

Interactive Cartoon Rendering and Sketching of Clouds and Smoke

Eduardo J. Álvarez¹, Celso Campos¹, Silvana G. Meire¹, Ricardo Quirós²,
Joaquin Huerta², and Michael Gould²

¹ Departamento de Informática, Universidad de Vigo, Spain
ccampos@ei.uvigo.es

² Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I, Spain
{quirós, huerta, gould}@lsi.uji.es

Abstract. We present several techniques to generate clouds and smoke with cartoon style and sketching obtaining interactive speed for the graphical results. The proposed method allows abstracting the visual and geometric complexity of the gaseous phenomena using a particle system. The abstraction process is made using implicit surfaces, which are used later to calculate the silhouette and obtain the result image. Additionally, we add detail layers that allow improvement of the appearance and provide the sensation of greater volume for the gaseous effect. Finally, we also include in our application a simulator that generates smoke animations.

1 Introduction

The automatic generation of cartoons requires the use of two basic techniques in expressive rendering: a specific illumination model for this rendering style and the visualization of the objects silhouettes. This style is known as “Cartoon rendering” and its use is common in the production of animation films and in the creation of television contents. Cartoon rendering techniques in video games is also growing as they can produce more creative details than the techniques based on realism.

There are several techniques to automatically calculate silhouette -outline- and cel-shading [1][2][3]. Shadowing and self-shadowing, along with the silhouettes, are fundamental effects for expressing volume, position and limits of objects. Most of these techniques require general meshes and they do not allow representation of amorphous shapes, which are modeled by particle systems as in the case of clouds and smoke.

Our objective is to create cartoon vignettes for interactive entertainment applications, combining cartoon techniques with a particle system simulator that allows representation of amorphous shapes such as clouds and smoke. Special attention should be paid to the visual complexity of this type of gaseous phenomena, therefore we use implicit surfaces in order to abstract and simplify this complexity [4][5]. To obtain the expressive appearance, we introduce an algorithm that enhances silhouette visualization, within a cartoon rendering. For the simulation of smoke, we use a particle system based on Selle’s [6] hybrid model.

2 Previous Work

Clouds are important elements in the modeling of natural scenes, both if we want to obtain high quality images or for interactive applications. Clouds and smoke are gaseous phenomena very complicated to represent because of several issues: their fractal nature, the intrinsic difficulty of its animation and local illumination differences.

The representation of cloud shapes has been treated by three different strategies: volumetric clouds (explicit form [7] or procedural [8]), using billboards [9] [10], and by general surfaces [12][13]. The approach based on volume, in spite of the improvements of graphics hardware, is not yet possible at interactive speed because of the typical scene size and the level of detail required to represent the sky.

The impostors and billboards approach is the most widely used solution in video games and, although the results are suitable, their massive use slows the visualization due to the great number of pixels that must be rendered.

On the other hand, the use of general surfaces allows efficient visualization however it generates overly coarse models for representing volumetric forms. Bouthors [11], extends Gardner's model [12][13] by using a hierarchy of almost-spherical particles related to an implicit field that define a surface. This surface is later rendered to create a volumetric characteristic that provides realistic clouds.

In expressive rendering, the relevant works on gaseous phenomena are scarce in the literature. The first works published in this field are from Di Fiore [14] and Selle [6] trying to create streamlined animations of these phenomena. The approach of Di Fiore combines a variant of second order particle systems to simulate the gaseous effect movement using 2D billboards drawn by artists, which are called 'basic visualization components'.

Selle introduces a technique that facilitates the animation of cartoon rendering smoke. He proposes to use a particle system whose movement is generated with the method presented by Fedkiw [15] for the simulation of photorealistic smoke. To achieve the expressive appearance, each particle is rendered as a disc in the depth buffer creating a smoke cloud. In a second iteration of the algorithm, the silhouette of the whole smoke cloud is calculated reading the depth buffer and applying the depth differences. This method obtains approximately one image per second and has been used by Deussen [16] for the generation of illustrations of trees.

McGuire [17], presents an algorithm for the real-time generation of cartoon rendering smoke. He extends Selle's model incorporating shading, shadows, and nailboards (billboards with depth maps). Nailboards are used to calculate intersections between smoke and geometry, and to render the silhouette without using the depth buffer. The particle system is based on work recently presented by Selle, Rasmussen, and Fedkiw [18], which introduces a hybrid method that generates synergies using Lagrangian vortex particle methods and Eulerian grid based methods.

3 Scene Modeling

The rendering process necessarily requires an abstraction and simplification of the motif. This is made evident in the generation of hand-drawn sketches, even more so when representing gases. By means of several strokes the artist adds detail to the

scene creating a convincing simplification of the object representation which can be easily recognized by the viewer. Our method provides the user with complete freedom to design the shape and the aspect (appearance) of the cloud.

In a first approach, we propose the possibility to model clouds as static elements in the scene, the same way it normally happens in animation films. The process of modeling clouds begins with the definition by the user of particles p_i that comprise the cloud, each one having a center c_i , a radius r_i and a mass m_i .

Once the set of particles is defined we perform the simplification and the abstraction of the geometric model of clouds. To calculate the implicit surface described by the total particle set, we use the function of density proposed by Murakami and Ichihara [19], and later used by Luft and Deussen[5] for the real-time illustration of plants with Watercolor style.

The influence of a particle p_i in a point q is described by a density function $D_i(q)$ defined as:

$$D_i(q) = \left(1 - \left(\frac{\|q - c_i\|}{r_i} \right)^2 \right)^2 . \tag{1}$$

For $\|q - c_i\| \leq r_i$, otherwise $D_i(q) = 0$.

In our model we include in the density function the mass m_i of each particle which allows the user to weigh the influence of each particle in the calculation of the implicit surface. The modified density function is expressed as:

$$D_i(q) = m_i * \left(1 - \left(\frac{\|q - c_i\|}{r_i} \right)^2 \right)^2 . \tag{2}$$

The implicit surface is generated from the summation of the density function of the set:

$$F(q) = \sum_i D_i(q) - T . \tag{3}$$

Therefore, the implicit surface $F(q) = 0$ is defined as those points q where summation of the density functions equals threshold T . The influence of the radius r_i and the mass m_i of particles, as well as the threshold T , are chosen empirically as they depend on the number and density of particles. Finally we triangulate the implicit surface and then we optimize it according to the level of subdivisions s_i chosen by the user. Fig. 1 and Table 1 provide a comparison between different levels of simplification.

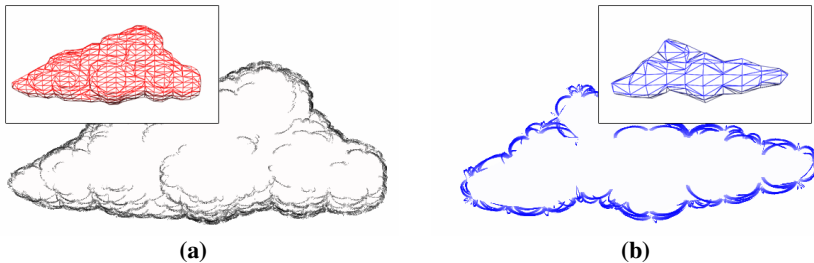


Fig. 1. Abstraction and simplification of two clouds models

Table 1. Comparison of the triangles count and the parameters used for the implicit surfaces

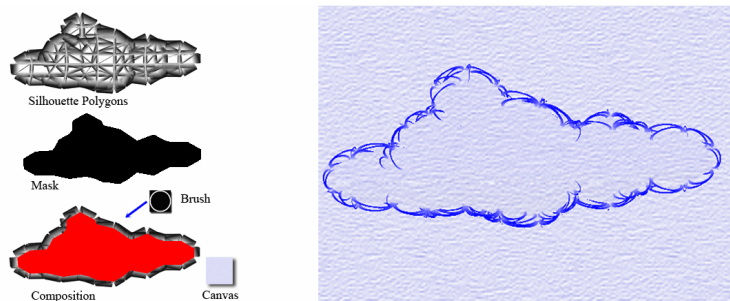
Figure	#particles	#tri implicit surface	T	m_i	s_i
1(a)	12	2360	0.05	8	22
1(b)	6	352	0.6	2	11

4 Rendering

Using an implicit surface allows calculation of the silhouette and to apply a illumination model for rendering. For silhouette detection and to achieve the cartoon appearance we use our previously published method [4]. Next, we describe the proposed method and discuss the visual results obtained thus far.

The detection algorithm allows silhouette extraction as an analytical representation obtaining interactive frame rates. As opposed to the methods proposed by other authors [14] [17] [6], the analytical description of the silhouette can be used to create new textured polygons which improve the appearance of the model. Our system allows us to define the height and the scale of the new polygons that form the silhouette.

The main drawback of this algorithm is that we need to remove those polygons that have been generated for interior hidden edges of the original polygon. A simple solution to this problem draws the mesh during a second iteration of the algorithm, once the hidden polygons have been removed, as shown by Fig. 2, left.

**Fig. 2.** Composing the final image for silhouette/based rendering

Finally, we select the texture to apply to the polygons of the silhouette and the background image to compose the final image. In Fig. 2 right, we show the Sketch of a cloud using this technique.

The illumination model used for the cartoon rendering allows a maximum of 32 gradations, which are applied on the mesh generated from the implicit surface as a 1D texture. The process of obtaining the final image is similar to the one described previously, however in this case we do not use the mask but instead the polygonal mesh textured with cartoon style, as shown in Fig. 3 left.

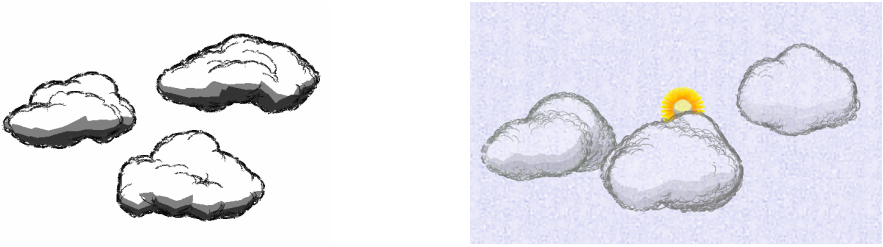


Fig. 3. Left, cartoon rendering image. Right, cartoon rendering with transparency.

Given the nature of gaseous phenomena it may be interesting to be able to define transparency levels at the same time that cartoon rendering is applied. In this case it is necessary to generate the mask of the cloud and to introduce it in a third step, as it is shown in Fig. 3, right.

5 Details Layer for Clouds

Once the general aspect of the cloud is defined, it may be interesting to incorporate greater level of detail to improve its appearance and to provide the sensation of greater volume. With this purpose, we propose to calculate a second implicit surface.

The calculation of the second implicit surface is made from particles p_i defined by the user in the scene modeling process (section 3). We reduce the value of the radius r_i and the mass m_i of each particle and we apply the density function $D_i(q)$ again, creating an inner cloud.

We use this new implicit surface to calculate its silhouette. Since the positions of particles used for its creation are the same for both surfaces, the second surface as well as its silhouette will be contained initially within the first surface.

Our system allows the user to independently modify the calculation parameters of both surfaces, making it is feasible to triangulate both surfaces with different number of polygons. Moreover, also the height and scale parameters of the silhouette can be changed for each surface. Thus the polygons that form the silhouette of the inner surface may be visible and cover part of the outer surface, enhancing the appearance of the final image. The result obtained for the example cloud can be observed in Fig. 4.

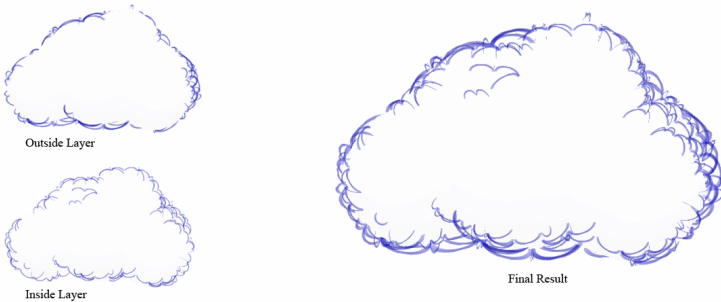


Fig. 4. A cloud with two layers of detail

Because the second surface is only necessary to add detail through the outline of the silhouette, and it is inside the outer surface, it is not necessary to visualize it nor to use it as a mask.

6 Smoke Simulation

Each particle of real smoke has very little mass or volume. Therefore, the smoke simulation is, in fact, the simulation of the instability of the air that contains the smoke particles. Expressive rendering is aimed at obtaining, first of all, a convincing shape of the object. In the case of the amorphous shape of the smoke, as with clouds, we use a particle set that is the base for calculating the surface that is used for rendering this effect.

Cloud models can be static, however in the case of smoke it is necessary to have a dynamic particle system. Our model uses a simplified version of the proposal made recently by Selle et al [18] for the particle system. It allows the user to fit the parameters pertinent to wind, turbulence, environmental forces and vortices, among others.

The positions of particles are calculated interactively according to the initial configuration defined by the user. Once the new positions are computed, we recalculate the implicit surface using the method described in section 3. Then we calculate the silhouette and we render it as we have described in the previous sections.

In the real world the smoke particles dissipate according to their speed. Although speed is a more objective criterion, it is more convincing to do the animation based on time. This approach allows us to maintain the number of particles steady during the simulation process. In this way we achieve that the speed of the visualization process of the smoke remains more or less constant.

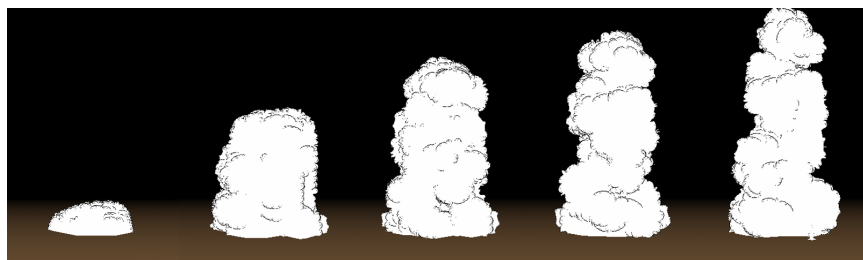


Fig. 5. Time evolution of cartoon smoke

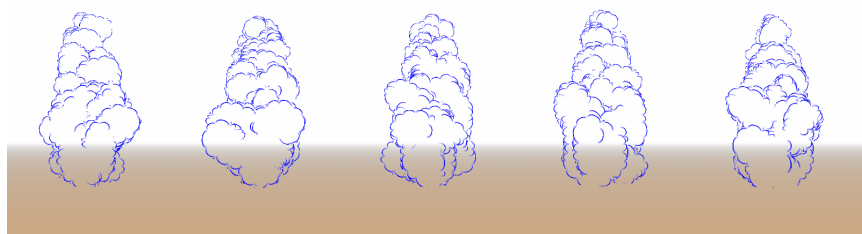


Fig. 6. Time evolution of sketch smoke

7 Results

The results obtained show a convincing imitation of hand-drawn sketches and drawings, although our approach is not strict in its physical foundations. We have given priority to the visual appearance with the purpose of simplifying the amount of information to represent while keeping the overall aspect and the capability of the user to identify the amorphous objects. With our approach we have obtained good results for interactive models rendering. Still, to obtain high resolution images intended for printing with good quality, we must optimize the algorithms developed.

The performance of our method has been demonstrated in a PC platform, with AMD's Athlon 64 X2 3800+ processor and a GeForce 7950 512 Mb graphics card, running Windows XP. Once we calculate the geometry of the objects to render, we set up different shape parameters for the clouds.

Different models have been created and different parameters have been applied, which entails the necessity to execute different number of iterations of the algorithm according to the desired target.

Table 2. Rendering times of clouds and smoke

Figure	#particles	#tri implicit surface	#iteration	s_i	fps
2	6	352	2	22	283
3(a)	47	2464	2	21	88
3(b)	21	4724	3	20	65
4	9	1928	3	21	154
5	150	$1600 < \#tri < 2100$	2	8	38
6	200	$1400 < \#tri < 1700$	2	7	48

8 Conclusions and Future Work

We present several techniques that allow representation of clouds and smoke with cartoon rendering and sketching. In contrast to the existing methods to date, our method provides results with interactive frame rates. The appearance of the gaseous phenomena is very stylized and incorporates greater level of detail depending on the user preferences; he can change several parameters affecting the results.

Temporal cost can be improved further by programming our functions in the hardware of the GPU which would allow greater realism to the process of simulation of the smoke. Also, it would be interesting to incorporate a model of behavior to generate particles of clouds with the purpose of generating animated sequences of its movement and metamorphosis. Finally, this method could also be enhancing by introducing multiresolution features that would improve the massive application of gaseous effects in computer graphics.

Acknowledgements

This work was partially supported by grant 05VI-1C02 of the University of Vigo, rant TIN2005-08863-C03 of the Spanish Ministry of Education and Science and by STREP project GameTools (IST-004363).

References

1. J. Buchanan, and M. Sousa. "The edge buffer: a data structure for easy silhouette rendering". In Proceedings of NPAR '00, (2000) 39–42
2. L. Markosian, M. Kowalski, D. Goldstein, S. Trychin, and J. Hughes. "Real-time non-photorealistic rendering". In Proceedings of SIGGRAPH '97, (1997) 415–420
3. R. Raskar and M. Cohen. "Image precision silhouette edges". In Proceedings of I3D., (1999) 135–140
4. C. Campos, R. Quirós, J. Huerta, E. Camahort, R. Vivó, J. Lluch. "Real Time Tree Sketching". Lecture Notes in Computer Science. Springer Berlin / Heidelberg, vol, 0302-9743, (2004) 197–204
5. T. Luft and O. Deussen, "Real-Time Watercolor Illustrations of Plants Using a Blurred Depth Test", In Proceedings of NPAR '06, (2006)
6. A. Selle, A. MOHR and S. Cheney, "Cartoon Rendering of Smoke Animations". In Proceedings of NPAR '04, (2004) 57–60
7. T. Nishita, E. Nakamae, Y. Dobashi. "Display of clouds taking into account multiple anisotropic scattering and sky light". In Proceedings of SIGGRAPH'96, (1996) 379–386
8. J. Schpok, J. Simons, D. S. Ebert, C. Hansen, "A real-time cloud modeling, rendering, and animation system". Symposium on Computer Animation'03, (2003) 160–166
9. Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, T. Nishita, "A simple, efficient method for realistic animation of clouds". In Proceedings of ACM SIGGRAPH'00, (2000) 19–28
10. M. J. Harris, A. Lastra, "Real-time cloud rendering". Computer Graphics Forum 20, 3, (2001) 76–84
11. A. Bouthors and F. Neyret, "Modeling clouds shape". In Proceedings Eurographics '04, (2004)
12. G. Y. Gardner, "Simulation of natural scenes using textured quadric surfaces". In Computer Graphics In Proceedings of SIGGRAPH'84, 18, (1984) 11–20
13. G. Y. Gardner, "Visual simulation of clouds". In Computer Graphics SIGGRAPH '85, Barsky B. A., 19, (1985) 297–303
14. F. Di Fiore, W. Van Haevre, and F. Van Reeth, "Rendering Artistic and Believable Trees for Cartoon Animation", Proceedings of CGI2003, (2003)
15. R. Fedkiw, J. Stam, and H. W. Jensen, "Visual simulation of smoke". In Proceedings of SIGGRAPH '01, ACM Press, (2001) 15–22
16. O. Deussen, and T. Strothotte, "Computer-generated pen-and-ink illustration of trees". In Proceedings of SIGGRAPH '00, (2000) 13–18
17. M. McGuire, A. Fein. "Real-Time Rendering of Cartoon Smoke and Clouds". In Proceedings of NPAR '06, (2006)
18. A. Selle, N. Rasmussen, R. Fedkiw, "A vortex particle method for smoke, water and explosions". ACM Trans. Graph., (2005) 910–914
19. S. Murakami and H. Ichihara, "On a 3d display method by metaball technique". Transactions of the Institute of Electronics, Information and Communication Engineers J70-D, 8, (1987) 1607–1615