Viewpoint Selection using Viewpoint Entropy

Pere-Pau Vázquez[‡], Miquel Feixas[‡], Mateu Sbert[‡], and Wolfgang Heidrich[*]

[‡] Institut d'Informàtica i Aplicacions, University of Girona, Email: {pvazquez|feixas|mateu}@ima.udg.es
[*] University of British Columbia, Vancouver, Canada, Email: heidrich@cs.ubc.ca

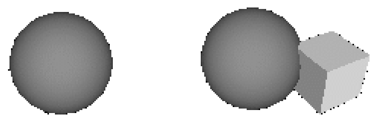
Abstract

Computation of good viewpoints is important in several fields: computational geometry, visual servoing, robot motion, graph drawing, etc. In addition, selection of good views is rapidly becoming a key issue in computer graphics due to the new techniques of Image Based Rendering. Although there is no consensus about what a good view means in Computer Graphics, the quality of a viewpoint is intuitively related to how much information it gives us about a scene. In this paper we use the theoretical basis provided by Information Theory to define a new measure, viewpoint entropy, that allows us to compute good viewing positions automatically. We also show how it can be used to select a set of N good views of a scene for scene understanding. Finally, we design an algorithm that uses this measure to explore automatically objects or scenes.

1 Introduction

The computation of good viewpoints has application in several fields: computational geometry, visual servoing, robot motion, graph drawing, etc. Moreover, it is becoming a key issue in computer graphics due to the increased interest in Image Based Rendering (IBR). IBR techniques use a set of images instead of a geometric representation of the scene [1].

It is difficult to define precisely the term *good view* in Computer Graphics. It seems intuitive to consider a view to be good if it provides a high amount of information for a scene. For instance Figure 1*b* seems to be better than 1*a*, as it gives us more information about the object seen. If the geometry of the scene is known, the set of faces of all objects can be regarded as the information we can work with. In this paper we present a measure based on Information Theory [2], which we call *viewpoint entropy*, that works on the projected areas of the faces and can be interpreted as the amount of information captured from a point. The best viewpoint will be the one with maximum viewpoint entropy. We will show that the new defined measure is useful to compute good viewpoints automatically and the problem of the selection of the minimal set of N good views will be addressed. This is useful for the problem of scene understanding. Another application for automatic exploration of objects or scenes will also been presented.



(a) (b)

Figure 1: A bad view and a good view of a scene containing a sphere and a cube.

The rest of the paper is organized as follows: In Section 2 we review related work and examine the meaning of the term *good viewpoint*. In Section 3 we present our approach. In Section 4 we address the problem of computing a set of N best views. Section 5 covers the application of automatic exploration of objects or scenes. Finally, in Section 6, we summarize the results obtained.

2 Previous Work

In the last decade viewpoint selection has become a very active area of research. In this section we present some methods and strategies that have been applied to different fields and objectives.

2.1 Computer Graphics

Kamada and Kawai [3] consider a viewing direction to be good if it minimizes the number of degenerated faces under orthographic projection. 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 details [4].

Barral et al [4] modify Kamada's coefficient in order to cope with perspective projections. Then they create a heuristic with some other parameters that weigh the number of faces seen from each point and the projected area, adding an exploration parameter which accounts for the faces already visited. An evaluation function is defined that allows to explore the scene in real time. However, they admit not having been able to determine a good weighting scheme for the different factors. This causes some problems with objects containing holes, as these are not captured properly by the algorithm.

Hlavac et al [5] use a set of images to represent an object. Their objective is obtaining an IBR representation to be rendered by interpolation. Consequently they choose a set of reference images positioned around the object in intervals that guarantee error bounds below some threshold during reconstruction of intermediate views. This method is not intended to measure the quality of a single view, as each image is compared with the previous one and only chosen if the degree of dissimilarity is high enough.

2.2 Other Fields

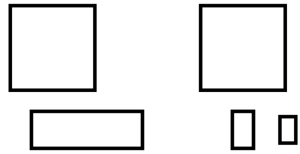
There are several related approaches being used in other fields.

Bourque and Dudek [6] define an interesting point in an image as the one different from the surrounding context. These regions are the ones on which the human attention would focus.

Arbel and Ferrie [7] use entropy for object recognition. In order to do so, they build entropy maps that are used to encode prior knowledge about the discriminability of objects as a function of viewing position. Takeuchi and Onishi [8] measure the entropy of an image based on histograms of intensities in order to find the complex parts of a scene. In both cases, the probability distributions used to compute entropy and the targets pursued are different from the ones addressed in this paper.

Roberts and Marshall [9] select a minimized number of views for complete coverage of the surface of three dimensional objects. They define a good view for a face to be one that is viewed head on, or as they put it, "the direction that lies within the faces visibility region and has the smallest angular offset from its inverse surface normal". For a set of faces they define a good view as the direction that is simultaneously at a minimum angle from each of the corresponding surface normals. A set of methods in Computer Vision address the problem of "Next Best View" selection, which rely on the information collected previously in order to choose the next interesting viewpoint. Unfortunately a unified quality criterion does not exist yet (see for example Massios and Fisher [10]).

The problem of computing good viewpoints is also related to visual servoing (see Marchand and Courty [11]).



(a) (b)

Figure 2: *(a)* and *(b)* are from the same scene, which one should we take?

2.3 What is a *Good View*?

As we have already mentioned, some definitions have been proposed, but there is no consensus about what a good viewpoint is. In spite of that, it seems intuitive that the best viewpoint is the one that obtains the maximum information of a scene. A good view must help us to understand as much as possible the object or scene represented.

From the previous work, we extract two parameters that are especially related to the quality of a viewpoint: the projected area and the number of faces seen. It seems necessary to obtain a quality function that weighs those two parameters. By itself, the projected area does not tell us about the amount of detail we can see and cannot be used for indoor scenes, because the projected area is constant. On the other hand, even though we have a high number of faces, they could be small and thus provide little information about a scene. In Figure 2*a* and 2*b* two projections of the same scene are shown. They exhibit different number of faces and different projected area. Which one should we take?

3 Viewpoint Entropy

One of the features we associate with the *goodness* (or *quality*) of a viewpoint is the amount of *information* it provides us with. We assume that the information we talk about is *visibility*. Recently, Information Theory tools have been used to study scene visibility [13]. Here we define *viewpoint entropy*, a new measure that allows us to obtain good viewpoints of a scene. We will see how the viewpoint entropy incorporates both the projected area and the number of faces, and can be understood as the amount of information captured by the viewpoint. In a novel work Rigau *et al* have arrived to an equivalent measure when studying the visibility complexity of 2D scenes [14].

3.1 Viewpoint Entropy

The *Shannon entropy* [15, 2] 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 $0 \log 0 = 0$ for continuity. 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*.

To define viewpoint entropy we use as probability distribution the relative area of the projected faces over the sphere of directions centered in the viewpoint. Thus, we define *viewpoint entropy* as

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

where N_f is the number of faces of the scene, A_i is 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 faces and consequently $A_0 = 0$. Hence, A_i/A_t represents the *visibility* of face i with respect to point p. It is important to remark that, with respect to the total area of face i, the projected 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. This justifies the use of projected area as the probability distribution to compute entropy.

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, in an open scene the maximum viewpoint entropy is $\log(N_f + 1)$, and in a closed scene it is equal to $\log N_f$. We define the *best* viewpoint as the one that has maximum entropy, i.e. maximum information captured.

One of the drawbacks of this measure could be the use of background as another face. This is required because when computing entropy we must use a probability distribution function, otherwise we would not have a consistent entropy measure. On the other hand, this is not the only reason, a probability distribution could be built by normalizing the measures without using the background, but such a measure cannot handle with distances: projecting the scene under the same direction but at a different distance would give the same value. The use of background gives the objects which are near higher entropy than the ones which are far. It is important to notice that this could lead to small errors, but this can only happen only if we are seeing all the faces (including background) with the same projected area (maximum viewpoint entropy). If starting from this position we move the camera in such a way that the background region decreases and the rest of faces increase their projected area, the total entropy would diminish, instead of growing. However, this does not happen in practice mainly because it is very difficult to obtain such a viewpoint (usually a number of faces are not visible). Furthermore, we work with many faces and the maximum error produced could be $log(N_f+1) - logN_f$, which for a big value of N is negligible.

3.2 Implementation

The computation of 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 up 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 view directions six different views are used. This is the general method for computing viewpoint entropy inside a scene.

Let us consider the specific example of a single object or a group of objects surrounded by empty space. In this case, the process can be simplified by just rendering one view (if all the projected area lies inside this view), dividing the relative projected area of each face by the area of the sphere, and adding the contribution of the background. The algorithm renders the scene from a set of points placed at regular positions over a bounding sphere of the object. At each point, the item buffer is read and the entropy is calculated. The distance of the camera and the number of viewpoints can be modified by the user. Algorithm 1 solves the problem of computing the best viewpoint of an object.

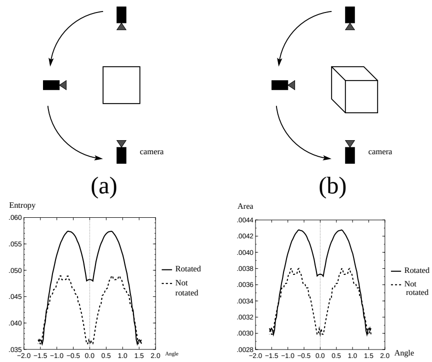
Algorithm 1 Computes the view with the highest entropy of an object.

Select a set of points placed in regular positions all around the object
maxI ← 0
viewpoint ← 0
for all the points **do**
  aux ← Compute the viewpoint entropy
  if aux> maxI **then**
    maxI ← aux
    viewpoint ← current point
  end if
end for
Write maxI and viewpoint

3.3 Results

We have implemented the method described above and tested it for several objects. We have also compared the results with the relative projected area. We show how these metrics differ and how entropy provides good results to get the best view. In the next section this measure will also be used to compute a set of N good views.

The algorithm computes the viewpoint entropy of 17-18 fps although no optimizations were made.



(a) (b)

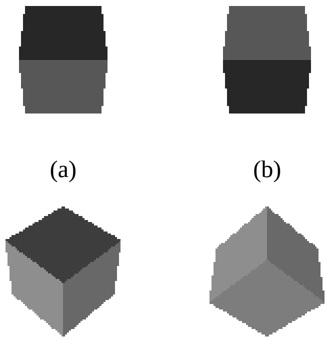
(c) Entropy (d) Proj. area

Figure 3: The diagrams in *(a)* and *(b)* show two capture paths of a camera around a cube. *(c)* and *(d)* plot the viewpoint entropy and the projected area respectively in the Y axis against the rotation angle. The dashed line corresponds to the first path and the continuous one to the second.

The method was tested on a Pentium III processor at 700 MHz.

To prove the quality of our measure, we computed the viewpoint entropy for several points lying on a sphere that surrounds a cube. The camera follows the paths depicted in Figures 3*a* and 3*b*. In the second example the cube has been rotated 45 degrees around the Y axis in order to compare the results. In both cases, when the camera is at the top of the object the rotation angle is $\Pi/2$ and at the bottom $-\Pi/2$. Figures 3*c* and 3*d* show the measures of viewpoint entropy and the projected area for both situations respectively. It is easy to see how entropy depends on the number of faces and the amount of area seen at every position. Although the first impression is that results are similar to the ones obtained from the projected area approach, we will show later that this does not happen in general. In Figure 4 we can see four snapshots of what is captured by the camera at the points of maximum entropy. Figure 4*a* and 4*b* belong to the first case (Figure 3*c*) and Figures 4*c* and 4*d* to the second (Figure 3*d*). As expected, the information received is largest at the points one would intuitively say that are the most *informative*.

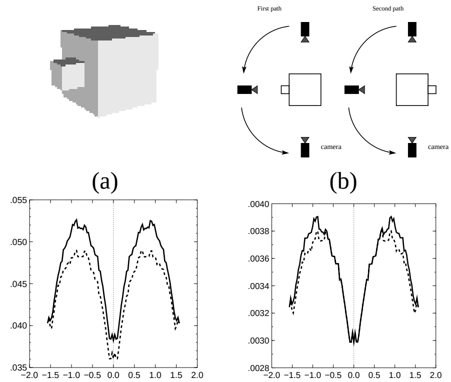
Let us consider another example for a more detailed comparison with the projected area approach. Figure 5*a* shows another scene we are studying, and 5*b* sketches the two paths that the camera fol-



(a) (b)

(c) (d)

Figure 4: *(a)* and *(b)* correspond to the points of maximum entropy for the path in Figure 3*(a)*, and *(c)* and *(d)* to the path in Figure 3*(b)*.



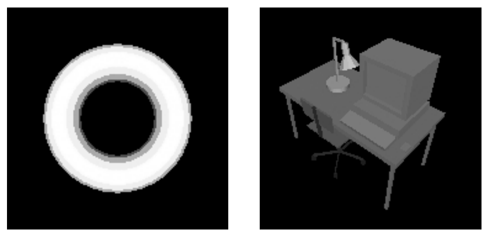
(a) (b)

(c) Entropy (d) Proj. area

Figure 5: *(b)* depicts the paths of the camera around a pair of cubes. *(c)* and *(d)* show the entropy and projected area in the Y axis respectively against the rotation angle. The dashed line corresponds to the second path and the continuous one to the first.

lows around the cubes. Figure 5*c* shows how entropy is sensitive to the number of faces in situations where the projected area by itself (Figure 5*d*) does not distinguish if the number of faces is different.

In Figures 6*a* and 6*b* we can see the points of maximum viewpoint entropy computed around a torus and a desk. Entropy decreases with increasing distance, because the projected area of each face is smaller. On the other hand, sometimes it is not enough with a single view, as it might not provide enough information from the scene and thus, we will need more images.



(a) (b)

Figure 6: The points of maximum viewpoint entropy of a torus *(a)*, and a desk *(b)*.

4 Selection of Good Views for Scene Understanding

So far, we have defined a measure to compute the quality of a view and to find out which is the best view according to visibility information. In this section we will address the problem of selecting a set of N views that may be best suited for representing the scene. This problem has to be faced in different ways according to the target pursued. We study here the problem of scene understanding. The purpose of scene understanding techniques is to select a minimal set of views that gives a good representation of the scene to the user. We have the following restrictions on the views:

- They should contain a high level of information about the scene.
- They should cover all the visible faces.

A naive method would be to choose a set of images with high viewpoint entropy. However, this does not ensure that all faces are covered. We avoid this problem with the use of bitmaps to encode the visibility of the faces from each point. Algorithm 2 depicts this method.

The algorithm stops when a certain percentage of the total visible faces are visited. By selecting views with high entropy and which show *new* faces we ensure that all the scene is covered and that we get views of good quality. Figures 9 and 10 show the resulting viewpoints of this algorithm for a molecule of isobutanolamine and a desk.

Algorithm 2 Computes the set of views with high entropy of an object.

Select a set of points placed in regular positions all around the object
for all the points **do**
  Compute the viewpoint entropy and store it
  Store a bitmap encoding the visibility of the faces from the point
end for
Order the points in decreasing viewpoint entropy
Select the first point {the one with max. entropy}
Accumulate the visited faces in a bitmap
$i \leftarrow 0$
while $i <$ totalPoints **and not** finished **do**
  if numFacesNotSeen(i) $>$ threshold **then**
    Select point i
    Accumulate the visited faces in a bitmap
    finished $\leftarrow$ isFinished(VisitedFacesno)
  end if
  $i \leftarrow i + 1$
end while

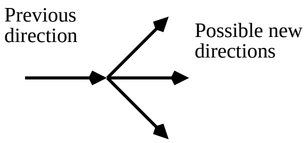
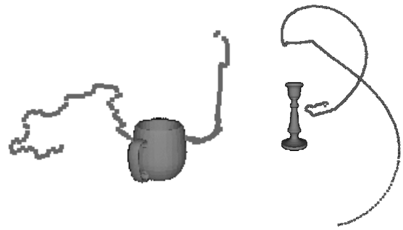


Figure 7: The possible directions of the navigation according to the previous direction.

5 Automatic Exploration of Scenes

In section 4 we have shown how to compute a minimal set of views that represents an object or scene. This method can be confusing to the user when changing from one view to another if the new one is completely different from the preceding one. This can be solved by simulating a walkthrough around a scene. There are two different approaches in bibliography: flying around the scene (over a bounding scene) and navigating through it. We have designed an algorithm that uses entropy to fly all around the objects but an extension inside the scenes can be implemented in a straightforward fashion.

We initially fix an arbitrary starting point and a direction of navigation. Similarly to Barral *et al* [4], subsequent moves are chosen between three possible new directions, according to the last movement, to ensure a smooth displacement of the camera (see Figure 7).



(a) (b)

Figure 8: Exploration path around a mug and a candlestick.

The next view is chosen according to the entropy and the number of faces not yet visited (as in the method in section 4). To evaluate the quality of the three possible next positions we multiply the viewpoint entropy of the next point and the bit difference of the possible new view and the accumulated one. In case none of the three possible views shows a new face we choose the one which lies furthest from the initial position. At this point other strategies could be selected, such as traveling towards non-explored regions, continuing in the same direction of the last movement, etc. The one we chose is simple and performs well in most cases. Figure 8 shows the path around a mug and a candlestick. The exploration stops when a certain percentage of the faces (90-99%) have been visited.

6 Conclusions and Future Work

In this paper we have defined a new measure, the *viewpoint entropy* of a point in a scene. This measure is based on Information Theory and can be interpreted as the amount of information seen from a point. Consequently, it can be used to determine good viewpoints. The best viewpoint has been defined as the one that has maximum entropy. Furthermore, viewpoint entropy is used to compute the minimal set of N good views which gives a good representation of the scene. This set of views is built according to the requirements of scene understanding: the set of views must be minimal, and it must cover all the visible faces.

We have developed an appearance-based algorithm suitable to addressing these problems. The algorithm is appearance-based in the sense that it only measures what we can really see. This means that we will apply equation (1) to the objects that project at least one pixel in screen, thus, which are perceivable by an observer. In future we will address the problem of navigating inside an scene. Although our measure works for indoor scenes, the camera orientation is relevant and thus, extending the method to navigation inside scenes is not straightforward. We also want to study optimization methods for viewpoint entropy computation, and explore artistic issues, which could deal to a totally different definition of viewpoint goodness.

7 Acknowledgements

This work has been supported by SIMULGEN Open ESPRIT project #35772 and research grant BR98/1003 of Universitat de Girona. The authors want to thank Hartmut Schirmacher, Philippe Bekaert, and the anonymous reviewers for their useful comments and help.

References

[1] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Proc. of SIGGRAPH 95*, pages 39–46, August 1995.

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

[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, *Computer Graphics Forum (Proc. Eurographics ’00)*, volume 19, 2000.

[5] V. Hlavac, A. Leonardis, and T. Werner. Automatic selection of reference views for image-based scene representations. In *Lecture Notes in Computer Science*, pages 526–535, New York, NY, 1996. Springer Verlag. Proc. of European Conference on Computer Vision ’96 (ECCV ’96).

[6] 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.

[7] T. Arbel and F. Ferrie. Viewpoint selection by navigation through entropy maps. In *Proc. of the 7th IEEE International Conference on Computer Vision (ICCV-99)*, volume I, pages 248–254, Los Alamitos, CA, September 20–27 1999. IEEE.

[8] Y. Takeuchi and N. Ohnishi. Active vision system based on information theory. *Systems and Computers in Japan*, 29(11):31–39, 1998.

[9] D.R. Roberts and A.D. Marshall. Viewpoint selection for complete surface coverage of three dimensional objects. In *Proc.of the Britsh Machine Vision Conference*, 1998.

[10] N.A. Massios and R.B. Fisher. A best next view selection algorithm incorporating a quality criterion. In *Proc.of the Britsh Machine Vision Conference*, 1998.

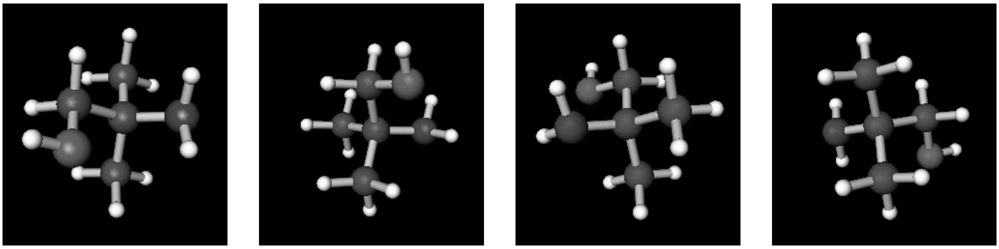
[11] E. Marchand and N. Courty. Image-based virtual camera motion strategies. In P. Poulin S. Fels, editor, *Proc.of the Graphics Interface Conference, GI2000*, pages 69–76, Montreal, Quebec, May 2000. Morgan Kaufmann.

[12] L. He, M. F. Cohen, and D.H. Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 217–224. ACM SIGGRAPH, Addison Wesley, August 1996.

[13] M. Feixas, E. del Acebo, P. Bekaert, and M. Sbert. An information theory framework for the analysis of scene complexity. In *Computer Graphics Forum (Proc. Eurographics ’99)*, volume 18, pages 95–106, Milan, Italy, 1999.

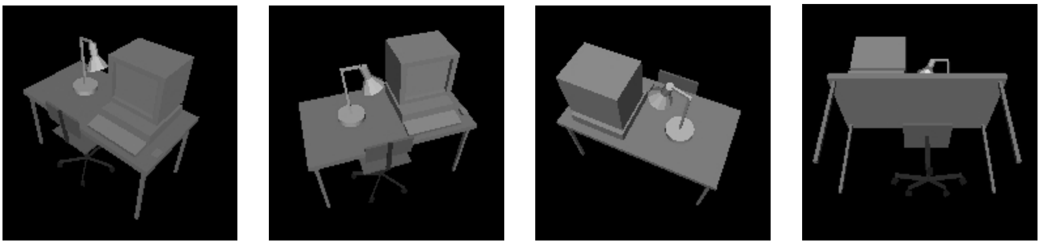
[14] J.Rigau, M.Feixas, and M.Sbert. Information theory point measures in a scene. Technical Report IIiA-00-08-RR, Institut d’Informàtica i Aplicacions, Universitat de Girona, Girona, Spain, 2000.

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

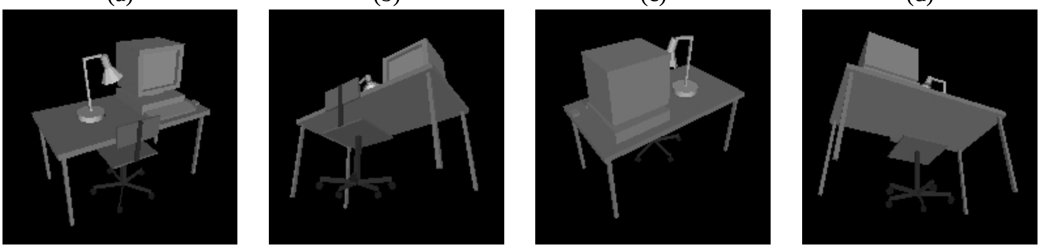


(a) (b) (c) (d)

Figure 9: The four view points of a molecule of isobutanolamine obtained with the bitmap method.



(a) (b) (c) (d)



(e) (f) (g) (h)

Figure 10: The eight view points obtained with the bitmap method for a desk.