Visualizing Collision Effects between Penetrating and Non-Penetrating Objects

Carlos D. Correa, Rutgers University   Deborah Silver, Rutgers University

1 Introduction

We present a novel method for visualizing collisions effects between penetrating or non-penetrating objects. The interaction of 3D objects has been traditionally divided into two stages: the detection of a collision, and the response or effects after the collision. In this work, we concern ourselves with the latter. The rendering of collision effects is useful for virtual sculpting, virtual reality, surgery simulation and games. The interaction between two colliding objects has been used for virtual sculpting, where one object (virtual "tool"), generates indentations in the other object (virtual "clay"). This has been implemented in real-time with the aid of a distance field representation of the virtual tool. One of the assumptions of virtual sculpting, reminiscent of real sculpting materials (e.g., clay), is that objects are non-penetrating. In this paper, we generalize the notion of collision effects by allowing penetrating "tools", where the object is cut as the tool penetrates the object. The collision effect is such that the object is still deformed, but a break appears where the tool is. In this work, we use an implicit representation of the deformed object (virtual "clay"), as opposed to traditional mesh-based approaches. The ability to deform objects implicitly was exploited by our previous work [Correa et al. 2006a; Correa et al. 2006b], but they were concerned with simple primitive deformations such as twists or peels.

2 Overview

For purposes of illustration, we consider the interaction of two objects: one rigid object, called "tool" (orange), which does not undergo deformation, and a deformable object, called "clay" (blue and green), which may be transformed. When the tool and clay objects collide, a collision effect is applied to the clay object, so that a desired rendering is obtained. We consider two cases: non-penetrating, where the clay object maintains its topology, and penetrating collision effects. Let us define two implicit representations f_c and f_t, for the clay and tool objects, respectively. A clay object is rendered as the isosurface $f_c(T(\mathbf{x})) = 0$, where T is a transformation, and $\mathbf{x}$ is a point in 3D space. Collision effects are then modeled as a transformation T. For a non-penetrating collision effect, the transformation is defined as $T(\mathbf{x}) = w(f_t(\mathbf{x}))d(\mathbf{x}) + \mathbf{t}$, where $w(d)$ is a weighting function that specifies the strength of the deformation, similar to [Gain and Marais 2005]. $d(\mathbf{x})$ is the maximum displacement, so that objects do not inter-penetrate, and $\mathbf{t}$ is a global translation. Examples are shown in Figure 1(a) where a hand model collides with and deforms a sphere, and Figure 1(c) where the Igea model collides and squeezes the bunny dataset. We can see the "sculpted" model in Figure 1(c,right) from a different perspective. For penetrating objects, the collision effect includes not only the displacement, but also the rendering of the interior of the object. A similar transformation is defined: $T(\mathbf{x}) = w(f_t(\mathbf{x}))\nabla_{f_t}(\mathbf{x}) + \mathbf{t}$. Instead of displacing in a particular direction, the cutting tool displaces the clay object in the direction orthogonal to the surface of the tool (i.e., ∇_{f_t}). In addition, we specify the interior of the object. For this, we exploit the fact that we are rendering an implicit surface. Rather than modifying the clay object's topology, we modify the implicit function to consider the space occupied by the tool object. In other words, we define the implicit function as the result of the difference between the clay object and the tool object: $min(f_c(T(\mathbf{x})), -f_t(\mathbf{x})) = 0$. Because the clay object is deformed while being cut, there is no loss of material in the subtraction operation. An example is shown in Figure 1(b) where a sphere is cut with a V shape. Texturing in the sphere helps to visualize the correct displacement due to collision (we do not concern about accurate texturing, but only as a guide). In Figure 1(d) a bunny dataset is cut with a blade object. The small inset shows the two objects while in the large image we disable the rendering of the tool object to reveal the collision effect. The interior of the object appears as light orange. In our current implementation, we exploit the GPU to render the surfaces implicitly, via raycasting [Correa and Silver 2007].

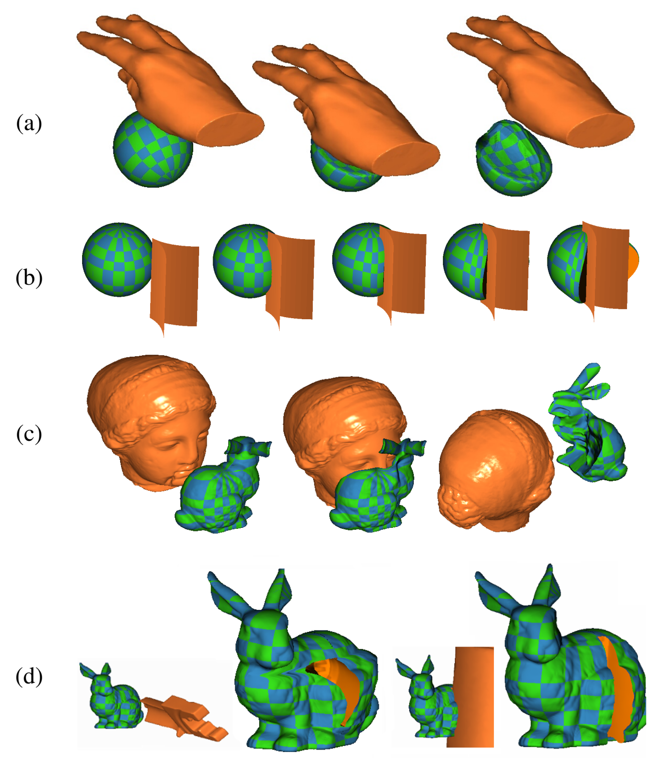

Figure 1: Example Collision Effects.

We have presented a novel mechanism for visualizing the effects of colliding objects. Unlike modeling approaches, where a new object is constructed based on collision response, we embed collision effects as part of the rendering process, which allows interactive exploration. We consider two cases: non-penetrating objects, where the collision response results in virtual sculpting, and penetrating objects, where the collision response results in cuts. This approach can be used in virtual reality, surgical simulation and the generation of special effects.

References

CORREA, C., AND SILVER, D. 2007. Programmable shaders for deformation rendering. In *Eurographics/SIGGRAPH Workshop on Graphics Hardware*.

CORREA, C., SILVER, D., AND CHEN, M. 2006. Feature aligned volume manipulation for illustration and visualization. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization) 12*, 5, sss–eee.

CORREA, C. D., SILVER, D., AND CHEN, M. 2006. Discontinuous displacement mapping for volume graphics. In *Volume Graphics 2006*, Eurographics Association, 9–16.

GAIN, J., AND MARAIS, P. 2005. Warp sculpting. *IEEE Transactions on Visualization and Computer Graphics 11*, 2, 217–227.