

Rapport d'avancement

Antoine Bouthors

7 février 2005

Table des matières

1	Travail effectué pendant le stage	1
1.1	Forme	1
1.1.1	La création et le placement des blobs sur la surface des blobs parents	1
1.1.2	La discrétisation de la surface du nuage	2
1.2	Shading	2
1.3	Résultats	2
2	Travail effectué en début de thèse	3
3	Travail futur	3
3.1	Forme	3
3.1.1	Méthode Hart	3
3.1.2	Méthodes proposées par Marie-Paule	3
3.1.3	Autres méthodes	4
3.2	Shading	4
3.2.1	Corolle et scattering	4
3.2.2	Sillons	4
3.2.3	Textures	4
3.2.4	Ombrage	5
3.2.5	Couleurs	5
3.2.6	Surfels	5
3.3	Animation	5
4	Proposition de planning	5

1 Travail effectué pendant le stage

Pendant le stage, mon travail s'est divisé en deux parties : la génération de la forme du nuage et la mise au point d'un premier shader.

1.1 Forme

Le principe de base est le suivant :

- Une hiérarchie de blobs est créée.
- Chaque blob définit une surface implicite (définie par une série de paramètres, comme un facteur d'aplatissement, un rayon, etc.) sur laquelle sont placés les blobs du niveau hiérarchique suivant.
- La surface des blobs du dernier niveau définit la surface du nuage

L'allure du nuage est alors déterminée par la formule et les paramètres choisis pour les surfaces implicites, qui permettent a priori de faire tout ce qu'on veut, pourvu qu'on trouve les formules qui correspondent. Il y a cependant deux points à résoudre :

1.1.1 La création et le placement des blobs sur la surface des blobs parents

Ceci a été fait via un système de particules (chaque particule correspondant à un blob) qui peuplent la surface implicite automatiquement.

Le problème de cette implémentation est qu'elle n'est pas tellement contrôlable : les blobs se créent et se placent tout seuls, avec comme seule contrainte que le peuplement doit être homogène (ils se repoussent les uns les autres).

Du coup, on ne contrôle pas vraiment la forme finale du nuage. Si on veut un nuage avec une tourelle par-ci et une excroissance par là, on ne peut le décider qu’au niveau racine où on place les blobs de base. Et les blobs des autres niveaux, générés et placés aléatoirement, peuvent très bien gommer notre architecture...

Par contre, le modèle peut parfaitement s’adapter à l’animation qu’on veut, il suffit de changer le modèle qui régit le placement des blobs : pour le moment ils bougent de façon à peupler la surface, mais on peut très bien les faire bouger de façon à ce qu’ils aient un mouvement convectif comme en [Ney97] tout en restant sur la surface du blob parent.

1.1.2 La discrétisation de la surface du nuage

C’est là où j’ai un peu triché puisque pendant le stage j’ai pris comme surface du nuage l’ensemble des surfaces des blobs, ce qui a éliminé toutes les questions de “comment la surface se comporte entre deux blobs (est-ce qu’elle garde bien les creux marqués ou non) ?”, mais a posé plusieurs gros problèmes :

- Les triangles de chaque blobs s’intersectaient entre eux, ce qui a donné des artefacts avec le blending
- La surface du nuage n’était pas au final un manifold
- Il y avait un tas de triangle qui ne servaient à rien (*e.g.*, tout ceux qui étaient à l’intérieur du volume du nuage)
- Il y avait un tas de triangles tout court, beaucoup trop

Concrètement, pour discrétiser un blob, je partais d’une sphère discrétisée de la taille du blob dont j’offsetais les vertex pour qu’ils collent à la surface implicite à la façon de [Gas93], et c’est tout.

1.2 Shading

Le travail sur le rendu n’a pas été particulièrement innovant mais m’a permis de prendre en main les langages de shading et d’expérimenter un peu les possibilités des shaders.

J’ai commencé par essayer de transposer le shader de [Ney00] en Cg, que j’ai reproduit plus ou moins fidèlement. Le résultat est tout de même moins bon, mais j’ai préféré faire quelques autres expérimentations pour voir quelles étaient les limites des shaders plutôt que de rester sur un seul problème.

J’ai donc travaillé sur (en vrac) :

- le bump mapping
- l’augmentation de la lumière dans les sillons, via un paramètre indiquant la distance du point à la surface du blob parent
- l’effet volumique de Gardner [Gar84, Gar85, Ney00], avec différents effets : prise en compte des normales des blobs parents, différentes fonctions de densité...
- l’ombrage, avec une shadow map un peu trafiquée pour ne pas mettre d’ombre sur la corolle

1.3 Résultats

Screenshots : <file:/home/evasion/aboutor/stage/code/forme/images/catalog.html>

C’est pas encore trop ça... Comme problèmes, on peut citer :

- Le bump mapping bugue
- La lacunarité n’est pas cohérente : la taille des blobs simulés par le bump mapping et celle des blobs modélisés par les meshes ne collent pas du tout. C’est à mon avis un gros facteur de non réalisme
- Les couleurs sont mauvaises
- La corolle n’est pas belle
- Les ombres sont moches
- C’est bien bien lent (≈ 0.3 FPS) à cause du nombre de faces
- Et surtout, il y a des tas d’artefacts à cause du blending et de l’intersection des faces

2 Travail effectué en début de thèse

Comme c'était la forme qui posait le plus gros problème, j'ai voulu repartir d'une forme clean et plus légère pour éliminer définitivement les deux derniers problèmes. Je n'ai donc pas avancé sur le rendu depuis.

Le but est de discrétiser en un seul mesh la surface décrite par l'ensemble des surfaces des blobs. Comme on a comme objectif d'avoir un nuage animé, j'ai cherché du côté des techniques qui permettent d'avoir un mesh animé, donc typiquement pas du marching cube de base.

Comme j'avais déjà un système à base de particules pour placer les blobs basé sur [CA97, WH94], j'ai utilisé cette technique pour discrétiser aussi la surface du nuage, d'autant plus que l'article décrivait comment générer un mesh à partir des particules.

Cependant, le paragraphe le décrivant est extrêmement vague et confus, et d'après ce que j'en avais compris ne pouvait pas marcher. Seulement, je viens juste de le relire et à présent je n'arrive pas à le remettre en défaut, donc j'avais peut-être compris quelque chose de travers avant.

Le principe est de tirer parti des informations de voisinage qu'on possède déjà sur les particules pour construire un mesh en utilisant les liens entre les voisins pour les arrêtes. On part d'une particule au hasard et on agrandit le mesh en allant d'un voisin à l'autre.

Pour ce qui est du placement des particules elles-mêmes, il y a possibilité de faire varier la densité de particules dans l'espace, donc typiquement d'en mettre plus dans les régions de plus grande courbure, ce qui permettrait de marquer les sillons.

Au final, ce que j'ai fait concrètement au début de ma thèse, c'est :

- D'une part, étudié la discrétisation de la surface en utilisant le système de particules que j'avais pour les blobs et en essayant d'implémenter (sans succès) l'algorithme de discrétisation de [CA97]
- D'autre part, amélioré le système de particules de façon à ce qu'il parvienne à l'équilibre plus rapidement, autant pour les particules définissant les positions des blobs que celles utilisées dans la discrétisation de la surface

J'en étais à considérer un algorithme de discrétisation à partir des particules utilisant la cohérence temporelle, soit en partant de [Har] (cf. 3.1.1), soit from scratch, quand le monitorat a finalement accaparé tout mon temps.

3 Travail futur

3.1 Forme

3.1.1 Méthode Hart

Vu la façon dont bouge un nuage, il y a moyen d'utiliser la cohérence temporelle, et c'est là que les particules montrent vraiment leur intérêt. John Hart propose sur son site une librairie de modélisation de surface à base de systèmes de particules [Har, HBJF02] qui inclut notamment la possibilité de discrétiser une isosurface avec des particules en tirant apparemment partie de la cohérence temporelle.

Le problème est que la méthode qu'ils utilisent n'est pas décrite dans leurs articles, ni dans la documentation, et donc il faut fouiller dans le code (ou leur demander) pour savoir comment il font, ce que je n'ai fait que grossièrement et sans succès pour l'instant.

A priori c'est une très bonne méthode pour ce qu'on veut faire.

3.1.2 Méthodes proposées par Marie-Paule

- **Adaptive Sampling of Implicit Surfaces for Interactive Modeling and Animation**

Méthode présentée par Mathieu Desbrun, Nicolas Tsingos et Marie-Paule dans [DTC96], utilisée/revue dans [HAC03, CH01]. Elle nécessite la présence d'un squelette pour placer des particules "graines" qui créeront la surface implicite. Les blobs du nuage pourraient être utilisés pour ce squelette. La discrétisation n'est pas régulière mais ce n'est pas forcément un problème. L'utilisation des blobs comme squelette est sans aucun doute un avantage puisqu'ils nous donnent des informations sur la position probable des particules.

Donc a priori une technique intéressante aussi, à étudier. Tirer parti de la cohérence temporelle n'est peut-être pas aussi simple qu'avec la technique 3.1.1.

- **Echantillonnage anisotrope et rendu par points différentiels pour les surfaces implicites**

Présenté à l'AFIG 2004 [LHRS04]. Ils ne génèrent pas un mesh, mais simplement un rectangle par particule, donc en gros des surfels. Et ça, je peux déjà le faire avec les particules que j'ai donc ça ne sert pas à grand chose, surtout qu'à première vue je vais plus vite qu'eux à placer mes surfels.

- **Marching triangles**

Il y a pas mal de papiers et de noms différents pour cette méthode [Cre02, KS01, AG01, HI96] qui consiste à partir d'une graine (un point ou un triangle) sur la surface implicite et de l'étendre en rajoutant des triangles petit à petit sur les arêtes libres du mesh. Les dernières variantes sont adaptatives. Le mesh généré est en général très régulier (voire Delaunay localement ou même globalement), mais pas à n'importe quel prix, et de toutes façons il n'y a aucune cohérence temporelle, donc je ne vois pas d'intérêt à se lancer là-dedans, à moins que tout le reste échoue.

3.1.3 Autres méthodes

Il n'y a pas que les surfaces implicites dans la vie. Si les primitives sont simplement des sphères, on peut imaginer qu'une simple CSG puisse faire l'affaire.

Et il serait aussi intéressant d'essayer avec des surfels, d'autant plus que c'est d'ores et déjà possible avec ce que j'ai fait jusqu'à présent.

3.2 Shading

3.2.1 Corolle et scattering

Un premier bon départ serait de réussir à refaire la corolle de [Ney00] de façon correcte, en reprenant bien les équations.

La grosse différence (et une grosse difficulté) est sur l'illumination de la corolle par derrière : comme on est pas en ray-tracing, on n'a pas moyen a priori de savoir localement si la corolle considérée est illuminée ou non, il faut donc intégrer dans le shader une technique d'ombrage (*cf.* 3.2.4).

Pour ce qui est de la simulation de la diffusion de la lumière, on peut trouver pas mal de formules et de simplifications différentes, suivant le bout par lequel on prend la chose, celles proposées dans [Ney00] n'en sont qu'un échantillon. Il faut donc chercher et tester les différentes possibilités, et ne pas oublier qu'on doit avoir quelque chose de robuste du point de vue de la justification physique.

Un exemple de technique possible : avec les shaders, il doit être possible, en une passe, d'obtenir dans le framebuffer la profondeur de nuage traversée par les rayons partant du point de vue. A partir de ça, on peut avoir beaucoup plus d'informations sur la fonction de dispersion à utiliser qu'en regardant simplement la normale.

J'aimerais aussi "valider" les essais en essayant de les rapprocher de tests effectués en réalité. Un exemple : observer la diffusion de la lumière dans un bloc de cire.

3.2.2 Sillons

Ici aussi, il faudrait se rapprocher de la réalité en faisant coller les résultats avec des choses observées dans la réalité ou simulées par exemple par radiosité.

D'autre part l'ajout de la composante apportée par les sillons peut se faire soit dans le shader de la surface directement (comme actuellement), soit dans un shader appliqué à un mesh particulier qui lui ne couvrirait que les sillons.

3.2.3 Textures

Il faudrait commencer par utiliser quelque chose de cohérent entre ce qui est représenté par le mesh et ce qui est représenté par la texture. L'idéal serait d'avoir la même formule pour les deux. L'inconvénient est que ça va changer pour chaque technique d'animation et de modélisation, et que du coup c'est à cheval entre les trois domaines. Mais je pense que c'est extrêmement important pour l'aspect et le réalisme.

3.2.4 Ombrage

Il y a moyen d'obtenir une ombre de qualité, utilisant l'effet de la corolle, peut-être même sans beaucoup d'effort. J'avais rapidement étudié la chose pendant le stage (voir figure 1), mais j'avais tout de suite arrêté quand j'ai compris que la non-convexité de la forme entraînait des problèmes non triviaux à résoudre.

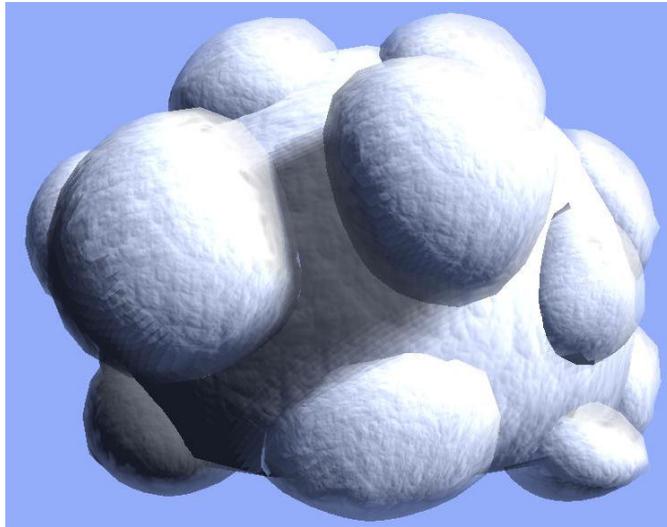


FIG. 1: Soft shadows

J'ai une solution si le nuage peut être découpé en tranches convexes, mais ce découpage peut coûter cher, et je pense qu'il y a moyen d'arriver à un résultat non physiquement exact mais visuellement satisfaisant pour beaucoup moins cher, donc je préfère commencer par chercher par là.

Sylvain Lefebvre a également réfléchi à la chose et dit qu'il a aussi une idée de méthode en découpant le mesh en tranche. Il faudrait donc en discuter avec lui (si ça se trouve c'est la même chose).

Ca c'est pour le shadow mapping. Avec d'autres techniques, il y a encore tout à voir. L'intérêt ici du shadow mapping est qu'il ne dépend pas de la complexité du nuage.

Une difficulté additionnelle sur la corolle est qu'il ne faut pas interférer avec la corolle mais au contraire en tirer parti (en ne mettant de l'ombre que là où le forward scattering n'atteint pas l'autre côté du nuage).

3.2.5 Couleurs

Les calculer par des formules valables plutôt que d'avoir des paramètres entrés à vue de nez ajouterait du réalisme.

3.2.6 Surfels

Là, tout est à étudier, je ne me suis jamais penché sur la question. Un avantage est qu'on a pas besoin de discrétiser la surface, un inconvénient est que s'il y a du blending, je ne vois pas comment ne pas avoir de problèmes, mais le cas a sûrement déjà été étudié.

3.3 Animation

Tout est à dire pour le moment : utilisation de [Ney97], de [Ney03], de la dernière soumission SIGGRAPH avec Alexis... Ce n'est de toutes façons pas la priorité pour le moment, donc : à voir plus tard.

4 Proposition de planning

Au vu de tout ça, on peut tirer les conclusions suivantes :

- Au niveau de la forme, on doit pouvoir obtenir une discrétisation sans énormément de recherche, en réutilisant des travaux déjà existants
- Il y a beaucoup de travail à faire et de choses à tester sur le shading

Je propose donc le planning suivant :

1. Je continue à travailler sur la forme pour obtenir une surface de travail convenable, mais pas trop longtemps (pas plus qu'un mois par exemple), en utilisant les méthodes Crossno (2) et/ou Hart (3.1.1)
2. Au bout de ce temps, j'utilise ce que j'ai réussi à obtenir pour me mettre au shading, et un stagiaire continue à travailler sur la forme, qui en aura sans aucun doute besoin

Je pense que le stagiaire ne doit pas être un TER, parce qu'en 3 semaines de stage, il ne sortira rien, à moins qu'il continue durant l'été. Un ingénieur 2^e ou 3^e année me semble plus approprié.

Références

- [AG01] Samir Akkouche and Eric Galin. Adaptive implicit surface polygonization using marching triangles. *Computer Graphics Forum*, 20(2) :67–80, 2001. 3.1.2
- [CA97] Patricia Crossno and Edward Angel. Isosurface extraction using particle systems. In *IEEE Visualization '97*, pages 495–498, November 1997. 2
- [CH01] Marie-Paule Cani and Samuel Hornus. Subdivision curve primitives : a new solution for interactive implicit modeling. In *Shape Modelling International*, Italy, May 2001. 3.1.2
- [Cre02] B. Crepin. Dynamic triangulation of variational implicit surfaces using incremental delaunay tetrahedralization. In *VVS '02 : Proceedings of the 2002 IEEE symposium on Volume visualization and graphics*, pages 73–80. IEEE Press, 2002. 3.1.2
- [DTC96] Mathieu Desbrun, Nicolas Tsingos, and Marie-Paule Cani. Adaptive sampling of implicit surfaces for interactive modeling and animation. *Computer Graphics Forum*, 15(5), dec 1996. Published under the name Marie-Paule Gascuel. 3.1.2
- [Gar84] Geoffrey Y. Gardner. Simulation of natural scenes using textured quadric surfaces. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 11–20, July 1984. 1.2
- [Gar85] Geoffrey Y. Gardner. Visual simulation of clouds. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 297–303, July 1985. 1.2
- [Gas93] Marie-Paule Gascuel. An implicit formulation for precise contact modeling between flexible solids. In *Proceedings of SIGGRAPH 93*, pages 313–320, August 1993. 1.1.2
- [HAC03] Samuel Hornus, Alexis Angelidis, and Marie-Paule Cani. Implicit modelling using subdivision-curves. *The Visual Computer*, 19(2-3) :94–104, May 2003. 3.1.2
- [Har] John C. Hart. Surface constrained particle systems library. <http://graphics.cs.uiuc.edu/surface/>. 2, 3.1.1
- [HBJF02] John C. Hart, Ed Bachta, Wojciech Jarosz, and Terry Fleury. Using particles to sample and control more complex implicit surfaces. In *Proceedings of SMI 02*, 2002. 3.1.1
- [HI96] A. Hilton and J. Illingworth. Marching triangles : Delaunay implicit surface triangulation. In *IEEE International Conference on Image Processing*, 1996. 3.1.2
- [KS01] T. Karkanis and A. J. Stewart. Curvature-dependent triangulation of implicit surfaces. *IEEE Computer Graphics & Applications*, 21(2) :60–69, mar / apr 2001. 3.1.2
- [LHRS04] Florian Levet, Julien Hadim, Patrick Reuter, and Christophe Schlick. Echantillonnage anisotropique et rendu par points différentiels pour les surfaces implicites. In *Actes des 17ièmes Journées de l'Association Française d'Informatique Graphique (AFIG)*, 2004. 3.1.2
- [Ney97] Fabrice Neyret. Qualitative simulation of cloud formation and evolution. In D. Thalmann and M. Van de Panne, editors, *8th Eurographics Workshop on Computer Animation and Simulation (EGCAS'97)*, pages 113–124. Eurographics, Springer Wein, September 1997. 1.1.1, 3.3
- [Ney00] Fabrice Neyret. A phenomenological shader for the rendering of cumulus clouds. Technical Report RR-3947, INRIA, May 2000. 1.2, 3.2.1
- [Ney03] Fabrice Neyret. Advected textures. *Symposium on Computer Animation'03*, july 2003. 3.3
- [WH94] Andrew P. Witkin and Paul S. Heckbert. Using particles to sample and control implicit surfaces. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, pages 269–278, July 1994. 2