The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage
VAST (2005)
M. Mudge, N. Ryan, R. Scopigno (Editors)

Viewpoint quality and scene understanding

Dmitry Sokolov
sokolov@msi.unilim.fr

Dimitri Plemenos
plemenos@unilim.fr

Université de Limoges, Laboratoire MSI
83 rue d'Isle, 87000 Limoges, France

Abstract

Virtual worlds exploration techniques become nowadays more and more important. The methods are used in wide variety of domains — from graph drawing to robot motion. This paper is dedicated to virtual world exploration techniques which have to help a human being to understand a 3d scene. Improved methods of viewpoint quality estimation are presented in the paper. Using these methods, a real-time technique allowing to choose a "good" viewpoint for a virtual scene is also proposed.

Keywords: Scene understanding, Automatic virtual camera, Good point of view, Visibility.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Viewing algorithms

1. Introduction

The last decades have been marked by the development of the computer as a tool in almost every domain of human activity. One of the reasons for such a development was the introduction of human-friendly interfaces which have made computers easy to use and learn.

The increasing exposure of the general public to technology means that their expectations of display techniques have changed. The increasing spread of the internet has changed expectations of how and when people are to access information. The whole way in which heritage is viewed is undergoing a change.

Virtual Reality technology broadens the role of the museum, it can provide several advantages:

1. Simulation of the physical layout of the museum and the artefacts,
2. Removal of time and distance constraints on access to displays,
3. Transformation of abstract data into a virtual artefact,
4. Ability to interact with artefacts.

Visitors are not limited by opening hours for observing artefacts in the virtual museum. Internet access also ensures that visitors can view the artefacts without having to travel great distances. Virtual worlds are not limited by physical laws, so it's possible to transform general information to an artefact, which is easy to view and understand, but which has not any prototypes in real world. Also, an exact copy of the real artefact can be handled by the user without fear of breakage or loss.

As a consequence, a lot of problems raised, one of them is automatic exploration of a virtual world. During these last years people pay essentially more attention to this problem. They realized necessity to have fast and accurate techniques for better exploration and clear understanding of various virtual worlds. However, there are few papers which consider this problem from the computer graphics point of view. On the other hand, many 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.

Ability to automatically compute a good exploration path could allow to create a virtual guide, a councillor, which will not limit a user to a precomputed trajectory, but will give

recommended direction of moving, taking into account a set of already visited sights. Figure 1 shows an example.

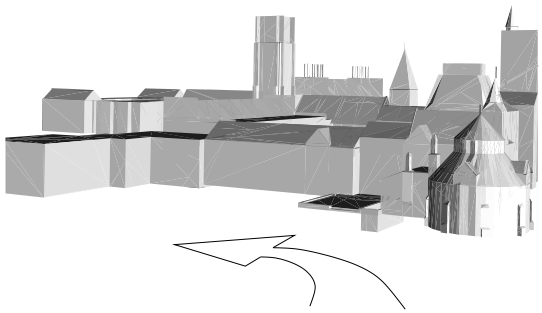


Figure 1: *A user walks in a virtual city, the arrow indicates a recommended direction.*

In this paper we are mainly concerned by global virtual world exploration, where the camera remains outside the scene, but the proposed methods may help to the indoor exploration researches too.

Viewpoint quality is an important criterion for good comprehension of a scene. In this paper we shall try to propose new methods for more accurate estimation of viewpoint quality, in order to improve scene understanding.

The rest of the paper is organized as follows: in section 2 related works are reviewed. Their drawbacks, which we would like to eliminate in our approach, are under discussion in section 3. Section 4 presents our approach and invents new criteria of quality. Section 5 shows results of applying new technique in comparison with previous methods. Finally, we address the problem of rapid estimation qualities for a set of viewpoints in section 6.

2. Background

Kamada and Kawai [KK88] have proposed a fast method to compute a point of view, which minimizes the number of degenerated edges of a scene. They consider a viewing direction to be good if parallel line segments are as far away one from another as possible at the screen. Obviously, this means minimizing an angle between a direction of view and a normal of a considered face. Or, for a complex scene, this means minimization of the maximum angle deviation for all faces of the scene.

Colin [Col88] has proposed a method, initially developed for scenes modeled by octrees. The method is to compute a good viewpoint for an octree. The main principle of the method can be described as follows:

1. Choose three best directions of view d_1, d_2 and d_3 among the 6 directions corresponding to 3 coordinate axes passing through the center of the scene.
2. Compute a good direction in the pyramid defined by the 3 chosen directions, taking into account an importance of each of the chosen directions (see figure 2).

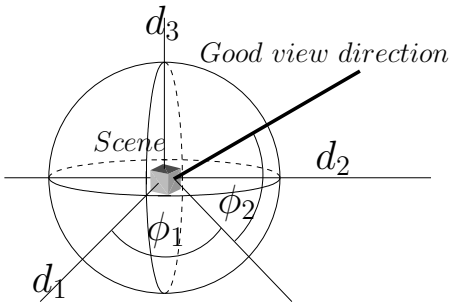


Figure 2: *Direct approximate computation of a good direction of view.*

The viewpoint is considered to be good if it shows high amount of voxels.

In [PB96] Plemenos and Benayada have proposed heuristic that extends definition given by Kamada and Kawai. Their heuristic considers a viewpoint to be good if it minimizes maximum angle deviation between direction of view and normals to the faces and gives a high amount of details. The viewpoint quality according to [PB96] can be computed by the following formula:

$$C(p) = \frac{\sum_{i=1}^{n} [\frac{P_i(p)}{P_i(p)+1}]}{n} + \frac{\sum_{i=1}^{n} P_i(p)}{r}, \qquad (1)$$

where:

1. $C(p)$ is the viewpoint quality for the given viewpoint p,
2. $P_i(p)$ is the number of pixels corresponding to the polygon number i in the image obtained from the viewpoint p,
3. r is the total number of pixels of the image (resolution of the image),
4. n is the total number of polygons in the scene.
5. $[a]$ means the ceiling function, i.e the smallest integer number $a_c \in \mathbb{N} : a_c \geq a$.

In [VFSH01] Vázquez et al. have provided an information theory-based method evaluating viewpoint quality. To select a good viewpoint they propose to maximize the following function, they have called a "viewpoint entropy":

$$I(S,p) = -\sum_{i=0}^{N_f} \frac{A_i}{A_t} \cdot \log \frac{A_i}{A_t}, \qquad (2)$$

where N_f is the number of faces of the scene, A_i is the projected area of the face number i, A_0 represents the projected area of the background in open scenes and A_t is the total area of the projection.

For more details, a state of the art paper [Ple03] on virtual

world global exploration techniques is available, whereas viewpoint quality criteria and estimation techniques are presented in [PSF04].

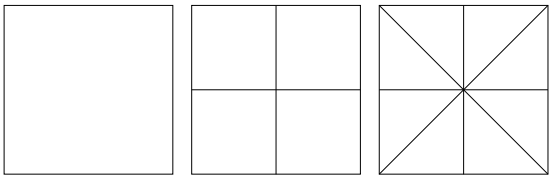
3. Discussion

Directly or implicitly, all the proposed methods use only two global parameters as input:

1. Size of a visible surface (projected area, amount of voxels, angle between direction of sight and normal to the face),
2. Number of visible faces.

In other words, all of them consider a viewpoint quality as a sum of qualities of separate faces, but don't take into account *how* a polygon is connected to the adjacent ones.

The number of visible faces is quite weak criterion for a viewpoint quality estimation, because it may give non-accurate results. For example, if we consider a very simple scene, that consists of one square (figure 3(a)), then equation 2 gives us $I(S,p) = 0$ for the viewpoint p lying at a perpendicular to the square's center. If we subdivide the square (figure 3(b)), topology of the scene does not change, but $I(S,p)$ grows.



(a) A scene consisting of a single square.

(b) The square is subdivided into 4 faces.

(c) The square is subdivided into 8 faces.

Figure 3: *Three scenes represent the same square **(a)**, it is divided into 4 parts **(b)** and 8 parts **(c)**. Equation 2 gives us $I(S,p) = 0$ for **(a)**, $I(S,p) = \log 4$ for **(b)** and $I(S,p) = \log 8$ for **(c)**.*

Thus, the methods using a number of faces to evaluate a viewpoint quality, depend on initial scene subdivision. Using the projected area of a face as a criterion of quality, the dependence will appear also if we don't use an additive metric, i.e., a sum of areas.

4. New heuristic

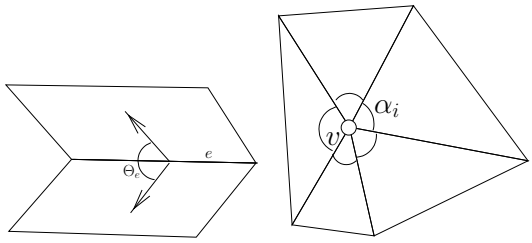
In this section a new heuristic using a new criterion of a viewpoint quality is presented. As we have said before, it would be good if the estimation routine could be broadened by the knowledge of how a face is connected to the adjacent ones. We propose to use a curvature of a surface as such a knowledge. Intuitively, this choice is good since more mesh bends are visible from the viewpoint, more details of a scene could be seen. Unfortunately, we can't apply differential geometry definitions due to a discrete nature of our scene, so, the equations should be redefined. There are many ways to define a total curvature for a discrete mesh — for example, a curvature along edges, so-called extrinsic curvature (equation 3) and intrinsic curvature for a set of vertices (equation 4). A concave mesh brings to a user the same amount of information as a convex one, so, we propose to consider absolute values instead of signed ones. Then the first one can be written as:

$$C_{ext} = \sum_{e \in E} (1 - \cos\Theta_e) \cdot |e|, \tag{3}$$

where E is the set of edges of the scene, $|e|$ is the edge length, Θ is the angle between normals to the faces sharing the edge e. The second one can be defined as follows:

$$C_{int} = \sum_{v \in V} \left| 2\pi - \sum_{\alpha_i \in \alpha(v)} \alpha_i \right|, \tag{4}$$

where $\alpha(v)$ is the set of angles adjacent to the vertex v (see figure 4).



(a) Extrinsic curvature is equal to the product of edge length and angle Θ between faces sharing the edge.

(b) Intrinsic curvature is equal to the sum of angles adjacent to the vertex minus 2π.

Figure 4: *Two ways to define total curvature for a discrete mesh: curvature along edges and curvature in vertices.*

Thus, our heuristic could be expressed as follows:

$$I(p) = C(F(p)), \tag{5}$$

where $F(p)$ means the set of visible faces for the viewpoint p, $C(F(p))$ is the total curvature for the mesh visible from p. In order to calculate the curvature, both equations 4 and 3 may be used. We prefer to use intrinsic curvature equation, because it requires significantly less amount of calculations than the other one. It implies point-to-point visibility problem, while the usage of extrinsic curvature requires point-to-segment visibility computations.

The proposed heuristic is invariant to any subdivision of a

scene keeping the same topology. Indeed, if we subdivide a flat face to several ones, then all inner edges and vertices will not be taken into account due to zero angles. The heuristic works for the outdoor and indoor explorations equally. An important property of such a kind of viewpoint quality definition is the possibility to extend it, using the total integral curvature $\int_{\Omega} |K| dA$, into the class of continuous surfaces, such as NURBS etc., which nowadays become more and more usable.

If we are allowed to use point-to-region visibility calculations, the projected area could be considered as an input parameter also:

$$I_2(p) = C(F(p)) \cdot \sum_{f \in F(p)} P(f),$$

where $P(f)$ means the projected area of the face f. This definition is invariant to the scene changes keeping the topology, but it is more expensive than the previous one due to large amount of computations necessary to get properly projected areas. As a cheaper solution keeping an eye on projected sizes, we could propose to use extrinsic curvature in formula 3 with projected edge lengths instead of absolute ones. Sometimes it would be convenient — distant objects will give less contribution to the viewpoint quality than closer ones.

5. Comparison with previous results

5.1. Projected area versus our heuristic

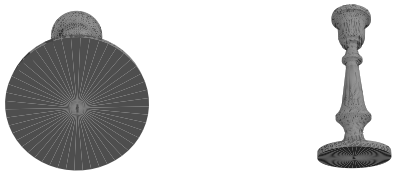
Figure 5 shows us that maximum of projected area may give to a user large amount of pixels drawn, but few details of a scene.

(a) (b)

Figure 5: *Maximum area **(a)** versus our method of viewpoint selection **(b)**.*

5.2. Viewpoint entropy versus our heuristic

A candlestick from SGI Open Inventor models is drawn at figure 6. Figure 6(a) is taken from [Vaz03], this viewpoint maximizes the viewpoint entropy. A view direction, maximizing our heuristic is shown at figure 6(b).



(a) (b)

Figure 6: *The best views of the candlestick are computed by entropy-based method **(a)** and our method **(b)**.*

5.3. Viewpoint complexity versus our heuristic

A martini cup from SGI Open Inventor toolkit was tested. The viewpoint complexity [PB96] and the viewpoint entropy [Vaz03] techniques give the same result, which is shown at figure 7(a). The result of our method application is given at figure 7(b).

(a) (b)

Figure 7: *The viewpoint complexity and the entropy-based methods give the same result **(a)**; our method application **(b)**.*

5.4. Stand-alone results

Figures 8 and 9 present results of viewpoint quality estimations for two models: the Utah teapot and a virtual office model. The figures show qualities of viewpoints lying on surrounding sphere. To distinguish values in black-and-white picture, we have connected viewpoints by lines; more the line is dark and thick, more the viewpoints are good. The figures show three images in order to represent better the 3D data: the first image (the large one at the top of the figure) gives a scene from the best point of view. The small images allow to see how good each viewpoint is.

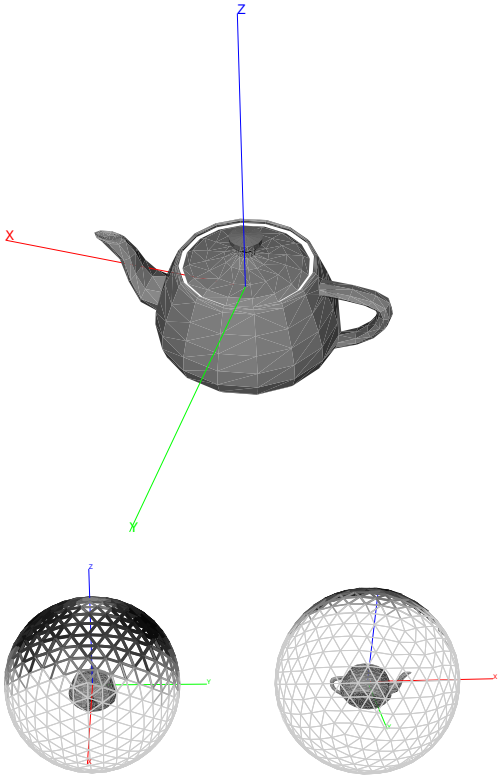


Figure 8: *The figure shows the scene from different points of view. First image gives the scene from the best viewpoint, the second and third ones show the viewpoint qualities (see the section 5.4).*

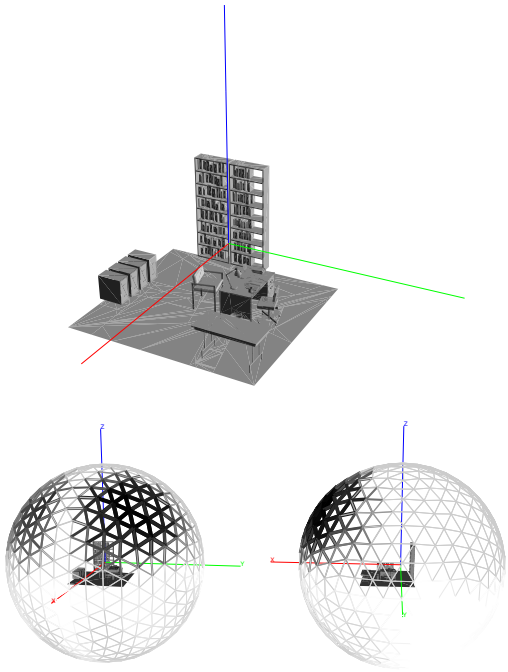


Figure 9: *The figure shows the scene from different points of view. First image gives the scene from the best viewpoint, the second and third ones show the viewpoint qualities (see the section 5.4).*

6. Selection of good views for scene understanding

Having defined a viewpoint quality measure we would like to find view directions, maximizing the heuristic.

6.1. Previous approaches

Plemenos in [Ple91] and [PB96] has proposed an iterative method of automatic calculation of a good viewpoint. The scene is placed at the center of a sphere whose surface represents all possible points of view. The sphere is divided into 8 spherical triangles (see figure 10). The best spherical triangle is chosen by the viewpoint qualities of the triangle vertices. Then, the selected spherical triangle is recursively subdivided and the best vertex is chosen as the best point of view at the end of the process (figure 11).

The good view criteria in this approach were the number of visible polygons and the projected area of visible parts of a scene.

Vázquez in [Vaz03] proposes similar technique, applying

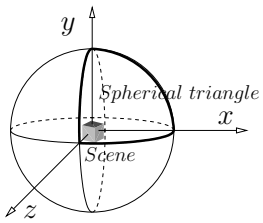


Figure 10: *The sphere of points of view is divided into 8 spherical triangles.*

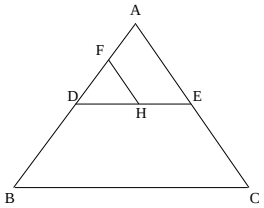


Figure 11: *Recursive subdivision of the "best" spherical triangle.*

a heuristic designed to predict new viewpoint entropy using values of already processed points.

Both works operate with quite expensive point-to-region visibility and calculate visible parts using the Z-Buffer method.

In these papers, the scene is rendered from a viewpoint, coloring each face with a unique color ID and using flat shading. In the resulting rendered scene, each pixel represents the color code of the face that is visible in that pixel — OpenGL allows to obtain a histogram which gives informa-

tion on the number of displayed colors and the ratio of the image occupied by each color. See [BDP99] and [NRTT95] for more detailed description.

Quality estimation routine for a single viewpoint, applying these methods, runs in $O(N_f \cdot Z)$ time, where N_f is a number of faces in a scene and Z is the resolution of Z-Buffer. For N_s viewpoints the running time is $O(N_f \cdot N_s \cdot Z)$. Note that Z should be significantly greater than N_f in order to have at least few pixels to display each face. Complexity of the algorithm forces the authors to use adaptive search algorithms which may give inexact results. In the following section a new approach, which allows us to use even brute-force method in real-time, is presented.

6.2. New approach

Our new heuristic does not use visibility of faces, it uses visibility of vertices instead. This allows us to reduce the problem of computing viewpoint quality from point-to-region to point-to-point visibility. Using algorithm 1, we reduce the running time from $O(N_f \cdot Z \cdot N_s)$ to $O\left(N_f \cdot N_v \cdot \sqrt{N_s}\right)$ operations, where N_v means the number of vertices of a scene, $Z \gg N_f$.

Let's rasterize our surrounding sphere; each element represents a viewpoint. To evaluate viewpoint quality, we have to find all vertices of a scene which are visible from each element of the sphere. In order to perform it rapidly, we consider a reverse problem by finding all visible viewpoints from each vertex of the scene. It allows to use structuredness of the rasterized sphere for fast elimination of hidden areas. Without loss of generality we can consider a rasterized plane instead of the sphere and a triangulated scene (we can triangulate polygonal mesh in linear time using the algorithm presented in [Cha91]). Now we iterate through each vertex of a scene and find all pixels of the plane which are visible from a given vertex (see figure 12).

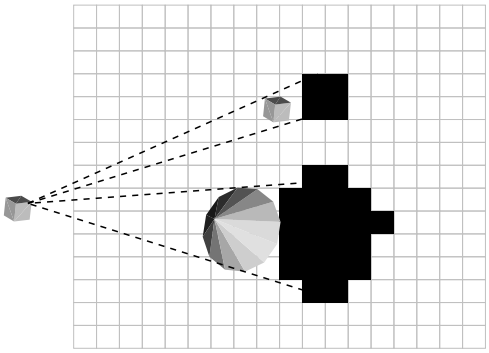


Figure 12: *The scene consists of two cubes and a cone, the rasterized plane represents a set of viewpoints. The given vertex of the scene is not visible from viewpoints colored black.*

Now description of the main step of the algorithm will be given. A vertex of the scene is given; for each triangle of the scene all its vertices are to be projected into the plane. Then, its boundary can be drawn on the rasterized plane using famous Bresenham's algorithm [Bre65] of digital line drawing. Having initially white plane, projection of each triangle is drawn with black color. Then we can easily delete inner parts of the projections. If our plane consists of N_s pixels, then the maximal boundary drawing time is $O(\sqrt{N_s})$ for a selected triangle. Having N_f triangles and N_v vertices in a scene, the total running time of the algorithm is $O(N_f \cdot \sqrt{N_s} \cdot N_v)$ operations.

This technique can be applied to the rasterized sphere as follows (see algorithm 1): let us suppose that there is a tessellated sphere; then a graph $G = (V, E)$ can be constructed, where the set of vertices V corresponds to the tessellation parts. Edge $(v_1, v_2) \in E$ if and only if the part v_1 is adjacent to v_2. For example, each pixel of usual screen has 8 neighbors. If there is such a kind of graph and two vertices of the sphere are given, then the shortest path in the graph corresponds to geodesic curve (or shortest line) connecting the vertices. Since the graph has very special structure, the shortest path of length l can be found in $O(l)$ operations using an adaptation of the Bresenham's algorithm. So, when we have projected 3 vertices of a triangle on the sphere, we can find 3 shortest paths representing the boundary of the projection. The paths are stored and next triangle is projected. After main loop, when the boundaries of all triangles have been computed, it's easy to remove from the graph all the vertices met, at least once, in the set of paths. Then, the graph

Input: Set of faces F, rasterized sphere S
Output: Set of qualities $Q = \left\{q_{\{s \in S\}}\right\}$

$q_s \leftarrow 0 \; \forall s \in S$
Construct graph $G = (S, E)$ as it is shown in section 6.2
for each vertex v of the scene **do**
  $B \leftarrow \emptyset$
  for each $f \in F$ **do**
    Find projection P_f of vertices of face f
    for i from 1 to $|P_f| - 1$ **do**
      Find shortest path γ in G from p_i to p_{i+1}.
      Store vertices are found: $B \leftarrow B \bigcup \gamma$
    end for
  end for
  $S \leftarrow S \setminus B$
  Remove broken edges from E
  Use depth-first search to remove from S inner parts of the projections.
  $q_s \leftarrow q_s + C_{int}(v) \; \forall s \in S$
  Restore S and $G = (S, E)$ to the initial state
end for

Algorithm 1: Rapid estimation qualities for a set of viewpoints lying at surrounding sphere.

is splitted into a set of linked components (for example, we can split it by constructing a depth-first search tree) and the components corresponding to internal parts of the triangles are to be removed.

7. Conclusion and future works

In this paper we have presented the new criteria of evaluating a viewpoint quality, which do not depend on changes in a scene keeping the original topology. The method is extendable into the class of continuous surfaces such as NURBS. A fast method of best viewpoint choosing is presented too.

The proposed techniques allow to get a good comprehension of a single virtual artefact or a general comprehension of a scene. In the future it would be interesting to extend our definitions. For example, we can consider not only curvature or number of faces, but also a number of visible objects. An example is shown at figure 13. The display is almost completely hidden by the case, but we clearly recognize it. If we could split properly (in human perception) a scene into a set of objects, then the number of visible objects becomes a very important criterion of a viewpoint quality estimation.

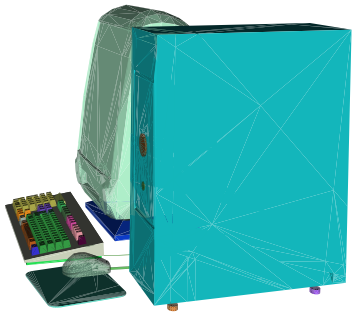


Figure 13: *The display is almost completely hidden by the case, but we clearly recognize it.*

This technique could be particularly helpful in virtual heritage projects, for example, in a virtual museum different objects should have different importances. Obviously, the artefacts should have significantly greater importances than walls, chairs and so on. Having a proper division of a virtual museum model into a set of objects we could obtain good heuristics for an automatic (or guided manual) exploration.

It would be also interesting to develop automatic methods of a scene exploration and to consider possibilities to develop methods allowing an interaction with a user, where the user can point which parts of a scene he (she) would like to explore in details.

References

[BDP99] BARRAL P., DORME G., PLEMENOS D.: Visual understanding of a scene by automatic movement of a camera. In *GraphiCon'99* (Moscow (Russia), September 1999). 6

[Bre65] BRESENHAM J. E.: Algorithm for computer control of a digital plotter. *IBM Systems Journal 4*, 1 (1965), 25–30. 6

[Cha91] CHAZELLE B.: Triangulating a simple polygon in linear time. *Discrete Comput. Geom. 6*, 5 (1991), 485–524. 6

[Col88] COLIN C.: A system for exploring the universe of polyhedral shapes. In *Eurographics'88* (Nice (France), September 1988). 2

[KK88] KAMADA T., KAWAI S.: A simple method for computing general position in displaying three-dimensional objects. *Comput. Vision Graph. Image Process. 41*, 1 (1988), 43–56. 2

[NRTT95] NOSER H., RENAULT O., THALMANN D., THALMANN N. M.: Navigation for digital actors based on synthetic vision, memory, and learning. *Computers & Graphics 19*, 1 (1995), 7–19. 6

[PB96] PLEMENOS D., BENAYADA M.: Intelligent display in scene modelling. new techniques to automatically compute good views. In *GraphiCon'96* (Saint Petersburg (Russia), July 1996). 2, 4, 5

[Ple91] PLEMENOS D.: A contribution to the study and development of scene modelling, generation and visualisation techniques. *The MultiFormes project., Professorial dissertation* (November 1991). 5

[Ple03] PLEMENOS D.: Exploring virtual worlds: Current techniques and future issues. In *International Conference GraphiCon'2003* (Moscow (Russia), September 2003). 2

[PSF04] PLEMENOS D., SBERT M., FEIXAS M.: On viewpoint complexity of 3d scenes. In *International Conference GraphiCon'2004* (Moscow (Russia), September 2004). 3

[Vaz03] VAZQUEZ P. P.: *On the selection of good views and its application to computer graphics.* PhD thesis, Barcelona (Spain), May 2003. 4, 5

[VFSH01] VAZQUEZ P. P., FEIXAS M., SBERT M., HEIDRICH W.: Viewpoint selection using viewpoint entropy. In *VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001* (2001), Aka GmbH, pp. 273–280. 2