



Interactive Virtual Relighting of Real Scenes

Céline Loscos, George Drettakis, Luc Robert

► **To cite this version:**

Céline Loscos, George Drettakis, Luc Robert. Interactive Virtual Relighting of Real Scenes. RT-0225, INRIA. 1998, pp.31. inria-00069946

HAL Id: inria-00069946

<https://hal.inria.fr/inria-00069946>

Submitted on 19 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interactive Virtual Relighting of Real Scenes

Céline Loscos, George Drettakis, Luc Robert

No 0225

Novembre 1998

————— THÈME 3 —————

 ***rapport
technique***


Interactive Virtual Relighting of Real Scenes

Céline Loscos, George Drettakis*, Luc Robert†

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet iMAGIS

Rapport technique n°0225 — Novembre 1998 — 31 pages

Abstract: Computer augmented reality (CAR) is a rapidly emerging field which enables users to mix real and virtual worlds. Our goal is to provide interactive tools to perform *common illumination*, i.e., light interactions between real and virtual objects, including shadows and relighting (real and virtual light source modification). In particular, we concentrate on virtually modifying real light source intensities and inserting virtual sources into a real scene. Real shadows already exist in the textures representing the real world; Thus, when real light sources are, for example, virtually switched off, the shadows they cause must be removed from the textures. Real shadows need to be reprojected using an alternative rendering approach, which will allow the modification of real and virtual light intensities, as well as interactive manipulation of additional virtual objects. To achieve this, we present a three-step method. We first reconstruct a simplified representation of real scene geometry using semi-automatic vision-based techniques. Using the simplified geometry, and adapting recent hierarchical radiosity algorithms, we construct an approximation of real scene light exchanges. We next perform a preprocessing step, based on the radiosity system, which removes shadows from real world textures. Since our goal is to achieve a convincing relighting effect, rather than an accurate solution, we present a heuristic correction process which results in visually plausible renderings. Finally, we perform an interactive process to compute new illumination with modified real and virtual light intensities. Our results show that we are able to interactively relight real scenes, including additional virtual light sources and objects.

Key-words: Computer Graphics, Computer Augmented Reality, Virtual Reality, Common Illumination, Virtual Relighting, Hierarchical Radiosity, Global Illumination, Interactivity.

(Résumé : *tsvp*)

* Laboratoire iMAGIS/GRAVIR-IMAG, iMAGIS est un projet commun CNRS-INRIA-INPG-UJF, et une équipe de l'Unité Mixte de Recherche GRAVIR-IMAG, Grenoble, France

† Laboratoire Robotvis-Sophia-Antipolis, Nice, France

Modification Virtuelle Interactive des Scenes Réelles

Résumé : La réalité augmentée assistée par ordinateur (CAR) est un domaine en pleine expansion, qui permet à l'utilisateur de mélanger les scènes réelles avec les scènes virtuelles. Notre objectif est d'élaborer des outils pour obtenir un *éclairage commun* entre objets réels et objets virtuels. La notion d'éclairage commun inclut les interactions d'ombres et la modification (virtuelle) de l'éclairage réel. En particulier, nous nous intéressons à la modification virtuelle de l'intensité des lampes réelles, ainsi qu'à l'insertion de lampes virtuelles dans une scène réelle. Malheureusement les ombres dues aux objets réels sont déjà présentes dans les textures. Par conséquent, si nous voulons éteindre virtuellement des lampes réelles, les ombres réelles doivent être enlevées des textures. L'éclairage réel est ensuite représenté en reprojétant les ombres par un algorithme de rendu, permettant de modifier l'éclairage réel et virtuel, ainsi que de déplacer des objets virtuels dans une scène réelle. Notre algorithme est une méthode en trois étapes. Une scène réelle est tout d'abord reconstruite en une version géométrique simplifiée, en utilisant des techniques semi-automatiques de vision. Ensuite, les algorithmes de radiosit  hi rarchique sont adapt s pour repr senter approximativement les  changes lumineux de la sc ne r elle. Une  tape de pr -calcul consiste alors   enlever les ombres des textures repr sant la sc ne r elle, en utilisant le syst me de radiosit . Comme notre but est d'obtenir des effets convaincants plut t qu'une solution exacte, un algorithme bas  sur une correction heuristique permet d'obtenir un effet de rendu plausible. Enfin, un processus interactif permet de modifier l'intensit  de lampes r elles et virtuelles. Nos r sultats montrent que nous sommes capables de modifier virtuellement et interactivement l' clairage de sc nes r elles, incluant d'additionnels sources de lumi re et objets virtuels.

Mots-cl  : Synth se d'Images, R alit  Augment e Assist e par Ordinateur, R alit  Virtuelle,  clairage Commun, R  clairage Virtuel, Radiosit  Hi rarchique,  clairage Global, Interactivit .

1 Introduction

Computer augmented reality (CAR) is a rapidly emerging field which enables users to mix real and virtual worlds. Many applications of augmented reality already exist, for example in entertainment, film and television or help-systems for repair and manufacturing. Using real scenes or images of real scenes has many advantages over traditional, manually modeled virtual environments: the great effort required to model detailed objects is avoided, the resulting complexity in geometric primitives is greatly reduced, and the user is confronted with familiar, real-world references, allowing immediate and intuitive immersion.

The focus of previous research in the domain has mainly been on registration and calibration [Azu97]. Our interest however focuses on *common illumination* between real and virtual objects, i.e. the interaction of light (shadows, reflections etc.) between real lights and objects and virtual lights and objects. Previous work in common illumination [SHC⁺96, NHIN86, YM98, JNP⁺95, Deb98, FGR93, DRB97] has provided solutions which handle certain cases of light interactions between real and virtual objects, and especially the case of virtual objects casting shadows onto real objects. Most of them focus on the production of high-quality, off-line generation of films, or are limited in the effects they produce.

In this paper, we present a system which allows *interactive* modification of real or virtual lights, based on a simple model of the real scene and recent developments in interactive hierarchical radiosity. Modification of real lights is a difficult problem because real shadows are already present in textures representing the real world, which are mapped onto the reconstructed models of real objects. Previous solutions [NHIN86, YM98] allow the virtual modification of the sun position in outdoors real environments, but are based on inherently non-interactive algorithms. We build on the only previous interactive common illumination approach [DRB97], which was restricted in the effects it could treat to virtual objects casting shadows on reconstructed real surfaces.

Our solution has three main steps. We first reconstruct a 3D representation of a real scene, using advanced vision-based techniques. A hierarchical radiosity system is then initialised in a preprocess step, to represent real world illumination. The preprocess then automatically creates new textures with the real shadows *removed* from the textures, and finally reprojects real shadows using the modified radiosity estimation. The third step is an algorithm allowing interactive modification of real and virtual light source intensity.

For a designer, this system provides realistic tools to experiment with the illumination of an enhanced real environment. All that is required is a few photographs of the real scene, the reconstruction of a small number of objects, and the system preprocess as will be described in this paper; the designer can then interactively manipulate real light intensities, or insert and manipulate virtual lights and objects. In Figure 1, an example of a modeled real scene is shown in (a). In (b), its real illumination was modified, replacing two real light source intensities by coloured ones, and switching off two lights. Moreover, a virtual light source was inserted into the the real scene, modifying real object shadows. In (c), a virtual object was also inserted into the real scene, casting shadows onto real objects. This virtual object can be moved interactively at 3 frames per second.

In the following sections, we first present previous work on the several domains related to this work: augmented reality, 3D reconstruction of real scenes and hierarchical radiosity. Previous common illumination approaches are then discussed in more detail. We proceed to explain how we build a 3D geometric model representing the real scene, and present an overview of the algorithm for interactive re-lighting. The real shadow removal algorithm is then presented in detail, followed by implementation issues. Finally we describe results of relighting and interactive modification of real light intensities and the insertion of virtual lights.

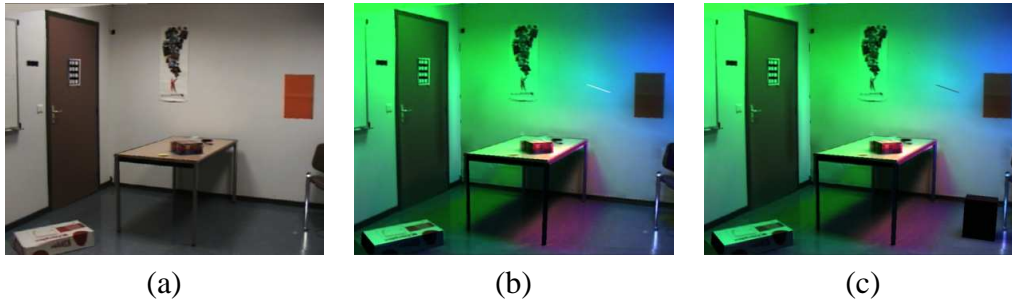


Figure 1: (a) Original real scene. (b) Virtual modification of the illumination of the real scene. (c) Real scene enhanced by a virtual object that moves at 3 frames per second.

2 Previous Work

Our work draws on multiple fields; in particular augmented reality, vision based reconstruction and global illumination. In the following, we will first give a rapid overview of augmented reality which concentrates, in general, on registration and calibration aspects. We next briefly discuss the 3D reconstruction method we use to build a simplified model of the real scene. We will use hierarchical radiosity to create a representation of real-world illumination, and also to permit interactive updates when moving virtual objects or modifying illumination; we thus introduce the basic concepts of this approach which are central to the understanding of our algorithm. We finally detail previous work on global common illumination, insisting in particular on the most closely related approaches which use radiosity methods.

2.1 Introduction to augmented reality

There are two main approaches to augmented reality: virtual and real environments can be combined by superimposing virtual objects on the real world viewed from semi-transparent glasses; alternatively, virtual and real environments can be merged with video images, and the result reprojected onto a screen. These two approaches are presented in detail in the survey of Azuma [Azu97] which also provides extensive references to related literature.

The first approach involves the calibration, registration and display of virtual objects in real time to avoid delays between projected images and the perceived real world. A limitation of this approach is the semi-transparency of the glasses which darkens the view of the real scene.

The second approach allows more interaction between real and virtual objects, because a geometric representation of the real scene is created from the images. We can therefore handle occlusion between real and virtual objects, as well as visual effects such as *common illumination*, which is the interaction of light between virtual and synthetic objects. Nevertheless, achieving real time or interactive display of these effects remains a challenging problem.

2.2 Reconstruction of Real Models

The simulation of common illumination effects requires a geometric representation of the real world. Much research on the subject exists in the field of Computer Vision; we have chosen to use an advanced vision-based technique, which allows semi-automatic reconstruction based on multiple views.

The approach we use is presented in [FLR⁺97]. In order to build a representation of a real scene, several vision techniques are combined: automatic calibration of the camera, mosaicing, computation of the epipolar geometry which results in a polygonal reconstruction of the scene, and the projection

of textures. The first step is the calibration of the camera which consists in retrieving the intrinsic parameters from a non-planar calibration pattern image using an automatic algorithm [Rob95]. The user provides approximate positions of 6 reference points. From this, the system retrieves intrinsic and extrinsic parameters of the camera. Then, four sets of three photographs each are taken, and a mosaic is built automatically for each set as presented in [ZFD97]. From the four mosaics, a 3D model is defined using the *TotalCalib* system [Tot] developed at the ROBOTVIS group. This system, shown in Figure 2, combines several techniques. Point correspondences are provided by a user, who clicks on one image to create a reference point. The matched points on the 3 other mosaics are given automatically by the system. From about 30 point correspondences, fundamental matrices are computed using a non-linear method [ZDFL95]. Polygonal regions are next manually selected by a user from point correspondences, and the system provides 3D coordinates of these polygons from the projection equations. Finally, textures are projected to allow correct perspective effects for a fixed viewpoint [FLR⁺97]. For each reconstructed polygon, a texture image is computed by de-warping the original image (from a given viewpoint), and mapping it to the plane of the polygon.



Figure 2: The TotalCalib system to build 3D models of real scenes, using automatic calibration, and epipolar geometry.

The main advantage of such a system is that user intervention is restricted to the choice of reference matches and polygon vertex selection. This system is however not without limitations: the resulting model of the real scene is approximate and may contain artifacts, since there is no guarantee that geometric properties such as parallel edges or orthogonal angles will be preserved. This drawback can be removed by taking into account additional user input, as presented in the work of Debevec et al. [DTM96] or Poulin et al. [POF98].

In the work by Debevec et al. [DTM96], reconstruction is based on a hierarchy of blocks. The main idea is to build polyhedra which include geometric constraints, such as parallelism, orthogonality, and size aspects. Polyhedra provide good approximations of many objects of the real world, especially for outdoor architectural scenes. This also allows the reconstruction of vertices which are invisible in the original images, but correspond to hidden vertices of the polyhedra. Another approach is described in [POF98] in which the primitives are points, lines and polygons, and constraints such as parallelism, orthogonality, or co-planarity are determined by the user.

If we use the projection of textures presented in [FLR⁺97] we are limited to a single point of view, i.e. one of the original views of the camera. View-dependent texture mapping [DTM96], combines

several original images from nearby viewpoints, by weighting them appropriately. A novel algorithm was recently presented to achieve view dependent texture mapping in real time [DBY98]. The first step of this algorithm is a clipping algorithm which splits the model into completely visible polygons for each view. Holes are then filled with Gouraud shading, and a view map is built for each polygon. For each new viewpoint, a new texture is created by modulating weighted textures from the three closest views. In [POF98], a method is proposed for view-independent textures, that weights and modulates textures from different views. There is thus a single texture per polygon, for any view point. A disadvantage of this approach is that parallax information is lost for all geometric detail which is not explicitly reconstructed, and specular effects are hidden. As a result, when the point of view changes, the resulting model looks less realistic.

2.3 Hierarchical radiosity

To achieve interactive relighting, we need an efficient description of light exchanges in the scene, including shadow information. We have chosen to use the hierarchical radiosity approach with clustering [HSA91, Sil95] with the extensions to dynamic environments [DS97]. We next introduce certain basic concepts of radiosity methods.

The radiosity method is based on energy exchanges, and has been used in computer graphics to simulate light interactions in synthetic environments [SP94], including indirect illumination. Since the radiosity method is a finite-element approach, a mesh representation of the scene is required, which is usually constructed with quadtrees.

The basic method consists in iteratively solving a system of equations using finite-element methods [GTGB84]. Form factors are first computed between elements or *patches*, and with the reflectance given for each element, a matrix is constructed. Using matrix solution techniques, radiosity values are updated at each new iteration. This “full-matrix” approach is costly in memory (n^2 where n is the number of patches) and in computation time. A first improvement was the progressive radiosity method [CCWG88] in which radiosity is first shot from the most “significant” patches, that is primarily light sources and patches that have the greatest radiosity values. The two methods converge to the same result, but we can have a more visually convincing approximation of the final solution before the convergence with progressive radiosity.

Hierarchical radiosity [HSA91] uses a multi-resolution representation of light, by creating a hierarchy of patches on each surface. Light exchanges are established at the appropriate levels at the patch hierarchy via a *link* data structure, resulting in an efficient solution. A generalisation of this approach can be achieved using clusters [SAG94, SDS95], which represent groups of objects. The entire scene is contained in a single, “root” cluster. Clusters and patches can be linked at the appropriate level, depending on the refinement criterion, which decides whether the link represents the light transfer at a suitable, user defined, level of accuracy. If the light exchange is not sufficiently well-represented, the link is refined and the patches or clusters are then subdivided.

When links are established, the incoming irradiance is *gathered* at each patch, followed by a *push-pull* step performed to maintain a coherent multi-resolution representation of radiant exchanges [HSA91]. The cluster-based hierarchical radiosity starts with the root cluster linked to itself. The algorithm described by Sillion [Sil95] performs a refinement step, establishing links at appropriate levels followed by the gather and push-pull steps. Irradiance is pushed down to the leaves of the patch hierarchy, and radiosity is pulled up by averaging [Sil95]. This is repeated until convergence.

Links store visibility information and form factors. The visibility information can be of three types: *VISIBLE*, *INVISIBLE* or *PARTIAL*. When computing radiosity exchanges between two patches, the incoming irradiance is multiplied by the form factor and an *attenuation factor*. If the patches are

entirely mutually visible the attenuation factor is one, which has no effect. However, if the two patches are completely invisible from each other, the visibility factor is set to zero, and therefore the light exchange between the two patches is null. When the visibility type is *PARTIAL*, the attenuation factor represents the degree of occlusion between two patches. The range of the attenuation factor is between one (when entirely *VISIBLE*), and zero (when *INVISIBLE*). The attenuation factor of a link is typically estimated by shooting rays between the two patches, and counting the percentage of rays blocked by occluders.

The hierarchical representation with links can be adapted to allow fast radiosity modification [DS97], by augmenting the links with a shaft data structure [HW91]. In addition, previously subdivided links, called *passive links* are maintained. The passive links contain all the necessary information allowing them to be reactivated at no cost, if it is required by a geometry change. See Figure 3 for an example.

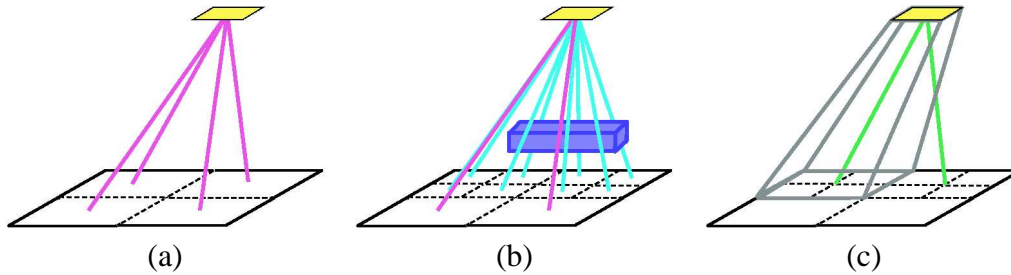


Figure 3: (a) Original subdivision and links in purple. (b) Adding a dynamic object, and updating the hierarchy of elements and links. Eight links shown in blue were created. (c) The passive links with their shafts are maintained in the hierarchy, allowing fast identification of the dynamic object movement. In this case, two passive links shown in green were maintained. The corresponding shaft is outlined in grey.

2.4 Common illumination in augmented reality

The retrieval and simulation of common illumination between virtual and real objects has been treated by several researchers in previous work [SHC⁺96, NHIN86, YM98, JNP⁺95, Deb98, FGR93, DRB97]. All use some form of a 3D representation of the real scene.

State et al. [SHC⁺96] use a composition of vision-based and magnetic tracking methods for accurate registration of the real environment. Virtual objects are inserted into a real scene and common illumination is performed, with a moving (real) point light source. Shadow maps are used allowing updates in real time, but only for direct illumination and sharp shadows from point sources.

Nakamae et al. [NHIN86] developed a solution for merging virtual objects into background photographs, and estimated the sun location to simulate common illumination effects in outdoors environments. More recently Yu [YM98] proposed a solution to virtually modify the illumination with different virtual positions of the sun in outdoors scenes. A pseudo-BRDF is first estimated, which is a function of the incident radiance on the reflected differential radiance. There are two measured incident radiances, one for the sun and one for the sky and the surrounding environment. Radiance is recovered by adapting the high-dynamic range method presented by Debevec et al. [DM97]. Diffuse and specular reflectances are retrieved using multiple images from multiple viewpoints. From various virtual positions of the sun and modified sky and environment illumination, the new outdoors illumination is performed pixel by pixel for each reconstructed triangle. However, for certain applications, an approximation of only the diffuse reflectance is sufficient.

For indoors environments, Jancène et al. [JNP⁺95] used vision-based techniques to retrieve the geometry of the real scene from a video sequence. Common illumination between virtual and real objects is simulated. This allows the creation of video sequences, with animated virtual objects such as a cloth, and the modification of the reflective properties of real objects. The final rendering is performed using a ray-tracing system, and images are merged using a masking algorithm.

Debevec [Deb98] also simulates common illumination effects using RADIANCE [War94], a ray tracing based global illumination system. In this work, the real environment is decomposed into the distant scene and the local scene. The distant scene is used to evaluate the global radiance, and the source emittance [DM97]. An approximate geometric model of the local scene is built using the methods previously developed by the same author [DTM96]. Since radiance is accurately retrieved from images, rendering with mixed images is done by using the difference of the wanted effects and the original image value. This method can be adapted for indoors or outdoors environments.

The common illumination methods presented above are geared towards high-quality image generation, requiring in the order of minutes per frame. Our approach is complementary: we want to provide *interactive* common illumination effects, which will allow a designer to modify and experiment with different lighting conditions. Digital prototyping or “mock-ups” require this type of interactive capability; for a “final” high-quality animation, one of the previous methods can always be used.

Radiosity-based systems for common illumination

The most closely related previous work is that of Fournier et al. [FGR93] and its interactive extension [DRB97]. The system presented permits the retrieval of radiosity parameters from the textures extracted from the real scene images. In our approach, we use Fournier et al.’s basic derivations for the extraction of the quantities required to initialise the radiosity equations. We thus describe this work in more detail.

First the real scene is modeled manually, using a simplified representation. Given this model which is subdivided into patches, the reflectance can be extracted from the image textures. The reflectance of each patch i is chosen to be:

$$\rho_i = \frac{\hat{B}_i}{\hat{B}_A} \times \hat{\rho} \quad (1)$$

where \hat{B}_i is the average intensity of the pixels in an image for patch i , \hat{B}_A the average intensity of the real image, and $\hat{\rho}$ is the average reflectance of the scene (given by the user). This estimation of the reflectance depends on the colour of the texture (i.e. the image of the real scene), which will be darker for patches in shadow. The emittance E_i of each source is estimated from the following equation:

$$\sum_i E_i A_i = (1 - \hat{\rho}) \hat{B}_A \sum_i A_i \quad (2)$$

with A_i being the area of patch i . This approximation is based on the estimated ambient term in the progressive radiosity algorithm [CCWG88]. To simplify, and as it is approximately the case for our scenes, we consider that all the sources have the same intensity. However a system of equations could be solved for non-homogeneous intensities.

Once the reflectance(s) ρ_i and the emittance(s) E_i have been estimated, a progressive radiosity solution is applied. The result of this simulation is the radiosity B_i of each patch. We use a *display correction factor* D_i of a patch i , which is first initialised to the radiosity B_i . When the scene is modified, the current radiosity B_i is updated to reflect the change. For example, if a virtual object is inserted, the patches on which a shadow is cast will have $B_i < D_i$. Modifications to the scene

(notably the addition of virtual lights and objects), are performed by modulating the texture T_i of a pixel as follows:

$$\frac{B_i}{D_i} \times T_i \quad (3)$$

It is important to note here that the accuracy of the radiosity estimation B_i is irrelevant. Since a ratio is being used (which is 1 if there is no change), the only requirement is that the modifications to B_i be *consistent*. Note that ray-casting is used for rendering in [FGR93].

This approach was adapted in [DRB97], in the context of a hierarchical radiosity system, which allows common illumination between a dynamic virtual object and a real scene. The interactive update of the illumination when the virtual object moves uses the dynamic hierarchical radiosity solution described in [DS97]. An example of the results obtained by the method of [DRB97] is shown in Figure 4, where a red dynamic virtual object was inserted into a real scene, on the top of the desk. The shadows are virtually projected onto the table, using the display ratio described above (Eq. (3)).

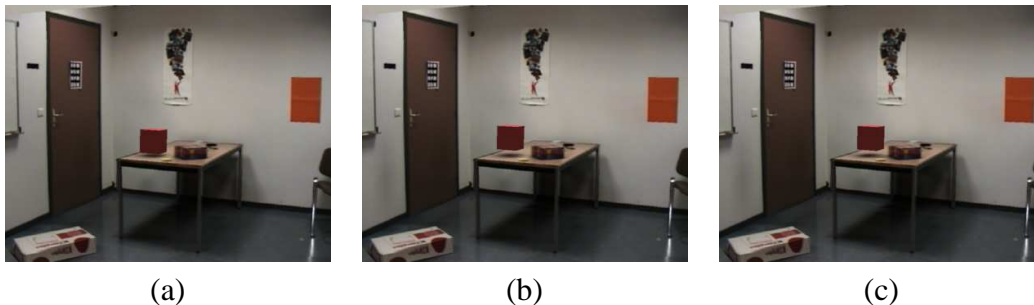


Figure 4: (a) A virtual object, floating above the table, was inserted into a real scene using [DRB97], in 5.65 seconds. Shadows are projected onto the table using the display ratio of Eq. (3). (b) and (c) The dynamic object moves above the table. Links and radiosity are updated interactively in 3 frames per seconds.

3 The Common Illumination System

The interactive common illumination system of Drettakis et al. [DRB97] was limited to the insertion of a virtual object into the scene, and the casting of shadows by the virtual object onto a real (reconstructed) surface; modification of the real or virtual lights was not possible (see Section 4.1).

The goal of our approach is to allow interactive modification of real source intensities, the insertion (and modification) of virtual sources, and the insertion and interactive manipulation of other virtual objects. All interactive updates will be performed with consistent update of shadows of real and virtual objects. Our system consists of 3 steps: a 3D reconstruction of the real scene, a preprocessing initialisation stage, and an interactive modification stage, during which the user can modify and enhance the real scene. The entire algorithm is summarised in Figure 5.

3.1 Representation of the real scene

The real scene is represented in our system with an approximation of its geometry and with projected textures. The model of the scene is built semi-automatically, using advanced vision techniques [FLR⁺97, DRB97] as described in Section 2.2. This process allows the reconstruction of the basic 3D model visible in the captured images (for example the mosaics shown in Figure 6).

3D reconstruction

Build a simplified 3D model of the real scene

Preprocess

Hierarchical radiosity system set up

Refinement for shadow boundaries

Removal of real shadows

System reinitialisation and shadow reprojection

Additional preprocess for the insertion of virtual objects

Interactive modification

Modification of real and virtual lights.

Update when a virtual object moves.

Figure 5: Complete algorithm.

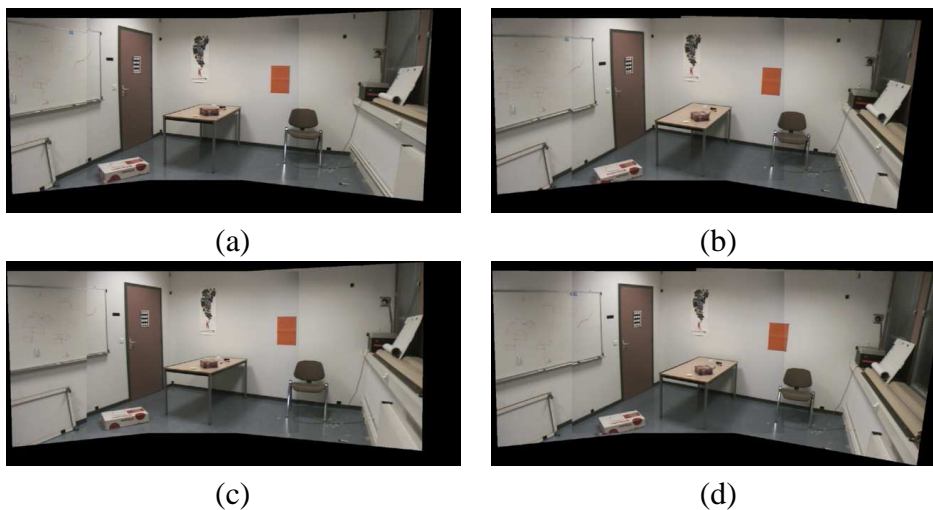


Figure 6: The four mosaics from four different points of view.

The rest of the scene, which is not visible in the images, is built with points measured manually. Approximate textures are used to map the polygons of this part. The positions of the light sources are also measured manually and inserted into the 3D model.

The model is then an approximation of the captured room, with a more precise model for the visible part of the scene, and a coarse model for the rest of the scene. An example of the resulting reconstruction is shown in Figure 7.

A limitation of this approach is that the projection of the textures is done only for a single point of view. We are therefore restricted to viewing the scene from a static viewpoint. In Figure 7(a), the model is viewed from our system, and in (b) the complete model is shown, including the non-visible part.

3.2 Preprocess to enable interactive re-lighting

The main contribution of our approach is the preprocessing algorithm which results in the removal of real shadows (i.e. due to real sources and real objects) which pre-exist in the real-world texture (i.e. the images of the real scene).

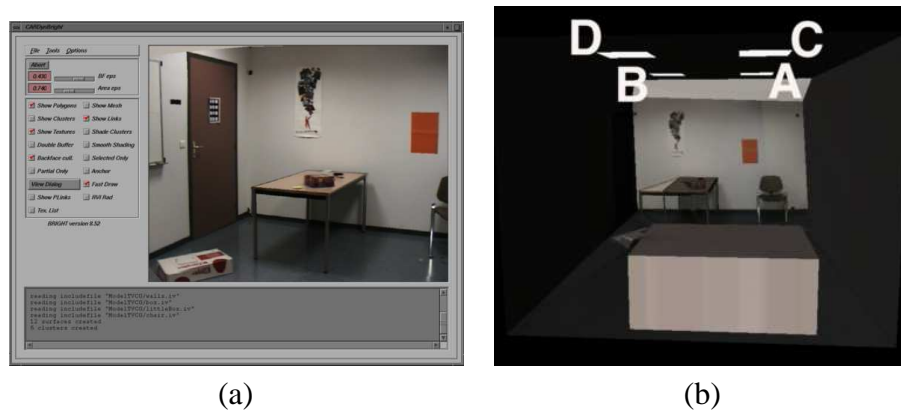


Figure 7: (a) The real scene viewed from our system. (b) The complete model including four lights (A, B, C, D).

The preprocess begins by setting up all necessary parameters for the estimation of the real scene illumination as in [DRB97]. An improved, texture-based refinement follows (described in Section 4.2), allowing better delimitation of real shadow boundaries (see Section 4.3). Real shadows are then removed from real world textures using a radiosity-based algorithm (see Section 4.4), and virtually reprojected using radiosity values (see Section 4.5).

Virtual objects and virtual light sources can then be inserted if desired. The insertion of dynamic objects is performed using the method of [DRB97]. In Section 5, implementation issues are explained which enable the insertion of virtual light sources and texture modulation when the virtually modified illumination is brighter than the original real illumination. Such effects could not be simulated by the algorithm described in [DRB97]. To brighten the textures, we use a multi-pass display described in Section 5.3, which is also applied during the preprocessing step to remove real shadows from textures.

3.3 Interactive modification process

When the entire preprocessing step (see Section 4) is completed, we can interactively modify the illumination of the light sources. Our interface, shown in Figure 8, allows the user to choose a new emittance in each RGB channel for real and virtual light sources.

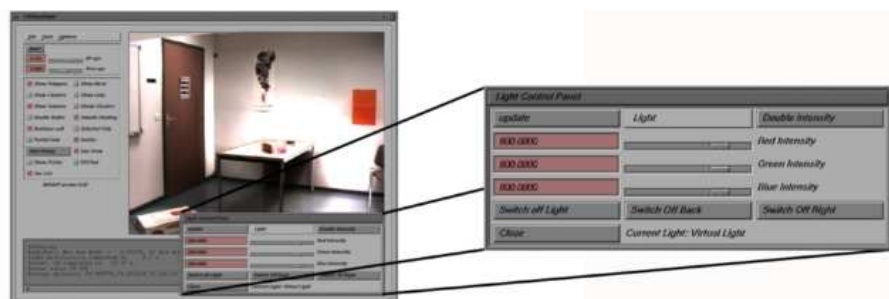


Figure 8: A screen snapshot of the interactive system. The user can manually select new light intensities for real or virtual light sources using the sliders shown in the inset.

As described in Section 5.4, the modification process consists on keeping the mesh subdivision and the hierarchy of links already established, cleaning previous radiosity values, and solving the radiosity system with new intensities by iterating gather and pushpull steps.

4 Preprocessing for Virtual Interactive Relighting

4.1 The problem of pre-existing real shadows

Figure 9 shows a scene illuminated by two sources. The sources A and B correspond to those shown in Figure 7. The radiosity equation for patch i is given by:

$$B_i = \rho_i V_{iA} F_{iA} B_A + \rho_i V_{iB} F_{iB} B_B + B^{ind} \quad (4)$$

where V_{iB} is the attenuation factor due to partial occlusion of light B, F_{iB} is the form factor between patch i and light B, and B_B is the emittance of light B before its modification, and respectively for light A. B^{ind} is the indirect illumination.

Assume that we turn off light source B, the radiosity B_i of a given patch i previously in penumbra will be reduced by the following amount:

$$B'_i = B_i - \rho_i V_{iB} F_{iB} B_B - B_B^{ind} \quad (5)$$

for B_B^{ind} the amount of indirect illumination due to the illumination from light B. In contrast, the radiosity of element j , entirely visible to source B, will be reduced by:

$$B'_j = B_j - \rho_j F_{jB} B_B - B_B^{ind} \quad (6)$$

since the attenuation factor V_{jB} is equal to one. For patch k completely in umbra, its radiosity is decreased by:

$$B'_k = B_k - B_B^{ind} \quad (7)$$

since the attenuation factor V_{kB} equals zero.

If the geometric reconstruction were perfect, and the reflectance estimation, visibility computations and radiosity solution were perfectly accurate, these two equations would suffice. All that would be required is an appropriate refinement to delimit shadow boundaries, and the application of the light modification algorithm described in the previous section.

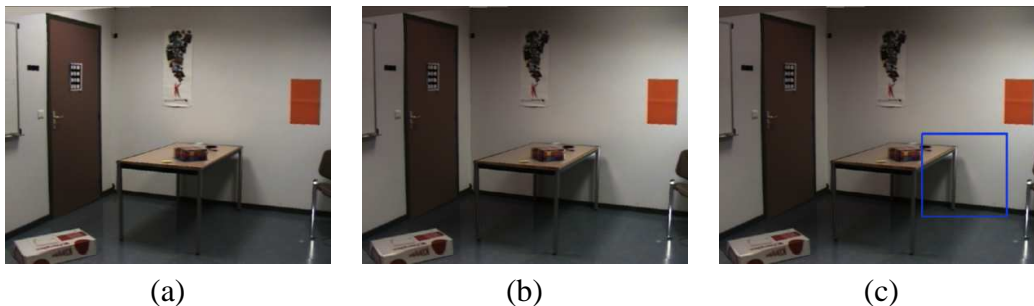


Figure 9: (a) The original illumination of the real scene. Sources A and B (see geometry of Figure 7) illuminate the wall, causing the shadow of the table to be cast on the wall. (b) Light B (on the left) is turned off and the intensity of light A on the right is doubled. (c) Using the method of [DRB97], pre-existing shadows are not removed and the quality of the relighting is unsatisfactory: the shadow due to light B is still clearly perceptible.

This is however far from being the case. Recall that the reflectance values were estimated from the texture values in the original images. Thus, the existence of a real shadow has already been taken into account, and consequently the value of the reflectance ρ_i of the patch i in penumbra will be much

lower than that of ρ_j , which is unoccluded. As a consequence, using the method of [DRB97], even after turning off light B, we still see the corresponding shadow (Figure 9(c)). To solve this problem, we need first to *remove* pre-existing shadows from the textures thus estimating more consistent reflectance values for the regions previously in shadow. Note that we do not require *accurate* reflectance estimation, but simply consistent values which will allow us to produce the desired effect.

During the shadow removal process presented in following sections, we modify the original image texture used to estimate the reflectance values. Intuitively, we identify the regions in (partial or complete) shadow for each source, and attempt to estimate the additional light received if the real blockers were removed. We then modify the hierarchy of radiosity elements, and finally change the texture using this estimation.

Before presenting the details of shadow removal, we present a texture-based refinement algorithm which improves delimitation of real shadows boundaries, necessary for an accurate shadow removal algorithm. We then present the details of each step of the shadow removal process.

4.2 Texture-based refinement for shadow boundaries

In order to remove shadows from original images, shadow boundaries need to be accurately defined. If we use usual refinement criteria, such as BF refinement [HSA91] or error-driven refinement [GH96] we do not obtain the desired result. The main problem is that these approaches do not always guarantee good shadow boundaries (even when using the visibility factor of [HSA91]). Since the geometry reconstruction and the visibility computation via ray-casting are not completely accurate we often get inappropriate subdivision. Discontinuity meshing [LTG93] is unsuitable for the same reasons, since discontinuity lines would be geometrically inaccurate. As a consequence, we use standard quadtree subdivision, with texture-refinement based on new criteria.

The main idea is to use colour information contained in original textures (i.e. the images of the real scene reprojected as texture onto the reconstructed polygons), combined with the visibility information provided by the radiosity system as initialised above. Real shadows already exist in the textures, and correspond to regions which are darker. By using the visibility type (see Section 2.3) contained in the links to patches in penumbra, and the colour differences between neighbouring patches, we can force refinement in regions corresponding to real shadow.

This refinement occurs after the first approximation of the radiosity solution of the real scene (see the algorithm in Figure 5). As shown in Figure 10(a), the initial subdivision, obtained using BF refinement, is coarse. Links from light sources have been attached to the leaves of the hierarchy of patches as in Figure 10(b), to provide accurately visibility information with respect to the light sources. Note however that indirect illumination is represented at higher levels in the hierarchy.

The texture-based refinement algorithm compares the visibility and the colour of two neighbouring leaves of the patch hierarchy. The visibility must be consistent with the colour differences. We consider two cases, for a patch and each of its neighbours (the meaning of “similar” for colour and visibility is defined below):

1. If the two patches have similar colours, they should also have the same visibility type with respect to all the real light sources. If it is not the case, then we subdivide the patch.
2. If the two patches have different colours, they should also have different visibility types. If not, we subdivide the patch.

If the patch has been subdivided, we examine the children created; if there is no change in visibility, the patch subdivision is cancelled and is again a leaf of the patch hierarchy.

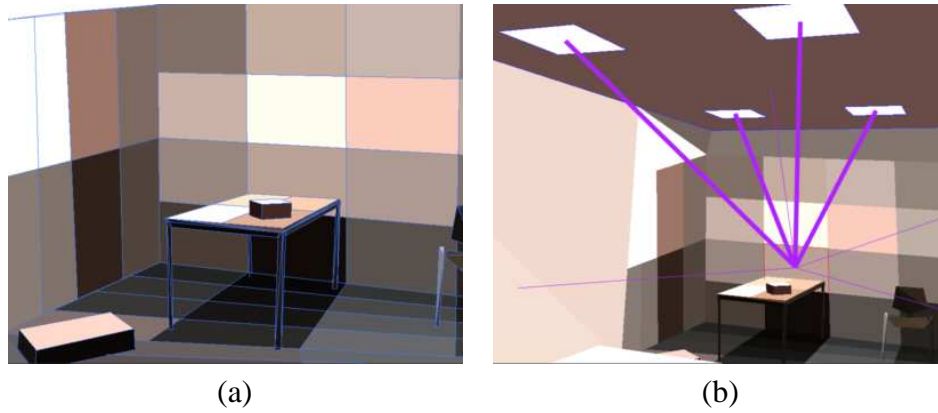


Figure 10: (a) Coarse mesh before refinement. (b) All links from light sources are at leaves.

Case 1 occurs at the limits of shadow boundaries, and helps in producing finer elements in these regions. The process will stop when we reach the maximum subdivision level or when the patches are separated into visible and “in shadow”.

Case 2 occurs when ray-casting has failed to identify the correct visibility type. The patch may be unsubdivided however when the colour difference is not due to a visibility change, but to a different texture. This is the case for the orange poster on the back wall in Figure 11(a).

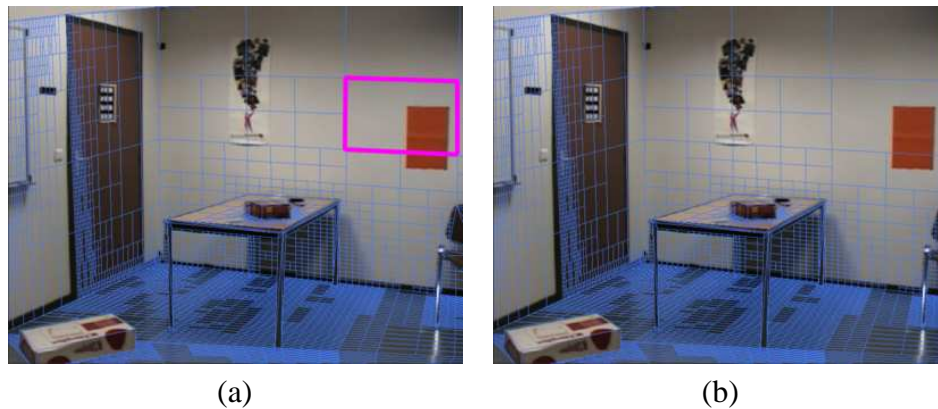


Figure 11: (a) A patch in pink with a different colour than neighbours but the same visibility. The patch was not subdivided. (b) Mesh after texture-based refinement with improved shadow boundaries. Compare to Figure 10.

Figure 12 shows how the refinement algorithm recursively traverses the hierarchy of elements and compares each pair of neighbouring hierarchy leaves. We consider that the visibility is similar if the difference of the attenuation factor is less than a visibility threshold fixed by the user. Similarly, we consider two patches to have the same colour if the difference in colour is less than a colour threshold also fixed by the user.

At the end of the refinement process, we set up all the necessary parameters: the reflectance, the display correction factor D^{orig} , which is equal to the original radiosity B^{orig} , and the texture T^{orig} , which is the original texture before any correction (i.e. still containing real shadows).

Links from real light sources are again fixed at the leaves of the patch hierarchy. A radiosity step (gather/push-pull) is then computed, corresponding to this new subdivision.

The texture-based refinement results in well-defined shadow boundaries, which is very important for the subsequent texture modification step. The resulting refinement is shown in Figure 11(b).

Refinement for shadow boundaries

```

for each leaf  $i$ , compare with its neighbour leaves  $n$ 
  if  $i$  has a similar colour to  $n$  and a different light source visibility
  then subdivide  $i$ 
  else if  $i$  has a different colour to  $n$  and similar light source visibility
  then subdivide  $i$ 
  else do nothing
if  $i$  has been subdivided then
  if  $i$  has no light source visibility differences with its children
  then remove the subdivision of  $i$  ( $i$  is a leaf again)
  else redo the process for each new child of  $i$ .

```

Figure 12: Image-based refinement for shadow boundaries.

4.3 Identification of regions in shadow

The main idea of our shadow removal algorithm is to add the radiosity blocked by real world objects, thus brightening the textures in the shadow regions.

We define \bar{E}_{is} to be the irradiance from source s that was blocked from patch i due to occlusion. This additional irradiance is the radiosity of light sources times the form factor F_{is} and the complement of the attenuation factor equal to $(1 - V_{is})$ for each source s . Considering each real light source, we have the additional irradiance \bar{E}_i for patch i :

$$\bar{E}_i = \sum_s F_{is}(1 - V_{is})E_s \quad (8)$$

The fact that all links from light sources are at the patch hierarchy leaves allows better estimation of \bar{E}_i , since the form-factor and visibility information are more accurate. For more accuracy, we should take into account the occluded indirect illumination. However, since we have not reconstructed every object of the scene, and since the geometric model is approximate, we chose to ignore the occluded irradiance due to indirect illumination.

Having all links from light sources at leaves also ensures the classification *partially or completely occluded* or *completely unoccluded* for a given patch with respect to a given source. We are therefore able to define a frontier composed of completely unoccluded patches that have occluded neighbours with respect to real light sources. This frontier usually encloses the regions where we need to modify the texture. However the algorithm does not depend on the creation of a closed region. The frontier elements will be used as references as explained in the following section.

4.4 Removal of real shadows

Once we have delimited and identified the shadow regions, we need to remove the real shadows from the texture (i.e. the real images), and initialise the system so we can perform interactive relighting.

This process will modify three main quantities: the display correction factor D^{orig} , the radiosity B^{orig} , and the texture T^{orig} . The shadow removal algorithm is summarised in Figure 13.

In Step 1, we perform the intuitive step of adding the radiosity which was blocked, to the accuracy provided by our radiosity approximation. This results in the generation of an intermediate texture T^{inter} , and after a radiosity step (iterative gather/push-pull) intermediate values for display correction

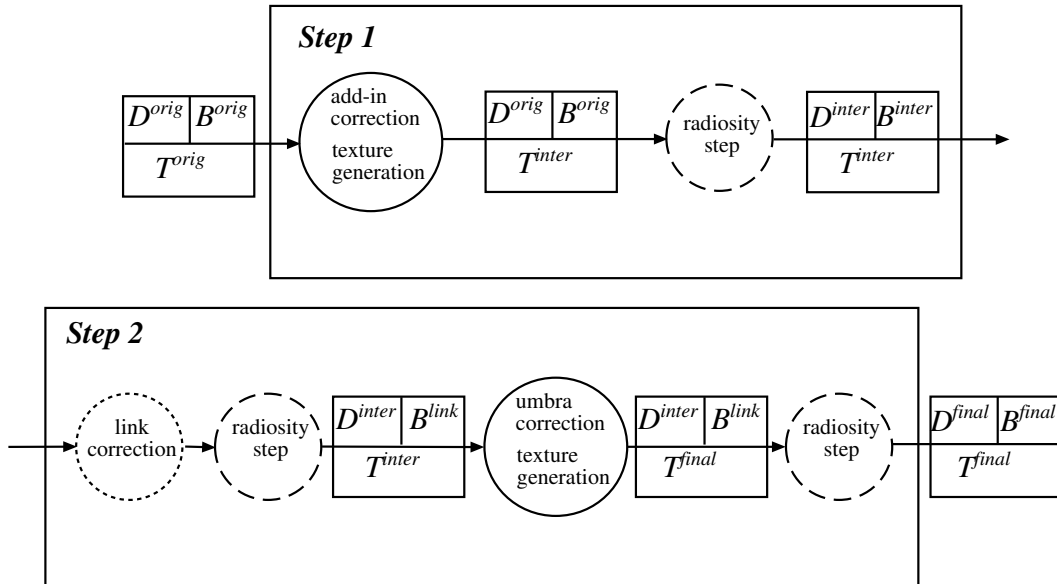


Figure 13: Overview of the shadow removal algorithm.

factor D^{inter} and radiosity B^{inter} . As we shall show in detail, this first intuitive step is insufficient due to the inaccuracies of the geometric reconstruction and the radiosity simulation. As a result, two further corrections are performed, resulting in the final values B^{final} for the radiosity and T^{final} for the textures.

4.4.1 Step 1: Texture generation by adding blocked light

To generate a new texture with the real shadows removed, we modulate the original texture by a correction factor computed at the vertices of the leaf radiosity patches. Modulating the texture at patch vertices results in smooth modified textures.

The correction factor is based on the additional irradiance described above in Eq. (8). To include the blocked radiosity, we could modulate the original texture T^{orig} as follows:

$$\frac{\rho_i E_i + \rho_i \bar{E}_i}{D_i^{orig}} \times T^{orig} \quad (9)$$

In this equation, \bar{E}_i is the potentially blocked irradiance, and $B_i^{orig} = D_i^{orig} = \rho_i E_i$. However, \bar{E}_i is computed with the approximate values F_{is} , V_{is} and E_s , and thus the modulation of Eq. (9) is insufficient. An additional correction factor is required which will brighten textures in the shadow regions.

To determine the correct colour required for the newly generated texture, we use a “reference reflectance” chosen from an appropriate unoccluded patch. Recall that reflectances were computed directly from the textures. Therefore they take into account the shadow colour in occluded regions. In unoccluded regions, the colour of the texture can be considered as a reference for the expected colour in occluded region, after shadow removal. The unoccluded reference reflectance can be used to modulate the texture instead of the initial reflectance of the occluded patch by replacing ρ_i in Eq. (9).

For each occluded patch i , we define a reference patch chosen in the frontier of unoccluded patches. A frontier patch is a non-occluded patch that is a neighbour of an occluded patch. An example of such frontier patches is shown in Figure 14(a). The reference patch r is chosen to have a

similar colour as the occluded patch and to be at a minimum distance from i . For the occluded red patch, shown in Figure 14(b), the algorithm chooses the blue patch as a “reference patch” from the frontier list of unoccluded elements shown in Figure 14(a). The blue frontier patch is the closest and brightest patch that has a similar colour to the red occluded patch (see the algorithm in Figure 15).

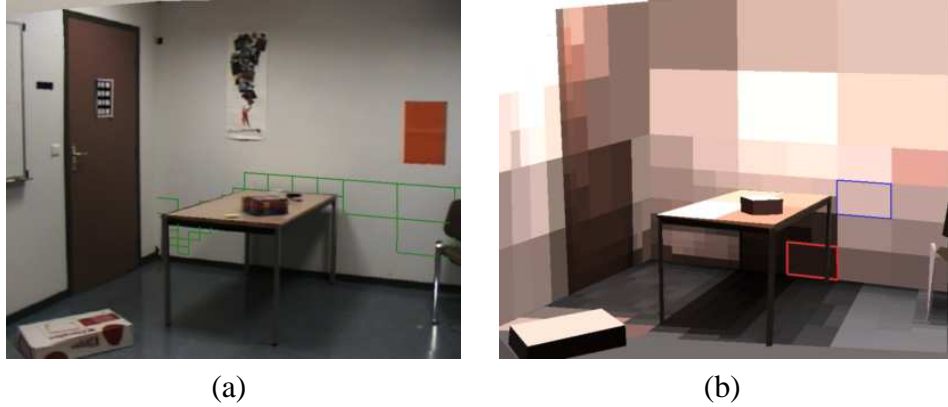


Figure 14: (a) Frontier in green composed of unoccluded patches, which enclosed shadow regions. (b) Blue patch chosen in the frontier as a reference for the red selected patch in shadow.

We define colours to be “similar” if their ratio is equal to $\alpha \sim \beta \sim \gamma$ in RGB defined in the following equation:

$$\begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} = \begin{pmatrix} \alpha R_r \\ \beta G_r \\ \gamma B_r \end{pmatrix} \quad (10)$$

$\alpha \sim \beta$ means that the absolute difference must be less than a certain threshold. We also add the condition that the highest-valued colour channel for i must be also the highest-valued channel for r . Moreover α , β , and γ must be smaller than one since the colour of the reference patch r has to be brighter than occluded patch i .

```

for each leaf  $i$ 
   $min_{distance} = \infty$ 
   $Reference = i$ 
  for each patch in frontier list  $n$ 
    if  $Distance(i, n) < min_{distance}$ 
      and  $n$  brighter than  $i$ 
      and  $n$  has a similar colour to  $i$ 
      and  $n$  brighter than  $Reference$ 
    then
       $Reference = i$ 
       $min_{distance} = Distance(i, n)$ 

```

Figure 15: Algorithm to choose reference patches.

We use the reflectance of the reference patch in the correction factor to brighten the texture. Since the reference patch is at a certain distance from the occluded patch, we modulate the reflectance of

the reference patch by the ratio of the form factors of patch i (F_{is}) and patch r (F_{rs}) with respect to the source s . Since we do not want to darken the texture, the reference patch r must also follow the rule:

$$\rho_r \frac{\sum_s F_{is}}{\sum_s F_{rs}} > \rho_i \quad (11)$$

If no patch in the frontier of unoccluded elements is found for a certain patch i , then the reference patch is a default reference patch selected by the user, or the patch i itself if the user selected patch does not verify the condition in Eq. (11).

For each patch i , we calculate the modified radiosity value B_i^{corr} which is used to generate a new texture:

$$B_i^{corr} = \rho_r \frac{\sum_s F_{is}}{\sum_s F_{rs}} \left(\sum_s (F_{is} V_{is} E_s + \bar{E}_{is}) + E_{ind} \right) \quad (12)$$

B_i^{corr} is a function of the reflectance of the reference patch ρ_r , of the ratio of the form factors, and of the sum of irradiance including the indirect irradiance E_{ind} (i.e. arriving from patches/clusters other than the primary sources).

We next generate a new, intermediate texture T^{inter} . To generate this texture, we render the textured leaf patches of the patch hierarchy with an appropriate modulation factor. For occluded patches only, the texture T_i^{orig} is modulated by the ratio of the correction factor B_i^{corr} of patch i and the display correction factor D_i^{orig} of patch i :

$$\frac{B_i^{corr}}{D_i^{orig}} \times T_i^{orig} \quad (13)$$

As expected, B_i^{corr} is greater than D_i^{orig} . To account for this brightening effect, we use a multi-pass display method described in Section 5.3 to generate a brighter intermediate texture T^{inter} . To create a new texture, we use an off screen window. In this window, we display the modified texture, and save it to a new texture. The texture window is shown in Figure 16(a), and the resulting textures T^{inter} are shown in (b).

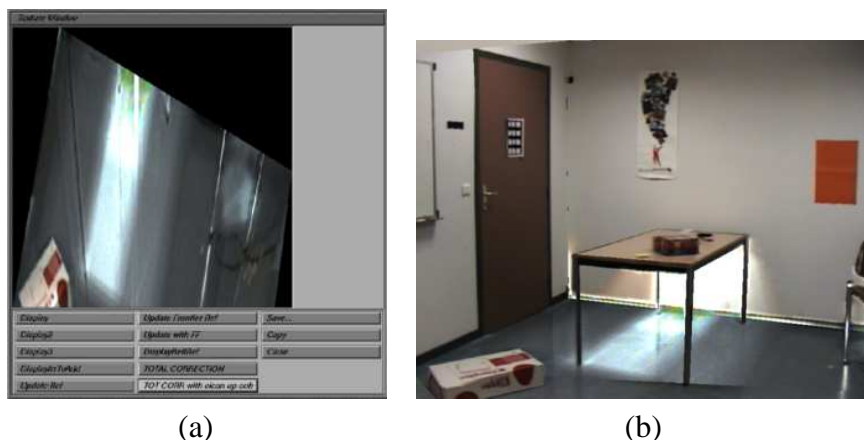


Figure 16: (a) Off screen window to create new textures. (b) The resulting texture T^{inter} , with real shadows removed, mapped onto the geometry of the real scene.

From the intermediate texture T^{inter} , we recompute new reflectance values for occluded patches and perform a radiosity step (see Figure 13), resulting in new radiosity values B_i^{inter} based on the new

reflectance, and new display correction factor D_i^{inter} , equal to the sum of the occluded radiosity B_i^{inter} and the additional irradiance \bar{E}_i (see Eq. (8)) times the new reflectance.

An illustration of D_i^{inter} is given in Figure 17(a), and B_i^{inter} is shown in Figure 17(b). We apply the ratio $\frac{B_i^{inter}}{D_i^{inter}}$ to the texture T_i^{inter} for displaying. The result of this process is shown in Figure 17(c).

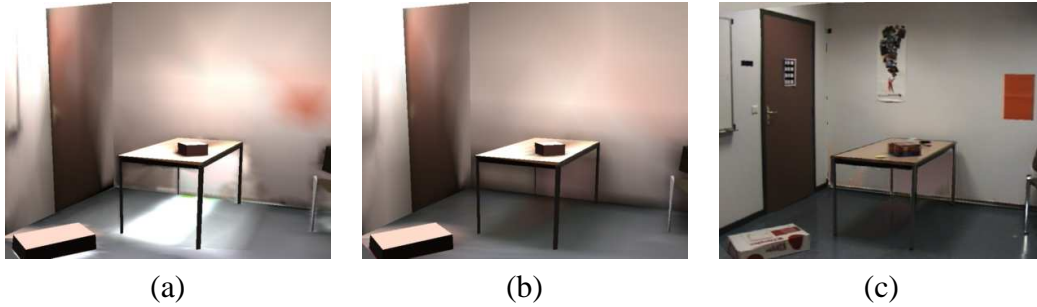


Figure 17: (a) Display correction D_i^{inter} corresponding to the new texture T_i^{inter} . (b) Radiosity B_i^{inter} corresponding to the new texture T_i^{inter} . (c) The resulting reprojection using these intermediate values.

As we can clearly see, the result is too bright and the resulting reprojected shadows are incorrect. This is due to the inaccuracies of the emittance and form-factor values which have been estimated using the radiosity solution (see Section 2.4). To compensate, we need to apply a subsequent correction step which in effect locally adapts the emittance values of the sources to be consistent with the original shadows.

4.4.2 Step 2: Link and umbra correction and final texture generation

We now have a configuration in which each (modified) surface has an intermediate texture T_i^{inter} and intermediate display correction factor and radiosity values. Our goal is to generate a new texture T_i^{final} which will be consistent with the original, real world image.

For each occluded patch i , we first compute a correction factor attached to each link with each real light source, using the radiosity parameters estimated from the intermediate texture.

Recall that to display the final image, we use the ratio B_i over D_i . Shadows are reprojected correctly because D_i is brighter than B_i . To achieve the desired effect, we need to compute a correction factor C_{link} for all sources s and patch i so that:

$$\frac{B_i^{link}}{D_i^{inter}} \times T_i^{inter} = \frac{\rho_i^{inter}(\sum_s C_{link} V_{is} F_{is} E_s + E_i)}{\rho_i^{inter}(\sum_s F_{is} E_s + E_i)} \times T_i^{inter} = T_i^{orig} \quad (14)$$

Using Eq. (14), if we consider that the correction factor C_{link} is the same for each light source, we get:

$$C_{link} = \frac{T_i^{orig}}{T_i^{inter}} \times \frac{\sum_s F_{is} E_s + E_i}{\sum_s F_{is} V_{is} E_s} - \frac{E_i}{\sum_s F_{is} V_{is} E_s} \quad (15)$$

The ‘‘link correction’’ C_{link} is computed for every link on every (partially) occluded patch. This correction is applied when computing the radiosity during the gather operation in the radiosity step. For example, the incoming irradiance E_i gathered at patch i from source s with emittance E_s is $E_i = C_{link} V_{is} F_{is} E_s$. After the completion of the radiosity step, each patch has a new radiosity value B_i^{link} (see Figure 13). Since $\sum_s F_{is} V_{is} E_s$ is null for *completely occluded* patches for all sources ($V_{is} = 0$), the

factor C_{link} can only be computed for *partially occluded* patches. Due to inaccuracies of the model, we do not use the correction factor when $\sum_s F_{is} V_{is} E_s$ is less than a certain (small) threshold value.

An additional correction factor C_{umbra} is thus required to compensate for the inaccuracy of interaction on completely (or almost completely) occluded (umbra) patches. This factor is applied after the computation of the new radiosity values B^{link} (see Figure 13). For each hierarchy patch leaf i in umbra the correction factor C_{umbra} is as follows:

$$C_{umbra} \frac{B_{link}}{D_i^{inter}} T_i^{inter} = T_i^{orig} \Rightarrow C_{umbra} = \frac{T_i^{orig}}{T_i^{inter}} \frac{D_i^{inter}}{B_{link}} \quad (16)$$

Since the correction factor C_{umbra} compensates for indirect illumination and is independent of the light sources, we use it to modify the intermediate texture to create the new, final texture T^{final} . This final texture, after the application of the correction factor, is shown in Figure 18(e). We see that the values are darkened.

Again, from this final texture, we recompute reflectance. A radiosity step follows resulting in the computation of final radiosity B^{final} , and display correction factor D^{final} which includes the additional irradiance \bar{E}_i in the same manner as for the generation of intermediate texture T^{inter} (see Figures 13 and 18).

4.5 Shadow reprojection

Once the final values for the display correction factor and current radiosity have been obtained, we can reproject shadows of the real objects, with the same display ratio (Eq. (3)). Since the correction factor values D_i^{final} are greater than the radiosity values B_i^{final} in occluded regions, the texture is darkened by the ratio $B_i^{final} / D_i^{final}$ and the display gives the impression of shadow.

The updated display correction factor D_i^{final} after texture modification is shown in Figure 18(a), and the radiosity B_i^{final} in (b). This radiosity is brighter than the radiosity in Figure 18(d) estimated from the original texture. This is due to the new reflectance values, which no longer take into account the colour of shadows. The resulting scene after texture modification is shown in Figure 18(f), which is very similar to the original image shown in Figure 18(c).

5 Implementation Issues

5.1 Insertion of virtual light sources and virtual objects

Insertion and dynamic modification are performed using the algorithm presented in [DS97], with an adaptation for virtual light sources. Moreover, we need to be able to brighten textures when the illumination is increased. Our solutions to these issues, as well as the interactive modification process, are described next.

5.2 Modified refinement to insert virtual sources

We adapt the method of [DRB97], in which a virtual object can be inserted into the real scene and interactively manipulated [DS97], to also treat the insertion of a virtual light source. This results in the projection of the shadows due to the virtual source on the real objects. The influence of a virtual light is significant, and thus we force additional refinement by establishing all links to virtual sources

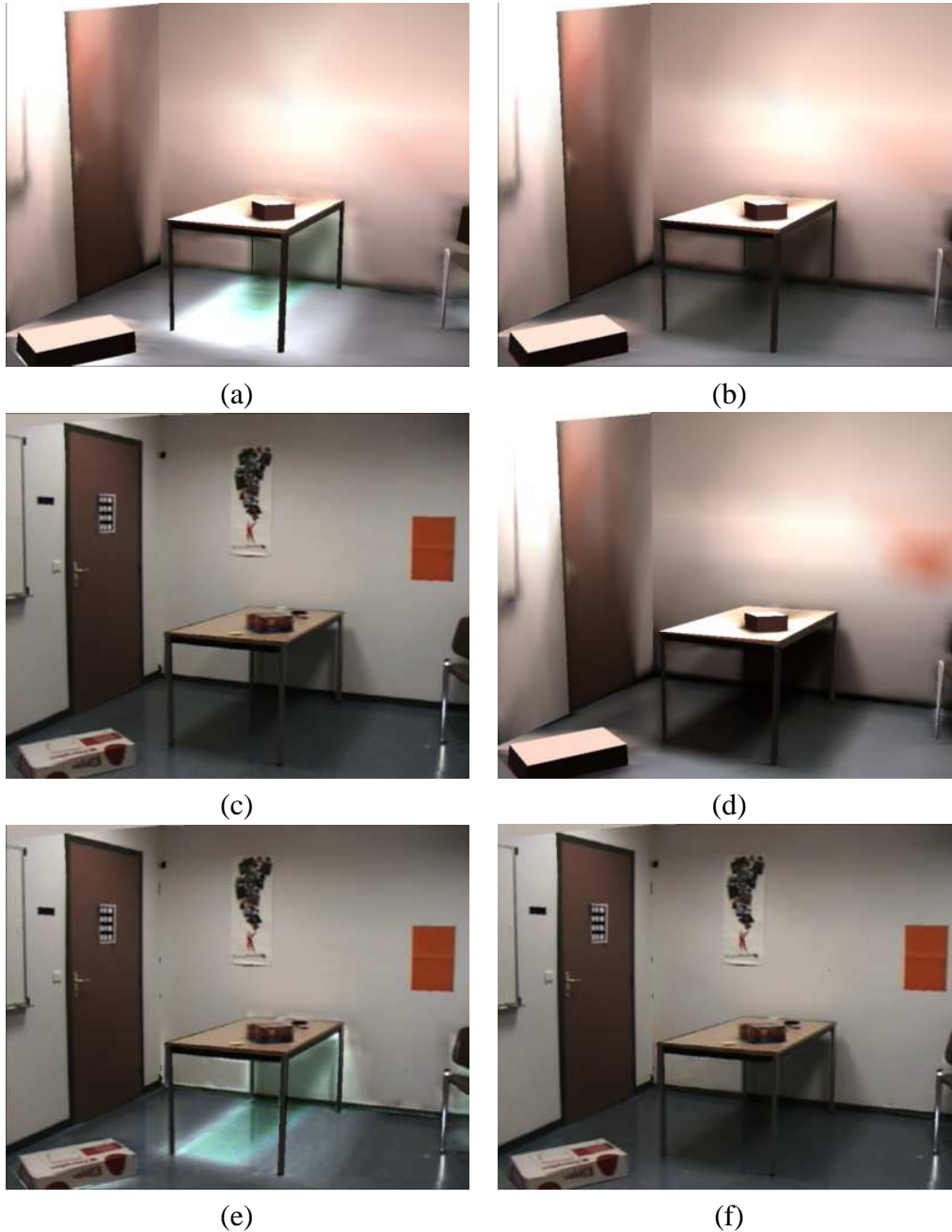


Figure 18: (a) Display correction D^{final} updated from the final textures T^{final} . (b) Radiosity B^{final} also estimated from the final textures T^{final} . (c) Original real scene before any modification. (d) Radiosity estimated from the original real scene. (e) Final textures after all modifications (with no radiosity, only textures). (f) Real scene after texture modification, with shadow reprojection.

on the polygons as opposed to allowing links from the virtual sources to the clusters. This is done on the polygons visible in the captured images; the polygons corresponding to the hidden parts of the scene are not affected by this forced refinement.

The additional light sources brighten the scene. To make this perceptible, it is necessary to overcome a limitation of the OpenGL graphics library used for our implementation. Virtual light source insertion is illustrated in Figures 20, 23, 24, 25.

5.3 Multi-pass display

The display ratio described in Eq. (3) is insufficient when B_i is greater than D_i . This is due to a limitation of the *glColor* function of OpenGL, which requires a colour value between zero and one. Therefore, if no special treatment is done when B_i/D_i is greater than one, the ratio is automatically clipped to one, and the desired illumination effect is not achieved.

In order to brighten the textures when source intensities increase, we use a n -pass display method, where $(n - 1)$ is the integer part of B_i/D_i . The n^{th} pass corresponds to the remainder of $B_i/D_i - (n - 1)$. To achieve the desired effect, we use the *glBlend* function of OpenGL. This enables us to modulate the texture by the colour $B_i/D_i - i$ or 1 if $(B_i/D_i - i) > 1$, n times. The multi-pass algorithm is described in Figure 19. The drawback of this approach is that the display time is increased during the light modification pass, taking between a 0.05 and 1.0 seconds. An example of the multi-pass display is shown in Figure 20, where a virtual light source has been added to the scene.

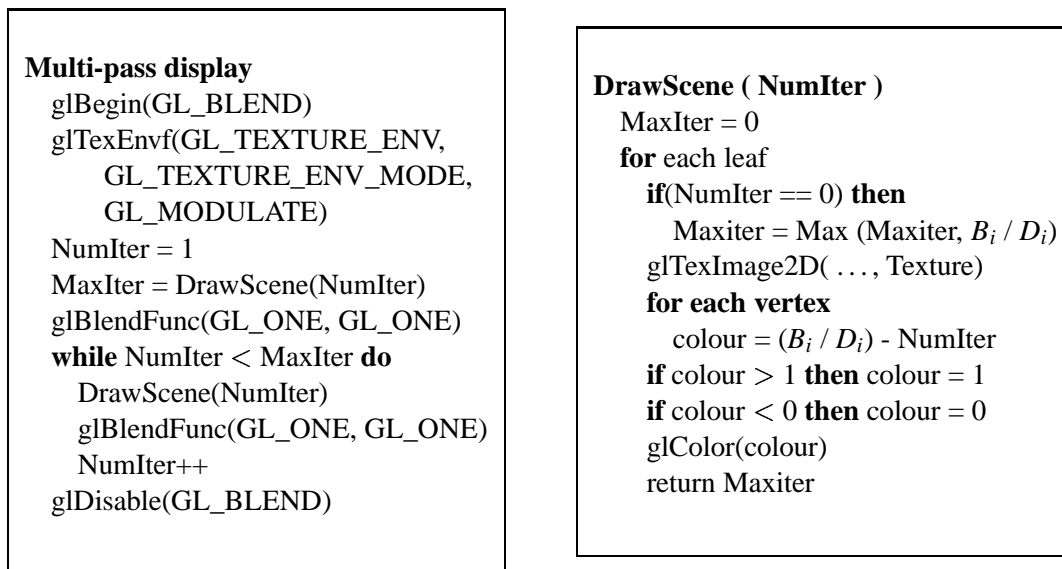


Figure 19: Multi-pass display.

Similar problems have been encountered by other researchers. Debevec et al. [DBY98] use a similar approach when combining several weighted textures to create a single one, while Soler and Sillion [SS98] also used a multi-pass display to correctly modulate textures with direct illumination.

5.4 Final relighting

Recall that the radiosity system has been previously set up, and that real shadows were removed from real world textures. Links between patches and clusters in the radiosity hierarchy have already been established, including the form factor computation and the visibility determination. In order to avoid

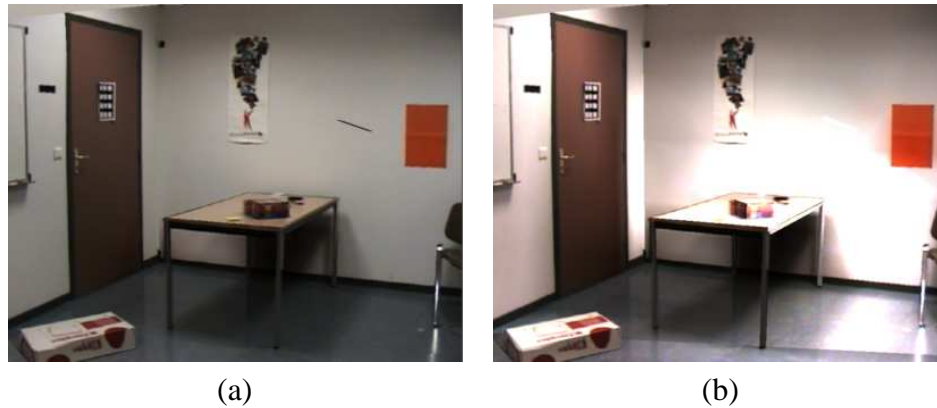


Figure 20: (a) Insertion of virtual light source, with no multi-pass display. (b) Using the multi-pass display (5 passes in 0.3 seconds).

supplementary costs, the subdivision and the links are maintained. Since we only modify the intensity of the light sources, the subdivision and links still fit to the illumination even after modification. Keeping the same hierarchy may provide superfluous mesh subdivision if lights are switched off; since the user may switch them on again later however, we prefer to maintain the mesh subdivision. The modification process consists in recursively removing radiosity stored at each level of the hierarchy. We then perform a complete *radiosity step*: without performing additional refinement, we gather radiosity across the links, and perform the push-pull step to maintain a coherent representation of radiosity in the hierarchy. The iterative process is stopped when the global illumination is stable.

This process is interactive since the costly refinement step (which includes visibility and form-factor computation) is avoided. The update time depends on the initial level of subdivision. Note however that the insertion of a virtual object may result in additional subdivision. The update rate is the same if we modify one or several lights. Example update rates are shown in Figure 21. As expected, the time for each modification increases when the number of leaves of the subdivision increases. Times are computed on an SGI Onyx2 Infinite Reality workstation.







Refinement			
Decrease light intensity			
Times for modification	0.3 sec.	0.5 sec.	1.0 sec.
Times for display	0.1 sec.	0.3 sec.	0.5 sec.
Number of leaves	513	942	1830

Figure 21: Interactive modification of a virtual light source intensity. The time rates depends on the level of the subdivision.

6 Results

Once the preprocessing step is complete, we can interactively modify the illumination from real and virtual light sources. We have tested the algorithm on a scene with four real lights. We first switch off the two back lights (C, D) shown in Figure 7. In the resulting image Figure 22(b), the scene is darker than the original illumination shown in Figure 22(a) but with no change in shadows.

We then switch off the front left light (A) and double the intensity of the right light (B) (see Figure 22(c)). The resulting shadow of the table is homogeneous in the umbra regions. As expected, the shadow due to the left light has disappeared, since the part of the scene which was illuminated by this light source is darker. Compare the new result with that of the method of [DRB97] previously shown in Figure 9(c), which was inexact, since real shadows were not removed from textures.

We now switch on the left light with double the original intensity and switch off the right light (see Figure 22(d)). Again, shadows are as expected (i.e. the shadow boundary of the right light is no longer visible). For each light modification, the whole process (radiosity step and display) takes 0.8 seconds. The accompanying video shows these light modifications, recorded in real time on an SGI Onyx2 Infinite Reality workstation.

We can also insert a virtual source, and modify its intensity as described above. An interesting test is to switch off all the real lights, and to illuminate the real scene only by a virtual source (see Figure 23(a) and (b)). Notice that real shadows from real light sources are no longer perceptible. However, real objects such as the table cast new shadows on the floor and the walls, due only to the virtual light.

With this new illumination, we are still able to interactively move a dynamic virtual object, such as the orange box on the floor in Figure 24. Updates take 0.3 sec. per frame, when moving the virtual object, with the subdivision shown in Figure 24(a). With both real and virtual illumination, this virtual object casts shadows onto the real scene.

If we want to modulate lights, we can relight the real scene in mixed colours (see Figure 25).

We have also tested our method on another real scene, shown in Figure 26(a). In (b), we have removed the real shadows from textures of this scene. We have performed real relighting in (c) by switching off five lights (out of six). In (d), we have inserted a virtual light source, and we have relighted with mixed colours.

7 Future work

Since the majority of the work is done during a preprocessing step, especially for real shadow removal from textures, the relighting process is interactive and allows fast manipulation of the real scene. Two main issues need to be addressed: the speed of the updates and the quality of shadow removal.

To increase update rates, recall that we maintain the subdivision intact when performing the light modification process, and that the rates are directly related to the subdivision. We therefore perform the modification with the subdivision used for real shadow removal, which is undoubtedly excessive. The multi-pass display also takes time, and it could also be optimised. The radiosity steps could also be optimized by avoiding the complete traversal of the hierarchy in the spirit of [DS97].

The quality of the shadow removal is directly related to the subdivision effected by the hierarchical radiosity algorithm. Our texture-based refinement has greatly improved the resulting quadtrees compared with traditional refinement approaches, but it is still susceptible to problems mainly due to inaccurate geometric reconstruction. Another limitation of our system is the constraint to a fixed view point. Users modifying the illumination of a real scene, would like to change the point of view

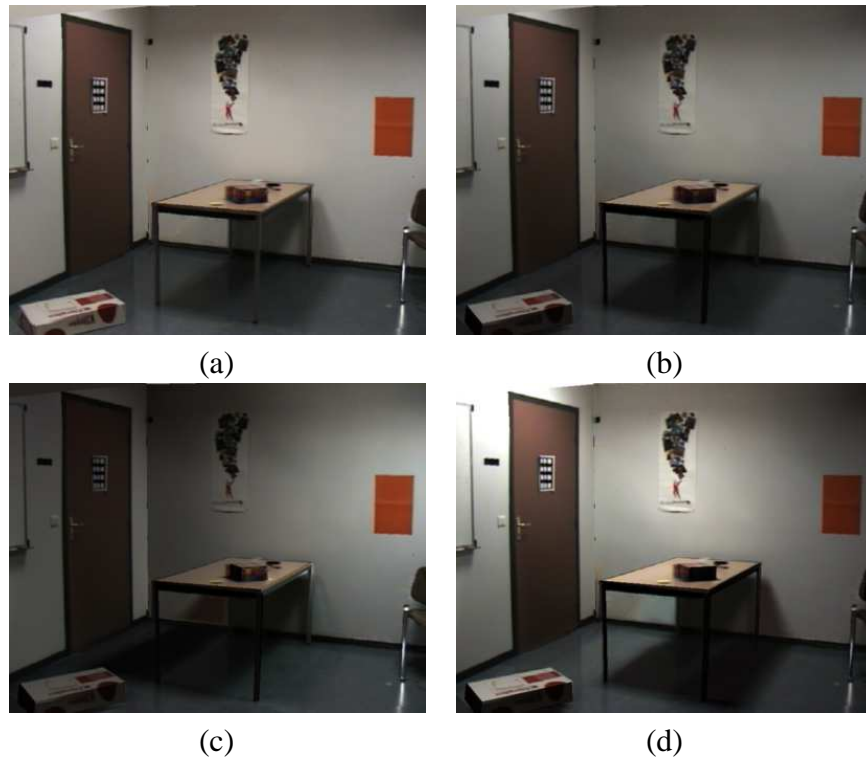


Figure 22: (a) Original scene lit with shadow reprojection. (b) Back lights are switched off. (c) Left light is switched off, and right light has double intensity. (d) Right light is switched off, and left light has double intensity.

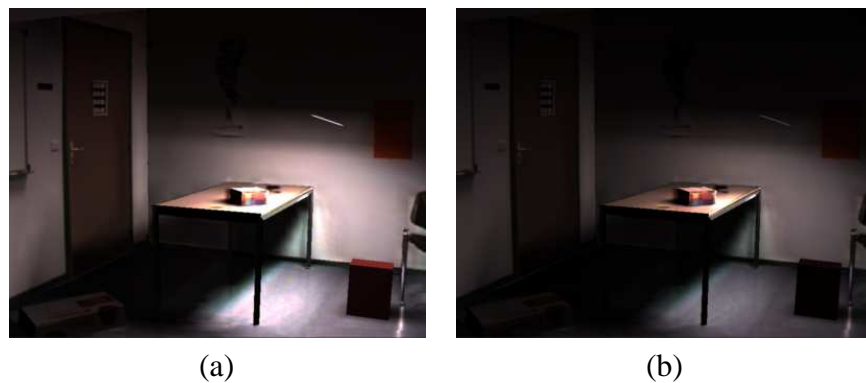


Figure 23: (a) Insert a virtual light. Switch off all real lights. The real scene is lit only by the virtual light. (b) Decrease the intensity of the virtual light.

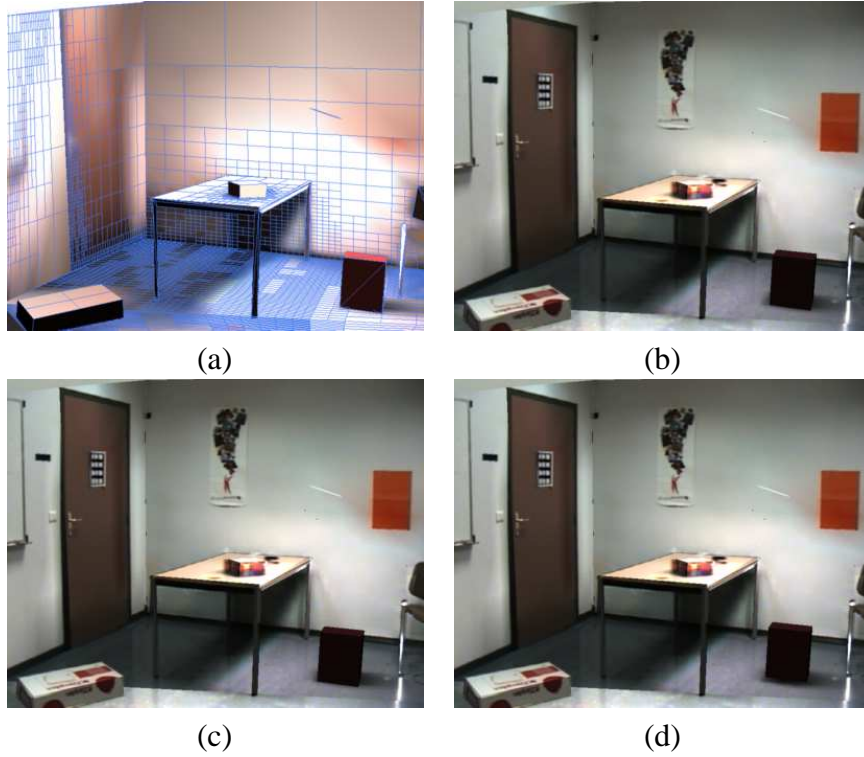


Figure 24: (a) Insertion of virtual object and the consequent subdivision. (b), (c), (d) The red virtual object is moving at interactive rates.



Figure 25: Different colour for each light.

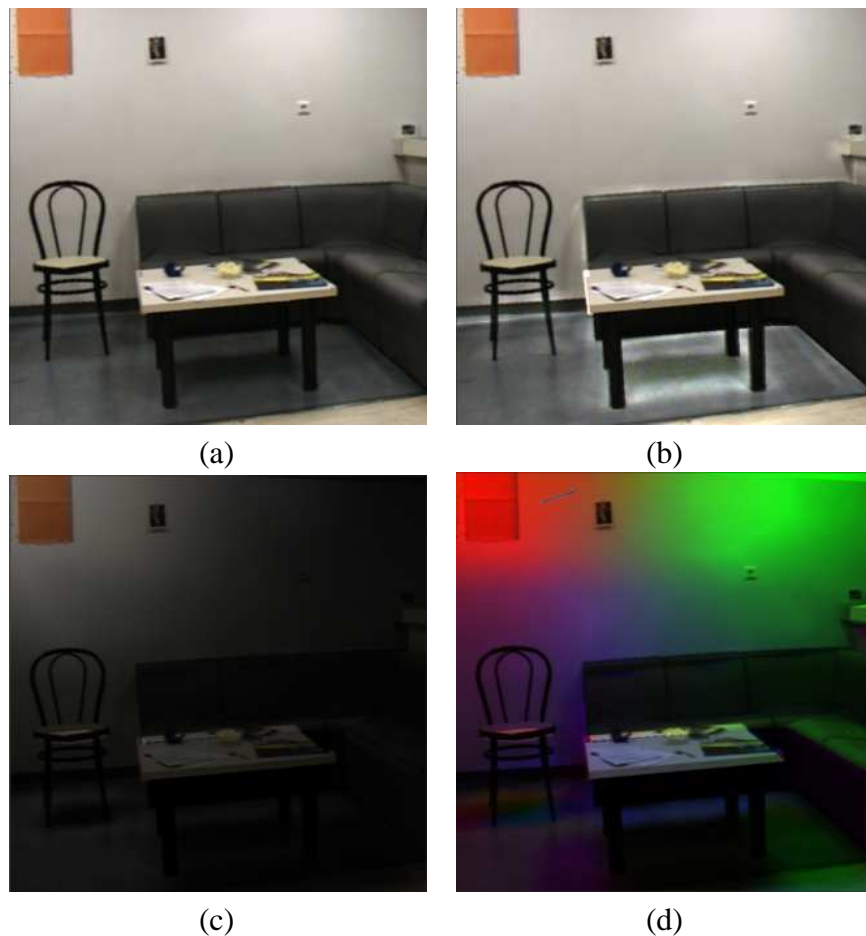


Figure 26: (a) Original real scene viewed from our system. (b) Real shadows were removed from real world textures. (c) Five of the six real lights were interactively switched off and the scene is lit only by the front left light source. (d) A virtual light source was inserted and different colour intensities were set for the two front real lights and the virtual light.

to better appreciate light effects. Building on recent work [POF98, DBY98] we believe that we could develop a solution at least for a limited set of viewpoints.

Another interesting research direction is to allow the removal or replacement of real objects. For example, consider that we want to remove the real table of our scene (Figure 7) and replace it with a different virtual table. There are several problems. First we have to extract the shadows of the real table from the real world textures. We also have to remove its reflections on specular surfaces or mirrors, such as the reflection of the table on the floor. Another problem is the extraction of the image of the table from the textures of the wall and of the floor. A combination of our illumination-based algorithms and more traditional vision-based approaches could be used to correctly fill holes or unwanted object images when completely removing a real object from the real scene.

8 Conclusion

We have presented a solution to remove real object shadows, thus enabling interactive modifications of both real and virtual illumination for reconstructed real scenes. The algorithm we presented has three main steps. The first step is the real scene reconstruction of a 3D geometric model using advanced

vision-based techniques. The second step is a preprocessing algorithm which removes real shadows from the images of the real world. We use the structure provided by hierarchical radiosity to detect shadow regions. Once these regions are identified, we modify real world textures to include the real light source illumination blocked by real occluders, thus brightening the dark shadow regions. Real shadows are then reprojected using the radiosity system. Once the preprocess is finished, the real room illumination is very similar to the original image even though real shadows have been removed from textures.

We then have the ability to select and change the intensity of each real and virtual light source. Since real shadows were previously removed, the obtained solutions are consistent: i.e. shadow boundaries due to switched-off light disappear. Modification of light source intensities, and displacement of virtual objects is achieved interactively, correctly incurring all (diffuse) common illumination effects such as shadows between real and virtual objects.

Acknowledgments

This work was funded in part by the Reactive LTR Project ARCADE (#24944) of the European Union. We would like to thank Sylvain Bougnoux for providing the TotalCalib system. We also would like to thank Cyril Soler and François Sillion for the many discussions and help.

References

- [Azu97] Ronald T. Azuma. A survey of augmented reality. In *Presence: Teleoperators and Virtual Environments 6*, pages 355–385, August 1997. Earlier version in Course Notes #9: Developing Advanced Virtual Reality Applications, ACM SIGGRAPH (LA, 1995), 20-1 to 20-38.
- [CCWG88] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 75–84, August 1988.
- [DBY98] Paul Debevec, George Borshukov, and Yizhou Yu. Efficient view-dependent image-based rendering with projective texture-mapping. In *Rendering Techniques '98*, 9th EG workshop on Rendering, Vienna, Austria, June 1998. Springer Verlag.
- [Deb98] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Computer Graphics (ACM SIGGRAPH '98 Proceedings)*, 1998.
- [DM97] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 369–378. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
- [DRB97] George Drettakis, Luc Robert, and Sylvain Bougnoux. Interactive common illumination for computer augmented reality. In J. Dorsey and P. Slusallek, editors, *Rendering Techniques '97*, pages 45–56, 8th EG workshop on Rendering, Saint Etienne, France, June 1997. Springer Verlag.
- [DS97] George Drettakis and François Sillion. Interactive update of global illumination using a line-space hierarchy. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 57–64. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
- [DTM96] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 11–20. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

- [FGR93] Alain Fournier, Atjeng S. Gunawan, and Chris Romanzin. Common illumination between real and computer generated scenes. In *Proceedings of Graphics Interface '93*, pages 254–262, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.
- [FLR⁺97] Olivier Faugeras, Stéphane Laveau, Luc Robert, Gabriella Csurka, Cyril Zeller, Cyrille Gauclin, and Imed Zoghliami. 3-d reconstruction of urban scenes from image sequences. *CVGIP: Image Understanding*, 1997.
- [GH96] S. Gibson and R. J. Hubbard. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15(5):297–310, 1996. ISSN 0167-7055.
- [GTGB84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 212–22, July 1984.
- [HSA91] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, July 1991.
- [HW91] Eric Haines and John Wallace. Shaft culling for efficient ray-traced radiosity. In *Eurographics Workshop on Rendering*, 1991.
- [JNP⁺95] Pierre Jancène, Fabrice Neyret, Xavier Provot, Jean-Philippe Tarel, Jean-Marc Vézien, Christophe Meilhac, and Anne Vérioust. Res: computing the interactions between real and virtual objects in video sequences. In *Second IEEE Workshop on Networked Realities*, pages 27–40, Boston, Massachusetts (USA), October 1995. <http://www-rocq.inria.fr/syntim/textes/nr95-eng.html>.
- [LTG93] Daniel Lischinski, Filippo Tampieri, and Donald P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 199–208, 1993.
- [NHIN86] Eihachiro Nakamae, Koichi Harada, Takao Ishizaki, and Tomoyuki Nishita. A montage method: The overlaying of the computer generated images onto a background photograph. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 207–214, August 1986.
- [POF98] Pierre Poulin, Mathieu Ouimet, and Marie-Claude Frasson. Interactively modeling with photogrammetry. In *Rendering Techniques '98*, 9th EG workshop on Rendering, Vienna, Austria, June 1998. Springer Verlag.
- [Rob95] L Robert. Camera calibration without feature extraction. *Computer Vision, Graphics, and Image Processing*, 63(2):314–325, March 1995. also INRIA Technical Report 2204.
- [SAG94] Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 435–442. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [SDS95] F. Sillion, G. Drettakis, and C. Soler. A clustering algorithm for radiance calculation in general environments. In *Eurographics Rendering Workshop 1995*. Eurographics, June 1995.
- [SHC⁺96] Andrei State, Gentaro Hirota, David T. Chen, Bill Garrett, and Mark Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 429–438. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.
- [Sil95] F. X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):240–254, September 1995. ISSN 1077-2626.
- [SP94] François Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, San Francisco, 1994. excellent coverage of radiosity and global illumination algorithms.
- [SS98] Cyril Soler and Francois Sillion. Automatic calculation of soft shadow textures for fast, high-quality radiosity. In *Rendering Techniques '98*, 9th EG workshop on Rendering, Vienna, Austria,

- June 1998. Springer Verlag.
- [Tot] <http://www.inria.fr/robotvis/personnel/sbournou/TotalCalib/>.
- [War94] Gregory J. Ward. The RADIANCE lighting simulation and rendering system. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 459–472. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [YM98] Yizhou Yu and Jitendra Malik. Recovering photometric properties of architectural scenes from photographs. In *Computer Graphics (ACM SIGGRAPH '98 Proceedings)*, 1998.
- [ZDFL95] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, October 1995.
- [ZFD97] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2d mosaic from a set of images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997. IEEE.



Unit e de recherche INRIA Lorraine, Technop ole de Nancy-Brabois, Campus scientifi que,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS L ES NANCY
Unit e de recherche INRIA Rennes, Irista, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit e de recherche INRIA Rh one-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

 diteur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399