed as a simple expression, considering the restitution constant ε and friction coefficient μ of the collision, using Newton and Coulomb laws.

Newton's law relates the normal components of the relative velocity before and after the collision, as

$$v_{rel\_n}^{+} = -\varepsilon v_{rel\_n}. \tag{9}$$

Coulomb's law treats dynamic and static friction, respectively, through the following equations:

$$\text{if } v_{rel\_t} \neq 0 \text{ then } J_t = \mu J_n, \tag{10}$$

$$\text{if } v_{rel\_t} = 0 \text{ then } J_t < \mu J_n, \tag{11}$$

where $\boldsymbol{J} = J_n.\hat{\boldsymbol{n}} - J_t.\hat{\boldsymbol{t}}$. The unit tangencial relative velocity vector $\hat{\boldsymbol{t}}$ is orthogonal to $\hat{\boldsymbol{n}}$.

Therefore, in the case of static friction, the relative velocity after collision is expressed as

$$\boldsymbol{v}_{rel}^{+} = v_{rel\_t}^{+}.\hat{\boldsymbol{t}} - \varepsilon.v_{rel\_n}.\hat{\boldsymbol{n}};$$

by replacing it in (8), considering the condition $v_{rel\_t}^{+} = 0$, we obtain

$$\boldsymbol{J} = -\boldsymbol{K}^{-1}(\boldsymbol{v}_{rel} + \varepsilon.v_{rel\_n}.\hat{\boldsymbol{n}}). \tag{12}$$

For static friction, it must be verified that $J_t < \mu J_n$; else, we must compute the impulse for dynamic friction, from (8) considering $J_t = \mu J_n$ and (9), computing J_n first. With the values of J_n and J_t, we can compute the velocities of A and B after collision, using (6).

Continuous contact is treated also by impulses, considering that it results from micro-collisions [16]. The condition $v_{rel\_n} = 0$ (approximately) is perturbed to $v_{rel\_n} < 0$, by adopting the classical equation of mechanics $v_f^2 = v_i^2 + 2g\delta$, with $v_f = 0$ and $v_i = v_{rel\_n}$, yielding

$$v_{rel\_n} = -\sqrt{2g\delta},$$

where $0 \leq \delta \leq \lambda$ is the size of the penetration in the relative tolerance λ of contact point, as shown in Figure 5.

The restitution coefficient, ε_c, for contact is given by

$$\varepsilon_c(\delta) = \frac{\varepsilon_{max} - \varepsilon}{\lambda^2}\delta^2 + \varepsilon,$$

where ε is the normal restitution coefficient, λ is the tolerance relative to $f_j(t_f)$, and $\varepsilon_{max} \geq \varepsilon$ is specified by the user. Notice that $\varepsilon_c(\delta)$ varies from ε to ε_{max} as a quadratic function of δ, this permits to compute great value of impulse when the penetration of contact point on tolerance is greatest.

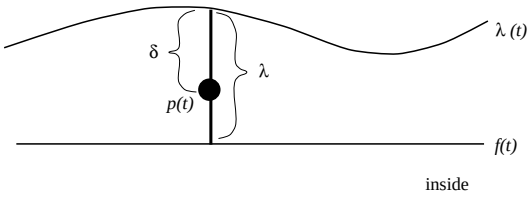
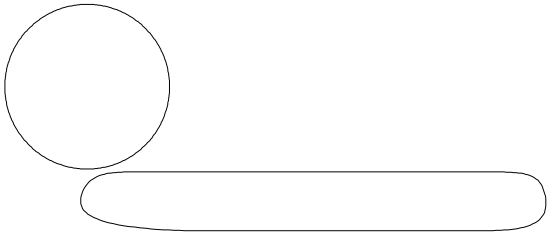


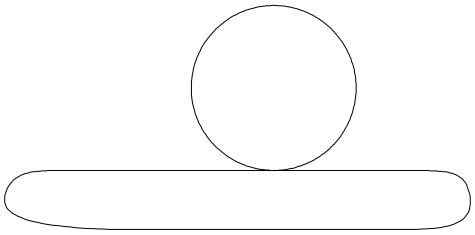
Figure 5. Penetration degree of contact point on tolerance interval.

4.4. Examples

To observe the resulting effects of the approach proposed here we analyse the dynamic behaviour of two simple objects. We consider one fixed object as lying horizontally, and allow the second circular object, having initial horizontal velocity, to move freely. In this situation, we could display the variation of friction coefficient depending of rugosity degree on each contact point. Figure 6 shows the simulation scene of the two objects.



(initial state)



(state during simulation)

Figure 6. An example of dynamic contact between two simple objects. Horizontal block is fixed, and the spherical object is free to move over the horizontal object.

To create the rugosity over the object surfaces, we use a library of several rugosity patterns. We consider the values defined in Figure 7 to test our model.

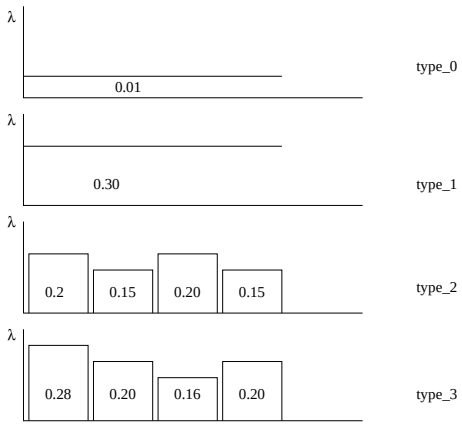


Figure 7. Example of four types of library for tolerances.

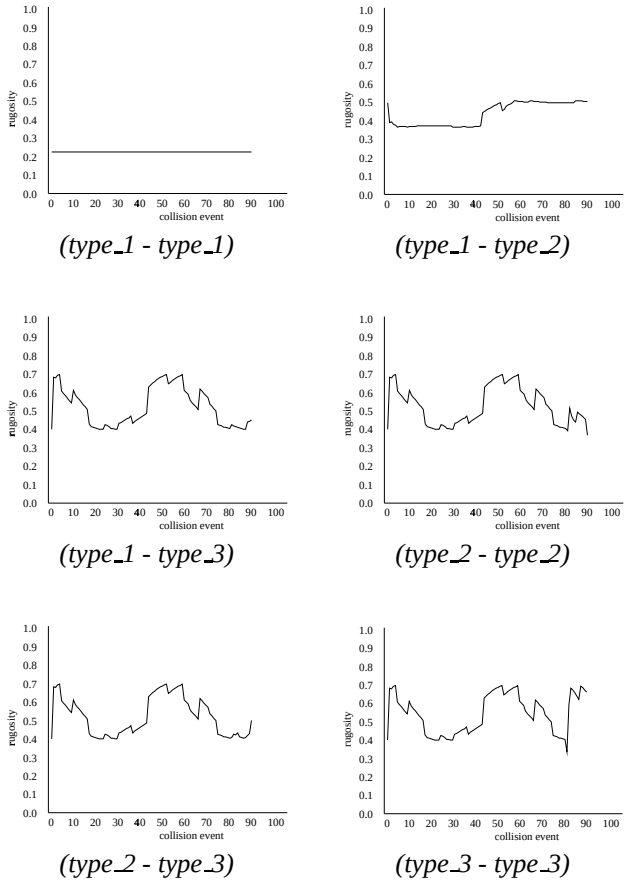


Figure 8. Friction value in each collision event between two objects (fixed horizontal block - free spheric object).

We applied each type of rugosity given in Figure 7 to each object and observed the variation of the friction coefficient in each collision between the two objects. In the first Figure 8, we observe a constant behavior of the friction, because both objects were covered by constant rugosities (*type_1* of Figure 7). The other ones were obtained by combining rugosities from library; for example the last figure represents rugosities *type_3*, where the tolerance values are non-homogeneous, distributed over both boundaries.

Figure 9 shows another example where we considered an object of arbitrary shape with initial horizontal velocity, moving over a fixed object lying horizontally. There are several collision points in each time step. Figure 10 and Figure 11 show their velocities behavior reparametrized to the unit interval for different distribution of rugosities over each object. In the first case, the rugosity covering both object is constant; and in the second case, the objects have rugosity *type_2* and *type_3*, respectivaly.

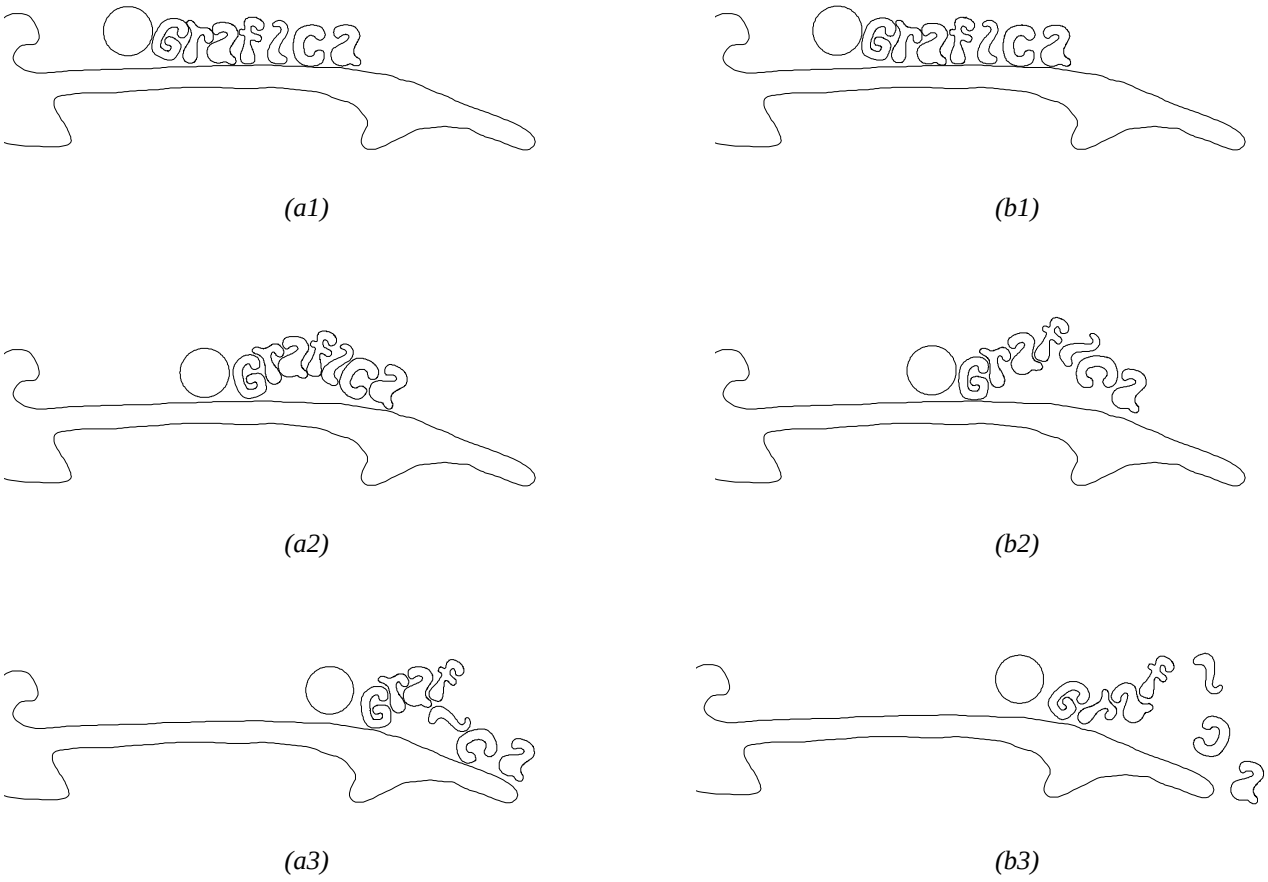
Figure 12 shows two columns of animation sequences of several irregular objects. In each case the objects are covered with several types of rugosity, and have differents dynamic behaviors.

5. Conclusion

Realistic simulation must integrate the morphologic aspect of the object as occur in the real word, because real world objects have arbitrary shapes and heterogeneous distribution of rugosities on their surfaces.

The rugosity degrees relative to the contact points of objects in collision permits the computation of a friction coefficient, in each collision event; for that is necessary to include the parameter of rugosity on the geometric definition of the object, because their explicit geometric representation is not practical. The use of statistical estimatives helps us to characterize the rugosity behavior of each local area of the surface using the rugosity tolerance value as a parameter. The consideration of a non-homogeneous distribution of rugosities over object surfaces allows us to obtain animations with interesting movements, because they have different constrainst for each collision event.

An immediate extension of this work is its use for 3D environments. For that we can use B-splines surfaces to model objects, all others considerations remaining the same as in this work. For realistic visualization of the objects it may be necessary to generate the rugosities over each object, using the statistical distributions associated with the tolerance values.



(a1) *(b1)*

(a2) *(b2)*

(a3) *(b3)*

Figure 12. Animation of objects with different types of rugosity, sequence of scenes *(a1-a3)* **and** *(b1-b3)***, respectivaly, for each case.**

Acknowledgements

This research has been developed in laboratories Tec-Graf, at PUC-Rio, and VISGRAF, at IMPA. This project is sponsored by CNPq in Brazil.

References

[1] D. Baraff. *Dynamic Simulation of Non-penetrating Rigid Bodies.* PhD. Thesis, Department of Computer Science, Cornell University, Ithaca, 1992.

[2] R. Bartels and J. Beatty. A technique for the direct manipulation of spline curves. *Proceeding of Graphics Interface*, pages 33–39, 1989.

[3] R. Bartels and B. Fowler. Constraint based curve manipulation. *IEEE Computer Graphics and Applications*, pages 43–49, september 1993.

[4] S. Coquillart and P. Jancène. Animated free-form deformation: an interactive animation technique. *Computer Graphics Proceedings, SIGGRAPH91*, 25(4):23–26, july 1991.

[5] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[6] L. H. Figueiredo and J. Stolfi. *Métodos numéricos autovalidados e aplicações.* 21o Colóquio Brasileiro de Matemática, Brazil, 1997.

[7] A. Finkelstein and D. Salesin. Multiresolution curves. *Computer Graphics Proceedings, SIGGRAPH'94*, pages 261–268, 1994.

[8] J. Gomes and L. Velho. *From Fourier Analysis to Wavalets*. Course Notes, SIGGRAPH'98, 1998.

[9] S. Gottschalk, M. Lin, and D. Manocha. Obbtree: A hierarchical structure for rapid interference detection. *Computer Graphics Proceeding, SIGGRAPH'96*, pages 171–180, 1996.

[10] J. Hahn. Realistic animation of rigid bodies. *Computer Graphics*, 24(4):299–308, 1996.

[11] D. Halliday and R. Resnick. *Physics.* John Wiley & Sons, inc., 2 ed., New York, 1967.

[12] W. Hsu, J. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. *Computer Graphics Proceedings, SIGGRAPH'92*, pages 177–184, 1992.

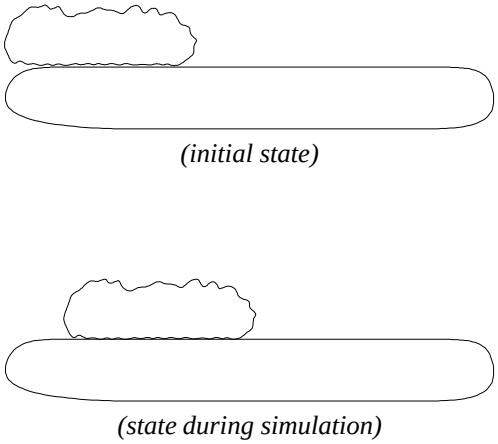


Figure 9. An example of dynamic contact between two irregular objects.

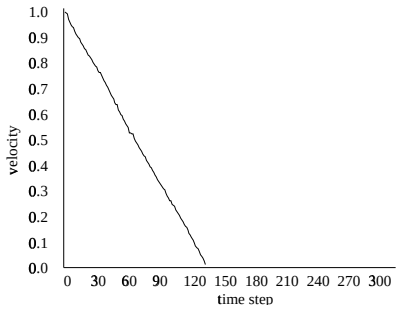


Figure 10. Velocity evolution of the object moving over the fixed object, both having constant distribution of rugosity.

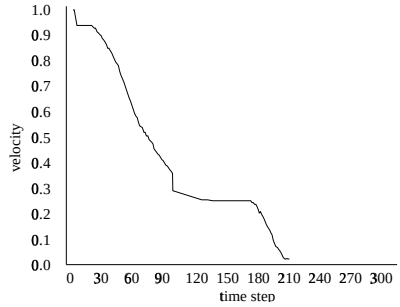


Figure 11. Velocity evolution of the object with rugosity *type_2* moving over the fixed object with rugosity *type_3*.

[13] P. Hubbard. *Collision Detection for Interactive Graphics Aplication.* PhD. Thesis, Department of Computer Science, Brown University, Berkeley, 1995.

[14] M. Lin. *Efficient Collition Detection for Animation an Robotics.* PhD. Thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1993.

[15] W. D. MacMillan. *Dynamics of Rigid Bodies.* Dover publications, inc, New York, 1936.

[16] B. Mirtich. *Impulse-Based Simulation of Rigid Body System.* PhD. Thesis, Department of Computer Science, University of California, Berkeley, 1996.

[17] M. Moore and J. Wilhelms. An collision detection and response for computer animation. *ACM Computer Graphics*, 22(4):289–298, 1988.

[18] D. F. Morrison. *Multivariate statistical methods.* McGraw-Hill series in probability and statistics, 2 ed., New York, 1976.

[19] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University, 2 ed., Cambridge, 1992.

[20] L. Rivera. *Análise de Interferências de Objetos de Geometria Arbitrária com Perturbações em Animação Baseado em Simulação Dinâmica.* Dsc. Thesis, Departamento de Informática, Pontifícia Universidade Católica de Rio de Janeiro, Brazil, 2000.

[21] L. Rivera, P. C. Carvalho, and L. Velho. Interactive manipulation of multiresolution curves. *Porceeding of IASTED International Conference in Computer Graphics and Imaging (CGIM'99)*, 22(1):94–99, october 1999.

[22] L. Rivera, P. C. Carvalho, and L. Velho. *Modelagem e manipulação de objetos de geometria complexa*. Technical Report n. 6/99, Departamento de Informática, Pontifícia Universidade Católica de Rio de Janeiro, Brazil, 1999.

[23] D. Rogers and J. Adams. *Mathematical Elements for Computer Graphics.* McGraw-Hill International Editions, 2da ed., New York, 1990.

[24] J. M. Snyder. An interactive tool for placing curved surfaces without interpenetration. *Computer Graphics Proceedings, SIGGRAPH'95*, pages 209–218, 1995.

[25] E. Stollnitz, T. Derose, and D. Salesin. *Wavelets for Computer Graphics: Theory and applications*. Morgan Kauffmann Publishers, inc., San Francisco, 1996.

[26] W. Welch. *Serius Putty: Topological Design for Variational Curves and Surfaces.* PhD Thesis, Department of Computer Science, Carnegie Mellon University, Pittsburgh, 1995.

[27] W. Wesselink. *Variational Modeling of Curves.* PhD Thesis, Department of Computer Science, Eindhoven University of Technology, Eindhoven, 1996.

[28] A. Witkin and P. Herbert. Using particles to sample and control implicit surfaces. *Computer Graphics Proceedings, SIGGRAPH'94*, pages 269–277, 1994.