Journal of Software Engineering and Applications, **2011, 4, 253-258**
doi:10.4236/jsea.2011.44028 Published Online April 2011 (http://www.SciRP.org/journal/jsea)

Scientific Research

Application of Genetic Algorithm for Computing a Global 3D Scene Exploration

Oana Livia Apostu, Karim Tamine

Xlim Laboratory, Department of Mathematics and Computer Sciences, University of Limoges, Limoges, France.
Email: {oana.livia, karim.tamine}@unilim.fr

Received March 15th, 2011; revised April 7th, 2011; accepted April 10th, 2011.

ABSTRACT

This paper is dedicated to virtual world exploration techniques, which have to help a human being to understand a 3*D scene. A new method to compute a global view of a scene is presented in the paper. The global view of a scene is determined by a "good" set off points of view. This method is based on a genetic algorithm. The "good" set of points of view is used to compute a camera path around the scene.*

Keywords*: *Genetic Algorithm, Global Virtual World Exploration

1. Introduction

In the recent years, the concept of virtual world has evolved and become more and more important. As a consequence, one of the most important problems that were raised consists in developing fast and accurate techniques for a good exploration and understanding of these virtual worlds.

The purpose of a virtual world exploration in computer graphics is to permit a human being, a user, to acquire enough information in order to better understand the environment he or she is faced with. This is done by guiding a virtual camera using an automatically computed path, depending on the nature of the world. The trajectory of the virtual camera is usually obtained using a set of "good" points of view that are either predetermined or calculated during the actual movement. This is directly connected to the notion of viewpoint quality, which plays an important role in path computation and therefore in scene understanding.

There are two classes of methods for a virtual world exploration:

1) Global exploration class, where the camera remains outside the world to be explored.

2) Local exploration class, where the camera moves inside a scene and becomes a part of the scene.

Global exploration is used in acquiring a general knowledge of the scene, whereas local exploration becomes necessary if further insight of the world is needed.

On the other hand, we can explore a virtual world in two different modes:

1) Real time online exploration, where the virtual world is visited for the first time and the path of the camera is determined in real time, in an incremental manner.

2) Offline exploration, where the camera path is pre-computed in a preliminary step. This means that the virtual world is found and analysed by the program guiding the camera, in order to determine interesting points of view and the paths linking these points. When necessary, the camera will explore the world, following the already determined path. In this mode, it is less important to use fast techniques in order to determine the camera path.

In this paper we are mainly concerned with obtaining a set of optimal viewpoints, for of an offline exploration. We remain in the context of global virtual world exploration, where the camera is outside the scene. We propose a new method which employs genetic algorithms in order to evolve a given set of viewpoints into another that offers a better view of the scene. We shall also try to propose a new definition of quality of a given set of viewpoints.

The paper is organized in the following manner: in Section 2 we review related works, Section 3 details the definition of the quality of a set of viewpoints, Section 4 takes a closer look at how genetic algorithms are used in our implementation, Section 5 presents some of the results we achieved, and finally, in Section 6 a conclusion is given with a brief description of future works.

2. Background

2.1. Static Explorations

The first works on visual scene understanding were published at the end of years 1980. Thus, Kamada *et al.* [1] proposed a fast method to compute a viewpoint that minimizes the degenerated edges of a scene.

Colin [2] has proposed a method initially developed for the scenes modeled by octrees. The aim of the method was to compute a good point of view for an octree. The method uses the principle of "direct approximate computation" to compute a good direction of view.

Plemenos [3] proposed an iterative method of automatic viewpoint calculation. The scene is placed at the center of a sphere, whose surface represents all the possible points of view. The sphere is divided into eight spherical triangles and the best one is chosen according to the view qualities of the vertices of a triangle. Then the selected spherical triangle is recursively subdivided. The best vertex is chosen to be the best point of view at the end of the process. The heuristic considers a viewpoint to be good if it gives a high amount of details and it minimizes the maximum angle deviation between the direction of view and the normals to the faces.

Sbert and Vasquez [4,5] introduced an information theory-based approach to estimate the quality of a viewpoint. This quality is computed as a viewpoint entropy function:

$$I(p) = \sum_{i=0}^{N_f} \frac{A_i}{A_t} \cdot \log_2 \frac{A_t}{A_i} \quad (1)$$

where:

p is the viewpoint.

N_f is the number of faces of the scene

A_i is the projected area of the face number i

A_t is the total area of the projection.

A_0 is the projected area of background in open scenes.

2.2. Dynamic Exploration

A single good point of view is generally not enough for complex scenes, and even a list of good viewpoints does not allow the user to understand a scene, as frequent changes of viewpoint may confuse him. To avoid this problem, virtual world exploration methods were proposed.

Plemenos and Dorme [6-8] proposed a method, where a virtual camera moves in real time on the surface of a sphere surrounding the virtual world. The exploration is online; the scene is being examined in incremental manner during the observation. All the polygons of the virtual world are taken into account at each step of the exploration. The method is based on the following heuristic:

$$w_c = \frac{v_c}{2} \cdot \left(1 + \frac{d_c}{p_c}\right) \quad (2)$$

where:

w_c is the weight of the current camera position,

v_c is the viewpoint complexity of the scene from the camera's current position,

p_c is the path traced by the camera from the starting point to the current position,

d_c is the distance from the starting point to the current position.

In order to avoid fast returns of the camera to the starting position, the importance of a viewpoint is made inversely proportional to the camera path from the starting to the current position. Also, for a smooth movement of the camera, only three new viewpoints are considered while computing the next position.

Vasquez [4] proposed an exploration method that is similar to the previous one. The difference is that the next viewpoint is chosen in dependence of the entropy (see Equation (1)) and the number of faces not yet visited. To evaluate the qualities of the next three possible positions, they multiply the viewpoint entropy of each point by the number of new faces that appear with respect to a set of faces already visited. In the case where none of the three possible viewpoints show a new face, they choose the one lying furthest from the initial position.

In many cases, the online exploration is not necessary because there is enough time to pre-compute interesting points of view for a virtual world and even interesting trajectories. Thus Jaubert [9] and Sokolov *et al.* [10] proposed an offline exploration method based on the pre-computing of a minimal set of good viewpoints.

In [11,12] image-based techniques are used to control the camera motions in a changing virtual world.

For more details, a state of the art paper on virtual world global exploration techniques is available [12], whereas viewpoint quality criteria and estimation techniques are presented in [13].

3. Quality of a Set of Viewpoints

3.1. Definition

Here we shall present a new heuristic using new definitions of viewpoint quality. Since we place our work in the context of 'parisien method' of genetic algorithms [14], each viewpoint is regarded as part of a group, and not as an individual. We define the view quality, or the amount of the scene visible from a set of viewpoints, as a relation between all the points in the set.

First of all, we consider that each viewpoint has a certain visibility of each one of the polygons composing the scene. We use the following notation:

$V_P(i)$ is the visibility information the viewpoint P has of the polygon number i.

Next, we define the view quality of the group of viewpoints, noted VP, in respect to a scene S, as follows:

$$Q(S,VP)=\frac{1}{n}\sum_{j}^{n}\max_{i=1,\cdots m} VPi(j)) \quad (3)$$

where

n is the total number of polygons

m is the total number of viewpoints

$VP_i(j)$ is the visibility information the viewpoint i has of the polygon j

The first advantage of our approach is that each viewpoint contributes to the quality of the group only with its strongest visibility information. A viewpoint can fail to see several faces of the model, but manage to have a very good view of one or two polygons. The given definition ensures that the point in question will bring to the group exactly the relevant information.

Moreover, in the context of the genetic algorithms, this definition will allow the elimination of the weakest elements of the group, without altering the general view quality of the scene. This means that we can easily select and keep only those viewpoints that provide the best visibility information. This will be further explained in Section 4.2.

3.2. Viewpoint Visibility Computation

In order to calculate the visibility of each polygon from a given viewpoint, we use a ray-tracing algorithm. A number of rays are traced between the viewpoint and each polygon. If all the rays reach the destination polygon without intersecting other polygons, than the viewpoint is considered to have a complete visibility of the polygon. If none of the rays reach their destination, the polygon is hidden for the viewpoint. If some of the rays reach the polygon, their number is expressed as a proportion of the total of the rays that have been traced.

The number of considered rays depends on the surface of the destination polygon. A bigger surface will require more rays, whereas a smaller one can be analyzed with only a few. Moreover, the destination points on the polygon's surface are part of a uniform distribution [10].

4. Genetic Algorithms

4.1. General Guidelines

First of all, let us suppose that there is an unknown scene, and the user would like to get a general comprehension of its structure. Since we are in the context of exploring the exterior of a scene, it is reasonable to restrict the space of possible viewpoints to a surrounding sphere. Therefore, the scene is placed in the center of the sphere, whose surface represents all the possible points of view.

The first step of our algorithm is to choose a random distribution of distinct viewpoints situated on the surrounding sphere. Since we are interested in evolving a set of points into an optimal one, we place no conditions on the initial set. This is considered as the start population.

This population is optimized through a series of genetic operations (selection, crossover and mutation). After each "cycle of life", the weakest individuals are being eliminated from the set. The number of viewpoints is permitted to increase or decrease up to certain limits, and will stop evolving after a predefined number of operations. Moreover, the process also stops if no improvement has been achieved after a certain number of iterations. The general routine is summarized in **Algorithm 1**. Each step of the genetic algorithm is explained in the following sections.

4.2. Selection of a Breeding Population

In order to decide which viewpoints will be affected by the genetic operators, we use a fitness proportionate selection, also known as roulette-wheel selection. A fitness function is used to associate a probability of selection with each viewpoint. In this current implementation, the fitness function is based on two parameters: the number of polygons that a viewpoint is capable of seeing, partially or completely, and the total amount of surface of the scene visible from the viewpoint.

The first parameter has the advantage of being an exact value, since it counts the visible polygons. Nevertheless, a partially visible polygon can be one that is visible only as far as 1%, and therefore less significant than one

Algorithm 1: Evolution of a set of viewpoints

```
1: procedure Evolve(scene S, set of viewpoints VP)
2: begin
3:     iterations ← 0
4:     bad_iterations ← 0
5:     while (iterations < max_iterations) and (bad_iterations <
max_bad_iterations) do
6:     begin
7:     //Start a new cycle of life
8:     //Eliminate the weakest individuals, in order to obtain a
breeding population
9:     eliminate_weakest(new_VP)
10:    old_quality ← Q(S, VP)
11:    //Perform a genetic operation (crossover, mutation)
12:    new_VP ← genetic_operation(VP)
13:        new_quality ← Q(S, new_VP)
14:        //Test the new view quality
15:    if (new_quality > old_quality) then
16:        begin
17:            VP ← new_VP
18:            bad_iterations ← -1
19:        end if
20:    end while
21: end
```


with a higher visibility. On the other hand, the second parameter gives an estimation of the amount of visible surface, but suffers from being an approximation, and not an exact value.

In this context, it would be interesting to find new criterion of fitness evaluation, that considers the viewpoint as part of the set, and evaluate its viability in relation with the other points and with the general quality of visibility.

An important step in selecting a breeding population consists in eliminating those viewpoints that are redundant to the global visibility **(Algorithm 1**, Line 9). Theses are points that either have equal visibility values (Definition 1) or have smaller visibility values for all the polygons in the scene, when compared to another viewpoint (Definition 2).

Let us suppose that ***n*** is the total number of polygons in the scene ***S*** and ***m*** is the total number of viewpoints. $\boldsymbol{P_i}$ and $\boldsymbol{R_i}$ are the visibility information that the viewpoints ***P*** and ***R*** respectively have of the polygon number *i*. We have the following two definitions.

Definition 1.

Two viewpoints are equal if their visibility values are equal.

$$P \approx R \Leftrightarrow P_i = R_i, \quad \text{for all} \quad i \in \{1,\cdots,n\}$$

Definition 2.

A viewpoint is smaller than another if all its visibility values for the same polygons are smaller than the visibility values of the second viewpoint.

$$P \le R \Leftrightarrow P_i \le R_i, \quad \text{for all} \quad i \in \{1,\cdots,n\}$$

Let VP be the set of viewpoints and S the considered scene. We can observe the following two results.

Lemma 1.

If $VP' = VP - \{P \mid \exists R \in VP : P \approx R\}$ then $Q(S,VP) = Q(S,VP')$

Proof.

Let A and B be two viewpoints with $A \approx B$. This implies that $\max(A_i, B_i) = B_i$, for all $i \in \{1,\cdots,n\}$.

$$\begin{aligned} nQ(S,VP) &= \sum_j^n \max_i \left(VP_i(j)\right) \\ &= \sum_j^n \max\left(\max(A_i,B_i), \max_{p \in VP-\{A,B\}} VP_i(j)\right) \\ &= \sum_j^n \max\left(B_i, \max_{p \in VP-\{A,B\}} VP_i(j)\right) \\ &= \sum_j^n \max_{p \in VP-\{A,B\}} VP_i(j) \\ &= nQ(S,VP) \end{aligned}$$

We can therefore eliminate all points that are equal to another viewpoint, without affecting the global visibility.

Lemma 2.

If $VP' = VP - \{P \mid \exists R \in VP : P \le R\}$ then $Q(S,VP) = Q(S,VP')$

Proof.

Let A and B be two viewpoints with $A \le B$. This implies that $\max(A_i, B_i) = B_i$, for all $i \in \{1,\cdots,n\}$. The rest of the demonstration is identical to the one provided for Lemma 1. This proves that we can eliminate all points that are smaller than another viewpoint without affecting the global visibility.

4.3. Crossover and Mutation

A viewpoint can mutate into a new point, which will be situated randomly in its neighbourhood. The neighbourhood of a viewpoint is defined as a percentage of the scene's surrounding sphere.

The crossover operation represents a barycentric interpolation of two viewpoints.

$$NewVP = \frac{\alpha}{\alpha + b} \cdot VP1 + \frac{\beta}{\alpha + \beta} \cdot VP2 \qquad (4)$$

where

a is the fitness evaluation of the viewpoint $VP1$

β is the fitness evaluation of the viewpoint $VP2$

The resulted point is afterwards projected on the surrounding sphere.

Since mutation is a unary operator, it raises no problems with viewpoint selection. On the other hand, the crossover operator requires the individuals to be grouped in pairs. This can be done by either a random process or by arranging viewpoints using some sort of criterion. In the latter case, two situations have been considered. The first one was based on the assumption that two good viewpoints will result into a third which has a visibility that is at least as good as that of its parents. So the viewpoints were grouped in the decreasing order of their fitness evaluation. A second approach tried to create balanced pairs, by grouping weak points with strong ones.

4.4. Virtual Camera Path

Once the «good» group of viewpoints computed, we shall compute the trajectory of à «virtual camera» through all these viewpoints. We use a optimization method such as "Heuristic algorithm for the Traveling-Salesman Problem" [15] on a complet graph consisting of viewpoints (nodes) computed by the genetic algorithm. The cost of each edge of the graph connecting two viewpoints P_i and P_j is equal to $d_{ij}/tetha_{ij}$ where d_{ij} is the euclidian distance between P_i and P_j, and $tetha_{ij}$ is the angle formed by the edges (P_{i-1},P_i) and (P_i,P_j).

5. Results

This section summarizes the results we obtained using our method. We have tested the genetic algorithms on

two types of scenes: a random distribution of triangles of various sizes (**Figure 1**) and scenes containing various models (**Figure 2**)

In each case, the algorithm started with a random group of viewpoints that were allowed to evolve up to a certain number. The choice was to allow the initial group to double its cardinal. One reason behind this was to test if there exists a maximal population whose visibility cannot be further improved.

The first observation that could be made is that in the case of the models, the improvement in visibility is two times more significant than in the case of the random distribution of triangles. For an average scene of 1000 polygons and a population of 10 viewpoints, the visibility was improved from a proportion of 60% to 70% in the case of random triangles and to 80% in the case of arbitrary models. One reason for this is the fact that random distributions of triangles may contain small faces that are almost hidden by other objects. Another factor that plays an important part is the coherence present in the case of actual models. First of all, mutation is basically choosing a new point in the neighbourhood of the first. Therefore it is obvious that in the case of a model, the new point will have a visibility similar to that of its ancestor; whereas in the case of random polygons, moving one unit may very well mean seeing a complete different arrangement. The same logic applies to the crossover operator; placing a new viewpoint somewhere between its ancestors will guarantee nothing in terms of visibility in the case of randomly placed polygons.

We were also interested in how the two genetic operators influence the evolution of the population of viewpoints. Since evolution was significant especially in the case of the scenes containing coherent models, we have studied the behaviours of crossover and mutation only on those types of scenes. **Table 1** summarizes the average results obtained for scenes of 1000 polygons and a population of maximum 10 viewpoints. Although it is obvious that applying both genetic operators yields better results than applying only one, we remark that crossover operations achieve the same improvements as mutation operations, but with fewer viewpoints. Therefore, choosing a greater percentage of individuals for the crossover operation and fewer ones for mutation will ensure better results. **Table 2** offers a comparison between the results obtained when applying the genetic operators on different proportions of the population of viewpoints.

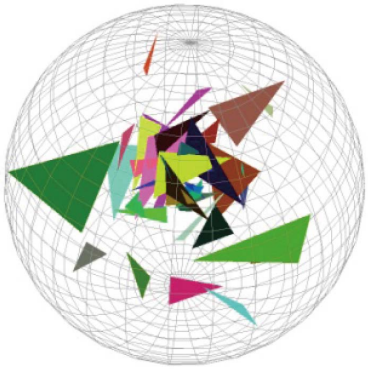

Figure 1. A set of random triangles.

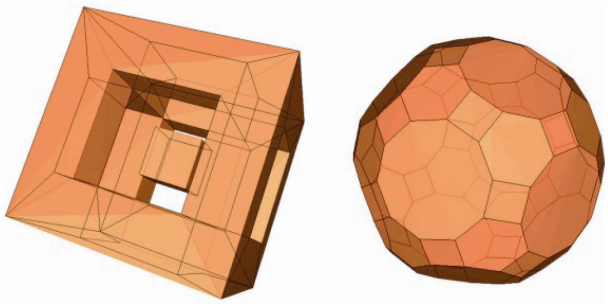

Figure 2. Test scenes.

Table 1. Average results obtained for scenes containing approximately 1000 polygons and a set of maximum 10 viewpoints.

Genetic operator	Initial visibility (%)	Final visibility (%)	Difference	Final number of points
crossove	63	75	12	7,625
mutation	67	81	14	9,875
both	66	88	22	10

Table 2. A comparison between results obtained when applying the genetic operators (crossover, mutation) on different proportions of the set of viewpoints.

% of viewpoint to do crossever	% of viewpoint to mutate	Initial visibility (%)	Final visibility (%)	Difference
50	50	66	88	22
40	60	66	88	22
70	30	60	90	30

6. Conclusion and Future Work

In this paper we have proposed a new method for evolving a group of random viewpoints into one that provides a better visibility of the considered scene. The novelty of our approach consists in using genetic algorithms for obtaining the new viewpoints. We have also introduced a definition of the quality of a set of viewpoints, that does not consider each point as a individual, but rather as part of the whole set.

This work leaves several interesting problems, such as providing new definitions for evaluating the view quality

of a single viewpoint, but in relation to its neighbours or with the whole group. This would allow for a better application of genetic algorithms and maybe for better results and faster results.

Moreover, it would be interesting to implement other methods for computing the visibility of viewpoints. In our work, a ray tracing algorithm was used. This lead to an increase in time complexity, as well as in a decrease in accuracy. In fact, there is a direct proportion between the computation time and the accuracy of the process. In order to have a more accurate degree of visibility, more rays need to be traced, which implies a higher time complexity. One possible solution to this problem would be the use of Z-buffer algorithm to determine an exact or approximate visibility for each polygon and each viewpoint.

REFERENCES

[1] E. Marchand and N. Courty, "Image-Based Virtual Camera Motion Strategies," *Graphics Interface Conference*, Montreal, May 2000.

[2] C. Colin, "A System for Exploring the Universe of Polyhedral Shapes," *Proceedings of Eurographics'*88, Nice, Vol. 11, 1988.

[3] D. Plemenos, M. Sbert and M. Feixas, "On Viewpoint Complexity of 3D Scenes," *International Conference on Computer Graphics and Vision*, Moscow, September 2004.

[4] P. P. Vazquez, M. Feixas, M. Sbert and W. Heidrich, "Viewpoint Selection Using Viewpoint Entropy," *Proceedings of the Vision Modeling and Visualization Conference*, Stuttgart, 21-23 November 2001.

[5] J. Louchet, "Using an Individual Evolution Strategy for Stereovision," *Genetic Programming and Evolvable Machines*, Kluwer Academic Publishers, Vol. 2, No. 2, 2001, pp. 101-109. doi:10.1023/A:1011544128842

[6] P. Barral, G. Dorme and D. Plemenos, "Visual Understanding of a Scene by Automatic Movement of Camera," *The* 9*th International Conference on Computer Graphics and Vision*, Moscow, 1999.

[7] P. Barral, G. Dorme and D. Plemenos, "Scene Understanding Using a Virtual Camera," *Eurographics'*2000, Interlagen, 20-25 August 2000.

[8] M. Feixas, "An Information Theory Framework for the Study of the Complexity of Visibility and Radiosity in a Scene," Ph.D. Thesis, University of Catalonia, 2002.

[9] T. Kamada and S. Kawal, "A Simple Method for Computing General Position in Displaying Three-Dimensionnal Objects," *Computer Vision*, *Graphics and Image Processing*, Vol. 41, No. 1, 1988, pp. 43-56. doi:10.1016/0734-189X(88)90116-8

[10] G. Turk, "Generating Random Points in a Triangle," In: A. Glassner, Ed., *Graphics Gems I*, AP Professional, 1990.

[11] H. Noser, O. Renault, D. Thalman and N. M Thalman, "Navigation for Digital Actors Based on Synthetic Vision, Memory and Vision, Memory and Learning," *Computer & Graphics*, Vol. 19, No. 1, 1995, pp. 7-19. doi:10.1016/0097-8493(94)00117-H

[12] D. Plemenos, J. Grasset, B. Jaubert and K. Tamine, "Intelligent Visibility-Based 3D Scene Processing Techniques for Computer Games," *International Conference on the Computer Graphics and Vision*, Novosibirsk, June 2005, pp. 84-91.

[13] D. Sokolov and D. Plemenos, "Viewpoint Quality and Scene Anderstanding," *The* 6*th International Symposium on Virtual Reality, Archaeology and Cultural*, Pisa, 8-11 November 2005, pp. 173-185.

[14] J. Louchet, M. Guyon, M. J. Lesot and A. Boumaza, "Guider un Robot Par Evolution Artificielle en Temps Réel," *Revue Extraction des Connaissances et Apprentissage et évolution*, Editions Hermès, December 2001, pp. 115-130.

[15] S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem," *Operations Research*, Vol. 21, No. 1, 1973, pp. 498-516. doi:10.1287/opre.21.2.498

