Exploring virtual worlds: current techniques and future issues

Dimitri PLEMENOS
University of Limoges, MSI laboratory, 83, rue d'Isle, 87000 Limoges (France)

Abstract

The purpose of this paper is to discuss about existing methods allowing to explore virtual worls for a human user. Clearly, we are interested in methods allowing a real human user to understand more or less complex unknown virtual worlds. The main techniques corresponding to our criteria are presented and their main qualities and drawbacks are commented. Finally, taking into account the qualities and drawbacks of the presented methods, several improvement ideas are proposed.

Keywords: *Virtual worlds, exploration techniques, virtual camera, good view, heuristic search, viewpoint entropy.*

1. INTRODUCTION

Virtual worlds exploration techniques become nowadays more and more important. When, more than 10 years ago, we have proposed the very first methods permitting to improve the knowledge of a virtual world [2, 3], many people thought that it was not an important problem. People begun to understand the importance of this problem and the necessity to have fast and accurate techniques for a good exploration and understanding of various virtual worlds, only during these last years. However, there are very few papers which face this problem from the computer graphics point of view, whereas several papers have been published on the robotics artificial vision problem.

The purpose of a virtual world exploration in computer graphics is completely different from the purpose of techniques used in robotics. In computer graphics, the purpose of the program which guides a virtual camera is to allow a human being, the user, to understand a new world by using an automatically computed path, depending on the nature of the world. The main interaction is between the camera and the user, a virtual and a human agent and not between two virtual agents or a virtual agent and his environment.

There are two kinds of virtual worlds exploration. The first one is global exploration, where the camera remains outside the world to be explored. The second kind of exploration is local exploration. In such an exploration, the camera moves inside the scene and becomes a part of the scene. Local exploration can be useful, and even necessary, in some cases but only global exploration can give the user a general knowledge on the scene. In this paper we are mainly concerned by global virtual world exploration, whereas interesting results have been obtained with local exploration techniques. In any case, the purpose of this paper is visual exploration of fixed unchanging virtual worlds.

The paper is organised in the following manner: In section 2 we try to justify the need of virtual worlds exploration. In section 3 we present some early techniques for understanding simple virtual worlds, whereas in section 4 some very late sophisticated techniques for complex virtual worlds exploration are presented. In section 5 we will expose our ideas about the future evolution of virtual worlds exploration techniques. Finally, we will conclude is section 6.

2. WHY EXPLORE VIRTUAL WORLDS?

Exploring virtual world, seen as a computer graphics problem, is quite different from the robotics problem where an autonomous virtual agent moves inside a virtual world whose it is a part. In computer graphics, the main purpose of a virtual world exploration is to allow the user to understand this world. Here, the image of the world seen from the current point of view is very important because it will permit the user to understand the world. In robotics, the problem is different and the use of an image is not essential because the virtual agent is not a human beeing.

At about the second half of the 80's, we have localised a problem which appeared to us as very important. Very often, the rendering of a scene, obtained after a long computation time, was not possible to exploit because the choice of the angle of view was bad. In such a case, the only possibility was to choose another angle of view and to try again by running the time consuming rendering algorithm once again. We thought that it is very difficult to find a good angle of view for a 3D scene when the working interface is a 2D screen. On the other hand, we thought that the choice of a angle of view is very important for understanding a scene. In figure 1, one can see two different views of the same scene. Only the second one could help to understand it.

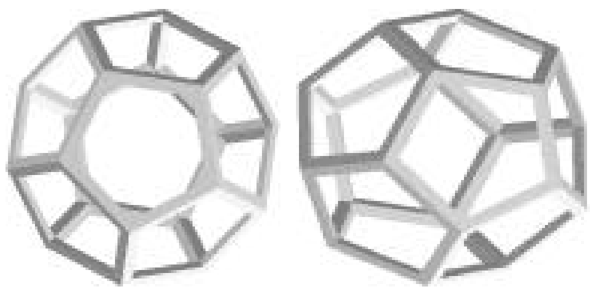

Figure 1: Bad and good angle of view for a scene

As the program (modeller, renderer) has a full knowledge of the geometry of the world to visualise, we thought that it could be more interesting to ask the program to find a good angle of view for this world.

Obviously, it is not possible to try all the possible angles of view and to choose the best one because the camera's movement space is continuous. Even if we decide to sample the camera's space, we are typically in a case where there are many possibilities of solution and we need a method to choose a good enough solution. In such cases the use of artificial intelligence technique of heuristic search is recommended.

This simple computer-aided way to understand a virtual world is not always sufficient. The world to understand can be very complex and, in such a case a single point of view is often insufficient. Rather than computing several good points of view and showing the world from each of them, we think that it is much more interesting for the user to use a moving virtual camera in order to permit him to explore the world by computing an interesting path for the camera.

3. SIMPLE VIRTUAL WORLD UNDERSTANDING

The very first works in the area of understanding virtual worlds were published at the end of 80's and the beginning of 90's. There were very few works because the computer graphics community was not convinced that this area was important for computer graphics. The purpose of these works was to offer the user a help to understand simple virtual worlds by computing a good point of view. In this section we will use the term of scene to point out a simple virtual world.

3.1 Non degenerated view

Kamada et al. [2] consider a direction as a good point of view if it minimises the number of degenerated images of objects when the scene is projected orthogonally. A degenerated image is an image where more than one edges belong to the same straight line. In figure 2, on can see a good view of a cube together with degenerated views of it.

The used method avoids the directions parallel to planes defined by pairs of edges of the scene.

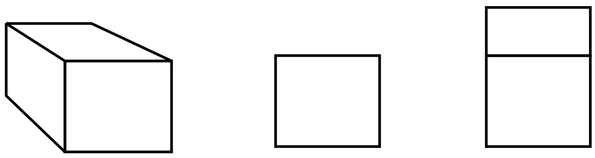

Figure 2: good and degenerated views

If L is the set of all the edges of the scene and T the set of unit normal vectors to planes defined by couples of edges, let's call $\vec{P}$ the unit vector of the direction of view to be computed.

In order to minimise the number of degenerated images, the angles of the vector $\vec{P}$ with the faces of the scene must be as great as possible. This means that the angles of the vector $\vec{P}$ with the elements of T must be as small as possible. The used evaluation function is the following:

$$f(\vec{P}) = \mathrm{Min}_{\vec{t} \in T}(|\vec{P} \cdot \vec{t}|)$$

As the purpose is to minimise the angle between $\vec{P}$ and $\vec{t}$, we must maximise the angle between these two vectors. So, we must maximise the function $f(\vec{P})$. To do this, a vector $\vec{P}$ which minimises the greater angle between itself and the elements of T must be found.

In order to compute this vector, the authors have proposed two methods. The first one is simple and its main advantage is to be very fast. The second method is more precise because it is based on the computation of all the normal vectors to all the faces of a scene and all the angles between these vectors. This method is time consuming. Only the first method will be explained here.

As the purpose of this method is to decrease the computation cost of the function $f(\vec{P})$ for all the unit normal vectors, a set E of uniformly distributed unit vectors is chosen on the unitary sphere defined at the centre of the scene.

The function $f(\vec{P})$ is computed for each element of the set E and the vector with the maximum value is chosen.

This method is interesting enough due to its rapidity. Its precision depends on the number of elements of the set E.

The technique proposed by Kamada is very interesting for a wire-frame display. However it is not very useful for a more realistic display. Indeed, this technique does not take into account visibilities of the elements of the considered scene and a big element of the scene may hide all the others in the final display.

3.2 Direct approximate viewpoint calculation

Colin [4] has proposed a method initially developed for scenes modelled by octrees. The purpose 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. This principle can be described as follows:

1. Choose the three best directions of view among the 6 directions corresponding to the 3 coordinates axes passing through the centre of the scene.
2. Compute a good direction in the pyramid defined by the 3 chosen directions, taking into account the importance of each one of the chosen directions.

How to estimate the importance of a direction of view? What is a good criterion to estimate the quality of a view? The idea is to maximise the number of visible details. A direction of view is estimated better than another one if this direction of view allows to see more details than the other.

In the method of Colin, each element of the scene owns a "weight" with reference to a point of view. This weight is defined as the ratio between the visible part of the element and the whole element. It allows to attribute an evaluation mark to each direction of view and to use this mark to choose the three best directions.

When the three best directions of view d1, d2 and d3 have been chosen, if m1, m2 and m3 are the corresponding evaluation marks, a "good" direction of view can be computed by linear interpolation between the three chosen directions of view, taking into account the respective evaluation marks. So, if m1>m2>m3, the final "good" direction is computed (figure 3) by rotating the initial direction d1 to the direction d2 by an angle ϕ1=90*m1/(m1+m2), then to the direction d2 by an angle ϕ2=90*m1/(m1+m3).

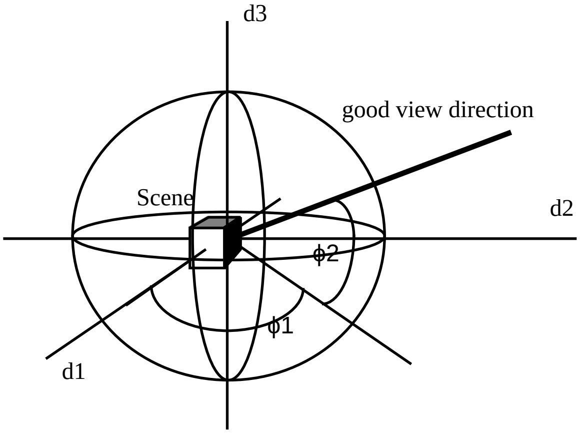

Figure 3: direct approximate computation of a good direction of view

3.3 Iterative viewpoint calculation

The good point of view computing method proposed by Plemenos [2, 3] was developed and implemented in 1987 but it was first published only in 1991.

The good view criterion used by this method is the number of visible details combined with the projected area of the visible parts of the scene. More precisely, the importance of a point of view will be computed using the following formula:

$$I(V) = \frac{\sum_{i=1}^{n} [\frac{P_i(V)}{P_i(V)+1}]}{n} + \frac{\sum_{i=1}^{n} P_i(V)}{r}$$

where: I(V) is the importance of the view point V,
$P_i(V)$ is the projected visible area of the polygon number i obtained from the point of view V,
r is the total projected area,
n is the total number of polygons of the scene.

In this formula, [a] denotes the smallest integer, greater than or equal to a.

In practice, these measures are computed in a simple manner, with the aid of graphics hardware using OpenGL [5, 11]. A different color is assigned to every face, an image of the scene is computed using integrated z-buffer and a histogram of the image is computed. This histogram gives all information about the number of visible polygons and visible projected area of each polygon.

The process used to determine a good point of view works as follows:

The points of view are supposed to be on the surface of a virtual sphere whose the scene is the centre. The surface of the sphere of points of view is divided in 8 spherical triangles (figure 4).

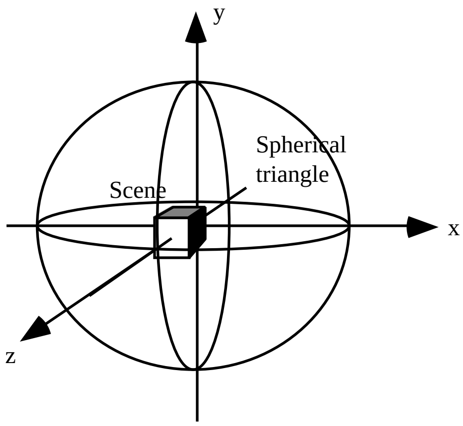

Figure 4: sphere divided in 8 spherical triangles

The best spherical triangle is determined by positioning the camera at each intersection point of the three main axes with the sphere and computing its importance as a point of view. The three intersection points with the best evaluation are selected. These three points on the sphere determine a spherical triangle, selected as the best one.

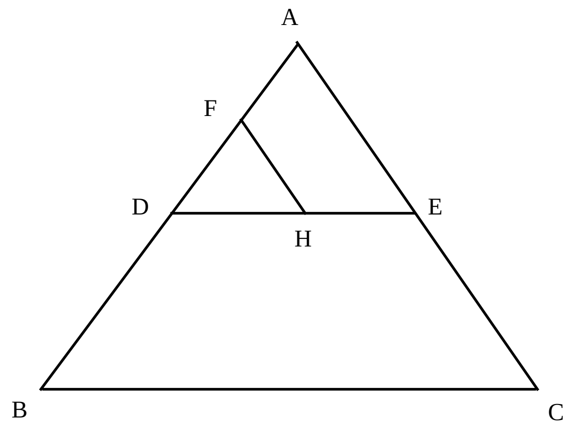

Figure 5: Heuristic search of the best point of view by subdivision of a spherical triangle

The next problem to resolve is selection of the best point of view on the best spherical triangle. The following heuristic search technique is used to resolve this problem:

If the vertex A (figure 5) is the vertex with the best evaluation of the spherical triangle ABC, two new vertices E and F are chosen at the middles of the edges AB and AC respectively and the new spherical triangle ADE becomes the current spherical triangle. This process is recursively repeated until the quality of obtained points of view does not increase. The vertex of the final spherical triangle with the best evaluation is chosen as the best point of view.

3.4 Good view and information theory

Sbert et al. [6] proposed to use information theory in order to establish an accurate criterion for the quality of a point of view. A new measure is used to evaluate the amount of information captured from a given point of view. This measure is called *viewpoint entropy*. To define it, the authors use the relative area of the projected faces over the sphere of directions centred in the point of view (figure 6).

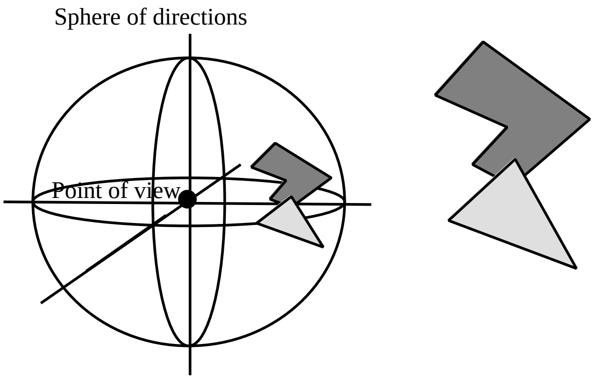

Figure 6: the projected areas of polygons are used as probability distribution of the entropy function

The viewpoint entropy is then given by the formula:

$$H_p(X) = - \sum_{i=0}^{N_f} \frac{A_i}{A_t} \log \frac{A_i}{A_t}$$

where N_f is the number of faces of the scene, A_i is the projected area of the face i and A_t is the total area covered over the sphere.

The maximum entropy is obtained when a viewpoint can see all the faces with the same relative projected area A_i/A_t. The best viewpoint is defined as the one that has the maximum entropy.

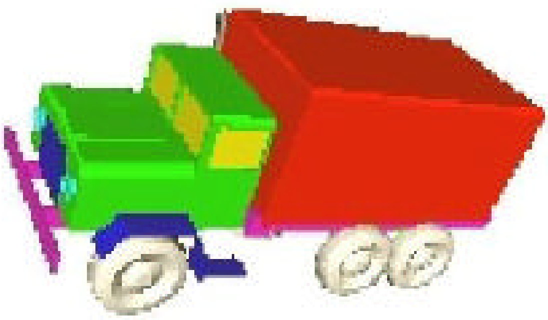

Figure 7: point of view based on viewpoint entropy

To compute the viewpoint entropy, the authors use the technique proposed in [5], based on the use of graphics hardware using OpenGL.

The selection of the best view of a scene is computed by measuring the viewpoint entropy of a set of points placed over a sphere that bounds the scene. The point of maximum viewpoint entropy is chosen as the best one. Figure 7 presents an example of results obtained with this method.

4. VIRTUAL WORLD EXPLORATION

When we have to understand a complex virtual world, the knowledge of a single point of view is not enough to understand it. Computing more than one points of view is generally not a satisfactory solution in most cases because the transition from a point of view to another one can disconcert the user, especially when the new point of view is far from the current one. Of course, the knowledge of several points of view can be used in other areas of computer graphics, such as image-based modelling and rendering [10, 14] but it is not suitable for virtual world understanding. The best solution, in the case of complex virtual worlds is to offer an automatic exploration of the virtual world by a camera that chooses its path according to the specificities of the world to understand.

4.1 Incremental outside exploration

An important problem in automatic virtual world exploration is to make the camera able to visit the world to explore by using good points of view and, at the same time, by choosing a path that avoids brusque changes of direction.

In [5, 7] an initial idea of D. Plemenos and its implementations are described. The main principle of the proposed virtual world exploration technique is that the camera's movement must apply the following heuristic rules:

- It is important that the camera moves on positions which are good points of view.
- The camera must avoid fast returns to the starting point or to already visited points.
- The camera's path must be as smooth as possible in order to allow the user to well understand the explored world. A movement with brusque changes of direction is confusing for the user and must be avoided.

In order to apply these heuristic rules, the next position of the camera is computed in the following way:

- The best point of view is chosen as the starting position for exploration.
- Given the current position and the current direction of the camera (the vector from the previous to the current pasition), only directions insuring smooth movement are considered in computing the next position of the camera (figure 8).

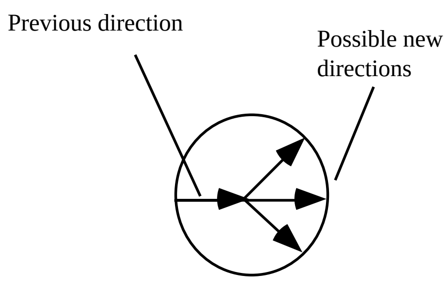

Figure 8: Only 3 directions are considered for a smooth movement of the camera

- In order to avoid fast returns of the camera to the starting position, the importance of the distance of the camera from the starting position must be inversely proportional to path of the camera from the starting to the current position (figure 9).

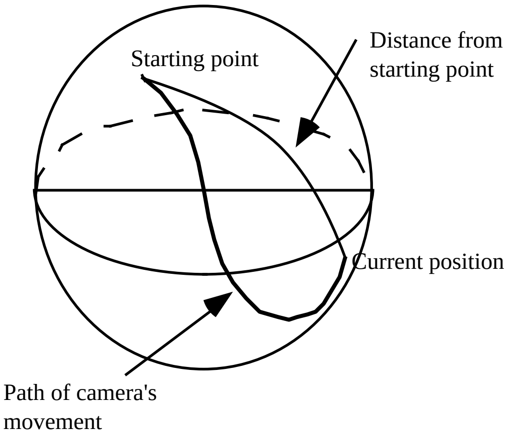

Figure 9: Distance of the current position of the camera from the starting point.

Thus, the following evaluation function is used to evaluate the next position of the camera on the surface of the sphere:

$$w_c = \frac{n_c}{2}\,(1+\frac{d_c}{p_c})$$

In this formula:

- w_c is the weight of the current camera position,
- n_c is the global evaluation of the camera's current position as a point of view,
- p_c is the path traced by the camera from the starting point to the current position,
- d_c is the distance of the current position from the starting point.

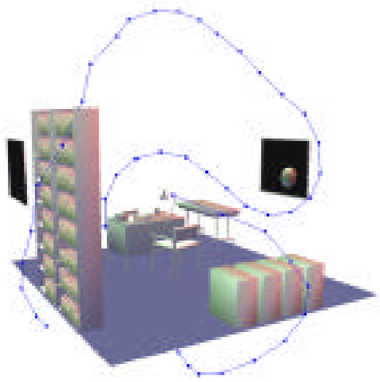

Figure 10: exploration of a virtual office by incremental outside exploration.

Several variants of this technique have been proposed and applied. In figure 10 one can see an example of exploration of a simple virtual world representing an office.

4.2 Viewpoint entropy-based exploration

In [9] and [10], the authors propose two methods of virtual world exploration: an outside and an indoor exploration method. The two methods are based on the notion of viewpoint entropy.

The *outside exploration method* is inspired from the incremental outside exploration described in the previous subsection. The camera is initially placed at a random position on the surface of a bounding sphere of the world to explore. An initial direction is randomly chosen for the camera. Then, the next position is computed in an incremental manner. As in [5, 7], only three possible next position are tested, in order to insure smooth movement of the camera (figure 11).

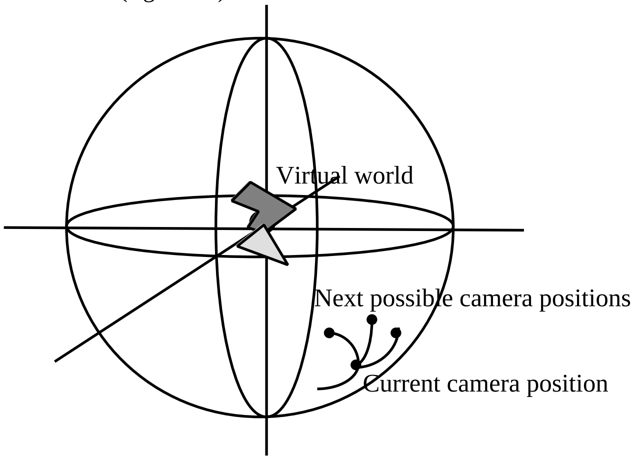

Figure 11: three possible next positions evaluated

An evaluation function computes the best next position of the camera, taking into account only the viewpoint entropy of the not yet visited polygons of the scene. The exploration is finished when a given ratio of polygons have been visited. The retained criterion of position's quality seems good enough and the presented results seem interesting. However, no measure is available to compare this method to the one of sub-section 4.1. Figure 12 shows viewpoint entropy-based exploration of a glass.

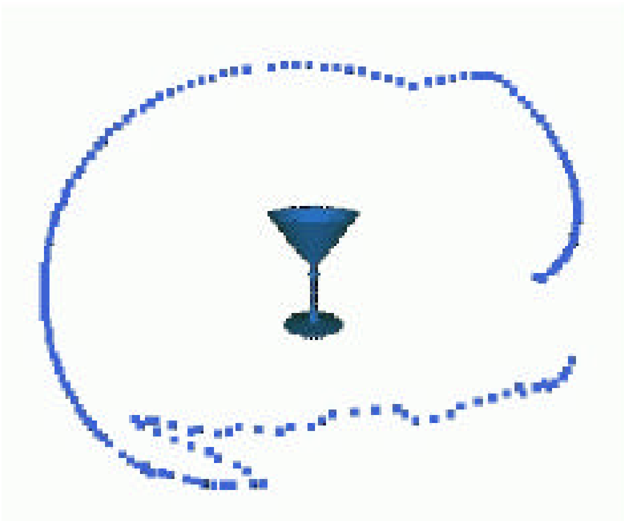

Figure 12: viewpoint entropy-based exploration

The *indoor exploration method* is also an incremental exploration technique. The camera is initially placed inside the virtual world to explore and it is supposed to move in this closed environment using each time one of the three following possible movements: *move forward*, *turn left* and *turn right*. Moreover, the camera is restricted to be at a constant height from the floor.

During the virtual world exploration, the path of the camera is guided by the viewpoint entropy of the polygons not yet visited. The next movement of the camera, selected from all the possible movements, is the one with maximum viewpoint entropy coming from the not yet visited polygons. In order to know, at each moment the polygons not yet visited, an array is used to encode the already visited polygons.

The exploring algorithm is stopped when an important part of the polygons of the virtual world has been visited. This threshold value is estimated by the authors to be of about 80% or 90% of the polygons of the explored world.

The main drawback of the method is that collisions are taken into account very roughly by the evaluation function of the camera movement.

4.3 Other methods

There are really very few papers on the virtual world exploration area which face the problem with the computer graphics point of view. In [12] image-based techniques are used to control the camera motions in a possibly changing virtual world. The problem faced in the paper is adaptation of the camera to the changes of the world. In [8] the authors propose a method allowing to obtain a set of camera motions and virtual world motions satisfying user-defined constraints specified in an easy declarative manner. Both approaches are interesting but their purpose is quite different from what we consider as virtual world exploration techniques, because they are concerned by moving objects. The problems they try to resolve are not really computer graphics problems but rather robotics ones.

5. FUTURE ISSUES

Virtual worlds exploration techniques are very useful techniques for the virtual world designer as well as for the user wishing to understand a complex scene found on the Internet or to visit a virtual site as reconstructed ancient cities, museums, etc. We think that indoor virtual world exploration is unable to give the user a global view of the explored world. It must be used only when some parts of the explored world are not accessible with an outside exploration technique.

Two kinds of virtual world exploration can be envisaged in the future:

1. Real time *on-line* exploration, where the virtual world is visited for the first time and the path of the camera is determined in incremental manner. In this case, it is important to have fast exploring techniques in order to allow the user to understand in real time the explored world.

2. *Off-line* exploration, where the user does not visit the virtual world together with the camera. The virtual world is found and analysed by the program guiding the camera's movement, in order to determine interesting points to visit and path(s) linking these points. The user will visit the virtual world later, following the already determined path. In such a case, it is less important to use fast techniques to determine the camera's path.

5.1 On-line exploration of virtual worlds

In on-line exploration it could be interesting to elaborate plans, instead of using purely incremental exploration. For example, for some scenes it could be known that the camera must reach some pre-defined points. In such a case, an intermediate goal of the camera could be to reach one of these points, while applying the other criteria to determine its path from the current point of view.

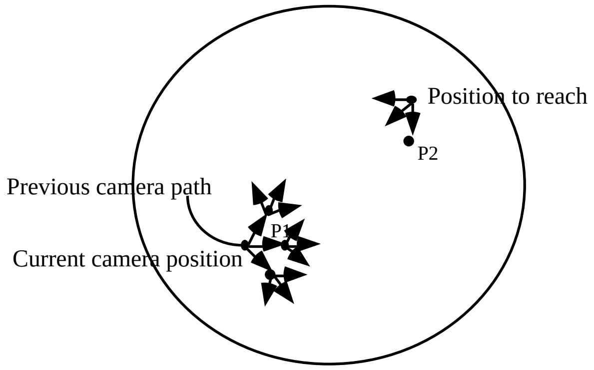

Figure 13: Plan-based exploration of virtual worlds

In figure 13, the camera has to reach a pre-defined position. In order to reach this position, a next position P1 of the camera and an intermediate position P2 to reach are computed as close as possible to each other. Encouraging results have be obtained in this manner in the MSI laboratory of the University of Limoges.

5.2 Off-line exploration of virtual worlds

In off-line exploration, the process could be decomposed in two steps: computation of a minimal number of points of view for the world to be explored and computation of an optimal path for the camera, taking into account the computed points of view.

1. Computation of a minimal number of points of view. The number of computed points of view should be enough to well understand the virtual world to explore. There are two ways to compute these points. The first one computes a big number of points of view on a sphere surrounding the virtual world and then suppresses any point of view that does not add visual information about the world to explore. The second way to compute a minimal number of points of view directly computes new points of view on the surface of the surrounding sphere, which adds visual information about the world to explore. The process is finished when no additional visual information is added. This can be done by using heuristic comparison techniques and recursive sphere subdivision.
2. Computation of an optimal path for the camera. A way to compute an optimal path for the camera is to choose as starting position one of the points to visit and then try to reach successively the remaining points in an order determined by two criteria: distance from the current

position and quality of view.

6. CONCLUSION

We have presented some techniques used to explore virtual worlds. We have chosen to present the most interesting among the proposed techniques and only those whose purpose is exploration for computer graphics use, that is techniques allowing a human user to understand the virtual word visited by a camera.

Two kinds of techniques have been presented. The first group of techniques is based on the computation of a good point of view and allows to understand relatively simple scenes. The purpose of the second group of techniques is to understand or to visit complex virtual worlds, where a single point of view is not enough to understand them. These techniques are generally based on the movement of a camera around or inside the world to explore.

Some future issues have been proposed in order to improve the exploration process which can be on-line or off-line. Some of these proposals have been implemented and first results are available. They will be presented in more details in other publications, when they will be well tested.

Currently, does not exist any measure to compare various exploration techniques. Comparison is only visual. We think that it would be very interesting to define a kind of distance, permitting to compare all the proposed techniques by computing the distance of the computed camera paths from an ideal path. Another idea, allowing comparisons between various exploration techniques, would be automatic or semi-automatic generation of test virtual worlds with well chosen properties, in order to permit easy visual comparisons. For example, a sphere with some holes, containing a more or less complex virtual sub-world, would be an interesting test virtual world.

Of course, a lot of techniques presented in this paper can easily be adapted to give solutions to the automatic light source placement problem. Indeed, some papers [3, 13] have proposed partial solutions to some aspects of this problem. However, the light source placement problem has first to be well formulated before being able to say what part of the problem is addressed by a proposed solution.

7. REFERENCES

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

[2] PLEMENOS D., A contribution to the study and development of scene modelling, generation and visualisation techniques. The MultiFormes project., Professorial dissertation, Nantes (France), November 1991.

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

[4] COLIN C., A System for Exploring the Universe of Polyhedral Shapes., Eurographics'88, Nice (France), September 1988.

[5] BARRAL P., DORME G., PLEMENOS D., Visual understanding of a scene by automatic movement of a camera, GraphiCon'99, Moscow (Russia), August 26 - September 3, 1999.

[6] SBERT M., FEIXAS M., RIGAU J., CASTRO F., VAZQUEZ P.-P., Applications of the information theory to computer graphics, International Conference 3IA'2002, Limoges (France), May 14-15, 2002.

[7] BARRAL P., DORME G. PLEMENOS D., Scene understanding techniques using a virtual camera, Short paper, Eurographics'2000, Interlagen (Switzerland), August 20 - 25, 2000.

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

[9] VAZQUEZ P.-P., SBERT M., Automatic indoor scene exploration, proceedings of the International Conference 3IA'2003, Limoges (France), May 14-15, 2003.

[10] VAZQUEZ P.-P., On the selection of good views and its application to computer graphics., PhD Thesis, Barcelona (Spain), May 26, 2003.

[11] DORME G., Study and implementation of 3D scene understanding techniques., PhD thesis, University of Limoges (France), June 2001. In French.

[12] MARCHAND E., COURTY N., Image-bases virtual camera motion strategies., Graphics Interface Conference, GI'2000, Montreal (Canada), May 2000.

[13] GUMHOLD S., Maximum entropy light source placement., IEEE Visualization, October 27 - November 1, 2002.

[14] VAZQUEZ P.-P., FEIXAS M., HEIDRICH W., Image-based modeling using viewpoint entropy. In J. Vince and R. A. Earnshaw, editors, Advances in Modelling, Animation and Rendering (proc. Computer Graphics International), pages 267-279, Springer 2002. ISBN 1-85233-654-4.

About the author

Dimitri PLEMENOS is a full professor at the University of Limoges (France) and director of the MSI research laboratory. His research area is intelligent techniques in computer graphics.
plemenos@unilim.fr