Temporal coherence in progressive radiosity

D. Tost and X. Pueyo

Report LSI–93–5–R

TEMPORAL COHERENCE IN PROGRESSIVE RADIOSITY

D. Tost and X. Pueyo

Software Department. Polytechnical University of Catalonia. ETSEIB-
Avda Diagonal 647, Pta 8, Barcelona 08028 , Spain.

ABSTRACT

In this paper we firstly present a study of the use of temporal coherence in radiosity algorithms. We underline the fact that this may be used even in static environments given that the behavior of a radiosity algorithm, when computing the form-factors of a surface is the same as the behavior of an algorithm computing visibility for a sequence of view points corresponding to a linear movement. Secondly, we study the proposed solutions for dynamic environments. Finally, we propose an algorithm for dynamic environments which target is to reach a wide use of temporal coherence.

INTRODUCTION

Radiosity based lighting models have proved to accurately simulate the interreflections of light between diffuse reflecting surfaces [1]. The balance of energy between N surface patches obtained from the division of a whole, energetically closed environment can be expressed as:

$$B_i = E_i + \rho_i \sum_{j=1}^{N} B_j F_{ij}$$

$$i = 1, ..., N$$

where B_i is the radiosity of surface i, E_i is the energy directly emitted from surface i, ρ_i is the reflectivity of surface i and F_{ij} is the form factor that represents the fraction of energy leaving surface j and landing onto surface j.

The expression of the form factor F_{ij} is:

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{cos\phi_i cos\phi_j}{\pi * r_{ij}^2} * V * dA_j * dA_i$$

where, A_i and A_j are the area of the surface patches, V is 1 or 0 depending on whether dA_j is visible to dA_i or not; r_{ij} is the distance between the two patches and ϕ_i and ϕ_j are the angles between the surface patches normals and the line passing by the centers of the two patches. Several methods have been proposed in order to compute them, including:

- Analytical solutions
- Hemispace projection
- Raytracing

A survey of these methods can be found in [2].

The computation of the form-factors is the bottleneck of the algorithms. However, being view independent, they only need to be computed once in order to obtain a sequence of images of an observer moving continuously around a static scene. In dynamic environments, form-factors may change so they must be recomputed at each frame. This is a serious drawback for using extensively the radiosity in computer animation. However, the scenes in which radiosity is applyied are usually closed enviroments, for instance rooms. We believe that animation in such environments will often deal with a reduced number of *actors* in front of a large number of static *furniture*. Thus, from one frame to the next one, the number of changes in the geometry and the attributes of the scene will be small and they will provoque predictable modifications of the form-factors.

In a previous paper [3], we presented a frame-to-frame coherent algorithm for hidden surface removal that allows significant per-frame computations savings. We herein propose to extend this method to the radiosity determination in animated scenes where the objects motion is piecewise linear and is known a priori.

The paper is divided into 3 sections: we first analyse how temporal coherence may be used for radiosities computations (section 1), we next describe the existing solutions (section 2), then we propose an algorithm based on progressive radiosity and intended to make a wide use of temporal coherence (section 3). Finally, we give our conclusions and future work mainlines.

1. TEMPORAL COHERENCE IN RADIOSITY

Temporal coherence capitalizes on the environment continuity existing over time. It may be aplied to visibility computations [4][5] as well as shading [6]. In radiosity it may have several meanings. First, the interest of radiosity itself is based on temporal coherence; i.e. it is based on the fact that diffuse environments don't change their aspect when the viewer changes his position, as stated in the introduction. If we think on temporal coherence as a way to improve the efficiency of radiosity algorithms, so to speed up the form-factor computation, then we may find different types of temporal coherence. In a first level, these may be divided into three classes depending on the features of the environment:

- static scene - view independence (classic case).
- static scene - view dependence
- dynamic scene

Static scene - view independence

Here the concept of time is virtual, i.e. we think on a virtual observer moving from one patch to another. We may include in this class the well known uses of objective coherence based on view independent presorted structures of objects or surfaces like BSP trees [7],[8],[9] and octrees [10].

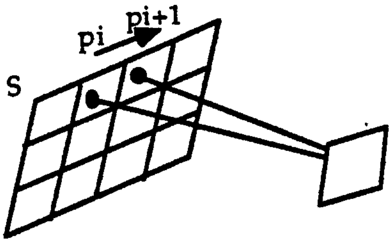


Figure 1. Virtual motions in form-factors computations: linear movement.

On the other hand, we find two kinds of virtual movements, when computing form factors, where predictable coherence may be used. First, when we proceed to compute the form factors of patches of a given surface, we usually "move" from a patch to its neighbour following a linear movement (figure 1). Secondly, the form factor of a patch is computed by means of performing the visibility function for several neighbouring directions. We may think, in this case, on several different circular movements around the center of the patch (figure 2). We may find exemples of the use of these kinds of coherence in [11] and [9].

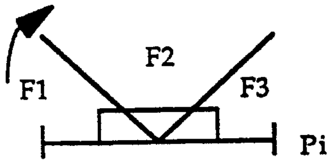


Figure 2. Virtual motions in form-factors computations: circular movement.

Static scene - view dependence

This class of algorithms includes the tecniques proposed to speed up the radiosity process in complex architectural models [12]. In these papers the authors propose strategies to interactively update the radiosity results during walkthroughts. The main idea is to take advantage of the fact that the walls divide the environment into subenvironments with restricted visibility; i.e. that a viewer in a given subenvironment may only see a reduced number of subenvironments (figure 3). Then radiosity may be computed using only these subenvironments and some of their neighbours. Finaly, the update of the radiosities may efficiently be done when the observer moves in a given direction towards a neighbouring subenvironment, using predictable coherence.

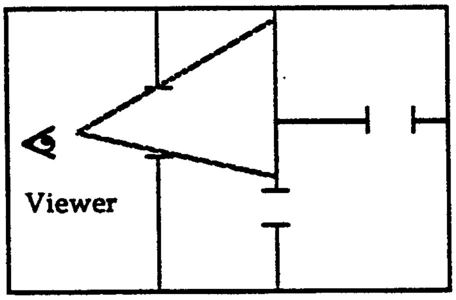


Figure 3. Visibility coherence during a walkthrought.

Dynamic scenes

Now the goal is to efficiently compute radiosities at interactive rates in environments where the visibility relationships change. This happens when new objects appear in the scene or when the scene contains moving objects [13], [14]. Main concepts to be used in these cases are: precomputing static-to-static form-factors and considering the new occluding objects of a given frame, as surfaces shooting negative energy to the occluded patches. In the next section we describe these algorithms before introducing the proposed method which belongs to this class of temporal coherence uses.

2. RADIOSITY IN DYNAMIC SCENES

We describe in this section the two main approaches proposed to deal with an efficient computation of radiosity in dynamic scenes. Baum et. al. [14] used predictive coherence based on the knowledge of the paths of the dynamic objects, whereas Chen's algorithm [13] uses objective coherence and it is based on the progressive refinement approach.

The Back-buffer algorithm [14]

This technique is presented as a method for exploiting temporal coherence from the fixed "view point" of a static patch in a dynamic environment. In this case, the intervisibility only changes over time for a subset of patches. This subset is defined by the paths of the moving objects. More accurately, by their swept volumes. Three types of geometrical relationships are found in these environments:

- static patch to static patch.
- static patch to dynamic patch.
- dynamic patch to dynamic patch.

The temporal coherence is applied to the former relationship given its invariance over time.

A temporally occluded surface (patch) is defined to be a static surface (patch) that is partially or fully occluded from the "observer" (the reference patch) by a dynamic object at some point in time. These patches are recorded in a buffer (the back item buffer) so that a temporally occluded patch is known at each pixel (the hemicube technique is assumed). The back-buffer algorithm consist of two steps:

- Preprocessing: It computes all the static-to-static form factors and initializes the back-buffer. It is executed only once.
- Update: It computes visibility for the temporally occluded patches as well as for the dynamic patches. The former is obtained by comparing the current position of the dynamic object with the back buffer.

Incremental radiosity [13]

The progressive radiosity approach, differs from the classical one in that, at each cycle, a patch is chosen *(shooting patch)* and its unshot radiosity, B_i^u, is distributed to the other ones.

$$\begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ \cdot \\ B_i \end{bmatrix}^{new} = \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ \cdot \\ B_i \end{bmatrix}^{old} + B_i^u * \begin{bmatrix} \rho_1 * F_{1i} \\ \rho_2 * F_{2i} \\ \cdot \\ \cdot \\ \cdot \\ \rho_n * F_{Ni} \end{bmatrix}$$

The process is performed iteratively until it converges. The shooting patches are chosen among those that contribute to the environment with the most energy, in order to enable the system to quickly converge.

The advantage of this method is that, as in other progressive refinement methods [15], in the first iterations, it allows to obtain, with a small cost, images of a low quality but sufficient in order to detect unwanted visual effects and to stop the process. This is particularly desirable in an animated sequence where many frames are to be produced and thus errors should be early filtered.

Incremental radiosity is an extension of progressive radiosity that allows to introduce changes in the environment at each solution cycle. The allowable changes are:

- surface attributes (emission or reflectance): solved by substracting new and the old radiosity:

$$\Delta B_i = E_i' - E_i + \frac{(\rho_i' - \rho_i) * (B_i - E_i)}{\rho_i}$$

- scene geometry: solved by removing the contribution to the old geometry of the previously shooted patches and reshooting their radiosity to the new geometry:

$$B_j' = B_j + \rho_j * (B_i^s \Delta F_{ji} + B_i^u F_{ji}')$$

where $\Delta F_{ji} = F_{ji}' - F_{ji}$ is the incremental form factor and $B_i^s = B_i - B_i^u$ is the shot radiosity of patch i.

3. A FRAME-TO-FRAME COHERENT RADIOSITY ALGORITHM

Frame-to-frame Coherence in visibility computations

We first describe the main features of the frame-to-frame coherent algorithm of hidden surface removal proposed in [3]. The algorithm incrementally computes, for each frame, a depth sorted list of the scene faces which is used as input of the Painter's algorithm. It is based on a double structure: each object is represented by a temporal BSP-tree *(tBSP)* of its faces and the objects themselves are structured in a temporal graph of the depth priority relationships between them. In a pre-process, these structures are created and traversed in order to obtain the first depth sorted list of the scene faces. In successive frames, the list can be easily updated with partial traversals and with no geometrical computations.

We use BSP trees as representation schemes of the objects because they store information on the relative priorities of the objects faces, which, if no deformations are allowed, is independent on the geometry of the objects and therefore, valid throughout the whole animation sequence [7]. tBSP-trees are conventional BSP-trees in which, at each node, the sign of the dot product between the surface normal and the viewing direction is stored along with the time at which this dot product will change. Figure 4 illustrates this structure: f points at the faces list, s is the current sign, and t is the associated instant of change. These two fields are computed in the first traversal of the BSP tree and need only to be consulted at successive traversals. These traversals may even be avoided at some frames if the first instant of time in which a sign change will occur, is stored at the root level.

The temporal graph of the depth priority relationship between objects is computed based on their spacetime extents. It is represented by an adjacency list associated to each object, where, along with each adjacency the interval of time in which the occlusion exists, is stored (figure 4). It is showed in [17] that, a traversal of this graph can be performed, that maximizes the level (or number of ascendents along a tree path) of all graph nodes and thus structures the graph by levels. It is also demonstrated that the depth order of two nodes of same level is irrelevant. Therefore, the depth sorted list of objects is implicit in the levels of the structured graph. Two algorithms have been developed: the*maximum-level* topological sorting and the update of the graph.

A complete analysis of the complexity of the algorithms and implementations results may be found in [17], from which is it concluded that, if the rate of changes is moderate, they allow significant reductions in the computational time.

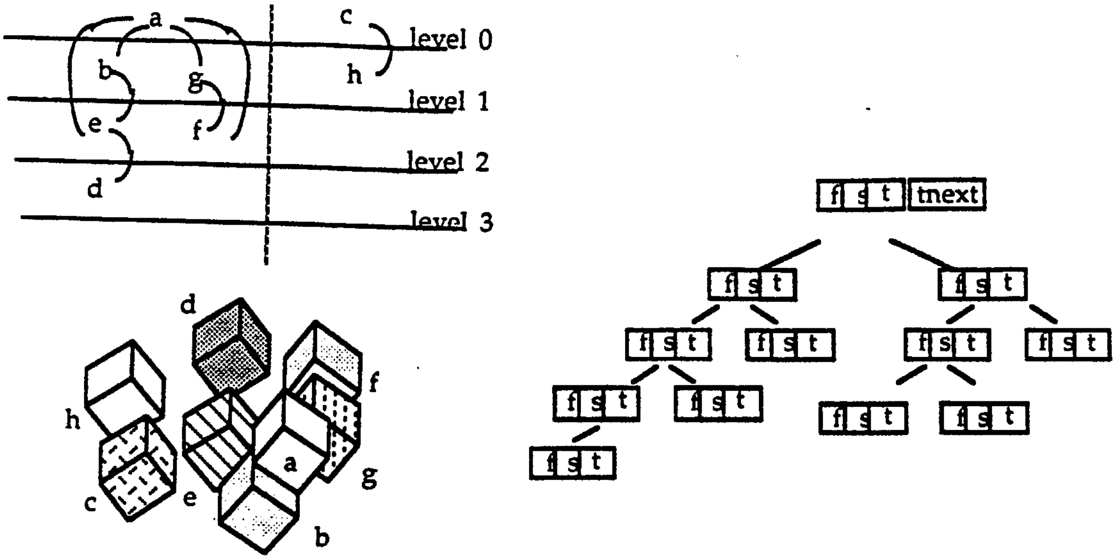


Figure 4. Data Structures of the Algorithm.

Extension of the algorithm to radiosities computation

As mentioned before, the computation of the form-factors involves the determination of intervisibilities between patches. Thus, an extension of the algorithm described above could be developed. We herein focus on the computations of the static-to-static and static-to-dynamic form-factors.

Given a static patch and a patch viewing direction, a conventional form-factor computation is first performed without taking into account the moving objects. The visible static patch in the patch viewing direction is the *background patch*. Its associated delta form-factor is computed and stored. Next, a temporal graph of occlusions between moving objects in the patch viewing direction is calculated. If none of the moving objects may occlude the background patches during the sequence, then the form-factors from the patch are definitely valid. Otherwise, the graph must be stored. At successive frames an update of the graph and a traversal of the t-BSP trees of the corresponding moving objects will give a depth sorted list of moving patches which last element is the background patch. A FTB traversal of the list will determine the visible patch. If it is not the background patch, its corresponding delta form-factor is computed but not stored. The temporal graphs may be computed at the surface level and at the object level. Research is currently being carried on in order to minimize the number of graphs.

Heuristic for choosing the shooting patch

As mentioned before, in the progressive approach, the shooting patches are chosen according to their potential contribution to the scene radiosities. A simple estimation of this contribution is: $B_i^u A_i$, where A_i is the patch area and B_i^u is the unshot radiosity of patch i [16].

For the incremental radiosity, Shenchang [13] suggests a similar test based on the following aproximation of the form-factors:

$$F_{ji} = A_i / \sum_{j=1}^{N} A_j$$

We herein propose a different heuristic that, along with the patches contribution, takes into account two additional criteria:

- the visibility of the patches towards the observer
- the motion of the patches

As mentioned before, the computation of the radiosities is view-independent. The final picture is obtained in a separate process, generally a Z-buffer algorithm, in which only a small subset of the scene patches, the visible ones, is rendered. However, the radiosities of hidden patches must be computed because they contribute to the visible patches radiosities. We believe that, in a progressive approach, in order to choose the shooting patches, their visibility can be taken into account in two different ways by: first, at a same potential contribution, choosing visible patches in front of hidden ones, and second, avoiding to distribute the radiosities to low emittance hidden patches.

A main reason account to prefer visible patches rather than hidden ones: the probability that the energy shooted by a visible patch reaches, with one iteration, the other patches seen by the observer is higher. On the contrary, the energy leaving a hidden patch has a higher proability to be intercept when shooted to visible patches. This assertion is specially true when it is applied to culled patches.

In addition, if a hidden patch has a small potential contribution to the scene radiosities, then, not only it musn't be chosen as a shooting patch but even, the radiosities of the shooting patches may not need to be distributed to it. Hidden patches with a low potential radiosity are called *passive* patches, by opposite to the remainers, *active* ones. Passive patches are taken into account for visibility since the beginning of the process but, regarding to lighting, they are introduced progressively at given cycles. The algorithm is progressive in two different ways: by shooting a new patch at each iteration and by introducing new patches.

The difference between conventional and progressive radiosity has been expressed [18] in terms of *gathering in* light versus *shooting it out*. Our algorithm combines both approaches (figure 5). At each cycle where a patch is shooted, its contribution to the *nac* active patches is computed according to the formula:

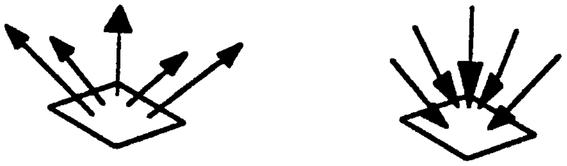
$$for\ j = 1, .., nac,$$

$$B_j\ due\ to\ B_i = \rho_j * B_i^u * F_{ij} * \frac{A_i}{A_j}$$

At each cycle where a passive patch is activated, its radiosity is computed as:

$$B_i = E_i + \sum_{j=1}^{nac} \rho_i B_i * F_{ij} * \frac{A_i}{A_j}$$

$$nac = nac + 1$$



A: Shooting an active patch

B: Activating a new patch

Figure 5. Active and Passive patches.

This approach is meaningless in static environments, where it is often desirable to change the viewpoint once all the radiosities have been computed, but, on the contrary, it is useful in a dynamic environment with a fixed viewpoint if a sequence of frames is to be obtained.

The scene structure allows to classify the patches at the object level and at the surface level. If an occluded object of low emittance is found, its faces, and consequently its surface patches, are classified as passive. Thus, it can be considered that low emittant back-faces are *culled* for lighting purposes althought not for visibilty.

Figure 6 illustrates our heuristic. In the perspective projection, objects A, B, C, H ang G are not visible because they are clipped off the viewing prism or they are occluded by other objects. The lamp C will be taken into account as an active object because, being a light source, it has a high potential radiosity. The same criterion can be applied at the surface level, to the computer's screen. Similarly, the hight reflectivity of table H will determine that it is an active object. On the contrary, A, B and G as well as all their patches are to be considered as passive.

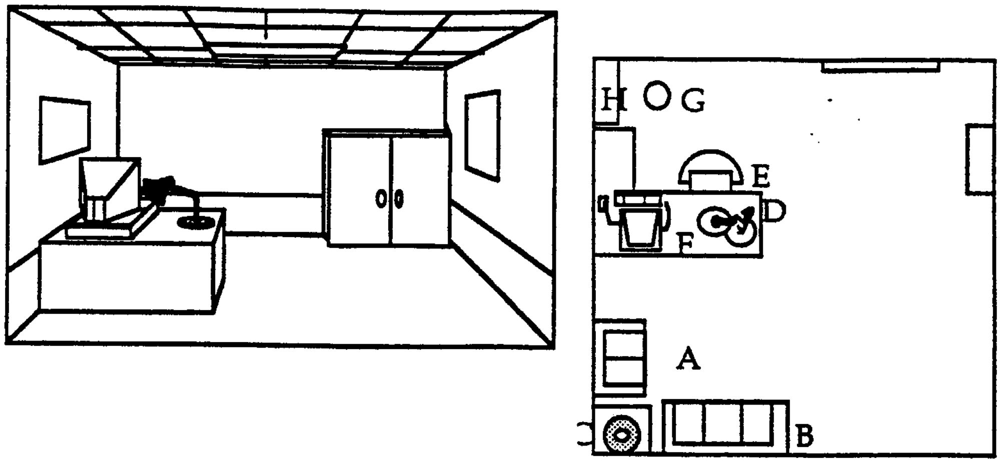


Figure 6. Heuristic for the shooting patch: a sample scene.

The application of this criterion supposes that the visibility status of each patch is known a priori. The three following steps must be realized in order to compute a single frame:

- determination of the patches visibility
- radiosities computations
- rendering and display of the visible patches

The visibility computation (step 1) can be performed identically as at the patch level: by traversing a temporal graph of the objects, constructed from the observer viewpoint, and the t-BSP trees. The resulting depth sorted list of the objects is the input of a FTB ray-casting that outputs a depth sorted list of the surfaces that really project onto the image plane. The third step of the process can here use this list as input for a FTB Z-Buffer. This allows to reduce the number of required radiosities interpolations.

In a dynamic environment, the three steps must be performed in turn at each frame. However, the visibilities determination is based on an update of the temporal graph. In addition, the radiosities can be computed according to the incremental approach developed in [13]. Motion will therefore be an additional criterion for choosing shooting patches, as visible, highly reflectant, moving patches will be selected in the first place, in order to substract their previous contribution and add their new one.

4. THE ALGORITHM

We outline below the main steps of the proposed algorithm

```
Preprocessing
    Compute the tBSP trees
    Compute the temporal graph from the viewers viewpoint
    Compute static form-factors and visibilities intervals
First Frame
    Visibilities computations
    maximum-level topological sorting of the graph
    compute a depth sorted list of surfaces by traversing the objects tBSP-trees
    compute a visiblesurface list by ray-casting the depth sorted list of surfaces
    mark active patches
    Progressive radiosity computation
    while (not converged)do
        if ( non previously shooted active patches exist) then
            choose an active shooting patch distribute its radiosity to other
            active patches
        else
            activate a passive patch
            compute its radiosity
        endif
    endwhile
    Rendering
    Render visible surfaces
Next frames
    incremental visibilities computations
    incremental radiosities computations
    rendering of visible surfaces
```

CONCLUSIONS AND FUTURE WORK

Temporal coherence have been shown as a tool to improve radiosity computation not only in scenes with moving objects but also for walkthroughs and even in static environments with static observers.

We have analysed the solutions proposed up to now for an efficient computation of radiosity in dynamic environments confirming that only a few solutions have appeared.

We have proposed a framework for an extensive us of temporal coherence in dynamic environments. More accurately we propose to used it for computing the form-factors, by storing those are constant overtime and allowing to quickly compute the visibilities changes, and for speeding up the radiosities computation by reducing the number of patches involved in the environment's energy balance.

As future work we envisage to implement the different concepts developed in this paper in order to assess them empirically. We also foresee to apply temporal coherence to static environments and walkthroughs. Finally, we will try to combine temporal coherence with other types of coherences applied to form-factor computations.

REFERENCES

1 Goral, C, Torrance K., Greenberg D., Battaile B., 'Modeling the interaction of light between diffuse surfaces', *ACM Computer Graphics* , Vol. 18(3), pp. 213-222, 1984

2 Pueyo, X., 'Diffuse interreflections: techniques for form-factor computations: a survey',*The Visual Computer* , (7) pp. 200-209, 1991

3 Tost, D., 'An algorithm of hidden surface removal based on frame-to-frame coherence', *Eurographics'91*, Edited by Post,F. and Barth,W., North Holland, pp. 261-273, 1991

4 Hubshmann, H. Zucker, S., 'Frame-to-frame coherence and the hidden surface computation: Constraints for a convex world', *Computer Graphics* , Vol. 15 (3), pp 45-54, 1981

5 Shelley, K., Greenberg, D., 'Path specification and path coherence', *ACM Computer Graphics*, Vol. 16 (3), pp 157-166, 1982.

6 Groller, E., Purgathofer, W., 'Using temporal and spatial coherence for accelerating the calculation of animation sequences', *Eurographics'91*, Edited by Post,F. and Barth,W., North Holland, pp. 203-213, 1991

7 Fuchs, H.,Kedem, Z.,Naylor,B., 'On Visible Surface Generation by a priori tree structure', *ACM Computer Graphics*, Vol. 14(3), July 1980, pp 124-133.

8 Campbell, A., Fussell, D. 'Adaptative Mesh Generation for Global Diffuse Illumination',*ACM Computer Graphics*, Vol 24(4), August 1990

9 Vilaplana, J., Pueyo, X.,'Multi-level use of coherence for complex radiosity environment',*Research Report of LSI Dept.,Polytechnical University of Catalonia*, 1992

10 Wang, Y., Davis, W., 'Octant Priority for Radiosity Image Rendering'*Graphics Interface'90*, pp 17-30, 1990.

11 Tellier, P., Maiselle, E., Bouatouch, K., Languenou, E., 'Using coherence to accelerate radiosity',*Irisa, Publication Interne* , 616, November 1991

12 Teller, S., Sequin, C.'Visibility Preprocessing for Interactive Walkthroughs' *ACM Computer Graphics* Vol 24(4), pp 61-69, July 1991.

13 Chen, S.,'Incremental radiosity: an extension of progressive radiosity to an interactive image synthesis system', *ACM Computer Graphics*, Vol 24(4), August 1990

14 Baum, D., Wallace, J., Cohen M., Greenberg D., 'The Back-Buffer algorithm: an extention of the radiosity method to dynamic environments', The Visual Computer (2) pp.298-306, 1986.

15 Maillot, J., Carraro, L., Peroche, B.,'Progressive Ray Tracing',*Third Eurographics Workshop on Rendering*, Bristol, pp 9-19, 1992.

16 Cohen, M. F., Cheng S.,E., E.C.,Wallace, J.R., Greenberg, D.P., ' A Progressive Refinement Approach to Fast Radiosity Image Generation, '*ACM Computer Graphics*, Vol. 22(4), August 1984, pp 75-84.

17 Tost, D. 'Utilització de la coherència entre imatges successives per a la visualització d'escenes 3D en animació', PhD Thesis, Polytechnical University of Catalonia, March 1991.

18 Wallace, J.R., Elmquist, K.A., Haines, E.A. 'A Ray-Tracing Algorithm for Progressive Radiosity' *ACM Computer Graphics*, Vol. 23(3), July 1989, pp 315-324.

Departament de Llengatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

List of research reports (1993).

LSI-93-1-R "A methodology for semantically enriching interoperable databases", Malú Castellanos.

LSI-93-2-R "extraction of data dependencies", Malú Castellanos and Fèlix Saltor.

LSI-93-3-R " The use of visibility coherence for radiosity computation", X. Pueyo.

LSI-93-4-R "An integral geometry based method for fast form-factor computation", Mateu Sbert.

LSI-93-5-R "Temporal coherence in progressive radiosity", D. Tost and X. Pueyo.

Internal reports can be ordered from:

Nuria Sánchez
Departament de Llenguatges i Sistemes Informàtics (U.P.C.)
Pau Gargallo 5
08028 Barcelona, Spain
`secrelsi@lsi.upc.es`