Comparison of Two Chosen Methods of Feature Dependent Views Generation

Monika Dąbkowska, Wojciech S. Mokrzycki
ICS PAS, Graphics & Image Processing Group
02–237 Warsaw, Ordona 21, monika, wmokrzyc@ipipan.waw.pl

Abstract. Comparison of two methods of feature dependent views generation created and implemented by our group. The criteria of comparison are: what views are generated by each method? How many views each method generates? Presented methods are used for preparing view models for a visual identification system.

Keywords: view model, view representation, model-based object identification, feature dependent views generation, standard view model, view sphere, view points, completeness of a set of views.

1. Introduction

Our goal is to work out a model of convex polyhedra for a visual identification system. The system is based on a model database and acts similarly to human memory. It perceives the reality by two cameras which act as eyes. Then it compares the analysed depth map with each model in the database. We want to make a model which would be possibly the best for comparing with the depth map so the system needs only little computation to answer the question whether it fits the model or not. Therefore we decided to create a view model. It means that an object is represented by a set of its views.

Two main problems have occurred while we were thinking. The first is how to place the view points? They can be disposed in a uniform way ([12, 15, 25]) or non-regularly in places depending on the object features ([5, 13, 19, 21, 23, 24]). Each object has an infinite number of views but for a computer database we can have only a finite number of views. We need to select as many wiews as necessary to have a good representation but, on the other hand, as few as possible to occupy a reasonable amount of memory. So, which views should be chosen to satisfy the conditions?

The second problem is to prove that the representation is complete. It means that for each view point the related view is in the database.

2. The View Sphere Concept

To simplify the task and to standardise the views we have decided to limit the set of views only to these views which can be seen from a sphere (called **the view sphere**), which has the centre O in the centre of the object and its radius R is known, fig. 1. The radius of the view sphere depends on the size of the object and on the angle of the cone of vision 2α (which is a technical parameter of the cameras). The exact relation is described in [13]. This radius is fixed so that the model of an object for which we try to create the views fills the viewing space but also so that maximum elements of the model are visible (not too near and too far) and it is constant during the process of generating views of the model. This implicates a restriction on the cameras (during the process of recognition) which have to be placed in a proper distance from the objects on the scene, because the perspective projection makes extreme elements of the

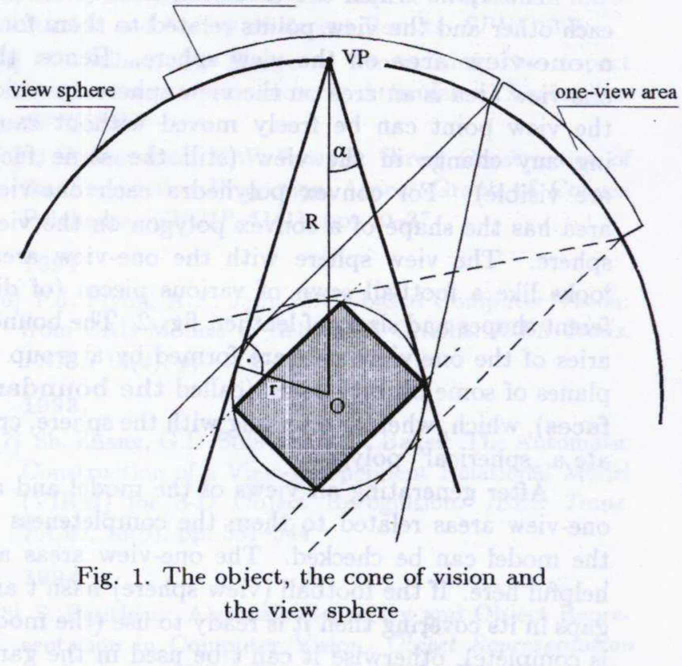

Fig. 1. The object, the cone of vision and the view sphere

visible part of the object disappear while increasing the distance. The distance between the cameras and the objects doesn't have to be exactly equal, but it should be comparable to the radius of the standard view sphere (used during creating the view model) of each object on the scene (it depends on the sizes of the objects and can be easily computed). This condition limits the usage of such a model database but it is still good for systems which can approach the scene, for example when the cameras are placed above the table and they can be moved down if necessary or in the case of a robot which can decrease the distance to the object.

The set of views which can be seen from the surface of the view sphere is only a set of some standard views because of the constant distance to the centre of the model.

3. The View and the Completeness of the View Model

To begin with we tried to construct a model of convex and opaque polyhedra. We considered only geometrical and not e.g. photometrical features (color, texture...) of the objects. According to this it is sufficient to match the visible faces, edges and vertices of the object with a view of a model to recognise the class of objects. Because the visible edges and vertices are elements of the visible faces, therefore we have defined **a view** as a set of the visible faces of the object. It occurred automatically that many views are represented by only one view, so called **the characteristic view**. The characteristic view can be any view of the group of identical views (identical in the sense of the above mentioned definition).

The views which are identical are adjacent to each other and the view points related to them form **a one-view area** on the view sphere. Hence, the one-view area is an area on the view sphere on which the view point can be freely moved without causing any change in the view (still the same faces are visible). For convex polyhedra each one-view area has the shape of a convex polygon on the view sphere. The view sphere with the one-view areas looks like a football sewn of various pieces (of different shapes and sizes) of leather, fig. 2. The boundaries of the one-view area are formed by a group of planes of some object's faces (called **the boundary faces**), which, when intersecting with the sphere, create a „spherical" polygon.

After generating all views of the model and all one-view areas related to them the completeness of the model can be checked. The one-view areas are helpful here. If the football (view sphere) hasn't any gaps in its covering then it is ready to use (the model is complete), otherwise it can't be used in the game — some new pieces of leather (some new views and new one-view areas related to them) are needed to complete the covering.

The algorithm of views generation ([19]) can be divided into five phases. The first phase is generating and storing of the set of standard views. The second — determining the one-view areas for each stored view. The third — checking of the covering of the view sphere. The fourth — determining of the gaps in the covering (indicating the gap boundaries). The fifth — generating of the missing views. The last phase isn't yet precisely specified. It may demand dividing the gap into smaller gaps and then generating the views. If it is possible to use here the sequence of instructions from the first phase then the algorithm could be taken in a loop.

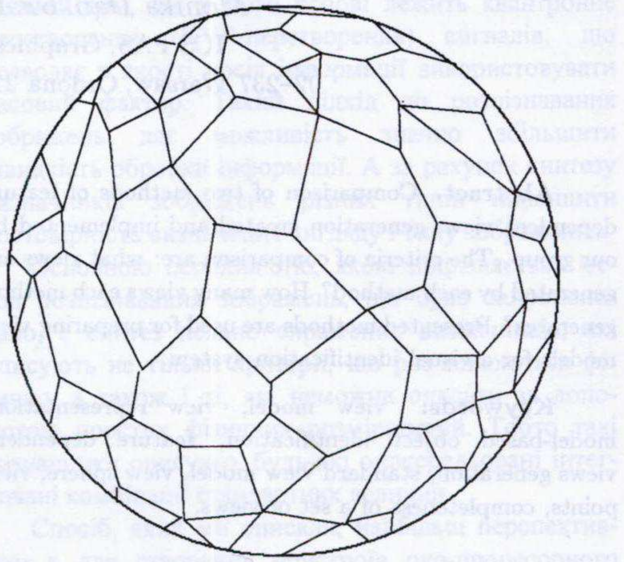

Fig. 2. Covering of the view sphere by the one-view areas

4. The Method of Three Types of Views

As written above we have made an assumption about the objects which can be recognised by the system: they can be convex and opaque polyhedra. According to the geometrical features of such objects — faces, edges and vertices — our first idea was to create the views dependent on those three features of the polyhedral object. The first method consists in determining the views of three types: face-centered, edge-centered and vertex-centered views, [13]. In the first case the viewing direction goes through the centre of the view sphere and the centre of the selected face. The view point lies on the sphere in the place where the so defined viewing direction intersects the sphere. In the case of the edge-centered views the viewing direction goes through the centre of the sphere and the centre of the selected edge. For the vertex-centered views it goes through the vertex. The sense of the viewing vector is always the same — toward the centre of the sphere.

After generating the face-centered view for each face of the polyhedron, the edge-centered view for each edge and the vertex-centered view for each vertex the one-view area for each view should be computed.

The reasoning for one-view areas, checking the covering, determining the gaps and the missing views should be repeated as in section 3.

5. The Method of Dispersing of the View Point

The second method is more accurate. At the beginning the first view is determined and registered. It can be any view computed by the first algorithm (described in section 4). In our implementation the face-centered view was chosen, because it is most possible that it won't be an incident view, [23].

After this the one-view area of that first view is computed. When all boundary faces (of the one-view area) are known, the view point is moved to the other side of each of them sequentially. The aim of this step is to find the neighbouring views for the first view. Firstly, a middling plane for the first view contour edge is computed. The middling plane is a plane containing the viewing direction (view point and the centre of the view sphere — point (0,0,0)) and the centre of the chosen (first) view contour edge. This plane intersecting with the view sphere determines the direction of the view point movement. The view point moves along the big circle formed by the middling plane and the sphere in the direction opposite to the chosen view contour edge. It stops after crossing the nearest boundary face (of the one-view area). Then the new view for the new location of the view point is computed and registered. Afterwards the view point comes back to its previous position, the middling plane for the next view contour edge is computed and the movement of the view point is continued along the middling plane.

When all view contour edges of the first view are used, the next view is chosen from the register and the whole procedure (determining the one-view area and the neighbouring views) is repeated until there are still any views in the register.

To be sure that the set of views is complete, the checking of the covering, as in section 4, should be done and if there are still any gaps, then the missing views should be computed.

6. Comparison

Both methods are implemented and have been tested on the same four objects: hexahedron (regular), pentahedron (saddle roof), heptahedron and octahedron. The results of the methods are in some cases different.

In the case of hexahedron and pentahedron both methods generate the same views. The hexahedron needs 26 views to be well represented (the view sphere is completely covered by the related one-view areas). Logically, the pentahedron, which has less faces than the hexahedron, should need less views: each method has generated the same 20 views of it, which also form the complete model. For more complex (with more faces and without symmetry axes) solids the situation is worse. The first method generates less views of such solids than the second one. But still it is very frequent that both models of nonsymmetric solids are incomplete. There are still some missing views.

7. Conclusion

Both described methods are efficient for regular or symmetric solids. For nonsymmetric solids the second method is better. It provides more views. To complete both methods an algorithm for detecting boundaries of the gaps and a method of generating the missing views is needed. It is quite possible that for generating missing views the original method of generating views could be used.

References

1987

[1] J.H. Connell, M. Brady: Generating and Generalizing Models of Visual Objects. *AI*, 31, pp. 159–183

1989

[2] C. Hansen, T.C. Henderson: CAGD-Based Computer Vision. *IEEE Trans. PAMI*, 11(11), pp. 1181–1193

1990

[3] K.W. Bowyer, Ch.R. Dyer: Aspect Graphs: An Introduction and Survey of Recent Results. *SPIE*, 1395

[4] M. Sallam, K. Bowyer: Generalizing the Aspect Graph Concept to Include Articulated Assemblies. *SPIE*, 1395

[5] J.H. Stewman, K.W. Bowyer: Direct Construction of the Perspective Projection Aspect Graph of Convex Polyhedra. *CVGIP*, 51(1), pp. 20–37

1991

[6] P.J. Flynn, A.K. Jain: CAD-based Computer Vision: from CAD Models to Relational Graphs. *IEEE Trans. PAMI*, 13(2), pp. 114–132

1993

[7] Sh. Zhang, G.D. Sullivan, K.D. Baker: The Automatic Construction of a View-Independent Relational Model (VIRM) for 3-D Object Recognition. *IEEE Trans. PAMI*, 15(6), pp. 531–544

1994

[8] S. Petitjean: Algebraic Geometry and Object Representation in Computer Vision. *Object Representation in Computer Vision. Proc. Int. NSF-ARPA Workshop*, NY, pp. 155–165, Dec.

1995

[9] P.A.R Cole, M.S. Khan: Modelling 3D Rigid Objects Using the View Signature II Representation Scheme. Proc. CAIP'95, *LNCS*, 970, pp. 154–161, Sep.

[10] A. Leonardis, S. Kovačič, F. Pernuš: Recognition and Pose Determination of 3D Objects Using Multiple Views. Proc. CAIP'95, *LNCS*, 970, pp. 778–783, Sep.

[11] W.S. Mokrzycki: System of Stereovisual Perception — SSP. Description of Conception and Implementation. *ICS PAS Reports*, 789, Warsaw (in Polish)

[12] O. Munkelt: Aspect-Trees: Generation and Interpretation. *Computer Vision and Image Understanding*, 61(3), pp. 365–386

1996

[13] M. Dąbkowska, W.S. Mokrzycki: Generating Views of 3D Objects for a Visual Identification System. *ICS PAS Reports*, 809, Warsaw (in Polish)

[14] M. Dąbkowska, W.S. Mokrzycki: Determining One-view Areas for Convex Polyhedra. *ICS PAS Reports*, 823, Warsaw (in Polish)

[15] V. Hlavač, A. Leonardis, T. Werner: Automatic Selection of Reference Views for Image-Based Scene Representations. 4th European Conference on Computer Vision, Cambridge, UK, Proceedings vol.1, *LNCS*, 1064, pp. 526–535, Apr.

[16] K. Tarabanis, R.Y. Tsai, A. Kaul: Computing Occlusion-Free Viewpoints. *IEEE Trans. PAMI*, 18(3), pp. 279–292

1997

[17] M. Dąbkowska, W.S. Mokrzycki: Structures and Completeness of View Models in the Visual Identification System. *ICS PAS Reports*, 833, Warsaw (in Polish)

[18] M. Dąbkowska: Spherical Representations of Objects for Visual Identification Systems. *Proc. of 3rd Symposium of Micro- and Optoelektronics Institute of Warsaw University of Technology „Technics of Image Processing" TPO'97*, Serock, Oct. 29–31 (in Polish)

[19] M. Dąbkowska, W.S. Mokrzycki: A Multiview Model of Convex Polyhedra. *MG&V*, 6(4), pp. 419–450

[20] C. Dorai, A.K. Jain: Shape Spectrum Based View Grouping and Matching of 3D Free-Form Objects. *IEEE Trans. PAMI*, 19(10), pp. 1139–1146

[21] C.B. Madsen, H.I. Christensen: A Viewpoint Planning Strategy For Determining True Angles On Polyhedral Objects By Camera Alignment. *IEEE Trans. PAMI*, 19(2), pp. 158–163

[22] I. Shimshoni, J. Ponce: Finite-Resolution Aspect Graph of Polyhedral Objects. *IEEE Trans. PAMI*, 19(4), pp. 315–327

1998

[23] M. Dąbkowska, W.S. Mokrzycki: A New View Model of Convex Polyhedra with Feature Dependent View. *MG&V*, 7(1/2), Proc. GKPO'98, Borki, Poland, pp. 325–334, May

[24] M. Dąbkowska, W.S. Mokrzycki: Generating of Convex Polyhedra Views with Direct Tracing of One-View Areas. *ICS PAS Reports*, 861, Warsaw, (in Polish)

[25] S. Kovačič, A. Leonardis: Planning Sequences of Views For 3D Object Recognition and Pose Determination. *PR*, 31(10), pp. 1407–1417