Locating sensors into a 3D polyhedral environment

Andrea Bottino
DAUIN – Politecnico di Torino
Corso Duca degli Abruzzi, 24
10128, Torino, Italy

andrea.bottino@polito.it

Aldo Laurentini
DAUIN – Politecnico di Torino
Corso Duca degli Abruzzi, 24
10128, Torino, Italy

aldo.laurentini@polito.it

ABSTRACT

In this paper we present an optimal 3D sensor location algorithms that can locate sensors into a polyhedral environment that are able to see the features of the objects in their entirety. Limitations due to real sensors can be easily taken into account. The algorithm has been implemented, and examples are also given.

Keywords

Inspection, surveillance, sensor planning, sensor positioning, art gallery problem, aspect graph, integer surface covering.

1. INTRODUCTION

Sensor planning is an important research area in computer vision. It consists of automatically computing sensor positions or trajectories given a task to perform, the sensor features and a model of the environment. Sensor panning problems require considering a number of constraints, first of all the visibility constraint. To this effect, the sensor is usually modeled as a point, and referred to as a "viewpoint". A feature of an object is said to be visible from the viewpoint if any segment joining a point of the feature and the viewpoint does not intersects the environment or the object itself and lies inside the frustum. In this paper we will deal with the problem of finding and locating the minimum number of sensors able to see at the same time all points of the surface of an object. The sensors are supposed to be omni directional. The major contribution of this paper is to present a 3D algorithm for finding a set of zones where a minimal set of viewpoints, able to observe the entire surface of the object, can be independently located. The algorithm works for multiply connected and unconnected general polyhedra, and can locate viewpoints able to observe the interior or the exterior surfaces of the polyhedra. For finding an optimal solution the view space needs not to be discretized, the only restriction being that each face must be completely observed by at least one viewpoint. It is also worth observing that, if the faces are subdivided into smaller areas, the solution provided by the algorithm converges towards the optimal solution of the unrestricted problem.

2. THE 3D ALGORITHM

The algorithm is an extension in 3D of the algorithm presented in [Bot04]. The outline of the 3D sensor positioning algorithm is the following:

Step 1 - Compute a partition Π of the viewing space into regions Z_i such that: a) the same set $\mathbf{F}_i = (F_p, F_q, ...F_t)$ of faces is completely visible from all points of $Z_i \ \forall \ i$; b) the Z_i are maximum regions, i.e. $\mathbf{F}_i \neq \mathbf{F}_k$ for contiguous regions. The list of the visible faces must be associated to each region of Π.

Step 2 - Select the *dominant and essential regions* of Π. A region Z_i is defined to be dominant if there is no other region Z_k of the partition such that $\mathbf{F}_i \subset \mathbf{F}_k$. A region is *essentials* if it covers some face not covered by the other dominant zones (and must be selected).

Step 3 - Select an optimal (or minimum) solution. A minimum solution consists of a set of dominant regions $\mathbf{S}_j = (Z_{k1}, Z_{k2}, ..., Z_{kn})$ which covers all the faces with the minimum number of members.

The environment is assumed to consist of simple polyhedrons. Regarding the complexity, step 1 and 2 can be performed in polynomial time; step 3 requires, in the worst case, exponential time. However, polynomial time can be achieved using greedy heuristics. Partition Π is built by means of a particular set of surfaces, called the *active visibility surfaces*.

A *visibility surface* (VS) relative to a face divides the space into areas where the surface is partially or totally hidden. A VS is an half-open planar surface starting at one of the edges or at a vertex of a face, lying in the half space opposite to the inner side of the face. Also, each potential VS has a positive side, which is the side closer to the originating face. The angle between the normal of this surface and the normal of the originating face is in the range $[0,\pi]$. Examples can be seen in Fig. 1, where the arrows mark the positive side of the VSs.
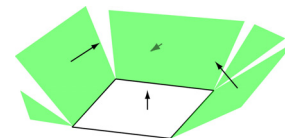

Fig. 1: example of VSs

Each VS can have an *active* and an *inactive* part. Only the active VSs are the effective boundaries of the visibility region of the corresponding surface. A VS is active when: 1) the angle with the normal of the originating face is 0 and the surface is not entering the object in the proximity of the originating face (VS of type I); 2) it is tangent to another part of the object (or to another object) and in the neighborhood of this part, the inner side of the object lies on the negative side of the potential VS (that is, the VE surfaces defined in [Gig91]). Those surfaces are defined by a vertex of the object and an edge of the face (VS of type II) or by an edge of the object and a vertex of the face (VS of type III). In both cases, the active part of the VS starts at the intersection point. Examples can be seen in Fig. 2. Each active VS has a cross operator `^j` associated, where `j` indicates the surface originating the VS. The operator has a direction, pointing to the area where the face is visible (see Fig. 2). In the

following we will use a result found in [Tar96], that is: if the face is convex (and simply connected), its visibility region is connected. This property allows pruning radically the initial set of potential VSs of a face. Therefore any concave face will be split into convex parts.
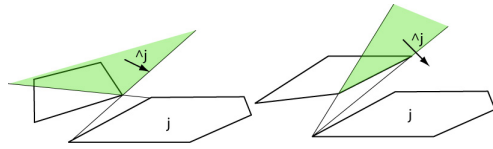

Fig. 2: VS of type II (left) and type III (right)

Finding the active part of a VS

For each initial VS, we must identify the part which is indeed active. In order to construct the active VSs of type I, we must find the regions of the plane P of a face F from where its 2D silhouette is completely visible. Forcing F to be convex, its 2D silhouette from a viewpoint V corresponds to the list of edges of F totally visible from the viewpoint. The active VS of type I can be constructed in the following way:

1. find on P the polygons corresponding to the intersection of the objects with P; let S, the initial active VS, be the union of all the regions in P where the 2D silhouette of F is completely visible;

2. for each edge, define as positive the side of the edge pointing to the internal part of the face; for each edge of the face closed by another face, or by another object intersecting the plane of the face, let H be the half plane in P bounded by the line containing the edge and corresponding to the positive side of the edge. Then S = S∩H (see Fig. 3)

Consider Fig. 3 where a face F and its active VS of type I are shown; edges e_1 and e_2 are closed by other objects.
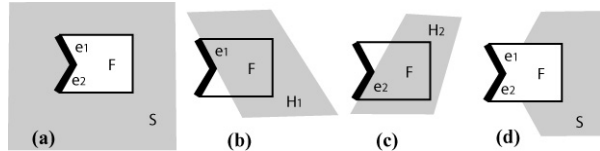

Fig. 3: edges e_1 and e_2 are closed by other objects

The initial active VS on P can be evaluated using a modified version of the 2D region labeling algorithm of [Bot04].

The active part of a VS of type II (III) can be found determining the parts of the initial unbounded triangular (trapezoidal) surface where the originating edge is entirely visible (see Fig. 4).

Additional rules for potential VS to be active

For a potential VS of type II or III, its orientation must be such that the plane containing the surface does not enter the inner side of F. For example, the VS of Fig. 5(a) is inactive.

Second, when the active parts of two VS relative to elements of the same face intersect somewhere, they both stop at their intersection (see Fig. 5(b)). The part of the VS that falls on the negative side of another VS becomes inactive.

Finally, consider a set of VS of type III (type II) insisting on the same edge e (vertex v). If one of the VS is found to be on the negative side of another VS, then it is inactive. See for instance Fig. 6(a) and (b).
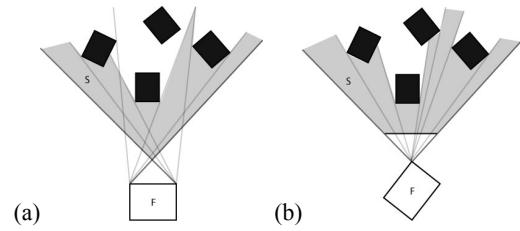

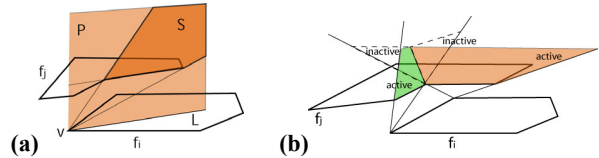**Fig. 4: active part of a VS of type II (a) and III (b)**

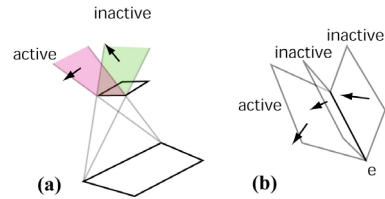**Fig. 5: inactive VS (a); partly inactive VS (b)**

Fig. 6: only the most external surface is active

3. COMPUTING PARTITION Π

The outline of the algoritm for computing Π and labeling its regions, is:

1. find all the active VSs and the associate operator ^
2. intersect all the active VSs and subdivide the space into regions
3. select one region and compute the set of visibile faces V(f1,…,fn) for that zone
4. visit all the regions and compute their set of visible faces with the following rules:
    a. when crossing a boundary between R1 and R2 in the direction of the operator ^, the operand (j) is added to the set of visible faces of R2
    b. when crossing a boundary between R1 and R2 in the opposite direction of the operator ^, the operand (j) is removed from the set of visible faces of R2

4. CONCLUSIONS

In this paper a method for positioning a minimum number of sensors into a 3D polyhedral environment has been presented for some sample cases. The approach has been implemented and results have been presented. Future work will be focused on extending the algorithm in order to consider the general case of face covering, and not only its integer face covering restriction.

5. REFERENCES

[Bot04] A. Bottino, A. Laurentini "Optimal positioning of sensors in 2D". Proc. 9th Iberoamerican Congress on Pattern Recognition, Puebla (Mexico), October 26-29, 2004

[Gig91] Gigus, J. Canny, R. Seidel, "Efficiently computing and representing the aspect graphs of polyhedral objects", IEEE Trans. PAMI, Vol. 13, no. 6, pp. 542-551, June 1991

[Tar96] K. Tarabanis, R.Y. Tsai, A. Kaul, "Computing occlusion-free viewpoint", IEEE Trans. PAMI, Vol. 18, no. 3, pp. 279-292, march 1996