Intelligent scene display and exploration

Dimitri PLEMENOS, Dmitry SOKOLOV
University of Limoges, XLIM laboratory, 83, rue d'Isle, 87000 Limoges (France)
plemenos@unilim.fr, s@skisa.org

Abstract

In this paper we present the main current techniques allowing intelligent scene display and exploration. These techniques are based on estimation of the visual pertinence of a view. Various geometric criteria, such as number of visible polygons, visible projected area of the scene, curvature or mesh salience are used to estimate visual pertinence. Other techniques try to take into account non-geometric criteria, such as lighting. Current techniques for estimation of visual pertinence of a view give satisfactory results, even if lighting parameters are not yet well integrated in estimation process.

Keywords: Visual pertinence of a view, Good point of view, Virtual world exploration, Computer games, Mesh saliency, Lighting.

1 Introduction

More and more complex virtual worlds may nowadays be discovered on the web and it is generally difficult to well understand these worlds without a tool able to choose a good view for each world and even to allow to explore it with a virtual camera.

This kind of tools, allowing a good visual understanding of a virtual world, are based on techniques able to evaluate the pertinence of a view. The very first of these techniques appeared at the end of the 80's but their importance was not well understood in their time.

With the fast development of computers capabilities this last decade, the problem of well understanding complex virtual worlds becomes more and more crucial and several recent papers try to take into account this problem.

Even if the real human perception is not perfectly taken into account with current visual pertinence evaluation techniques, these techniques give generally results good enough allowing to apply these techniques in several areas, such as computer games, virtual museums visiting, molecules visualization or realistic rendering.

How to evaluate the pertinence of a view? What are the main elements to take into account in this evaluation? It is generally admitted that geometry and topology of a virtual world are important elements to take into account but the problem is how to do it. The problem is difficult to resolve because its solution consists to quantify the human perception of an image.

On the other hand, the only knowledge of geometry and topology of a virtual world is probably not enough to allow precise quantification of the human perception. If the virtual world is illuminated, it is important to take into account illumination of its elements in evaluation of visual pertinence. Everybody knows that, even if there are lots of pertinent details in a dark room, no one of them is visible and it is not pertinent to choose a point of view allowing to see inside the dark room.

In this paper we will present the main techniques proposed until now to estimate the visual pertinence of a view for a virtual world, in order to well understand it. We will also present some ideas to improve existing techniques. It is supposed here that the camera always remains outside the scene. The term "visual world" will be sometimes used to point at a complex scene.

This paper will be organized as follows: In section 2 the main geometry-based proposed techniques for estimating the visual pertinence of a view and for understanding a virtual world will be presented. In section 3, new techniques based on the concept of viewpoint complexity will be presented. In this section, old viewpoint complexity criteria will be completed with new ones and new methods for virtual world understanding will be described. In section 4 the notion of mesh salience will be developed. In section 5 we will propose methods to take into account not only the geometry of a virtual world but also its lighting, that is the light source placement, when estimating the visual pertinence of a view. In section 6 conclusions on current visual pertinence and virtual world estimation techniques will be given, as well as some indication on possible future work.

2 Geometry-based techniques

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.

2.1 Best view computing for single display

Kamada et al. [Kamada and Kawai 1988] consider a position as a good point of view if it minimizes 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.

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

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 minimize 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:

$$f(\vec{P}) = \min_{\vec{t}\in T} \left|\vec{P}\cdot\vec{t}\right|.$$

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

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 center 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.

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.

The good point of view computing method proposed by Plemenos [Plemenos 1991; Plemenos and Benayada 1996] 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 equation:

$$I(V) = \frac{\sum_{i=1}^{n} \left\lceil \frac{P_i(V)}{P_i(V)+1} \right\rceil}{n} + \frac{\sum_{i=1}^{n} P_i(V)}{r}, \qquad (1)$$

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 equation, $\lceil a \rceil$ 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 [Barral et al. 1999; Dorme 2001]. 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 center. The surface of the sphere of points of view is divided in 8 spherical triangles (figure 1).

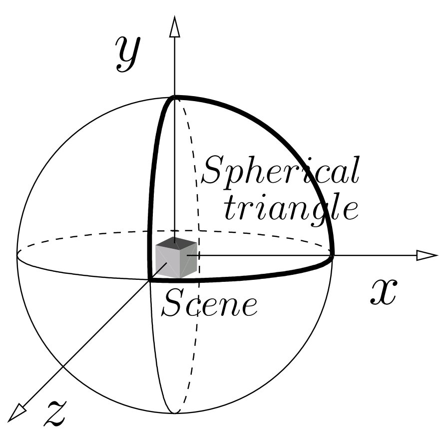


Figure 1: Sphere divided in 8 spherical triangles.

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.

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:

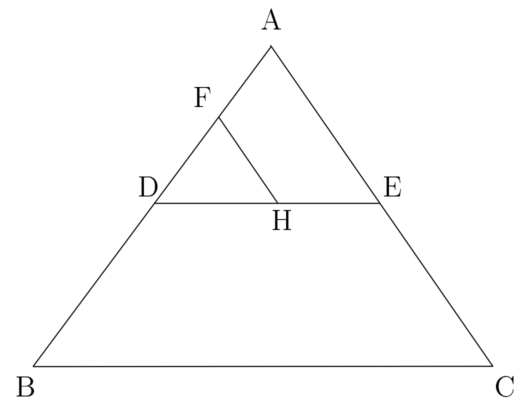


Figure 2: Heuristic search of the best point of view by subdivision of a spherical triangle.

If the vertex A (figure 2) 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.

Colin [Colin 1988] proposed a method to compute a good view for octree models. This method computes the “best” initial spherical triangle and then the “best” viewpoint is approximately estimated on the chosen triangle.

Sbert et al. [Sbert et al. 2002] 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 centered in the point of view.

The viewpoint entropy is then given by the formula:

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

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.

To compute the viewpoint entropy, the authors use the technique proposed in [Barral et al. 1999], 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 3 presents an example of results obtained with this method.

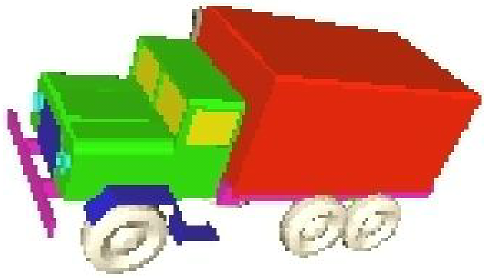


Figure 3: Point of view based on viewpoint entropy.

2.2 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 point 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 [Vázquez 2003; Vázquez et al. 2002] 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 properties of the world to understand.

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 [Barral et al. 1999; Barral et al. 2000] 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 position), only directions insuring smooth movement are considered in computing the next position of the camera (figure 4).
- 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 5).

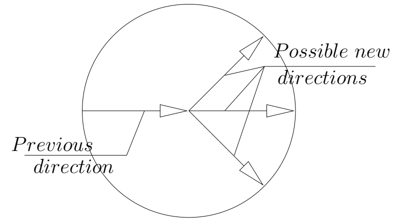


Figure 4: Only 3 directions are considered for a smooth movement of the camera.

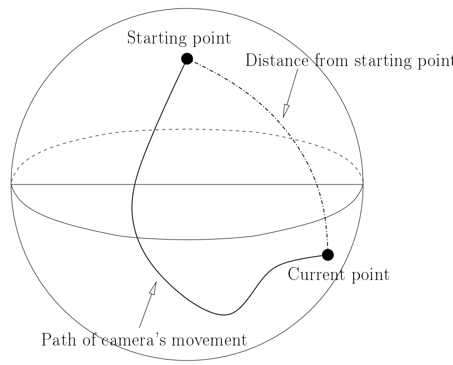


Figure 5: Distance of the current position of the camera from the starting point.

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

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

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.

Several variants of this technique have been proposed and applied. In figure 6 one can see an example of exploration of a simple virtual world representing an office.

Vázquez et al. [Vázquez and Sbert 2003a; Vázquez 2003] use a similar method for outside and indoor exploration of a virtual world. They use the viewpoint entropy to compute the pertinence of a view.

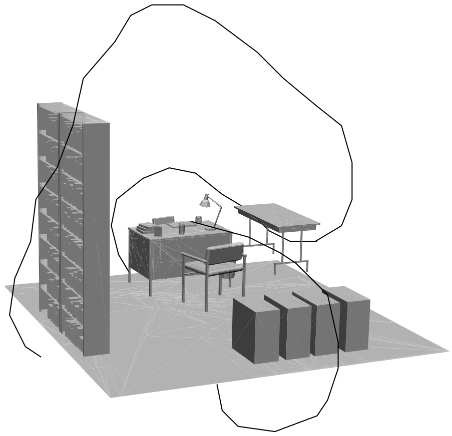
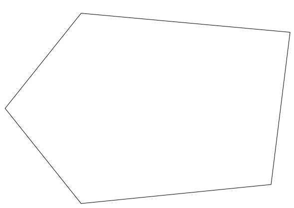


Figure 6: Exploration of a virtual office by incremental outside exploration.

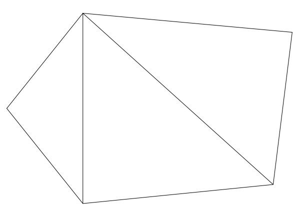
3 More accurate definition of viewpoint complexity

Most of the better known methods using the notion of viewpoint complexity to evaluate the pertinence of a view are based on two main geometric criteria: number of visible polygons and area of the projected visible part of the scene. Thus, equation (1) of section 2 is sometimes used to evaluate the viewpoint complexity for a given scene.

However, even if the methods using these criteria give generally interesting results, the number of polygons criterion may produce some drawbacks. Indeed, let us consider a scene made from a single polygon (see figure 7). This polygon may be subdivided in several other polygons and, in such a case, the number of visible polygons will depend on the number of subdivisions of the initial polygon. A viewpoint complexity evaluation function will give different results for the same scene, according to its subdivision degree.



(a) A single polygon scene



(b) Subdivided scene

Figure 7: The view quality for the same scene and from the same viewpoint depends on the subdivision level of the scene.

In order to avoid this drawback, another criterion was proposed by Sokolov et al. [Sokolov and Plemenos 2005; Sokolov et al. 2006], which takes into account the curvature of the scene. More precisely, the number of polygon criterion is replaced by the criterion of total curvature of the scene. In the proposed method, the importance of a view from a viewpoint p is given by the equation:

$$I(p) = \sum_{v \in V(p)} \left| 2\pi - \sum_{\alpha_i \in \alpha(v)} \alpha_i \right| \cdot \sum_{f \in F(p)} P(f), \qquad (2)$$

where:

- $F(p)$ is the set of polygons visible from the viewpoint p,
- $P(f)$ is the projected area of polygon f,
- $V(p)$ is the set of visible vertices of the scene from p,
- $\alpha(v)$ is the set of angles adjacent to the vertex v.

Equation (2) uses the curvature in a vertex (figure 8), that is the sum of angles adjacent to the vertex minus 2π.

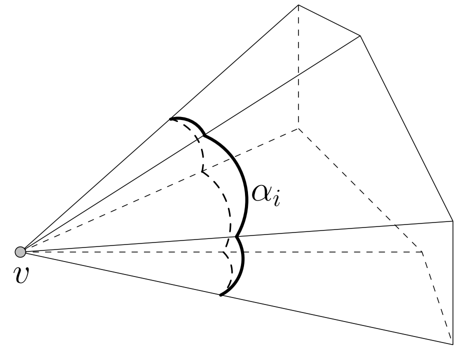


Figure 8: Curvature in a vertex.

The main advantage of the proposed criterion is that it is invariant to any subdivision of the scene elements maintaining the topology. Another advantage is that it can be extended in order to use the total integral curvature of curved surfaces.

The best viewpoint is computed by using a data structure, so-called *visibility graph*, which allows to associate to every discrete potential viewpoint on the surface of the surrounding sphere, the visual pertinence of the view from this viewpoint. Figure 9 shows the best point of view for a sphere with holes, containing various objects.

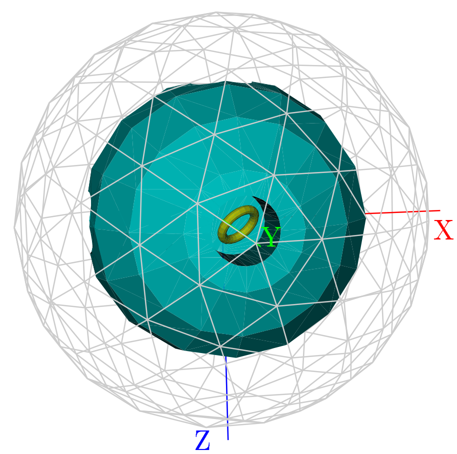


Figure 9: Best viewpoint computation for the scene, using the total curvature criterion instead of the number of polygons one.

The authors also propose a method to compute a pertinent trajectory for off-line exploration of the scene by a virtual camera. The method uses *the visibility graph* structure and a minimal set of viewpoints allowing to see all the vertices of the scene is computed in incremental manner. The main idea is to push the camera towards unexplored areas. Thus, having a trajectory from the starting point to the current camera position, the camera is pushed towards pertinent viewpoints allowing to see the maximum of not yet seen details of the scene. To do this, at each step a mass is assigned to each point of the discrete sphere and to the current position of the camera. The value of the mass assigned to a viewpoint is chosen according to the new information brought by the viewpoint. The camera position is then submitted to the Newton's law of gravity. The superposition of gravitational forces for the camera current position is the vector of movement. Figure 10 shows the camera trajectory for exploration of the scene of figure 9.

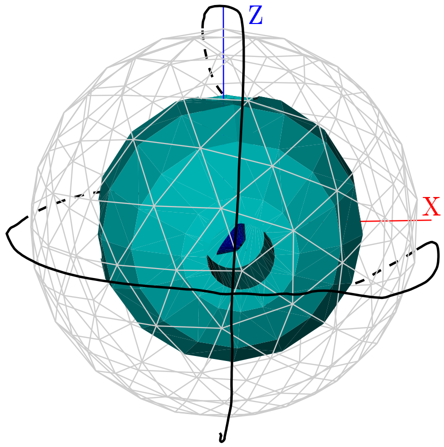


Figure 10: Trajectory of the virtual camera for exploring the scene of figure 9. All the holes of the sphere are visited.

Another method to compute a minimal set of good viewpoints in order to define a camera trajectory for off-line scene exploration was proposed by Jaubert et al. [Plemenos et al. 2005; Jaubert et al. 2006]. In this method a sufficient number of viewpoints is computed first and then the minimal set of good viewpoints is created by successively suppressing viewpoints which do not allow to see more details than the remaining ones.

4 Mesh saliency

The concept of mesh saliency was introduced in 2005 by Chang Ha Lee et al. [Lee et al. 2005]. The goal of this concept is to bring perception-based metrics in evaluation of the pertinence of a view. According to the authors, a high-curvature spike in the middle of a largely flat region is perceived to be as important as a flat region in the middle of densely repeated high-curvature bumps.

Mesh saliency is defined as the absolute difference between the Gaussian-weighted averages computed in fine and coarse scales. The following equation resumes the method to compute mesh saliency $\mathscr{S}(v)$ of a vertex v.

$$\mathscr{S}(v) = |G(\mathscr{C}(v),\sigma) - G(\mathscr{C}(v),2\sigma)|.$$

In this equation $\mathscr{C}(v)$ is the mean curvature of vertex v. $G(\mathscr{C}(v),\sigma)$ denotes the Gaussian-weighted average of the mean curvature and is computed by the following equation:

$$G(\mathscr{C}(v),\sigma) = \frac{\sum_{x \in N(v,2\sigma)} \mathscr{C}(x)\exp[-\|x-v\|^2/(2\sigma^2)]}{\sum_{x \in N(v,2\sigma)} \exp[-\|x-v\|^2/(2\sigma^2)]}.$$

$N(v,\sigma)$ in this equation is the neighborhood for a vertex v and may be defined as:

$$N(v,\sigma) = \{x : \|x-v\| < \sigma,\ x \textit{ is a mesh point}\}.$$

The mesh saliency can be computed at multiple scales. So, the saliency of a vertex v at a scale level i is computed by the formula:

$$\mathscr{S}_i(v) = |G(\mathscr{C}(v),\sigma_i) - G(\mathscr{C}(v),2\sigma_i)|,$$

where σ_i is the standard deviation of the Gaussian filter at scale i.

The authors use mesh saliency to compute interesting points of view for a scene. The shown examples seem interesting. An important advantage of the method is that the notion of mesh saliency is defined and may be computed at multiple scales.

5 What about lighting?

What is the lighting problem? There are rather two different problems which have to be resolved in different manners. The first problem is *absolute light source placement* and the second one is *taking into account light source position*.

5.1 Absolute light source placement

The problem is how to compute light source(s) position(s) in order to illuminate a scene in optimal manner. The resolution of this problem does not depend on the camera position. A good illumination of the scene should allow easier understanding by the user, if a camera explores the scene.

In the simple case of a single punctual light source, if only direct lighting is considered, the problem may be resolved in the same manner as the camera placement problem. What we have to do is to look for the best viewpoint from the light source.

In the general case, the problem is much more complex. Available today methods are not satisfactory. Most of them are based on inverse lighting techniques, where light source positions are deducted from the expected result. However, methods proposed by Poulingeas et al. [Jolivet et al. 2002] and Poulin et al. [Poulin and Fournier 1992; Poulin et al. 1997] are not entirely satisfactory, especially because it is not easy to well describe and formalize the expected results.

Design Galleries [Marks et al. 1997] is a general system to compute parameters for computer graphics but computation is not fully automatic. Another not fully automatic system to compute light source positions is presented in [Halle and Meng 2003]. The method presented in [Gumhold 2002] is based on the notion of *light entropy* and automatically computes lighting parameters but results are not entirely satisfactory without the help of the user.

5.2 Taking into account light source position

Up to now we have considered that the quality of a viewpoint is based on the geometry of the scene to be seen. However, a scene is often illuminated and several details, considered important according to the scene geometry, may be not visible for a given position of the light source, because they are shadowed. It is clear that, in such a case, it is important to take into account lighting in the computation of the quality of view from a viewpoint. If the number of scene details seen from a point of view is important, lighting of each visible detail has to be taken into account.

The problem of taking into account light source placement is quite different from the absolute source placement problem. Here the purpose is to take into account light source position in order to compute more precisely the pertinence of a view. The question to answer is: *Given a viewpoint P and a light source position L, how to compute the pertinence of the view from this viewpoint*? The problem is difficult to resolve in the general case but solutions may be proposed for some simpler cases.

Thus, Vázquez et al. [Vázquez and Sbert 2003b] have proposed a perception-based measure of the illumination information of a fixed view. This measure uses Information Theory concepts. The authors use with, as unit of information the relative area of each region whose colour is different from its surrounding.

It is possible to propose a method to compute the pertinence of a given view, taking into account the position of one (or more) punctual light source for direct lighting. This method is inspired from the method of viewpoint evaluation used in [Barral et al. 1999] and [Dorme 2001]. We have already seen that in the method proposed in [Barral et al. 1999] and [Dorme 2001], equation (1) is used to compute de viewpoint quality. In order to compute information needed by this equation, OpenGL and its integrated Z-buffer is used as follows:

A distinct colour is given to each surface of the scene and the display of the scene using OpenGL allows to obtain a histogram (figure 11) which gives information on the number of displayed colours and the ratio of the image space occupied by each colour.

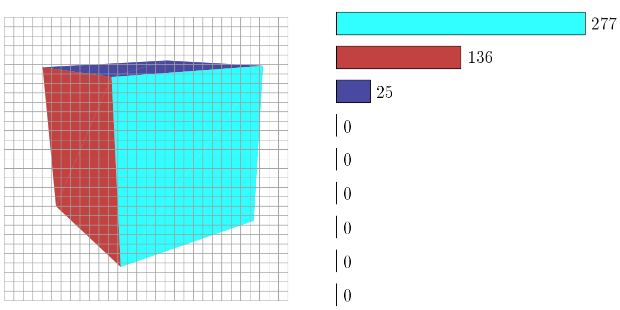


Figure 11: Fast computation of number of visible surfaces and area of projected viewpoint part of the scene by image analysis.

As each surface has a distinct colour, the number of displayed colours is the number of visible surfaces of the scene from the current position of the camera. The ratio of the image space occupied by a colour is the area of the projection of the viewpoint part of the corresponding surface. The sum of these ratios is the projected area of the visible part of the scene. With this technique, the two viewpoint complexity criteria are computed directly by means of an integrated fast display method.

Let us suppose that equation (1) is used to compute the quality of a point of view when only the geometry of the scene is used. In order to get an accurate estimation of the quality of view of a polygon of the scene from a given point of view it is important to integrate the quality of lighting of this polygon. A simple method to do this is to consider the angle of lighting from the light source to, say, the center of the polygon and to introduce the cosine of this angle in equation (1).

In practice we can use two Z-buffers, one from the point of view and one from the light source and approximate the cosine of the angle with the projected area of the polygon from the light source position (figure 12). For example, in equation (1), the considered visible projected area for a polygon will be de average value between the really visible projected area and the visible projected area if the light source position is taken as the center of projection. That is, the number of pixels corresponding to the colour of the polygon of the first Z-buffer will be added to the corresponding number of pixels of the second Z-buffer and divided by 2.

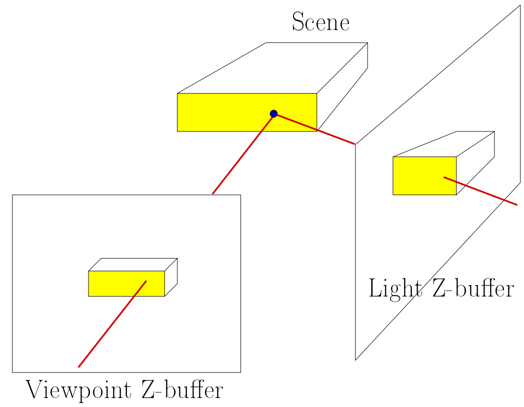


Figure 12: Two Z-buffers to estimate the quality of a viewpoint taking into account lighting of the scene.

This method may be easily generalized for *n* punctual light sources.

6 Conclusion and future work

In this paper we have presented the main current methods allowing intelligent scene display and exploration. The choice of camera position for a single display or of successive camera positions for scene exploration is based on the notion of visual pertinence of a view.

The purpose of all these methods is to use heuristics allowing to approximate the human perception of a picture, in order to choose interesting views of a scene. In all these methods it is supposed that the camera remains outside the scene.

Most of the presented methods take into account only the geometry of the scene to be displayed or explored. For these methods the most important criterion of visual pertinence is the number of visible details (polygons, curvatures), combined with the visible projected area of these details.

Some other methods try to take into account lighting parameters in order to, either choose a good lighting for a scene or allow more precise evaluation of the visual pertinence of a view by using these parameters together with scene geometry.

Geometry-based methods currently give good results whereas methods using lighting parameters are not yet entirely satisfactory. However we think that research on influence of lighting or other non-geometric parameters to the human perception, as well as modelling of these parameters, is a very exciting research area and should allow to obtain very interesting results.

Results of such techniques may have interesting industrial applications. We are particularly interested in their use in computer games applications.

7 Acknowledgments

This project has been supported and financed in part with funds of the European project GameTools. It has also been partly supported and financed by the Limousin Region (France). The authors would like to thank all people and organizations which have supported in any manner this project.

References

BARRAL, P., DORME, G., AND PLEMENOS, D. 1999. Visual understanding of a scene by automatic movement of a camera. In *GraphiCon'99*.

BARRAL, P., DORME, G., AND PLEMENOS, D. 2000. Scene understanding techniques using a virtual camera. In *Eurographics'2000*.

COLIN, C. 1988. A system for exploring the universe of polyhedral shapes. In *Eurographics'88*.

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

GUMHOLD, S. 2002. Maximum entropy light source placement. In *VIS '02: Proceedings of the conference on Visualization '02*, IEEE Computer Society, Washington, DC, USA, 275–282.

HALLE, M., AND MENG, J. 2003. Lightkit: A lighting system for effective visualization. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, IEEE Computer Society, Washington, DC, USA, 48.

JAUBERT, B., TAMINE, K., AND PLEMENOS, D. 2006. Techniques for off-line scene exploration using a virtual camera. In *International Conference 3IA'06*.

JOLIVET, V., PLEMENOS, D., AND POULINGEAS, P. 2002. Inverse direct lighting with a monte carlo method and declarative modelling. In *ICCS '02: Proceedings of the International Conference on Computational Science-Part II*, Springer-Verlag, London, UK, 3–12.

KAMADA, T., AND KAWAI, S. 1988. A simple method for computing general position in displaying three-dimensional objects. *Comput. Vision Graph. Image Process. 41*, 1, 43–56.

LEE, C. H., VARSHNEY, A., AND JACOBS, D. W. 2005. Mesh saliency. *ACM Trans. Graph. 24*, 3, 659–666.

MARKS, J., ANDALMAN, B., BEARDSLEY, P. A., FREEMAN, W., GIBSON, S., HODGINS, J., KANG, T., MIRTICH, B., PFISTER, H., RUML, W., RYALL, K., SEIMS, J., AND SHIEBER, S. 1997. Design galleries: a general approach to setting parameters for computer graphics and animation. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 389–400.

PLEMENOS, D., AND BENAYADA, M. 1996. Intelligent display in scene modelling. new techniques to automatically compute good views. In *GraphiCon'96*.

PLEMENOS, D., GRASSET, J., JAUBERT, B., AND TAMINE, K. 2005. Intelligent visibility-based 3D scene processing techniques for computer games. In *GraphiCon'05*.

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

POULIN, P., AND FOURNIER, A. 1992. Lights from highlights and shadows. In *SI3D '92: Proceedings of the 1992 symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, 31–38.

POULIN, P., RATIB, K., AND JACQUES, M. 1997. Sketching shadows and highlights to position lights. In *CGI '97: Proceedings of the 1997 Conference on Computer Graphics International*, IEEE Computer Society, Washington, DC, USA, 56.

SBERT, M., FEIXAS, M., RIGAU, J., CASTRO, F., AND VÁZQUEZ, P. P. 2002. Applications of the information theory to computer graphics. In *International Conference 3IA'2002*.

SOKOLOV, D., AND PLEMENOS, D. 2005. Viewpoint quality and scene understanding. In *VAST 2005: Eurographics Symposium Proceedings*, Eurographics Association, ISTI-CNR Pisa, Italy, M. Mudge, N. Ryan, and R. Scopigno, Eds., 67–73.

SOKOLOV, D., PLEMENOS, D., AND TAMINE, K. 2006. Viewpoint quality and global scene exploration strategies. In *International Conference on Computer Graphics Theory and Applications (GRAPP 2006)*.

VÁZQUEZ, P. P., AND SBERT, M. 2003. Automatic indoor scene exploration. In *International Conference 3IA'2003*.

VÁZQUEZ, P.-P., AND SBERT, M. 2003. Perception-based illumination information measurement and light source placement. *Lecture Notes in Computer Science 2669*, 306–316.

VÁZQUEZ, P. P., FEIXAS, M., SBERT, M., AND HEIDRICH, W. 2002. Image-based modeling using viewpoint entropy. In *Computer Graphics International*, 267–279.

VÁZQUEZ, P. P. 2003. *On the selection of good views and its application to computer graphics.* PhD thesis, Barcelona (Spain).

About the authors

Dimitri PLEMENOS is a full professor at the XLIM laboratory of the University of Limoges (France). His research area is intelligent techniques in computer graphics, including declarative modelling, intelligent rendering and intelligent virtual world exploration. He is author or co-author of several papers and member of the IPC of many international conferences and journals. Dimitri Plemenos is the organizer and general chair of the 3IA international annual conference on Computer Graphics and Artificial Intelligence.
E-mail: plemenos@unilim.fr

Dmitry SOKOLOV is a PhD student at the XLIM laboratory of the University of Limoges (France). His research area is Automatic Virtual World Exploration.
E-mail: s@skisa.org