Automatic Indoor Scene Exploration

Pere-Pau Vázquez†
Mateu Sbert‡
†Dept. LSI - Universitat Politècnica de Catalunya,
mail: ppau@lsi.upc.es
‡IIiA, Universitat de Girona
mail: mateu@ima.udg.es

Abstract

Automatic computation of best views of objects are very useful. For example, they can be used as the starting point of a scene exploration, or to enrich galleries of objects available through Internet by adding an image a model that may help to decide if it is worth downloading. Recently, a measure to evaluate the quality of a view has been proposed, it is called *viewpoint entropy* and has a basis on Information Theory. The best view is the one which gives the most information of the object being inspected. For large models, the selection of a set of good views that cover all the faces can aid the user to understand a certain object or scene. For very complex environments, however, a set of images may not suffice. For example, when examining buildings it may be difficult to locate in space the positions where the viewpoints were placed. In these cases, an interactive exploration of the model can better help the user to understand the structure of the scene. In this paper we present an automatic method for the exploration of scenes that uses viewpoint entropy.

Keywords: Interactive exploration, Information Theory, entropy.

1 Introduction

Many situations may require the inspection of a *whole* scene or object in order to have a correct understanding. For example, when modeling scenes (especially if the modeler is using a so-called declarative modeler [1]), or when it is important to properly communicate high amounts of information through photographs such as a virtual tour in a museum, or showing a flat to sale, or the possible design of a new building. These cases will take advantage of an interactive exploration of the scene. This way the user may get a proper idea of it. In this paper we present a new technique for the automatic indoor scene exploration. It uses an Information Theory based measure dubbed *viewpoint entropy* that can be used to evaluate the amount of geometric information present in a view.

The rest of the paper is organized as follows: In Section 2 we review previous work on good view selection and scene exploration. Section 3 introduces the viewpoint entropy measure. In Section 4 the algorithm for the automatic scene exploration is presented, and finally, Section 5 discusses the results and concludes our work pointing out some lines of future work.

2 Previous Work

Colin [2] presents a method to select a good view to observe a scene modeled with an octree. This method chooses the view which shows the highest amount of voxels, according to two different visibility measures, one exact and another one approximate. Kamada and Kawai [3] consider a viewing direction to be good if in a projection, parallel line segments on a plane in 3D project as far away from each other as possible. Intuitively, the viewer should be as orthogonal as possible to every face of the 3D object. As this is not possible, they suggest to minimize (over all the faces) the maximum angle deviation between a normal to the face and the line of sight from the viewer. However, this method fails when comparing scenes with equal number of degenerated faces and it does not ensure that the user will see a large amount of detail [4]. Plemenos and Benayada [5] extend Kamada's definition. They consider a direction to be good if it minimizes the maximum angle deviation between a normal to the face and the line of sight from the viewer *and* it also provides a high amount of detail. If only the first condition is satisfied, it does not ensure that the view will not hide important information of the scene. Therefore, they add another parameter to the maximizing function which accounts for the detail seen. The parameter added is the number of visible faces from a viewpoint. Then, a function which combines both parameters is created. Barral *et al.* [4] present a method for the automatic exploration of objects or scenes. This method is based on the good view concept definition presented in [5]. In this case, the goodness of a view is computed by defining a new importance function that depends on the visible pixels of each polygon. An extension of this method can be found in Dorme [6], where an extra parameter, the so-called *exploration parameter* is added in order to avoid taking into account faces that have already been shown to the user. However, there remain problems with objects containing holes, as these are not captured properly by the algorithm.

Marchand and Courty have presented a general framework that allows the automatic control of a camera in a dynamic environment [7]. The proposed method is based on the *image-based control* or visual servoing approach. Visual servoing consists in specifying a task (generally positioning or target tracking tasks) as the regulation in the image of a set of visual features. It consists in positioning a camera according to the information perceived in the image. The camera is placed in an arbitrary position and then must react in an appropriate and efficient way to modifications of the environment. Bourque and Dudek [8] describe a system for the automatic construction of image-based virtual realities to represent a real environment. The criterion used to determine the importance of a view is based on human attention.

Freeman has exploited the *generic viewpoint assumption* to address shape from shading problems. The *generic viewpoint assumption* states that an observer is not in a special position relative to the scene [9]. It is commonly used to disqualify scene interpretations that assume special viewpoints, thus, it can be used to avoid ambiguities [10]. The Design Galleries^{TM} (DG) system [11] is a method to automatically set parameters for computer graphics and animation. They focus on the problem of parameter tweaking, concretely: light selection and placement for image rendering; opacity and colour transfer-function specification for volume rendering; and motion control for particle-system and articulated-figure animation.

Vázquez *et al.* [12] have presented a measure, *viewpoint entropy*, based on Shannon's entropy [13]. In that paper, the authors present an algorithm for the exploration of objects that uses viewpoint entropy. Unfortunately, this method is only suitable for single objects as the navigation is carried over a bounding sphere. Moreover, there is the possibility of not exploring all important parts of an object if certain viewpoint is surrounded by very low informative regions.

Some commercial products such as QuickTime VR [14] offer panoramas, which are 360 degrees images taken from a single point of view. However, the user is restricted to camera movements of rotation and zooming, it is not possible to freely navigate and therefore the information shown depends on the initial photographs taken. Moreover, the user must jump from one to another viewing position and therefore it is possible to loose the notion of the structure of the scene after some jumps.

3 Viewpoint Entropy

The *Shannon entropy* [15, 13] of a discrete random variable X with values in the set $\{a_1, a_2, ..., a_n\}$ is defined as

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i,$$

where $p_i = Pr[X = a_i]$, the logarithms are taken in base 2 and $p_i \log p_i$ is equal to 0 for $p_i = 0$ for continuity reasons. As $-\log p_i$ represents the *information* associated with the result a_i, the entropy gives the average *information* or the *uncertainty* of a random variable. The unit of information is called a *bit.*

Based on this measure, Vázquez *et al.* [12] have presented a new measure, dubbed *viewpoint entropy*, that can be understood as the amount of information seen from a point from a scene (see also [16]). Given a scene S if we suppose the information we deal is the projected area of each face, viewpoint entropy from a point p can be calculated by projecting all the faces of the scene onto the viewing plane and adding the contributions of each face. Therefore, we can modify Shannon entropy taking as probability distribution the relative projected areas. This will result in formula 1:

$$I(S,p) = -\sum_{i=0}^{N_f} \frac{A_i}{A_t} \log \frac{A_i}{A_t}, \qquad (1)$$

where N_f is the number of faces of the scene, A_i is the solid angle of the projected area of face i over the sphere, A_0 represents the projected area of background in open scenes, and A_t is the total area of the sphere. In a closed scene, or if the point does not *see* the background, the whole sphere is covered by the projected areas and consequently $A_0 = 0$. Hence, A_i/A_t represents the *visibility* of face i with respect to the point p. It is important to remark that the area A_i/A_t is proportional to the cosine of the angle between the normal of the surface and the line from the point of view to the object, and it is inversely proportional to the square distance from the point of view to the face. Therefore, A_i/A_t grows when the face is seen at a better angle and at a shorter distance.

The maximum entropy is obtained when a certain point can *see* all the faces with the same relative projected area A_i/A_t. So, if the viewpoint does not see the background, we will have a maximum entropy of $\log N_f$. By optimizing the value of entropy in our images, we are trying to capture the maximum number of faces under the best possible orientation. We define the *best* viewpoint as the one that has maximum entropy, i.e. maximum geometric information captured. This measure has been used for the adequate selection of views of molecular models [17]. The computation of the viewpoint entropy can be done with the aid of graphics hardware using OpenGL, in a similar way to Barral *et al* [4]. The projected area of each face is computed by summing all the pixels that belong to that face, weighted by the solid angle subtended by the pixel. To distinguish between the different polygons, the faces are colour-coded in an item buffer, and to cover all the surrounding of a viewpoint six different views are used.

4 Automatic Indoor Scene Exploration

Viewpoint entropy has already been employed to automatically explore single objects. In that case, a camera was set to lie on a bounding sphere around the object and pointing to the origin, and therefore, only two degrees of freedom were allowed [12]. For indoor scenes, the camera ideally should have six degrees of freedom because there is information in all the directions of sight, but, as this would lead to perform not very natural movements, we will restrict the possible rotation directions. This way, the walkthrough simulates a human path. Moreover, there is also the problem of collision detection, as the camera should not pass through objects.

We have designed an algorithm that automatically explores an indoor scene. Its objective is to navigate through the environment to progressively show all

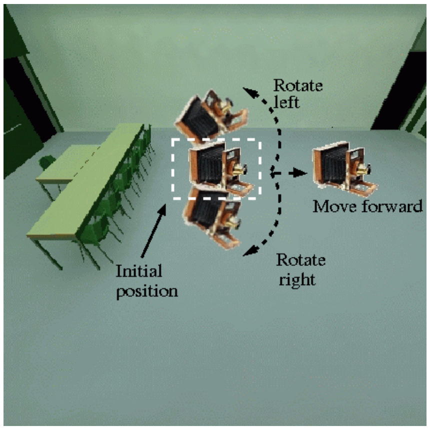


Figure 1: The initial camera and its possible new positions.

the present information to the user. The navigation stops when there is nothing new to discover, or when the algorithm is unable to find new information.

4.1 Camera Movement

As we have mentioned, when a viewer is placed in a closed environment, there is information in all 6 directions of sight. However, in order to avoid unnatural movements of the camera, we will restrict the user to be at a constant height from the floor and the three possible movements will be: move forward, turn right, and turn left, as shown in Figure 1.

Initially, the camera is placed somewhere in the scene and the algorithm automatically starts its navigation. The exploration path will be guided by viewpoint entropy. From the possible movements of a camera, we will select the one with maximum viewpoint entropy that comes from the not yet visited faces, that is, the new view which provides the maximum amount of *new* information. For equal values, we choose the first one. We store a bitmap that encodes the already visited faces. To select a new view we will consider *only the faces that have not been visited yet*, so the measure of entropy will only take into account the faces not visited, the other ones will be counted as background. This ensures the exploration path will be adding new information to the user. As stated in [12], viewpoint entropy properly weights number of faces and projected area.

In order to decide which view to choose, the possible views are first rendered using the OpenGL back buffer and their viewpoint entropy analyzed. Furthermore, to create a natural path, the rotation movement will be restricted not to go back again, that is, if the previous movement was a rotation to the right, the following one can not be a rotation to the left, so, in some cases we will only have to inspect two possible new directions. The original definition of viewpoint entropy measures the information that arrives to a point inside a scene from each direction of sight. As the information we need to measure is the one that arrives to the user of the walkthrough, the only that is really measured is the information that lies inside the viewing frustum, hence, only a rendering and entropy calculation per possible camera position and orientation is needed. This allows to perform an interactive navigation with a 400 × 400 window size in a modest PC (P-III with 320 Mb. and a 32Mb. NVidia Riva TNT2) with no acceleration optimization.

4.2 Stopping Condition

The algorithm must stop the exploration when all (or almost) of the information has been communicated to the user.

This can be detected using the array of visited faces. When its number is equal to the number of faces of the scene, the exploration has finished. However, it is difficult to reach this condition, because some of the faces may not be visible from any camera position. Recall that we are moving the camera at a constant height, some of the polygons may be invisible for such a camera. See for instance Figure 1, for a camera placed at the height of a human head, the polygons forming the bottom part of the table or the chairs will not be visible. On the other hand, they convey possibly not important information. Therefore, the condition must be somehow relaxed. We have found that a threshold of 80% to 90% of the faces is a good threshold for several models.

In some other cases, the scene is too complex because there are some unreachable regions. For these cases, as well as simple scenes with a lot of faces hidden by other objects, we need to avoid an infinite path. Our algorithm accounts for the number of recent steps where no new information was detected (see Section 4.3). When this number exceeds a threshold, it stops.

4.3 Exploration Algorithm Details

As it has been explained, our algorithm works the following way. The camera is placed at an initial position and, for each camera movement, the viewpoint entropy of the possible new positions is analyzed. The following position will be the one presenting a higher value of viewpoint entropy coming from faces that have not been visited. This will prioritize the new faces which project a larger area, because they are more *informative*. However, at this point a new problem arises.

Sometimes the camera has visited all visible faces of a region (say for example a room) but there is other information not reachable by a simple rotation or forward movement. This situation easily happens after a forward movement, as likely most of the faces present in the previous view are still visible. To cope with this situations we have added a distance parameter that determines how far the analyzed views are from the current one. Each time a new decision has to be made, two or three new views are inspected. Usually they are placed near the previous one (for example at 2 or 3 degrees of rotation or steps in world coordinates for forward movements). The term that indicates how far is placed the new camera position is called *distance*, and may be changed for each scene if a different one is required. For each new frame the real rotation of the camera is calculated as

$$rotY = currotY + incDist * distance$$

where $currotY$ is the current Y angle, and $incDist$ is a term with an initial value of 1. If no new information is added in a certain frame, we increment incDist in 1. That is, the following step, the views analyzed will be further than were the previous ones. This ensures a larger radius of exploration in search of new views than if the distance was constant. If *incDist* is not 1, when a new face is found, it is decreased. If *incDist* reaches a high value, it means that we are not going to find any new information, therefore we stop. The value it must reach depends on the value of *distance*. If *distance* is for instance 5, it means that each evaluation will be done at five degrees of our current orientation and five steps forward with the same actual orientation. If *distance* is 5 and *incDist* is 5 it means that the view that will be inspected is at 25 degrees from the current one.

4.4 Results

We have implemented the algorithm detailed above and in Figures 2 and 4, we show some results. We have drawn the camera positions with a square point and the direction of sight at each moment with a line starting from this point. Our exploration stops when more than 80% of the faces are visited. In Figure $2a$ we can see an exploration of the well-known model of the soda hall. The camera is initially placed at a height similar to the one of a person, so it flies over the tables. For forward movements, we must avoid to traverse the walls of the model. This has been solved using a fast simple bounding sphere collision detection algorithm. Figure $2b$ shows a detail of the path.

In Figure 3 we can see the exploration path of a classroom. The initial and final camera positions are marked. In Figures $4a$ and $4b$ show two details of the path. The camera positions are drawn as points and the orientations are drawn as lines and some of them are marked in white for the sake of the visual clarity. Notice how the camera avoids collisions with walls (Figure $4b$).

5 Conclusions and Future Work

We have presented a new method for the automatic exploration of indoor scenes. It is capable of automatically navigating inside a model until most of the polygons have been visited, or when it finds difficult to find any new polygon. To avoid missing important regions, when no information is found, the positions are inspected in the background are placed further than just a simple step. This system may avoid an early termination of the search if the neighborhood has already been visited, in this sense it is slightly better than previous approaches (see [6]). No previous knowledge of the scene is required, only a correct initial placement of the camera is required to give the exploration path a more "human" behaviour. In the future we want to improve our method with some kind of adaptive height, in order to allow the camera to go upstairs and downstairs if the model requires it so.

Acknowledgments

This work has been partially supported TIC 2001-2416-C03-01 project of the Spanish Government, and SGR2001-00296 grant from Catalan Government.

References

[1] D.Plemenos. Declarative modeling by hierarchical decomposition. the actual state of the multiformes project. In *GraphiCon'95*, 1995.

[2] C. Colin. Automatic computation of a scene's good views. In *Proc. MICAD*, February 1990.

[3] T. Kamada and S. Kawai. A simple method for computing general position in displaying three-dimensional objects. *Computer Vision, Graphics, and Image Processing*, 41(1):43–56, January 1988.

[4] P. Barral, G. Dorme, and D. Plemenos. Scene understanding techniques using a virtual camera. In A. de Sousa and J.C. Torres, editors, *Proc. Eurographics'00, short presentations*, 2000.

[5] D. Plemenos and M. Benayada. Intelligent display in scene modeling. new techniques to automatically compute good views. In *Proc. International Conference GRAPHICON'96*, July 1996.

[6] G. Dorme. *Study and implementation of 3D scenes comprehension techniques.* PhD thesis, Université de Limoges, 2001. In French.

[7] E. Marchand and N. Courty. Image-based virtual camera motion strategies. In P. Poulin S. Fels, editor, *Proc.of the Graphics Interface Conference, GI2000*, pages 69–76, Montreal, Quebec, May 2000. Morgan Kaufmann.

[8] E. Bourque and G. Dudek. Automatic creation of image-based virtual reality. In *Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, volume 3209, pages 292–303, Bellingham, WA, October 1997. SPIE - The International Society for Optical Engineering. ISBN 0819426415.

[9] W. T. Freeman. Exploiting the generic view assumption to estimate scene parameters. In *Proc. 4th International Conference on Computer Vision*, pages 347–356, Berlin, Germany, 1993. IEEE.

[10] A. L. Yuille, J. M. Coughlan, and S. Konishi. The generic viewpoint constraint resolves the generalized bas relief ambiguity. In *Proc. of Conference on Information Scienes and Systems (CISS 2000)*, Princeton University, March 15–17 2000.

[11] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: A general approach to setting parameters for computer graphics and animation. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 389–400. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.

[12] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In T. Ertl, B. Girod, G. Greiner H. Niemann, and H.-P. Seidel, editors, *Proceedings of the Vision Modeling and Visualization Conference (VMV-01)*, pages 273–280, November 21–23 2001.

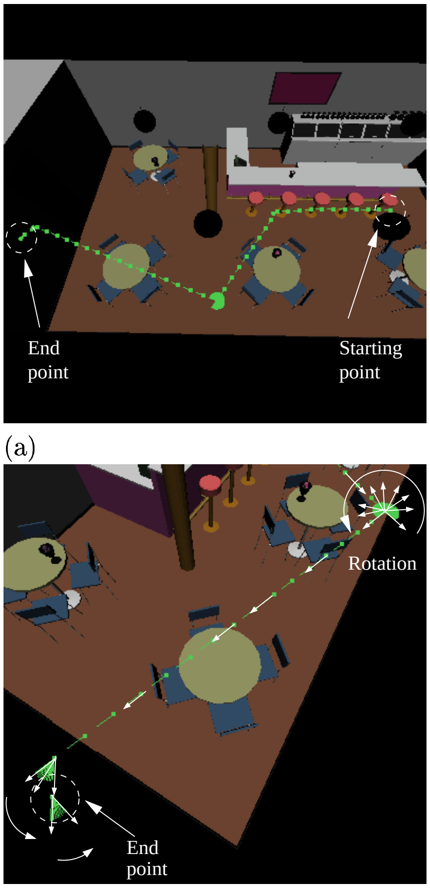
[13] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, 1991.

[14] S. E. Chen. Quicktime VR - an image-based approach to virtual environment navigation. In *Computer Graphics Proceedings (Proc. SIGGRAPH '95)*, pages 29–38, 1995.

[15] R. E. Blahut. *Principles and Practice of Information Theory.* Addison-Wesley, 1987.

[16] J.Rigau, M.Feixas, and M.Sbert. Information theory point measures in a scene. Technical Report IIiA-00-08-RR, Institut d'Informàtica i Aplicacions, Universitat de Girona, Girona, Spain, 2000.

[17] P.-P. Vázquez, M. Feixas, M. Sbert, and A. Llobet. Viewpoint entropy: A new tool for obtaining good views for molecules. In D.Ebert, P.Brunet, and I.Navazo, editors, *Data Visualisation 2002 (Eurographics/IEEE TCVG Symposium Proceedings)*. Eurographics/IEEE, May 27-29 2002.



(a)

(b)

Figure 2: Exploration of the Berkeley soda hall. The points in light green are the camera positions. For the sake of clarity, we have marked in white some of the camera positions and viewing directions.

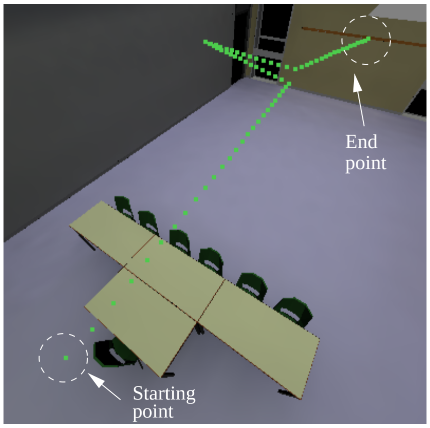
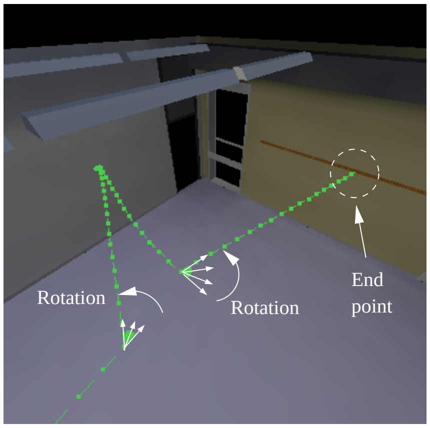
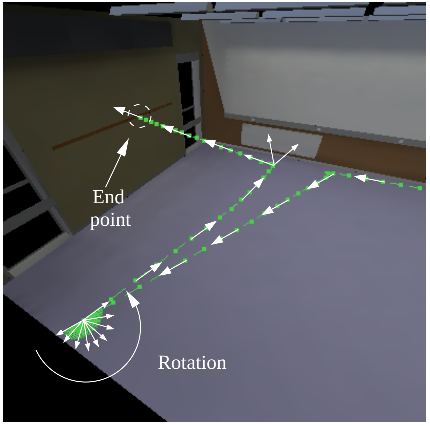


Figure 3: Classroom exploration path.



(a) Middle of the classroom exploration.



(b) Rotation of the camera near a wall.

Figure 4: Figures *(a)* and *(b)* show two close-ups of some parts of the path. Here some of the camera orientations are marked with a white arrow. Note how, in Figure *(b)*, the camera stops when is about to collide with the classroom wall and rotates until it finds a new way.