Heuristic search based improvements in computer graphics.

Dimitri PLEMENOS
University of Limoges, MSI laboratory
plemenos@unilim.fr

Abstract

Heuristic search is a well known Artificial Intelligence technique allowing to obtain satisfactory solutions for problems for which it is not possible to enumerate all the solutions. Several improvements can be done, using this technique, in computer graphics algorithms. Such improvements are presented in this paper in the area of radiosity, where the used evaluation function tries to approximate the visual complexity of a region of the scene, and in the area of 3D scene undrstanding, where evaluation function tries to incrementally evaluate both the position and the path of a virtual camera exploring a scene.

Keywords: *Artificial intelligence, heuristic search, radiosity, Monte Carlo sampling, scene understanding.*

1. INTRODUCTION

Computer graphics is an area where typically the number of possible solutions in many cases is very important or even infinite and it is not possible to enumerate all of them. In such cases, the use of heuristic search, a well known Artificial Intelligence technique, can improve results by permitting automatic computation of interesting solutions.

In Monte Carlo sampling based radiosity techniques, a well known problem is the problem of noise observed on the produced images, due to the stochastic nature of the sampling. From a given patch, all the regions of the scene receive almost the same number of rays. As the scene is not uniformly complex, some regions receive too many rays compared to their complexity while for other regions the number of received rays is not sufficient for an accurate sampling.

Another problem, often neglected by computer graphics researchers but very important in our opinion, is the problem of understanding a scene. The major part of scenes a computer graphics user has to manipulate are scenes created by other people and the knowledge the user has of these scenes is very coarse. As it is difficult to find a good view of a 3D scene, displayed on the (2D) screen of a computer, by changing, manually, the angle of view, it is important to develop methods allowing efficient exploration of the scene by a virtual moving camera.

The two above problems are of the same nature. In both cases, the problem to resolve is quite the same : how to guide the rays or the camera by automatically looking for a solution good enough according to specific criteria.

In section 2 of the paper, a brief presentation of the heuristic search technique will be made. In section 3, after a general presentation of Monte-Carlo radiosity, the main drawback of this technique will be explained. In section 4, heuristic search improvements of the Monte Carlo radiosity will be presented. Section 5 will be dedicated to the problem of understanding 3D scenes, while heuristic search based methods will be presented in section 6 to give satisfactory solutions to this problem.

2. THE HEURISTIC SEARCH

Artificial Intelligence uses 4 methods to resolve problems in any area:

1. If there exists a formula giving the solution of the problem, apply the formula.
2. If it is possible to enumerate all the possibilities of solution, enumerate them and evaluate each of them to verify whether it is really a solution or not.
3. If it is not possible to enumerate all the possibilities of solution, make a heuristic search to find interesting partial solutions.
4. If the problem is too complex to be resolved by one of the above methods, decompose it in a set of sub-problems and apply one of the 4 methods to resolve each of the sub-problems.

The use of heuristic search permits generally to find a solution good enough but not necessary the best solution. In a general manner, heuristic search uses the following steps from a given starting situation:

1. Choose an action among the possible actions. This choice can be based on experience, analysis of the situation, etc. or it can be a random choice.
2. Apply the action to modify the current situation.
3. Evaluate the situation using any method (numerical evaluation, analogy based evaluation, etc.).
4. If the current situation is a terminal one (a solution has been found for the problem), the heuristic search is finished. Otherwise, choose a new situation and apply heuristic search again.

The chosen new situation can be the last generated situation or it can be obtained by backtracking and selection of the most promising action.

Several strategies to control heuristic search can be applied.

3. THE MONTE CARLO RADIOSITY

In this section, some recalls will be made on the principle of Monte Carlo radiosity and its main drawbacks due to sampling problems.

3.1 PRINCIPLE OF THE MONTE CARLO RADIOSITY

The Monte Carlo based technique to compute radiosity tries to avoid explicit calculation of form factors. It is a shooting technique progressive technique, proposed by Shirley [1], where each patch of the scene distributes its energy to the remaining patches. A great number of rays are shot from randomly chosen points of the patch to directions chosen also randomly according to the cosine of the direction with the normal of the patch (see figure 1). The energy of the patch is distributed by the rays to the remaining parts of the scene, each ray transporting the same amount of energy.

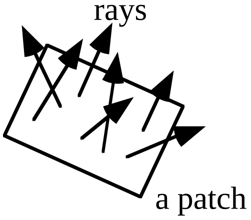

Figure 1: The energy of the patch is distributed by randomly chosen rays

The same process is repeated with all the patches of the scene, until complete distribution of the whole energy or until an image of good quality has been reached.

When using this technique, results are better if the next chosen patch is always the patch containing the greater amount of energy to distribute, because the image is more improved from one step to the next one.

In [2], Feda and Purgathofer introduced an additional loop to the classical Monte Carlo radiosity algorithm. At each step, additional rays are shot from each patch. If n is the number of rays shot until the step i and m the number of additional rays to be shot in the step i+1, the amount of energy already distributed and the amount of energy received by each patch are both divided by the number k=(n+m)/n before the beginning of the step i+1.

3.2 SAMPLING PROBLEMS WITH THE MONTE CARLO RADIOSITY

The Monte Carlo based radiosity computation is an elegant manner to compute the radiosity of a scene without having to explicitly compute form factors. However the Monte Carlo sampling, using rays to distribute the energy of each patch, is not entirely satisfactory because, on an average, the same number of rays is shot to all parts of the scene from a given patch.

Let us consider the scene of figure 2. As the scene is uniformly covered by the rays shot from the current patch, much more rays are shot to the part A of the scene than to the part B. For part B, it is even not sure that every patch will receive at least one ray.

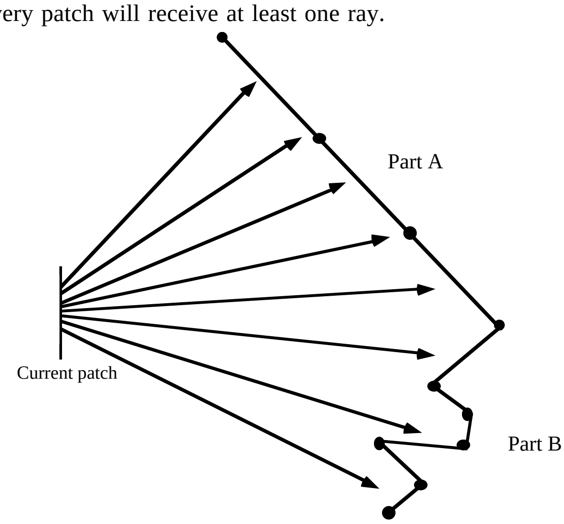

Figure 2: Monte Carlo sampling is not always the best one

In figure 3, as the car is stationed in front of the wall B, more rays should be shot against this wall than against the wall A, whose situation is much less complex, in order to obtain a satisfactory sampling of the whole scene.

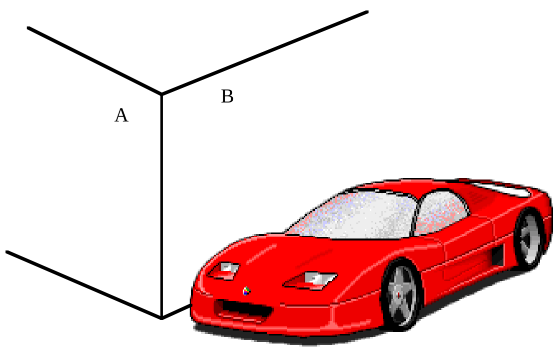

Figure 3: for an efficient sampling, more rays must be shot against the wall B

This sampling problem of the Monte Carlo radiosity may produce noisy images, especially in the case where the scene contains both simple and complex parts. In order to obtain a better distribution of the rays, it is necessary to have a mean to recognise the complexity of a region in a scene.

4. IMPROVEMENT OF THE MONTE CARLO RADIOSITY USING HEURISTIC SEARCH

In this section, a method of ray distribution, taking into account the complexity of a region of a scene, will be described. This method is based on heuristic search in the different regions of the scene, from each patch.

When searching to distribute rays in different regions according to the complexity of each region, it is necessary to have a criterion allowing to compute (or to estimate) the complexity of a region. In this paper, the problem of defining a complexity criterion for a region of a scene is not our problem. In other papers [6, 7, 8, 9], complexity criteria have been proposed and we are working on a new criterion which seems to us more accurate. However, the purpose of this paper is to describe a method permitting to divide a scene, seen from a patch, in regions of almost equal complexities and to distribute the rays in the scene according to the complexity of each region. The notion of complexity can be ambiguous. For our purpose, the only interesting complexity is the visual complexity, that is, the number of details seen in a region from a given patch.

Given a scene divided in patches, the first problem we have to face is to divide the scene in regions of almost equal visual complexity, for each patch. This part of the processing is performed during a preprocessing phase. It is the phase of the processing where heuristic search techniques are applied. The proposed method involves the following steps:

1. A hemisphere, divided in 4 spherical triangles, is associated with each patch of the scene (see figure 4).

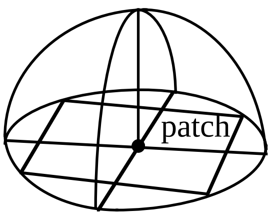

Figure 4: Initial subdivision of a hemisphere associated with a patch

2. At any phase of the processing, all spherical triangles of the hemisphere associated with each patch are processed independently on each other. Starting from the current situation, the visual complexity of the region of the scene contained in the pyramid defined by the centre of the patch and the current spherical triangle is measured and one of the following actions is done, according to the value of the visual complexity of the region:

   2.1 If the visual complexity of the region delimited by the centre of the patch and the current spherical triangle is greater than a threshold value, or if the maximum number of subdivisions fixed by the user is not reached, the spherical triangle is divided in 4 new spherical triangles (see figure 5) and the heuristic search starts again with each of the 4 new spherical triangles.

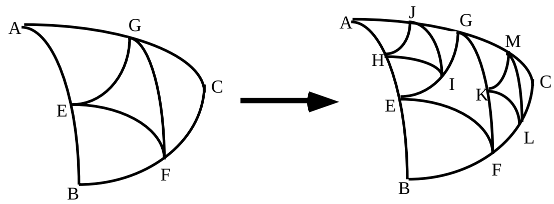

Figure 5: the spherical triangles AEG and CGF are divided in 4 new spherical triangles

   2.2 Otherwise, the heuristic search process is finished for the current spherical triangle of the hemisphere associated with the current patch.

The heuristic search based subdivision process is finished when no more subdivision is possible for any patch.

The second problem to resolve is the problem of distributing the rays shot from a patch according to the visual complexity of each region. In the ideal case, all spherical triangles of the hemisphere associated with a patch define regions with equal complexities. Unfortunately, it is generally not true because the cost of the region subdivision process greatly increases with each new subdivision level and must be stopped at a small number of levels (typically 3 or 4). Thus, the ray distribution method must take into account the visual complexity of each region. Unlike with the classical Monte Carlo sampling, where each ray contains the same amount of energy, the relationship between rays and transported energy is more complex in our case. The following rules are applied to determine the number of rays and the transported energy by a ray:

1. The number of rays shot in a region is proportional to the visual complexity of the region.
2. The amount of energy distributed in a region from a path is proportional to the area of the corresponding spherical triangle, according to the cosine of its average direction with the normal of the patch.
3. The amount of energy distributed by a ray in a region is proportional to the cosine of its direction with the normal of the patch.

Figure 6: detail of a reference scene obtained with classical Monte Carlo sampling at the end of the first step

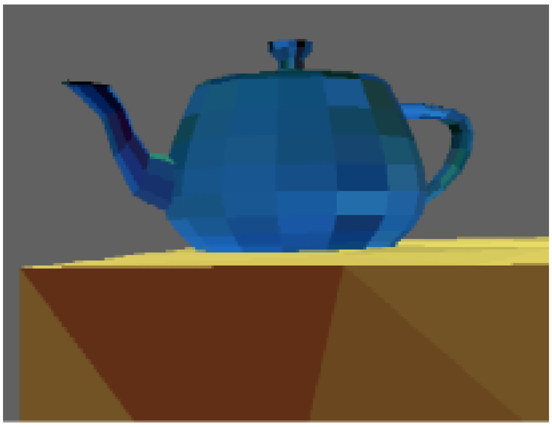

Figure 7: detail of a reference scene obtained with heuristic search based sampling at the end of the first step

Figures 6 and 7 illustrate improvements obtained with a heuristic search based sampling, compared to classical Monte Carlo sampling. Convergence with the new sampling is much faster.

5. THE PROBLEM OF UNDERSTANDING 3D SCENES

Complex 3D scenes are difficult to understand, especially scenes found on the web, because the scenes are three-dimensional and the screen two-dimensional and it is difficult to reach manually a good view position allowing to well understand them. The same problem is encountered when a scene is designed by a declarative modeller. As the scene is often described by means of high level properties, it is important to have a view of the scene showing these properties and permitting to evaluate their verification. Sometimes, even simple scenes are difficult to understand when the chosen point of view is bad, as seen in figure 7.

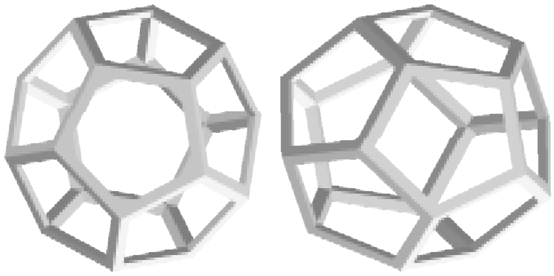

Figure 7: the same scene seen from a bad and from a good point of view

In order to understand a scene, designed with a modeller or found in the net, it is important to choose a view direction which shows its most important features. As such a view direction is very difficult to find interactively, it is very important that a scene modeller offers an automated computing of a good view direction. Indeed, the modeller has much more information about the scene than the user has and could use this information to automaticaly compute a good view direction. This is especially true for declarative modeling, where the designer has insufficient knowledge of the scene during the designing process.

Even if automatic computation by the modeller of a good view direction is very important for the user, it may be not sufficient to understand a complex scene. In such a case, it would be interesting to compute more than one points of view or, better, to compute a good trajectory for a virtual camera permitting to explore the scene.

6. HEURISTIC SEARCH BASED METHODS FOR UNDERSTANDING 3D SCENES

Several papers [3, 4, 5, 10, 12, 13, 14] have proposed criteria to estimate the quality of a point of view. Our purpose here is only to use a function giving an estimation of the quality of a given point of view in order to propose heuristic search techniques to compute a good point of view for a scene, as well as a good trajectory for a camera exploring the scene.

The first problem to face is the search of a good point of view. To resolve this problem, the scene is surrounded by a sphere representing the set of possible points of view. The 3 mains plans determine 8 initial spherical triangles on the surface of the sphere (see figure 8). The next problem is to choose the best initial spherical triangle, that is, the triangle where the final point of view will be computed. As the three main axes intersect the surface of the sphere in 6 points, the chosen spherical triangle is the one whose vertices are the three intersection points with the best values as points of view.

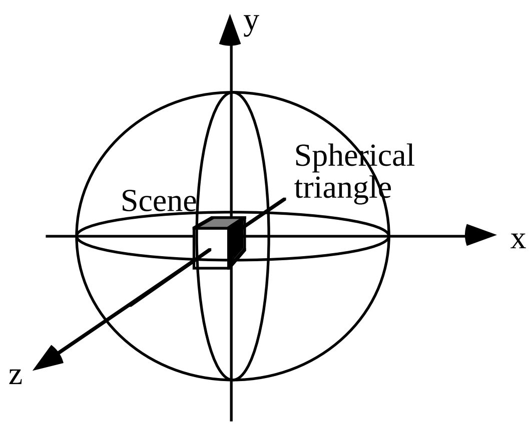

Figure 8: the scene is surrounded by a sphere divided in spherical triangles

Now, a good point of view has to be researched on the surface of the chosen spherical triangle. In this phase, heuristic search is introduced. The problem to resolve is the following: given the values of the evaluation function at the three vertices of a spherical triangle, determine a point of view, as good as possible, on the surface of this spherical triangle. Heuristic search is applied in the following manner, in many steps:

At any step, for a spherical triangle V1V2V3, if V1 is the vertex with the best value, the research of a good point of view is delimited in a new spherical triangle, V1M1M2, where M1 is the middle of the edge V1V2 and M2 the middle of the edge V1V3 (figure 9).

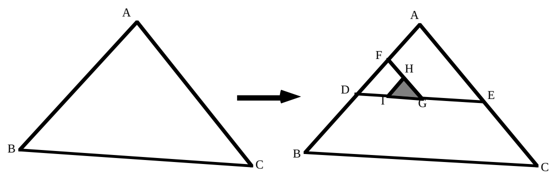

Figure 9: the heuristic search process to compute a good point of view

The choice of the next spherical triangle delimiting the search of a good point of view is based on the fact that a point of view, lying in the neighbouring of a good point of view, is probably a good point of view.

The heuristic search process is stopped after a fixed number of steps or when the best value of the vertices of the new triangle is worse than the corresponding value of the previous spherical triangle.

The second problem we have to face is the movement of a virtual camera exploring the scene. The criterion of "good position" for the camera has to take into account not only the quality of the position as a point of view but also the quality of the trajectory of the camera. As it is claimed above, our purpose is not to discuss about efficient evaluation functions for a position of the camera. Such functions have been already proposed and tested [12, 13, 14]. What we want to do here is to propose a heuristic search based incremental method to compute a good trajectory for the camera and taking into account an already defined evaluation function.

The sphere surrounding the scene (figure 8) is used again to materialise the possible positions of the camera. The camera starts its movement from an initial position which is a good point of view. From this initial position, the camera has 8 possibilities of movement (figure 10, on the left). As it is very important to have a smooth camera's movement, the camera will have only 3 possibilities of movement from any other position of the trajectory, depending on the current direction of the camera's movement (figure 10, on the right).

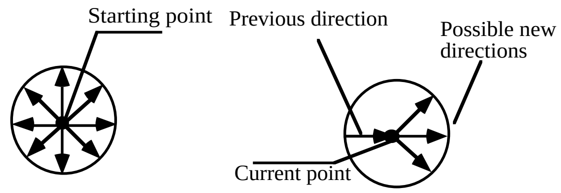

Figure 10: eight directions are possible for the camera from the starting point while only three directions are possible for any other position of the camera

The heuristic search process used to compute the trajectory of the camera is now the following:

1. The movement direction and the next position of the camera, for any current position, are determined from the value returned by the evaluation function for each possible next position.
2. If two or more possible next positions have equal quality values, the next position of the camera is randomly chosen among these positions.
3. The scene understanding process can be stopped when a given percentage of the scene has been explored. It can also be stopped by the user.

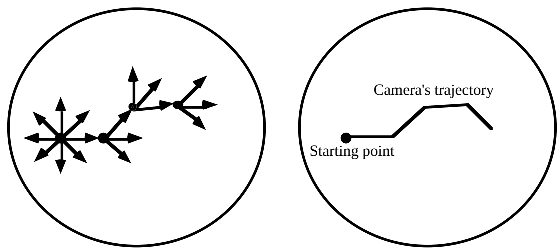

Figure 11: Heuristic search and camera's trajectory

Figure 11 shows incremental construction of the camera's trajectory from the possible choices at each step.

In figure 12, one can see the trajectory of a virtual camera around an office, incrementally computed with heuristic search.

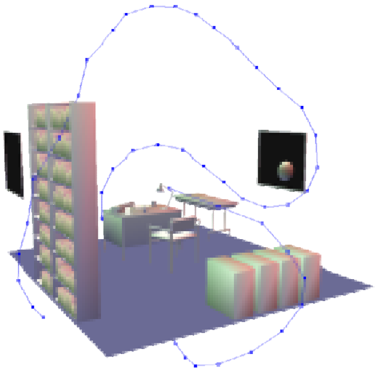

Figure 12: incremental computation of camera's trajectory using heuristic search

Other authors [11] have used a virtual camera to visit a scene but their purpose was quite different.

7. CONCLUSION AND FURTHER WORK

Heuristic search is a very interesting technique which can be applied to several areas of computer graphics. The use of this technique allows to find interesting approximated solutions to problems which cannot receive exact solutions. In this paper, we have presented two possible applications of heuristic search: Monte Carlo radiosity and scene understanding. We think that many other computer graphics areas can be improved by using heuristic search.

Of course, heuristic search cannot be efficient if the used evaluation functions are not good enough to characterise with precision a given situation. The evaluation functions we have defined to measure the visual complexity of a scene or the quality of a point of view or of a camera's trajectory are efficient enough and the obtained results are very satisfactory. We are currently working in definition of evaluation functions permitting to compute trajectories using global plans.

We have also faced the problem of exploration of a scene with a virtual camera moving **inside** the scene. We are also using heuristic search techniques but the used evaluation function has to be improved because there are several parameters to take into account (obstacles, vision field, etc.).

7. REFERENCES

[1] SHIRLEY P., Radiosity via Raytracing. Graphics Gem II, James Arvo editor, pp. 306-310, Academic Press, San Diego, 1991.

[2] FEDA M., PURGATHOFER W., Progressive ray refinement for Monte Carlo Radiosity,4th Eurographics Workshop On Rendering, pp. 15-25, June 1993.

[3] COLIN C., Automatic computing of good views of a scene, MICAD'90, Paris (France), February 1990 (in French).

[4] KAMADA T., KAWAI S., A Simple Method for Computing General Position in Displaying Three-dimensional Objects, Computer Vision, Graphics and Image Processing, 41 (1988).

[5] PLEMENOS D., Contribution to the study and development of techniques of modelling, generation and display of scenes. The MultiFormes project., Professorial dissertation, Nantes (France), November 1991 (in French).

[6] PLEMENOS D., PUEYO X., Heuristic Sampling Techniques for Shooting Rays in Radiosity Algorithms., 3IA'96 International Conference, Limoges (France), April 3-4, 1996.

[7] JOLIVET V. PLEMENOS D., The hemisphere subdivision method for Monte Carlo radiosity, GraphiCon'97, Moscow, May 21-24, 1997.

[8] JOLIVET V., PLEMENOS D., SBERT M., A pyramidal hemisphere subdivision method for Monte Carlo radiosity, Eurographics'99 (short paper), Milano (Italy), September 1999.

[9] JOLIVET V., PLEMENOS D., SBERT M., Pyramidal hemisphere subdivision radiosity. Definition and improvements, proceedings of the WSCG'2000 International Conference, Plzen (Czech Republic), February 2000.

[10] PLEMENOS D., BENAYADA M., Intelligent display in scene modelling. New techniques to automatically compute good views., GraphiCon'96, Saint Petersburg, July 1996.

[11] F. JARDILLIER, E. LANGUENOU, Screen-Space Constraints for Camera Movements: the Virtual Cameraman., Computer Graphics Forum, Volume 17, number 3, 1998.

[12] BARRAL P., DORME G., PLEMENOS D., Visual understanding of a scene by automatic movement of a camera, GraphiCon'99, Moscow (Russia), August 26 - September 1, 1999.

[13] BARRAL P., DORME G., PLEMENOS D., Visual understanding of a scene by automatic movement of a camera, Eurographics 2000, short paper.

[14] BARRAL P., DORME G., PLEMENOS D., Intelligent scene exploration with a camera, International conference 3IA'2000, Limoges (France), May 3-4, 2000.

Author:

Dimitri PLEMENOS is full professor at the University of Limoges. His research area is computer graphics and, more precisely, the use of artificial intelligence techniques in computer graphics. He is the director of the MSI research laboratory.

Dimitri PLEMENOS
University of Limoges
MSI Laboratory
83, rue d'Isle
87000 Limoges
France
Phone: (+33) 555 43 69 74
Fax: (+33) 555 43 69 77
E-mail: plemenos@unilim.fr