JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 15, 815-832 (1999)

Short Paper

Clustering-Based Hierarchical Radiosity for Dynamic Environments

WANG-YEH LEE AND JUNG-HONG CHUANG*
Department of Computer Science and Information Engineering
National Chiao Tung University
Hsinchu, Taiwan 300, R.O.C.
E-mail: wylee@csie.nctu.edu.tw
**E-mail: jhchuang@csie.nctu.edu.tw*

This paper extends hierarchical radiosity and clustering techniques to dynamic environments, in which dynamic manipulations, such as repositioning an object and changing surface attributes, are repeatedly applied. The cluster techniques can be effective in reducing the computational complexity of both initial linking creation and link updating of dynamic manipulations. When the object is repositioned in a dynamic environment, the affected links, both energy and cluster links, can be identified rapidly by means of our proposed new strategy. As the total net energy of the environment is decreased, we exploit the so-called *history links* to un-refine the affected patches. An effective progressive refinement strategy is also applied to further avoid the creation of unnecessary links.

Keywords: global illumination, hierarchical radiosity, dynamic environments, virtual environments, clustering, interactivity

1. INTRODUCTION

Greater realism requires global illumination models that take into account the subtle variations of indirect light between surfaces. Global illumination algorithms can be roughly classified into two categories: view-dependent methods, such as ray tracing [1], which compute images for a particular viewpoint, and view-independent methods, such as radiosity methods [2, 3], in which the specifications of the unknowns in radiosity equations are independent of the viewer position.

The radiosity methods have been proven useful for providing global illumination effects of static scenes in walk-through applications. Many interactive applications in visual simulation and virtual reality, however, address dynamic interactions between the objects in a scene. Hence, scene geometry and lighting can be dynamically changed during the simulation process.

Received January 5, 1998; accepted March 12, 1999.
Communicated by Zen Chen.

Baum et al. [4] proposed the first radiosity method for dynamic environments. This method was primarily directed toward computing an animation sequence, it requires that the path of object movement be known in advance and it faces storage and preprocessing problems.

After the progressive refinement technique [5] for radiosity methods was proposed in 1988, its extension to dynamic environments was introduced by Chen [6] and George et al. [7] in 1990. These two methods used the same concept but different terminology, where after the object geometry or attributes were altered, the shot radiosity was redistributed based on these changes. Unfortunately, these methods were applicable only to scenes of simple to moderate complexity. In 1994, the techniques were further extended by storing shadow and form-factor information to improve performance [8].

It was not until hierarchical radiosity [9] was proposed in 1991 that a new framework for dynamic environments made its debut. In 1994, Forsyth et al. [10] extended hierarchical radiosity to dynamic environments with no occlusion, where only links between the dynamic object and environments were updated using an extrapolation heuristic for each frame. In the same year, Shaw [11] also presented an extension of hierarchical radiosity to dynamic environments. The proposed method exploited the refinement process of the hierarchical radiosity algorithm to maintain an optimally-sized system matrix and employed efficient scene-partitioning techniques to speed up the construction of a new linear system after geometry changes.

This paper concentrates on extending clustering-based hierarchical radiosity to dynamic environments.

2. HIERARCHICAL RADIOSITY

The traditional radiosity methods discretize the environment into n elements and solve a linear system of n equations. The major limitation of these approaches is the $O(n^2)$ complexity of form factor computation.

The hierarchical radiosity algorithm (HR) proposed by Hanrahan et al. [9] was inspired by recent numerical method [12] developed to solve the N-body problem in physics. The basic idea behind it is to construct a hierarchical representation of the form factor matrix by adaptively subdividing each patch into elements according to a user-specified error bound, which guarantees that all form factors will be calculated to the same precision, and that image artifacts will almost be eliminated. Decomposing the form factor matrix into at most $O(n)$ blocks results in only $O(n)$ interactions needed to achieve the given error criterion, where n is the number of resulting elements.

In the multilevel hierarchy of the HR algorithm, energy can be transferred from node to node at different levels. Each surface is associated with a quadtree that keeps track of hierarchical subdivision, and the links representing the energy transports are built on-the-fly between the nodes of the quadtrees. If the energy is transferred to a non-leaf node, the nodes underneath will inherit the energy. On the other hand, the area-weighted average of radiosities on child nodes is assigned to their parent node. Because the hierarchy of interaction is equivalent to the block structured matrix, it was shown that only at most $O(n)$ links (interactions) are constructed to satisfy the user-specified error bound. Likewise, general iterative matrix techniques can be employed to solve the radiosity problem.

3. THE PROPOSED METHOD

Clustering-based hierarchical radiosity is proposed here to provide a simple but effective framework for identifying and updating the affected interactions caused by scene changes. The flow chart for this approach is depicted in Fig. 1.

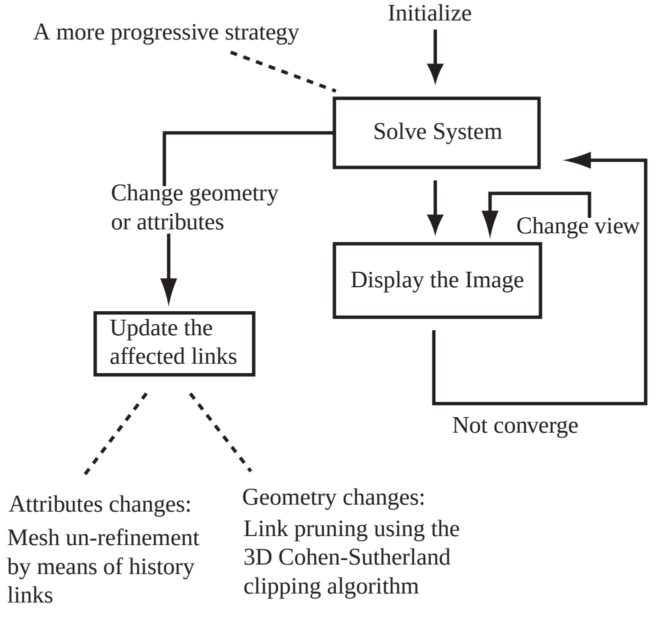


Fig. 1. The flow chart for the proposed method.

In dynamic environments, when there is an increase in the total net energy, for example when a light is turned on, the hierarchical radiosity method will spontaneously and automatically refine the patches as the solution cycle proceeds to reflect the change of energy. But this is not the case with a decrease in the total net energy. We propose the use of *history links* to record the information of each subdivision so that when the total net energy diminishes, the patches can be un-refined so as to retain the same accuracy. When an object is moved to a new position, we propose new schemes to prune the large portion of interactions which are immune from changes of geometry and update the necessary information for the influenced interactions. Thus, as the solution cycle progresses, new shadows will be generated, and old ones will be eliminated. Also, new links will be created, and old links will be broken, thereby entailing patch refinement. Because geometry changes only influences local energy propagation, we do not take patch un-refinements into account for the sake of efficiency. Finally, we propose one modified strategy to further improve interactivity in the entire simulation. This strategy offers improved criteria for avoiding the creation of unnecessary links.

3.1 Clustering-Based HR

The traditional hierarchical radiosity method has a complexity bound of $O(n^2+p)$, where n is the number of input surfaces and p is the number of the resulting elements. In our dynamic framework, we adopt *clustering* similar to that in Smits's work [13] to group surfaces into clusters or objects and to reduce the complexity bound to $O(n\log n+p)$. Because we use clustering, we do not need to check all pairs of surfaces during the initial linking stage. Thus, the complexity $O(n^2)$ is replaced by $O(n\log n)$.

Our approach is different from Smits's [13] with regard to the generation of clusters and the linking strategy. Smits's clustering algorithm uses ray tracing acceleration techniques to automatically generate the hierarchy of clusters, but we artificially define our surface clusters as objects in the data file, which is similar to the data formats embedded in several commercial modelers.

3.1.1 The linking strategy

Besides energy links in the traditional HR algorithm, *cluster links* represent the energy transfer between clusters or objects. In our framework, there are both surfaces and objects that coexist independently in environments. Each object consists of surfaces, and each surface may or may not belong to an object.

The linking strategy begins with checking all pairs of clusters or objects in the environment to see whether the error bound for the cluster link is acceptable. If it is, cluster links are created. Afterwards, the algorithm reverts to the conventional HR algorithm to check all the pairs of surfaces that conform to one of the following conditions:

1. The surfaces belong to different objects between which no cluster link is created.
2. Both surfaces exist by themselves. In other words, they do not belong to any of the objects or clusters.
3. One member of the surface pair belongs to an object or a cluster, and the other one exists by itself.

The HR algorithm then invokes surface subdivision to allow each link between surfaces to interact at an appropriate level in the hierarchy. The idea behind the linking strategy for our clustering-based hierarchy is that coarse cluster links are employed first; if this coarse approach does not produce the acceptable error bound, then more accurate techniques in the HR algorithm are invoked.

3.2 A Framework for Dynamic Environments

Our proposed method proceeds with standard hierarchical radiosity procedures. In the process of refining images, when there is a request for changes of geometry or surface attributes, a dedicated procedure called **AdjustSystem** replaces the traditional radiosity solver procedure. The major tasks performed by **AdjustSystem** are identifying the influenced interactions, updating the necessary information when the geometry is changed, and performing *mesh un-refinement* to raise the accuracy of energy propagation to a proper level when surface attributes are altered.

Before addressing links affected by geometry changes, we will first categorize and analyze thed links involve. To identify the affected links, we will extend the 3D Cohen-Sutherland clipping algorithm to prune un-affected links. Next, we will introduce *history links* which can be used to guide mesh un-refinement when there is a decrease in the total net energy.

3.2.1 Classification of links

As the geometry changes, shadow and light propagation which are in equilibrium in the environment, might be affected. From the point of view of the hierarchical radiosity method, such changes will affect links representing energy transfer; consequently, the information stored in these contaminated links, such as form factor and visibility information, should be recomputed.

In our framework, manipulation of geometry is restricted to rigid body translation and rotation. Also, an *object* is a group or a cluster of surfaces, and it may be either *convex* or *concave*. The environment consists of all surfaces except those of the *object*. In order to analyze and cope with links influenced by geometry changes, these links are categorized according to their intrinsic nature, the changes in them, and the ease of tackling them. Such links can be classified exclusively as environment-to-environment, environment-to-object, and object-to-object links according to their intrinsic nature (see Fig. 2). However, we ignore object-to-object links in our framework since illumination dramatically localizes on the *moving object* itself.

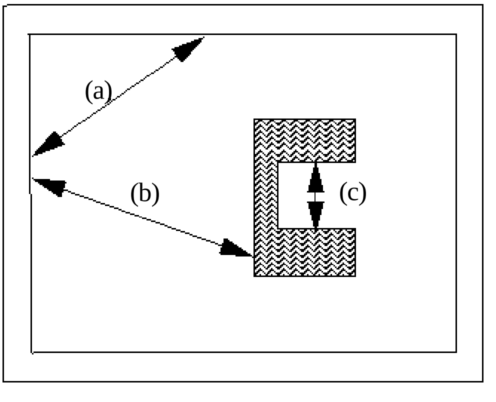


Fig. 2. Three intrinsic types of links: (a) the Env.-to-Env. link, (b) the Env.-to-Obj link, (c) the Obj-to-Obj link.

3.2.2 Identification of links

When there is a change in geometry, links in the environment may be either unchanged, updated, removed, or created. Links are considered easy to identify if they can be directly accessed without a priori knowledge; analogously, links are considered hard to identify if they must be accessed using some search processes. Operation versus tractability for these three types of links is summarized in Table 1.

Table 1. Operation vs. tractability for each kind of link.

Interactions	Env.-to-Obj.	Obj.-to-Obj.	Env.-to-Env.
Update	easy	easy	hard
Create	hard	easy	hard
Remove	easy	easy	hard

Environment-to-Object Links

When an object is repositioned in a dynamic environment, links emanating from the object to the environment must be updated because the form factor from the *moving object* to the environment will change. These environment-to-object links are easily accessed by determining where the links come from and end a priori. Visibility tests can be applied to these links to check whether totally occlusions exist. If they do exist, the link should be eliminated. The two operations, updating and elimination, are considered to be easy to perform.

If visibility between the surface of the *moving object* and the patch in the environment is newly introduced, a link between them should be established. To identify this kind of link, a linear search of the *unlink* patches on the *moving object* and the visibility between these patches and the environment must be computed. Thus, this process is considered difficult to perform.

When a convex *moving object* is considered, environment-to-object links are by no means occluded by the *moving object* itself. We will ignore this case in our treatise. Refer to Fig. 3 for the three different changes in environment-to-object links which occur due to a *moving object*.

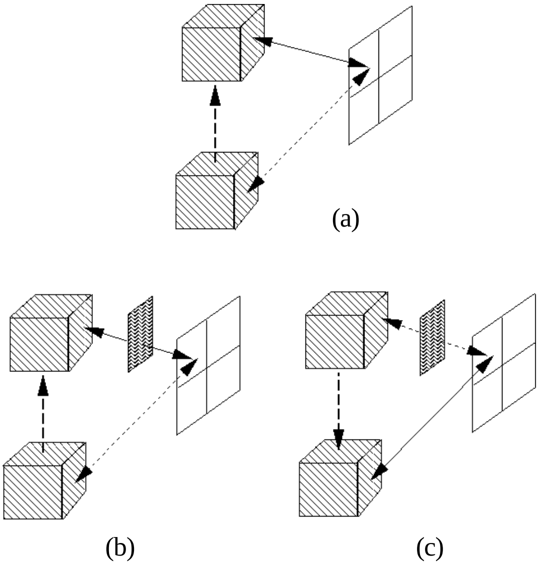
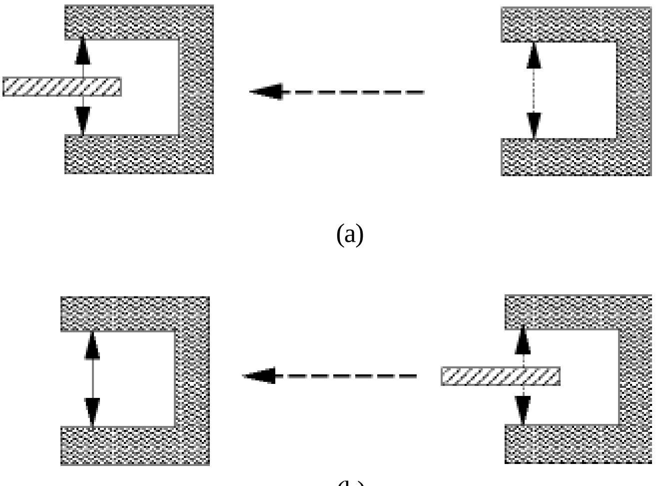


Fig. 3. Changes in env.-to-obj. links: (a) Links should be updated. (b) Old links should be eliminated. (c) New links should be created.

Object-to-Object Links

Object-to-Object links occur very infrequently in dynamic environments. Only *concave objects* may introduce this kind of link (see Fig. 4). When there is a change in geometry, object-to-object links may be either unchanged, updated, removed, or created. It is considered easy to perform these operations since the number of surfaces of the *moving object* is small comparable with the number of surfaces in the whole environment. However, these operations have little visual impact localized on the object itself. In our framework and implementation, all object-to-object links are assumed to be unchanged since feedback from the changes of these links is small, especially in dynamic environments.



(a)

(b)

Fig. 4. Changes in the Obj.-to-Obj. links of a concave object: (a) Old links should be removed. (b) New links should be introduced.

Environment-to-Environment Links

Environment-to-environment links may or may not be influenced by a geometry change. If a geometry change results in occlusion of environment-to-environment links, then this link must be further examined to decide whether it should be removed or updated depending on its visibility information. Similarly, if the change causes two *unlinked* surfaces in the environment to be mutually visible, then a new link should be established. Because the affect on this kind of link is not known a priori, the affected links can not be accessed immediately or directly.

Refer to Fig. 5 for changes in environment-to-environment links caused by a *moving object*. In the following section, we will focus on identification of environment-to-those environment links which are most difficult to deal with.

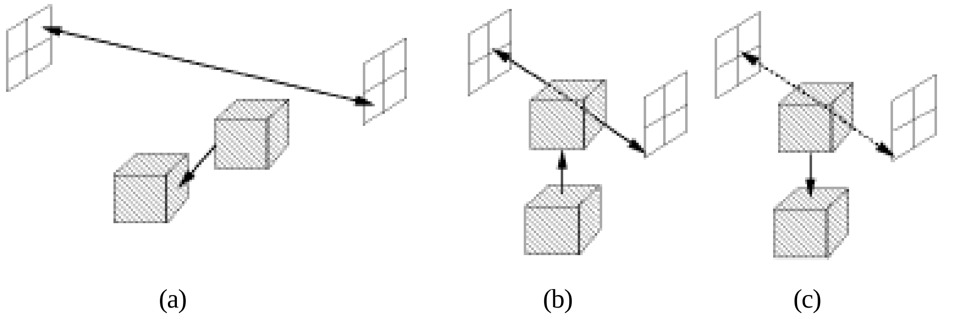


Fig. 5. Changes in Env.-to-Env. links: (a) Link remains unchanged. (b) Old links should be removed. (c) New links should be established.

3.3 Pruning Unnecessary Link Updates

In this section, we will specifically address the identification of environment-to-environment links which are affected when an object is moved. Env.-to-Env. links make up a large portion of the total links in an environment, but only a small percentage of them is affected when a geometry change occurs. A pruning process is needed.

In our approach, all the objects (surface clusters) in an environment are bounded using the axis-aligned bounding volume (AABV), not only to accelerate *ray-tracing* based visibility computation, but also to benefit the process of link identification. We use the AABVs of the *dynamic object* in the original position and the new position as two *clipping volumes* to prune links which can be *trivially rejected* and to identify links that may need to be updated. Link pruning using two bounding volumes is illustrated in Fig. 6.

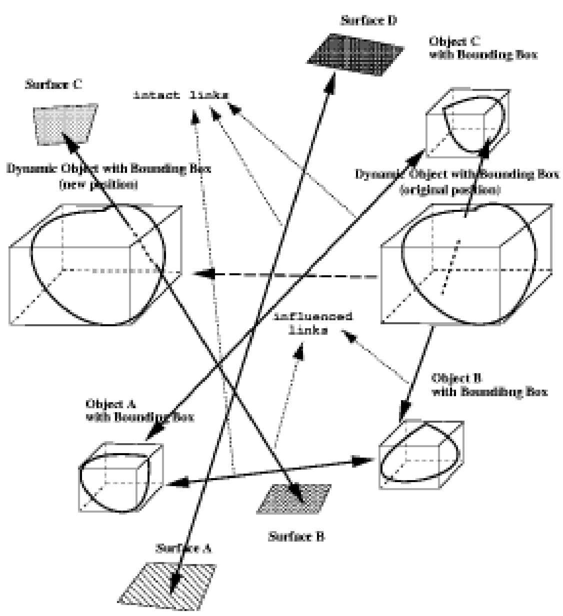


Fig. 6. Two bounding volumes of a dynamic object are used as two clipping volumes to prune links that remain intact and to identify links that may require modification.

The 3D Cohen-Sutherland clipping algorithm [14] is used to identify links affected by object movement. We extend the planes of the clipping volume to divide the space into twenty-seven regions. Each region is assigned a 6-bit code, based on where the region lies with respect to the outside half-spaces of the clipping volume planes. Each bit in the outcode is set to either 1 (true) or 0 (false). Each link, *whether an energy link or a cluster link*, can be represented as a line segment, and outcodes are assigned to the two endpoints. If the logical **AND** of the outcodes of the line segment is nonzero, then the link can be trivially rejected. In the case of an object or a surface, its gravitational center is used to represent the endpoint of the physical interaction.

To prune *intact* links and to identify links that merit further visibility computation, we first apply the trivial reject test to each pair of objects except for the *moving object* in the environment. If the line segment involved cannot be trivially rejected, then we check if a cluster link between these two objects exists. If a cluster link exists, the information associated with the link is updated. Otherwise, the trivial reject test is applied to the line segments between each pair of surfaces that belong to the two objects, respectively. Similarly, we perform further visibility computation on the links of those surface pairs that do not pass the trivial reject test and update the relevant information if necessary. Finally, we test links between all the other surfaces that do not belong to any objects in the environment and also identify the affected links. The most difficult and challenging environment-to-environment links can be sifted out efficiently and effectively using our proposed modified 3-D Cohen-Sutherland clipping algorithm as will be shown in section 4.

3.4 Mesh Un-Refinement

History links are used to guide mesh un-refinement when there is a decrease in the total net energy. Our history links are very similar to the *ghost links* proposed by Shaw [11] except that our *history links* govern both cluster links and traditional *energy links*, and employ more efficient binary search schemes.

History links are *inactive* links which record all necessary information, such as the form factor, the visibility, the identities of source and receive surfaces, etc., but do not represent the current energy transport because they are superseded by their children's links. When mesh un-refinement is triggered, these inactive links will become active. Note that history links are not constructed until the patches are subdivided.

For a quadtree of height h, the total number of nodes is $S_h = 1 + 4 + 4^2 + 4^3 + 4^4 + 4^5 + \cdots + 4^h = \frac{4^{h+1}-1}{3}$, where the total number of leaf nodes is 4^h and the total number of interior nodes is $\frac{(4^h-1)}{3}$, which is less than $\frac{1}{3}$ of the leaf nodes. Clearly, each interaction in the environment is associated with each leaf node, and each *history link* is associated with each interior node. Thus, adding additional *history links* will not increase the number of interactions since $O(n+\frac{1}{3}n) \equiv O(n)$. The order of the interactions remains the same.

There are two steps in the process of mesh un-refinement. First, all the unnecessary links in the environment are removed. Next, all the overrefined patches are recovered. We apply a binary search scheme to the history links of the patch hierarchy to find the desired history link. All the other interactions underneath each activated history link are then removed. Patch clean-up is performed in a bottom-up fashion; nodes are checked recursively from the leaf nodes to the root node to see if interactions exist, and those with no interactions are removed.

3.5 A More Progressive Strategy

In the HR approach, avoiding unnecessary subdivision and links results in efficiency speed-up and significant memory storage savings if determining whether or not subdivision is required costs less than the subdivision itself since both the total number of interactions and the number of prohibitively time-consuming visibility tests are reduced. We will present a modification of the strategy proposed in [16] to provide an improved criterion for testing if a pair of mutually visible surfaces requires further subdivision.

Consider two patches p and q. A bi-directional interaction (link) involves two form factors $F_{p,q}$ and $F_{q,p}$. If we refine patch p by subdividing it into smaller subpatches p_i with area A_i, then the definition of form factor

$$F_{p,q} = \frac{1}{A_p} \int_{A_p} \int_{A_q} Vis(p,q) \frac{\cos\theta_p \cos\theta_q}{\pi r^2} dA_q dA_p,$$

where $Vis(p, q)$ is the visibility function which determines occlusion between dp and dq, r is the distance between the two points, and θ_p and θ_q relate the normal vectors of dp and dq, respectively, to the vector joining the two patches, implies that the following equations are satisfied:

$$F_{p,q} = \frac{1}{A_p} \sum_i A_i F_{p_i,q}$$

and

$$F_{q,p} = \sum_i F_{q,p_i}.$$

In our strategy, we are only allowed to avoid unnecessary interactions between pairs of patches that are mutually visible. For a given threshold value ϵ, we check to see if the difference between the old form factor $F_{q,p}$ and the sum of F_{q,p_i}, and the difference between the old form factor $F_{p,q}$ and the area-weighted average of $F_{p_i,q}$ are both smaller than ϵ. We denote the sum of F_{q,p_i} and the area-weighted averaging of F_{p,q_i} as follows:

$$\hat{F}_{q,p} = \sum_i F_{q,p_i}$$

and

$$\hat{F}_{p,q} = \frac{1}{A_p} \sum_i A_i F_{p_i,q}.$$

Our strategy states that a patch p is refined if either of the following statements is true:

$$\frac{|\hat{F}_{p,q} - F_{p,q}|}{\hat{F}_{p,q}} > \epsilon$$

and

$$\frac{|\hat{F}_{q,p} - F_{q,p}|}{\hat{F}_{q,p}} > \epsilon.$$

Checking if the patch merits further refinement involves computation time for form factor calculation from and to the patch. If few refinements can be avoided, then little will be gained since the cost of actual subdivision is similar to that of checking in this strategy. However, if the reduction in the number of interactions happens *early* enough, then several levels of refinement can be eliminated. Therefore, in addition to significant memory storage savings, the computation time is reduced to a certain degree, and a more progressive paradigm is also achieved, which is consistent with the rationale behind the progressive dynamic system.

4. IMPLEMENTATION AND RESULTS

We used the C programming language and OpenGL, in conjunction with standard X libraries and toolkits, such as Xlib and OSF/Motif, to implement a prototype clustering-based hierarchical radiosity rendering system that supports dynamic manipulations. Our platform was a Silicon Graphics O2 with MIPS R5000 CPU and 128M main memory running IRIX 6.3. With regard to implementing the system, there are various issues worth noting. Some of them are form-factor determination, *BF* refinement, T-vertex and visibility coherence handling (see [15] for more details).

We tested our approach with three scenes having different levels of complexity for dynamic manipulations. In the box room scene, one of the boxes was moved from coordinates (114.4, 129.2, 50) to coordinates (150, 190, 50). In the chair room scene, the only chair in the scene was moved from coordinates (150, 270, 92) to coordinates (120, 150, 92), and was rotated around its local Z-axis about 25 degrees. In the full room scene, a chair was repositioned from coordinates (220, 295.3, 57.5) to coordinates (120, 300, 57.5) and was also rotated around its local Z-axis about 25 degrees. Fig. 7 to Fig. 9 show the rendered images.

When a *dynamic object* was repositioned, we compared the percentage of Env.-to-Env. links that needed changes and the link update time among the three different models. Statistics are presented in Table 2 to quantitatively illustrate the information obtained. We also tested the box room scene for the case of a decrease in the total net energy. The original light source of the box room on the ceiling had a radiosity of (12, 12, 12) for three wavelengths of RGB; then, we dimmed it to (2, 2, 2). The wireframe rendered images which further exhibit the current refinement level of accuracy with respect to the two states are shown in Fig. 10.

The top part of Table 2 lists the number of input surfaces, input objects, resulting elements, resulting links and its constituent parts, Obj.-to-Env. links and Env.-to.Env. links, and the total rendering time with/without the proposed progressive strategy for the three scenes, respectively. The results show that the proposed progressive strategy effectively avoided unnecessary links, thus reducing the total rendering time. The bottom section of Table 2 shows that the proposed dynamic framework rejected from 39.62% to 90.33% of the links in the scenes; hence, only 9.67% to 60.38% of the links required further visibility computation. The bottom line of Table 2 gives link update times, including the time needed for the reject test, visibility computation, and image updating, for each scene. We can conclude that the link updating time was in proportion to both the model complexity and the number of resulting links, and was relatively small compared to the time needed for refinement from scratch.

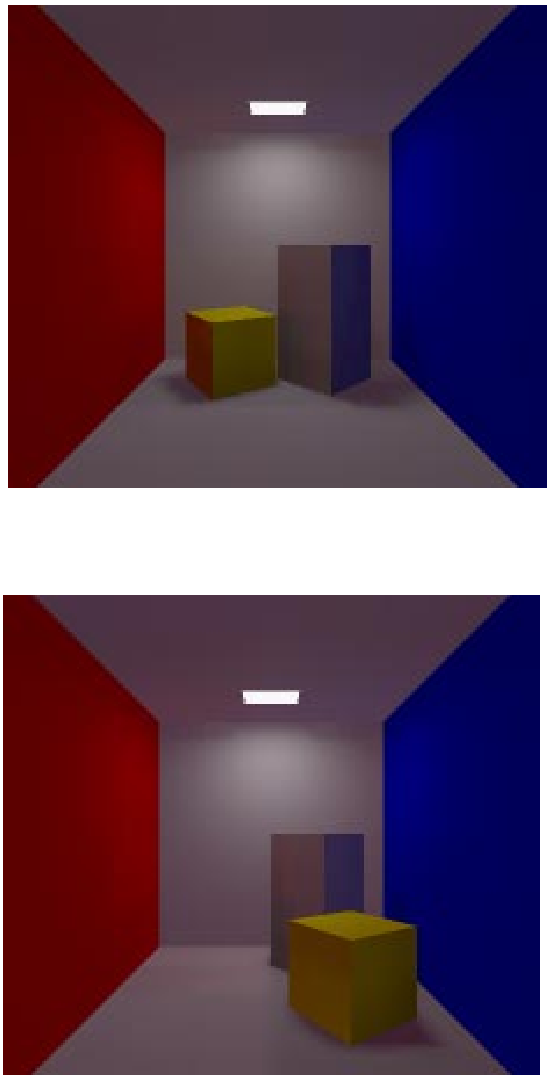


Fig. 7. A box room: before and after dynamic manipulation.

5. COMPARISON WITH OTHER APPROACHES

Previous work in the area of *dynamic radiosity* was conducted within the progressive refinement framework by Chen [6], George [7] and Müller [8], and within the hierarchical radiosity framework by Shaw [11] and Forsyth [10], respectively.

Direct comparison between the proposed approach and those using progressive refinement is somewhat difficult. Although an experimental comparison with the work proposed by Shaw has not been done yet, our approach is expected to exhibit some improvements. We employ a clustering-based hierarchy to reduce the complexity order of the initial linking stage and of link updates due to dynamic interactions; moreover, a more progressive strat-

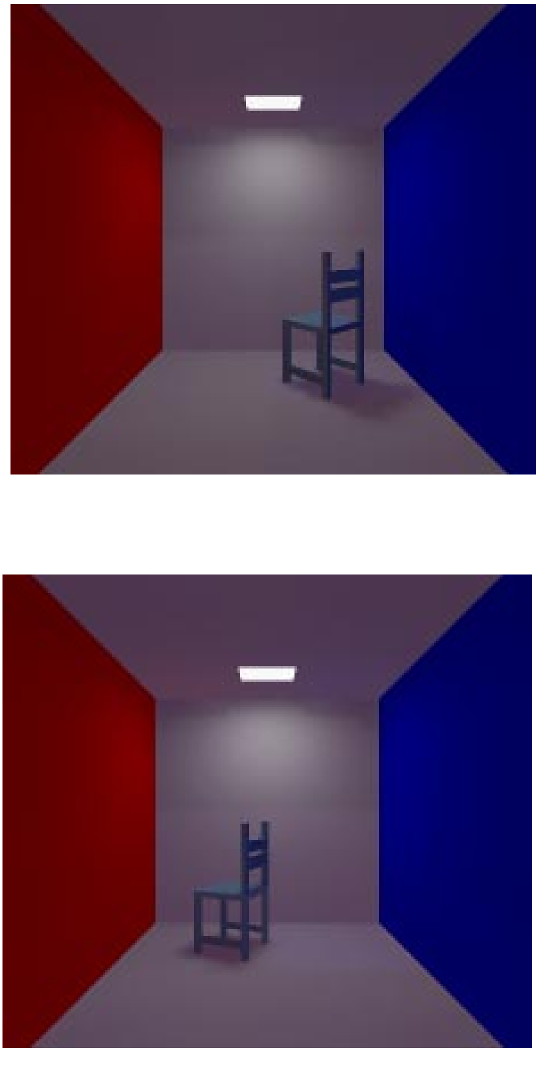


Fig. 8. A chair room: before and after dynamic manipulation.

egy for reducing the number of links is applied. Both contribute to improved efficiency in dynamic manipulation by decreasing the total number of links. Furthermore, our proposed framework provides an effective mechanism for identifying affected links more efficiently than does Shaw's *motion volume scheme*. Also, our use of *history links* that govern both *energy links* and *cluster links* is more effective than Shaw's use of *ghost links* since links differing in accuracy are considered and a binary search scheme for guiding mesh un-refinement can be applied.

Fig. 9. A full room: before and after dynamic manipulation.

6. CONCLUSIONS AND FUTURE WORK

Dynamic interactions between participants and objects is becoming essential in practical virtual environment applications. Traditional hierarchical radiosity methods result in links (interactions) between patches or elements in environments. In this paper, we have proposed a dynamic framework based on clustering-based hierarchical radiosity, in which the affected links, *both energy links and cluster links*, can be rapidly and effectively identified while an object is being repositioned. In addition to the traditional links, cluster links

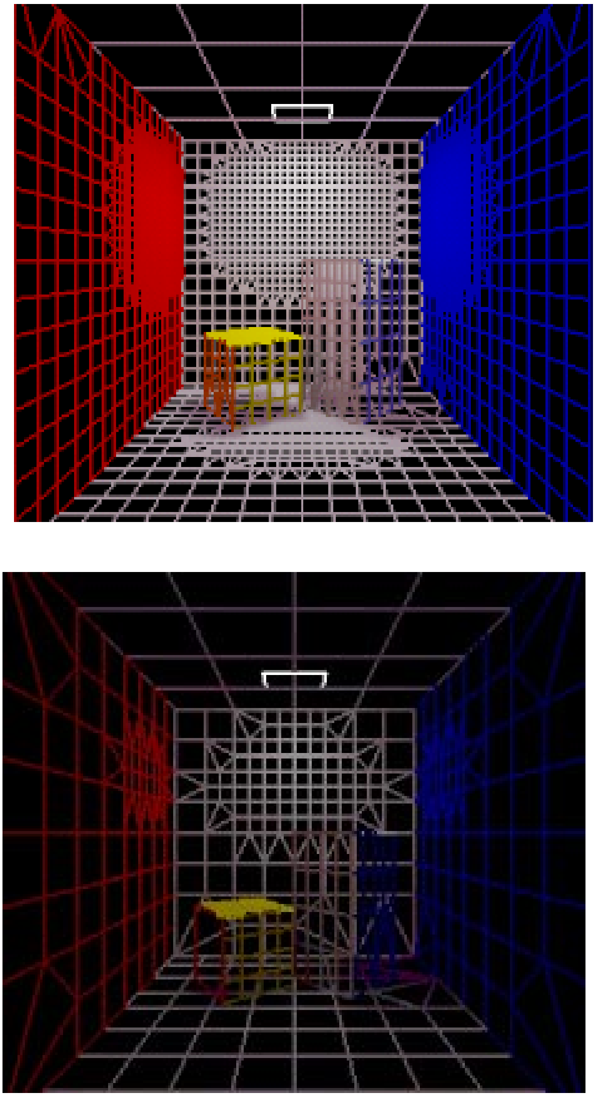


Fig. 10. A box room scene: before and after dimming the light source.

between objects are constructed so as to reduce the computation time of the initial linking stage. *History links* are also used to guide the mesh un-refinement process and to maintain suitable level of refinement accuracy when there is a decrease in the total net energy. Furthermore, a more progressive refinement strategy which avoids the creation of unnecesary links has been presented and applied in our approaches to provide more speed-up.

In our framework, we do not address the problem of the creation of new initial links. For example, when we move a chair from one room to another, the initial links from and to the chair should be constructed. This aspect should be further investigated. For better visual

Table 2. The statistical results for dynamic manipulations.

Results	Box room	Chair room	Full room
# of input surfaces	19	73	203
# of input objects	2	1	4
# of resulting elements	5723	5641	7943
# of resulting links	4785	7086	22726
# of Obj.-to-Env. links	294	708	3636
# of Env.-to-Env. links	4476	4233	16945
Total rendering time without the progressive strategy (in secs)	35.1	41.23	221.33
Total rendering time with the progressive strategy (in secs)	32.6	38.01	182.07
# of Env.-to-Env. links recomputed	987 (22.05%)	2556 (60.38%)	1639 (9.67%)
Link update time (in secs)	2.64	9.99	22.55

impact, a much more effective shadow generation algorithm is also needed for globally illuminated dynamic environments. Furthermore, environments involving multiple *moving objects* must be studied.

The approaches presented in this paper provide faster link updating in HR-based dynamic environments since links are not refined from scratch, yet updating in moderate environments is still too slow to be interactive. New techniques involving temporal coherence, visibility preprocessing [17], and discontinuity meshing [18, 19, 20] should be explored for their possible use in speeding up the updating process.

REFERENCES

1. A. S. Glassner, *An Introduction to Ray Tracing*, Academic Press, 1989.
2. C. M. Coral, K. E. Torrance, D. P. Greenberg and B. Battaile, “Modeling the interaction of light between diffuse surfaces,” *Computer Graphics*, Vol. 18, No. 3, 1984, pp. 212-222.
3. T. Nishita and E. Nakamae, “Continuous tone representation of three-dimensional objects taking account of shadows and interreflection,” *Computer Graphics*, Vol. 19, No. 3, 1985, pp. 23-30.
4. D. R. Baum, J. R. Wallace, M. F. Cohen and D. P. Greenberg, “The back-buffer algorithm: An extension of the radiosity method to dynamic environments,” *The Visual Computer*, Vol. 2, No. 5, 1986, pp. 298-308.
5. M. F. Cohen, S. E. Chen, J. R. Wallace and D. P. Greenberg, “A progressive refinement approach to fast radiosity image generation,” *Computer Graphics*, Vol. 22, No. 4, 1988, pp. 75-84.
6. S. E. Chen, “Incremental radiosity: An extension of progressive radiosity to an interac-

tive image synthesis system," *Computer Graphics*, Vol. 24, No. 4, 1990, pp. 135-144.

7. D. W. George, F. X. Sillion and D. P. Greenberg, "Radiosity redistribution for dynamic environments," *IEEE Computer Graphics and Applications*, Vol. 10, No. 4, 1990, pp.26-34.
8. S. Müller and F. Schöffel, "Fast radiosity repropagation for interactive virtual environments using shadow-form-factor-list," in *Proceedings of Fifth Eurographics Workshop on Rendering*, 1994, pp. 325-342.
9. P. Hanrahan, D. Salzman and L. Aupperle, "A rapid hierarchical radiosity algorithm," *Computer Graphics*, Vol. 25, No. 4, 1991, pp. 197-206.
10. D. Forsyth, C. Yang and K. Teo, "Efficient radiosity in dynamic environments," in *Proceedings of Fifth Eurographics Workshop on Rendering*, 1994, pp. 313-323.
11. E. S. Shaw, "Hierarchical radiosity for dynamic environments," Master's thesis, Program of Computer Graphics, Cornell University, Ithaca N.Y., 1994.
12. A. A. Appel, "An efficient program for many body simulation," *SIAM Journal of Scientific and Statistical Computing*, Vol. 6, No. 1, 1985, pp. 85-103.
13. B. Smits, J. Arvo and D. Greenberg, "A clustering algorithm for radiosity in complex environments," in Andrew Glassner, ed., *SIGGRAPH '94 Conference Proceedings*, Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, ACM Press, 1994, pp. 435-442.
14. J. D. Foley, A.van Dam, S. K. Feiner and J. F. Hughes, *Computer Graphics : Principle and Practice*, Addison-Wesley Publishing, 1990.
15. W. Y. Lee, "Effective clustering-based hierarchical radiosity for dynamic environments," Master's thesis, Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwain, Republic of China, 1997.
16. N. Holzschuch, F. Sillion and G. Drettakis, "An efficient progressive refinement strategy for hierarchical radiosity," in *Proceedings of Fifth Eurographics Workshop on Rendering*, 1994, pp. 357-372.
17. S. Teller and P. Hanrahan, "Global visibility algorithms for illumination computations," in J. T. Kajiya, ed., *SIGGRAPH '93 Conference Proceedings*, Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, ACM Press, 1993, pp. 239-246.
18. D. Lischinski, F. Tampieri and D. P. Greenberg, "A discontinuity meshing algorithm for accurate radiosity," *IEEE Computer Graphics & Applications*, Vol. 12, No. 4, 1992, pp. 25-39.
19. D. Lischinski, F. Tampieri and D. P. Greenberg, "Combining hierarchical radiosity and discontinuity meshing," in J. T. Kajiya, ed., *SIGGRAPH 93 Conference Proceedings*, Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, ACM Press, 1993, pp. 199-208.
20. F. Tampieri, "Discontinuity meshing for radiosity image synthesis," PhD thesis, Program of Computer Graphics, Cornell University, Ithaca N.Y., 1993.

Wang-Yeh Lee (李汪曄) received the BS and MS degrees in computer science from National Chiao Tung University, Taiwan, ROC, in 1996 and 1997, respectively. He is currently a lecturer in the Department of Mechanical Engineering and a PhD candidate in the Department of Computer Science and Information Engineering at National Chiao Tung University. His research interests include realistic image synthesis, computer animation, and data visualization. He is a member of ACM, ACM SIGGRAPH, and the IEEE Computer Society.

Jung-Hong Chuang (莊榮宏) is a professor of Computer Science and Information Engineering at National Chiao Tung University, Taiwan, ROC. His research interests include geometric and solid modeling, computer graphics, and virtual reality. Dr. Chuang received his BS degree in applied mathematics from National Chiao Tung University, Taiwan, in 1978, and the MS and PhD degrees in Computer Science from Purdue University in 1987 and 1990, respectively.