Interactive Re-lighting of Panorama

Tien-Tsin Wong[†] ttwong@acm.org
Chi-Wing Fu[‡] cwfu@cs.indiana.edu
Pheng-Ann Heng[†] pheng@cse.cuhk.edu.hk

[†] Dept. of Computer Science & Engineering, The Chinese University of Hong Kong
[‡] Dept. of Computer Science, Indiana University at Bloomington

Abstract— **Due to the simplicity and efficiency of panoramic image representation, it has been successfully applied to Internet applications to provide an immersive illusion within a real or synthetic environment. However, once a scene has been captured as image, the lighting condition can no longer be adjustable. As the illumination adjustment is one key capability in computer graphics, we describe in this article a panoramic image representation which allows us to incorporate the illumination information into the panorama. Scenes represented by this representation not just allow panning, tilting and zooming, but also allow interactive adjustment of lighting condition. An interactive panoramic viewer is developed to demonstrate the feasibility and applicability of the proposed representation.**

I. Introduction

Recent advances in image-based computer graphics technologies have induced various applications. One image-based technique which has been successfully adopted in various commercial applications is panoramic image representation. A well-known example is QuickTime VR [1]. It uses a single panoramic image to provide an immersive illusion within an environment. The panorama can either be captured from the real world or synthesized. To generate the perspective snapshot of the scene, part of the panorama is warped into the perspective image. The simplicity, efficiency and visual richness of this technique are the major reasons of its popularity.

The main advantage of image-based computer graphics is that the rendering time is independent of scene complexity as the rendering is actually a process of manipulating image pixels, instead of simulating light transport. On the other hand, the major drawback is its rigidity. Once the scene is captured as images, these images are no longer modifiable. Panoramic image representation also inherits the same rigidity. However, the ability to control the illumination of the modeled scene enhances the 3D illusion which in turn improves viewers' understanding of the environment. If the illumination can be modified by *re-lighting* the images instead of rendering the geometric models, the time for image synthesis should also be independent of scene complexity. By saying re-lighting, we mean generating desired images with novel lighting condition from pre-recorded reference images. This will save the artist/designer enormous time in fine tuning the lighting condition to achieve dramatic and emotional atmosphere. All the above reasons motivate us to develop an interactive image-based panorama viewer which incorporates illumination information for re-lighting.

A few techniques have been proposed to achieve re-lighting in the framework of image-based modeling and rendering. Nimeroff *et al.* [2] described an efficient technique to re-light images under various natural illumination. The intrinsic assumption of natural illumination may not be applicable for other kinds of illumination. Another approach [3] to capture illumination information is to use singular value decomposition (SVD) [4] to extract the principle components (eigenimages) from reference images. Unfortunately, since there is no direct relationship between the lighting geometry (the direction of light vector) and the principle components extracted, the illumination is basically *uncontrollable*. That is, one can change the illumination but *cannot* precisely specify the lighting direction.

In this article, we describe our solution for re-lighting panoramic images. We have proposed a concept of *apparent Bi-directional Reflectance Distribution Function of pixel* (pBRDF) [5] to represent the outgoing radiance distribution passing through the pixel window on the image plane. By treating each image pixel as an ordinary surface element, the radiance distribution of that pixel under various illumination conditions can be recorded in a table. If this table is incorporated into the panoramic image data structure, re-lighting can then be achieved. We have developed an interactive panorama viewer which not just allows panning, tilting and zooming but also allows modifying the lighting condition of the modeled environment.

II. Apparent BRDF of Pixel

The reflected radiance leaving the surface element can be calculated once the viewing vector, the light vector, the incident radiance and the reflectance of the surface element are known. The most general form to express the reflectance of a surface element is the Bidirectional Reflectance Distribution Function (BRDF) which is a four-dimensional table indexed by the light vector $\vec{L}$ and the viewing vector $\vec{V}$ (see Figure 1). Intuitively speaking, it tells us how the surface element looks like when it is illuminated by a light ray coming along the vector $-\vec{L}$ [1] and as viewed from the viewing direction $\vec{V}$. However, unlike the case in geometry-based computer graphics, we cannot (or refrain to) access the geometry details nor the surface properties of the scene objects. Accessing the geometric details implies the rendering time complexity no longer be independent of scene complexity. Nevertheless, we can adopt the same analogy as BRDF to express how a pixel (instead of surface element) will look like as viewed from $\vec{V}$ when the scene behind the pixel window is illuminated by a light ray coming along $-\vec{L}$. This kind of reflectance can be regarded as the *image-based reflectance*. We call this BRDF the *apparent BRDF of pixel* (pBRDF in short). It is the aggregate reflectance of all surface elements that are visible through the pixel window.

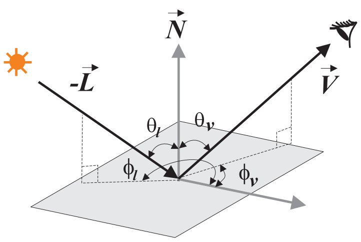


Fig. 1. BRDF.

Therefore, each pixel on the image plane is treated as an ordinary surface element and its apparent BRDF is measured and recorded. Figure 2 illustrates the idea in 2D cross section. Vectors $\vec{L}$ and $\vec{V}$ are the light and the viewing vectors respectively. Position $\dot{E}$ is the COP (eye). The apparent BRDF is a table indexed by vectors $\vec{V}$ and $\vec{L}$ (or the quadruple $(\theta_v, \phi_v, \theta_l, \phi_l)$).

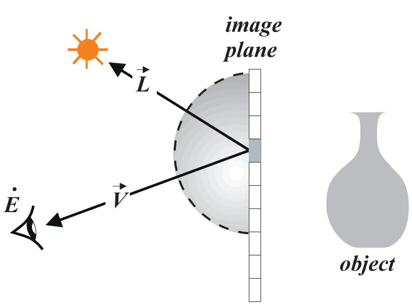


Fig. 2. Treating each pixel on the image plane as an ordinary surface element and measuring its apparent BRDF.

[1]The negative sign is used in order to be consistent with the convention that all vectors are denoted to be originated from the surface element.

The standard definition of BRDF of a real surface element is

$$\rho(\theta_v, \phi_v, \theta_l, \phi_l) = \frac{L_r(\dot{p}, \theta_v, \phi_v)}{L_r(\dot{p}, \theta_l, \phi_l)\cos\theta_l d\omega},$$

where ρ is the BRDF of the surface element,

(θ_v, ϕ_v) specifies the viewing direction, $\vec{V}$, in spherical coordinate,

(θ_l, ϕ_l) specifies the light vector, $\vec{L}$, in spherical coordinate,

$\dot{p}$ is the position of element,

$L_r(\dot{p}, \theta, \phi)$ is the radiance along the vector passing through $\dot{p}$ in the direction (θ, ϕ) (subscript r is used to avoid confusion with vector $\vec{L}$),

$d\omega$ is the differential solid angle.

It is the ratio between the radiance along the viewing direction and the radiance along the light vector weighted by the projected solid angle (the term $\cos\theta_l d\omega$). The projected solid angle is the area of differential solid angle after projecting onto the plane that contains the surface element. Since the differential solid angles may not be equal in size, the projected differential solid angle is mainly used to account for such unequal weight. In our first attempt [5] to include illumination information for the two-plane parameterized image-based representation [6], [7], we adopted this definition. It is also adopted by Yu and Malik [8] in capturing the reflectance of architectural models.

However, defining the pBRDF in this way may not be very useful. Since the pixel is not a true surface element but an imaginary window in the space, it does not physically reflect light energy. The actual reflection takes place at the real surfaces behind the pixel window. It is meaningless to scale the incident radiance by the projected differential sold angle (which is projected onto the image plane). Hence, we simply define the apparent BRDF of pixel as the ratio between the radiance along the viewing direction and the radiance along the light vector,

$$\rho_{\mathrm{pixel}}(\theta_v, \phi_v, \theta_l, \phi_l) = \frac{L_r(\dot{p}, \theta_v, \phi_v)}{L_r(\dot{p}, \theta_l, \phi_l)}.$$

This new definition simplifies the computation for relighting the images (described shortly) as the cosine term is dropped.

In the panorama application, the viewpoint is fixed in the space. Although the viewing vectors of different pixels in the panoramic image are different, they keep constant whenever the viewpoint is fixed. Hence we can drop the index (θ_v, ϕ_v) in the previous formulation,

$$\rho_{\mathrm{pixel}}(\theta_l, \phi_l) = \frac{L_r^{\dot{E}}(\dot{p})}{L_r(\dot{p}, \theta_l, \phi_l)}, \tag{1}$$

where $L_r^{\dot{E}}(\dot{p})$ is the radiance along the ray passing through the pixel window at $\dot{p}$ and arriving our eye fixed at $\dot{E}$.

In other words, the pBRDF of a pixel in the panoramic image is a spherical function which depends only on the direction of light vector.

III. Recording Illumination Information

To prepare the illumination information for panorama, we need to fill the two-dimensional spherical function with samples from the reference images. We have to decide what kind of light source should be used to illuminate the scene and how many images we need to take. The simplest way is to use a single directional light source as the illuminator because its light vector is constant for any point in the space. Note that the physical reflection is not taking place at the location of the pixel but at the surface elements behind the pixel window. Using a directional light source is more meaningful because the captured pixel value tells us what the surface elements behind the pixel window look like when all surface elements are illuminated by parallel light rays in the direction of $\vec{V}$.

Another issue is how to take samples (taking panoramic images). It is natural and simple to take sample with the light vector oriented on the spherical grid as shown in Figure 3.

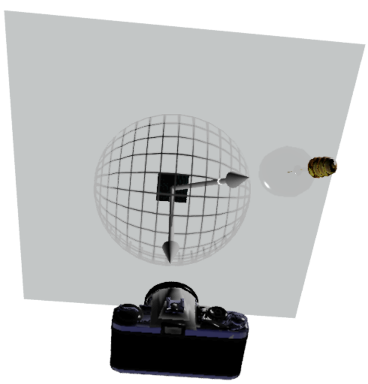
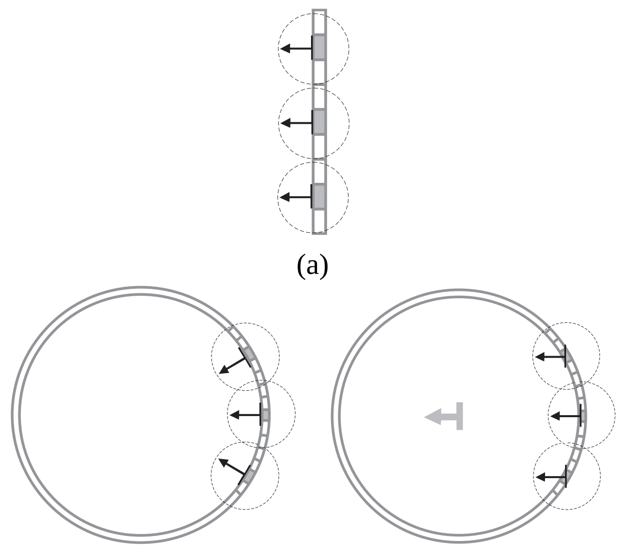


Fig. 3. Capturing the illumination characteristics by orienting the light vector on the spherical grid.

When measuring the pBRDF, it is convenient to specify the direction in a local coordinate system. Figure 4(a) shows that each pixel in a planar image is associated with a local coordinate system. Vectors have to be transformed to the local coordinate system during sampling and relighting. All coordinate systems are in the same orientation in this case. Moreover, the north poles of local frames are usually aligned with the normal of image plane. The major advantage with this orientation-aligned scheme is that, the light vector of a directional light source will always be transformed to the same local vector for any pixel. Hence only one transformation is needed.

In the case of panoramic image, projection manifold is not planar but spherical or cylindrical (Figure 4(b)). If the local coordinate system of each pixel is still aligned with the normal of projection manifold, the local coordinate systems will not be in the same orientation. In other words, different transformation of the light vector has to be done for different pixel. This is obviously inefficient. In fact, there is no need to align the local coordinate system with the normal of projection manifold. Instead, we align all local coordinate systems to a global frame as shown in the middle of Figure 4(c). Note that, unlike the real BRDF which describes reflectance over the hemisphere, we have to record the reflectance over the whole sphere as the real surface elements behind may not be parallel to the imaginary pixel window.



(a)

(b) (c)

Fig. 4. Alignment of local coordinate systems.

Therefore, for each lighting direction (each grid point (θ_l, ϕ_l) on the spherical coordinate system), we generate a panoramic image I_{θ_l,ϕ_l}. The sampling rate on the spherical grid depends on the rate of change of radiance on the illuminated scene which in turn depends on the surface properties and scene complexity. An empirical sampling rate is 70 $\times$ 140. The value $v_{\theta_l,\phi_l}(x,y)$ of pixel (x,y) in the image I_{θ_l,ϕ_l} is used to calculate pBRDF of pixel when the whole scene is illuminated by the light ray coming along (θ_l, ϕ_l) using the following equation (which is derived from Equation 1),

$$\rho_{\text{pixel}}(\theta_l, \phi_l) = \frac{v_{\theta_l,\phi_l}(x,y)}{L_r(\dot{p}, \theta_l, \phi_l)}$$

where v is the pixel value,

$L_r(\dot{p}, \theta_l, \phi_l)$ is the radiance emitted by the directional light source in the direction (θ_l, ϕ_l).

The above equation assumes the pixel value is linear to the radiance. This may not be a problem for synthetic images. For the case of real world images, a quantity which is linear to the radiance can be recovered from reference images using technique proposed by Debevec and Malik [9]. This quantity can then be used to replace the pixel value v in the above equation.

IV. Re-lighting

Once the pBRDFs are sampled and stored, they can be used for re-lighting. Using the property of superposition [10], the final radiance (or simply value) of each pixel can be computed. We proposed a local illumination model for image-based rendering (Equation 2) that makes use of superposition to re-light the image-based scenery under different lighting conditions.

$$\text{radiance through } p = \sum_{i}^{n} \rho_{\text{pixel}}(\theta_l^i, \phi_l^i) L_r(\dot{p}, \theta_l^i, \phi_l^i), \quad (2)$$

where n is the total number of light sources,

(θ_l^i, ϕ_l^i) specifies the direction, $\vec{L}_i$, of the i-th light source,

$\rho_{\text{pixel}}(\theta_l^i, \phi_l^i)$ is the reflectance of pixel (x, y) when the light vector is (θ_l^i, ϕ_l^i),

$L_r(\dot{p}, \theta_l^i, \phi_l^i)$ is the radiance along (θ_l^i, ϕ_l^i) due to the i-th light source,

$\dot{p}$ is the position of pixel (x, y).

Note the above illumination model is local in nature. That is, it only accounts for the direct radiance contribution from the light sources. No indirect radiance contribution is accounted for. One may say that the pixel is not a physical surface element but a window in space, how we can borrow the illumination model which models the reflection taken place on the real surface. This illumination model is *not* the result of borrowing the existing illumination model but is *a result of utilizing the image superposition properties*.

The pBRDF is actually an aggregate BRDF of all visible objects behind the pixel. We will show here that how this aggregate BRDF gives us the correct image. Consider k unoccluded objects [2] which are visible through the pixel window and are illuminated by n light sources. The radiance along $\vec{V}$ is,

$$\begin{aligned} & \sum_{i}^{n} \rho_i^1 L_r^i + \sum_{i}^{n} \rho_i^2 L_r^i + \cdots + \sum_{i}^{n} \rho_i^k L_r^i \\ = & \sum_{j}^{k} \rho_1^j L_r^1 + \sum_{j}^{k} \rho_2^j L_r^2 + \cdots + \sum_{j}^{k} \rho_n^j L_r^n \\ = & \rho_1 L_r^1 + \rho_2 L_r^2 + \cdots + \rho_n L_r^n \end{aligned}$$

where ρ_i^j is the reflectance of the j-th object surface when illuminated by the i-th light source,

L_r^i is the short hand of $L_r(\dot{p}, \theta_l^i, \phi_l^i)$, the radiance due to the i-th light source,

$\rho_i = \sum_{j=1}^{k} \rho_i^j$ is the aggregate reflectance of k objects. It is the pBRDF we have recorded.

The first row shows the sum of reflected radiances from all k unoccluded objects. Due to the linearity of illumination, we can reorder the terms to derive the result on the third row which is the sum of multiplications of aggregate reflectance and the radiance emitted by each light source.

A. Lighting Direction

With Equation 2, the lighting direction can be modified by substituting a different value of (θ_l, ϕ_l). Figures 5(a) and (b) show the perspective snapshots of a panorama (attic scene in Figure 12) by a directional light in two different directions. No geometric model is required during the re-lighting.

B. Light Intensity

Another parameter to manipulate in Equation 2 is the intensity of light source. This can be done by assigning the value of L_r^i for the i-th light source. Figure 6(a) shows another perspective snapshot of the same panorama illuminated by a blue spotlight.

C. Multiple Light Sources

Due to the linearity of light transport, arbitrary number of light sources can be included for illumination. The trade-off is the computational time. An additional multiplication and addition have to be computed in evaluating Equation 2 for each extra light source. In Figure 6(b), the same panorama is illuminated by the blue and the yellow spotlights simultaneously. Note that although the reference images are captured under the illumination of a single

[2] If there exist objects that occlude each other, we can always subdivide the objects into visible (unoccluded) portions and invisible (occluded) portions. Invisible portions will never contribute any radiance to the final image. Hence we consider only those unoccluded objects without loss of generality.

(a)

(b)

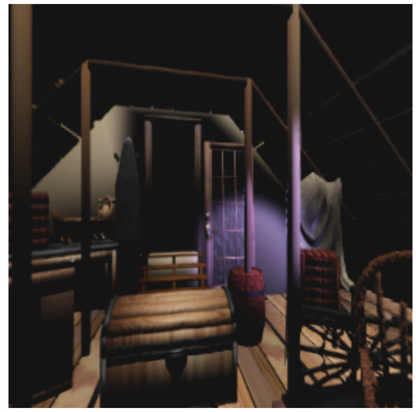
Fig. 5. Change of lighting direction.

directional light source, the re-lighted images can be illuminated with multiple light sources and each with different color.

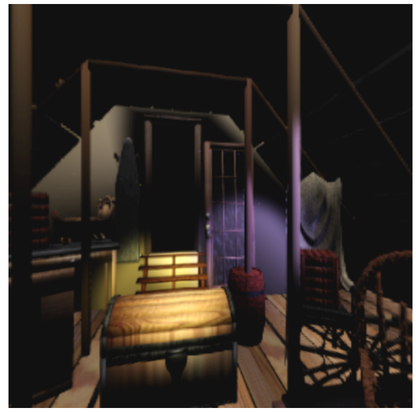
D. *Type of Light Sources*

It seems that the desired light source for re-lighting must be directional since the reference images are captured under the illumination of a directional illuminator. Re-lighting with directional light source is very efficient for evaluating Equation 2 because all pixels in the same image are illuminated by the same light vector (θ_l^i, ϕ_l^i). Only one transformation is needed to transform the light vector to the local frame of pixel. Moreover, no geometry information is required for re-lighting as all reference images are captured under a directional light source.

However, re-lighting is not restricted to directional light. It can be extended to point source, spotlight or more general area light source as well. It will be more expensive to evaluate Equation 2 for other types of light sources, as the light vector (θ_l^i, ϕ_l^i) has to be determined from pixel to pixel. Since the image plane where the pixels are located is only a window in 3D space (Figure 7), the surface element that actually reflects the light may be located on any



(a)



(b)

Fig. 6. Multiple light sources. (a)Top: Attic scene illuminated by a blue light. (b)Bottom: One more yellow spotlight is added.

point along the ray $\vec{V}$ as shown in Figure 7. To determine the light vector $\vec{L}$ correctly for other types of light source, the intersection point of the ray and the object has to be located first. There is no such problem in the case of directional source, since the light vectors are identical for any point in 3D space. One way to determine $\vec{L}$ is to use the depth image. While this can be easily done for synthetic scenes, real world scenes may be more difficult. Using a range scanner or computer vision techniques may provide a solution.

With the additional depth map, we can correctly re-light the scene with any type of light source by finding the correct light vector $\vec{L}$ using the following equation,

$$\vec{L} = \dot{S} - \dot{E} + \frac{\vec{V}}{|\vec{V}|} d \qquad (3)$$

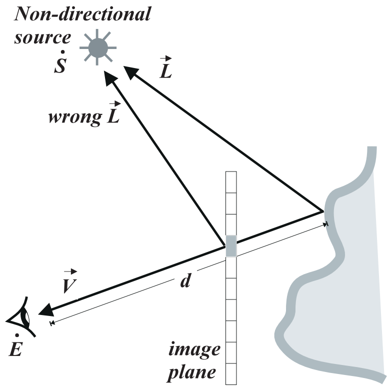


Fig. 7. Finding the correct light vector.

where $\vec{L}$ is the light vector,
$\dot{S}$ is the position of non-directional light source,
$\dot{E}$ is the position of eye,
$\vec{V}$ is the viewing direction, and $\vec{V} = \dot{E} - \dot{p}$
d is the depth value.

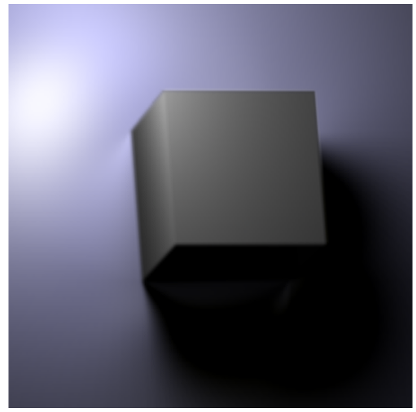
The illuminator can be a point source, a spotlight or even a slide projector source. Figure 8 shows an image-based scene containing a box lying on the plane. The scene is illuminated by a point source (Figure 8(a)) and a directional source (Figure 8(b)). Note that all input reference images are recorded with a single directional light source. Surprisingly, with this extra depth information, we can re-light the image-based scene with other types of light sources. Note that even a depth map is needed, the re-lighting is still independent of scene complexity because the depth map is also an image with the same resolution as the panorama.

The re-lighted image may contain shadow if the shadow is recorded in the pBRDFs during sampling. Note the difference in the shadow cast by different sources in Figure 8. Figure 9(a) demonstrates the result of re-lighting the attic scene with a spotlight. Note how the two pillars are correctly illuminated. Figure 9(b) shows the result of casting a slide (Figure 9(c)) onto the same scene. Theoretically, we can even illuminate the scene with area light source by trading off the computational time.

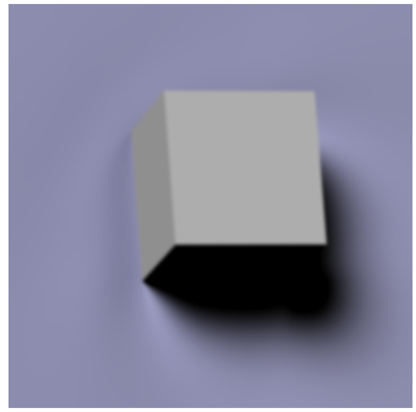
V. Implementation and Results

A. Implementation Details

To demonstrate the feasibility and practicability of the proposed image representation, we have implemented an interactive panorama viewer on both PC and UNIX platforms. Figure 10 shows the user interface of the viewer (SGI version). The lighting control panel on the left allows the user to control various parameters of illumination. The re-lighting result will then be shown through the right



(a)



(b)

Fig. 8. Point and directional light sources. (a)Top: shadow cast by a point source. (b)Bottom: shadow cast by a directional source.

window.

In our implementation, we use cylindrical panorama instead of spherical panorama to avoid the excessive sampling at the north and south poles. Since the proposed image representation is independent of projection manifold, using cylindrical panorama is indifferent from using spherical panorama. To display the cylindrical panorama, we first map the re-lighted panorama onto the surface of a cylinder. The texture-mapped cylinder is then drawn by hardware graphics accelerator. Hence, panning and zooming can be done in real time if graphics hardware is installed.

When the user modifies the lighting parameters, pixel values are re-computed using Equation 2. This computation is done purely by software.

Since at any given instance, only part of the panorama is visible through the display window, we tends to postpone the re-lighting of whole panorama. When the user changes the illumination, only the visible portion of the panorama is re-lighted as visualized in the unfolded panorama in Figure 11. The invisible portion will be re-lighted only when the user finishes the modification of lighting parameters.

We called this approach *lazy re-lighting*.

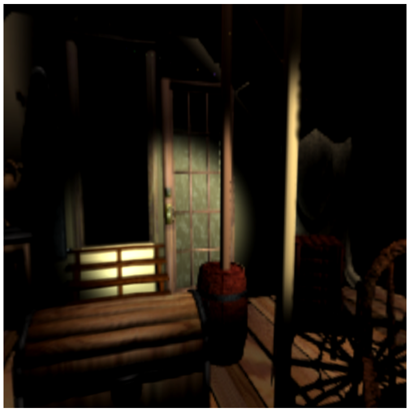
B. *Timing Statistics*

Table I shows the timing statistics of re-lighting on both PC and SGI platforms. Two complex scenes, attic (Figures 12 - 13) and city (Figures 14 - 15), are tested. Consider the attic scene, it contains more than 500k triangles and requires more than 2 minutes to render using Alias|Wavefront on SGI Octane with MIPS 10000 CPU. Using the image-based approach the scene can be re-lighted interactively on the same machine, even though the re-lighting is done purely by software. The column **Render** shows the rendering time of both scenes using traditional geometry-based renderer, Alias|Wavefront, on SGI Octane with MIPS 10000 CPU. Column **Re-light (PC)** shows the average re-lighting time (of the whole panorama) for the same scenes on Pentium III 667MHz. Similarly, column **Re-light (SGI)** shows the average re-lighting time (of the whole panorama) on SGI Octane. Since the panning, tilting and zooming can be done by hardware, their timing statistics are not included in the table. For a fair comparison, all testing (both geometry-based rendering and image-based re-lighting) are done by illuminating the scenes with a single directional light source. Since both image-based scenes are with the resolution of 1024×256, their time for re-lighting are the same.

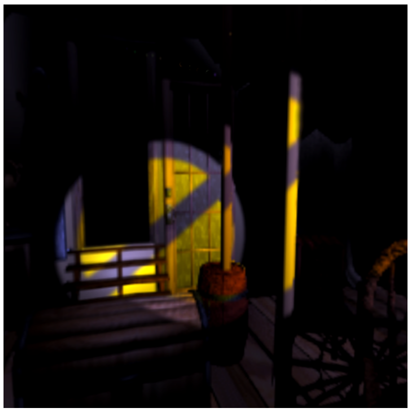
Scene	Render(SGI)	Re-light (PC)	Re-light (SGI)
Attic	133 sec	0.661 sec	1.27 sec
City	337 sec	0.661 sec	1.27 sec

TABLE I

TIMING STATISTICS.



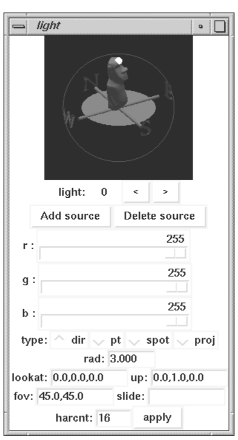
(a)



(b)

(c)

Fig. 9. Spotlight and slide projector sources. (a)Top: Scene illuminated by a spotlight. (b)Bottom left: Same scene illuminated by a slide projector source. (c)Bottom right: The slide used for slide projection.



(a)

(b)

Fig. 10. A panoramic viewer with controllable illumination.

The re-lighting results of these two data sets are shown in Figures 12 through 15. Figures 12(a) and 13(a) show the unfolded cylindrical panoramas of the attic scene re-lighted by a directional light source and spotlights respectively. The bottom three images are the perspective snapshots of panorama, they are obtained by warping the panorama. Note how the spotlights correctly illuminate the pillars and chair in the image-based attic. Figures 14 and 15 show the results of re-lighting the image-based city scene. Both images are illuminated by a directional light source. Figure 14 mimics the natural illumination of clear skylight while Figure 15 mimics the illumination during sunset. In the caption of each figure, the time for re-lighting the whole panorama (on Pentium III) is listed for reference. No depth map is needed in the synthesis of Figures 14 and 15 since only directional light sources are used.

C. Compression

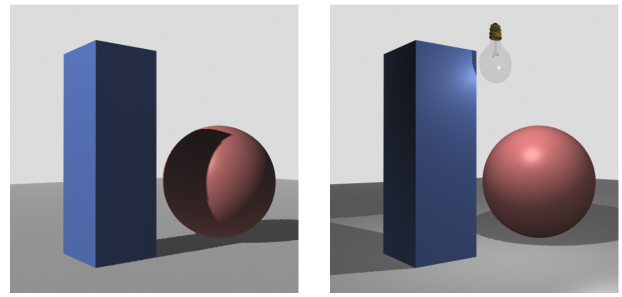
To reduce the data storage of image data, we compress the pBRDF using spherical harmonics [11] which is a common approach [12] to compress BRDF. For each pixel, we transform its spherical reflectance table to a coefficient vector. The higher the dimension of vector is, the more accurate the representation is. Normally, 25 coefficients are sufficient to represent the reflectance table. With spherical harmonics, the compression ratio is normally about 400 to 1 (assuming the sampling rate is 70×140). The actual compression ratio depends on the sampling rate of lighting direction and surface properties. During re-lighting, the reflectances have to be reconstructed from the spherical harmonic coefficients by calculating a summation of multiplications [5]. Hence the major computation time of the software re-lighting is spent on this reconstruction.

Further reduction can be achieved by exploiting the data correlation between adjacent pixels. It has been found that the coefficient vectors of neighboring pixels are very similar in values. By applying vector quantization on the spherical harmonic coefficient vectors, the overall compression ratio can be further improved to about 1800 to 1 (assuming the sampling rate is 70×140). A 1024×256 cylindrical panoramic image with illumination information incorporated requires about 3-4 MB of storage only, which is reasonable for current computer storage and network bandwidth.

VI. Limitations

One limitation of the proposed representation is that global illumination may not be correctly accounted. The re-lighting equation (Equation 2) assumes the only contribution to the reflected radiance through a pixel window is due to the directional light source. If part of the reflected radiance is due to interreflection, it is not possible to figure out as we don't have the detail scene geometry. Hence, global illumination cannot be correctly simulated due to the lack of geometry.

Another limitation is shadowing. Although the representation can correctly address shadowing due to directional light sources, shadowing may not be correctly accounted when non-directional light source is used. Figure 16(a) illustrates the perspective snapshot of a scene from the fixed viewpoint of a panoramic image. Shadow is cast by the rectangular box onto the sphere during scene capture as the directional light source is infinitely far away. During re-lighting, the result will be incorrect if a point light source is positioned in between the two objects. Our approach will *not* generate the correct image as in Figure 16(b). Instead, there will be incorrect shadow on the sphere. This is because the re-lighting algorithm assumes that there is *no occlusion* between the light source and the illuminated surface that is visible through the pixel window. If occlusion exists like the case in Figure 16(a), assumption is violated and re-lighting is hence incorrect.



(a) (b)

Fig. 16. When there is occlusion (between the directional light source and the surface), re-lighting for non-directional light source may be incorrect.

VII. Conclusions and Future Direction

In this article, we propose a panoramic image representation that allows us to incorporate the illumination information into panoramic images. The re-lighting algorithm depends solely on the image resolution and is independent of scene complexity. Based on this representation, we developed an interactive panorama viewer which not just allows panning, tilting and zooming, but also allows the modification of lighting condition. The introduction of illumination control enhances the 3D illusion while preserving its independency of scene complexity. One future direction is to generalize the representation to generalized panorama, concentric mosaics [13], which allows the viewpoint to move within a circular region.

Web Availability

Demonstrative movies and prototype viewer are available through the following web page:

`http://www.cse.cuhk.edu.hk/~ttwong/papers/pano/pano.html`

Acknowledgements

We would like to thank the anonymous reviewers for their review and constructive comments. This project is supported by the Research Grants Council of the Hong Kong Special Administrative Region, under RGC Earmarked Grants (Project No. CUHK 4186/00E) and RGC Co-operative Research Centres (CRC) Scheme (Project No. CRC 4/98).

REFERENCES

[1] Shenchang Eric Chen, "QuickTime VR - an image-based approach to virtual environment navigation," in Computer Graphics *Proceedings, Annual Conference Series, SIGGRAPH'95*, August 1995, pp. 29–38.

[2] Jeffry S. Nimeroff, Eero Simoncelli, and Julie Dorsey, "Efficient re-rendering of naturally illuminated environments," in *Fifth Eurographics Workshop on Rendering*, Darmstadt, Germany, June 1994, pp. 359–373.

[3] Peter N. Belhumeur and David J. Kriegman, "What is the set of images of an object under all possible lighting conditions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1996.

[4] G. Golub and C. van Loan, *Matrix Computations*, The John Hopkins University Press, 1989.

[5] Tien-Tsin Wong, Pheng-Ann Heng, Siu-Hang Or, and Wai-Yin Ng, "Image-based rendering with controllable illumination," in *Eighth Eurographics Workshop on Rendering*, Saint Etienne, France, June 1997, pp. 13–22.

[6] Marc Levoy and Pat Hanrahan, "Light field rendering," in Computer Graphics *Proceedings, Annual Conference Series, SIGGRAPH'96*, August 1996, pp. 31–42.

[7] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen, "The lumigraph," in Computer Graphics *Proceedings, Annual Conference Series, SIGGRAPH'96*, August 1996, pp. 43–54.

[8] Yizhou Yu and Jitendra Malik, "Recovering photometric properties of architectural scenes from photographs," in *SIGGRAPH 98 Conference Proceedings*. ACM SIGGRAPH, July 1998, Annual Conference Series.

[9] Paul E. Debevec and Jitendra Malik, "Recovering high dynamic range radiance maps from photographs," in Computer Graphics *Proceedings, Annual Conference Series, SIGGRAPH'97*, August 1997, pp. 369–378.

[10] I. W. Busbridge, *The Mathematics of Radiative Transfer*, Cambridge University Press, 1960.

[11] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Interscience Publisher, Inc., New York, 1953.

[12] Francois X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg, "A global illumination solution for general reflectance distributions," in *Computer Graphics (SIGGRAPH '91 Proceedings)*, July 1991, vol. 25, pp. 187–196.

[13] Heung-Yeung Shum and Li-Wei He, "Rendering with concentric mosaics," in Computer Graphics *Proceedings, Annual Conference Series, SIGGRAPH'99*, August 1999, pp. 299–306.

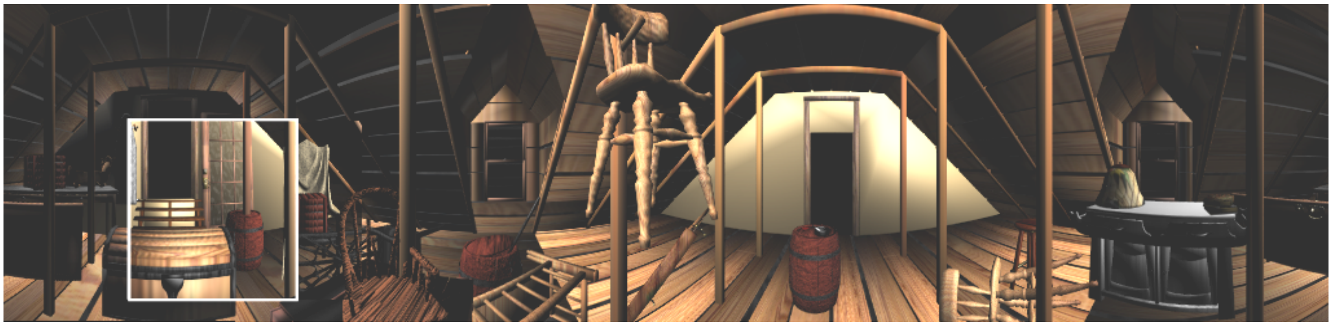
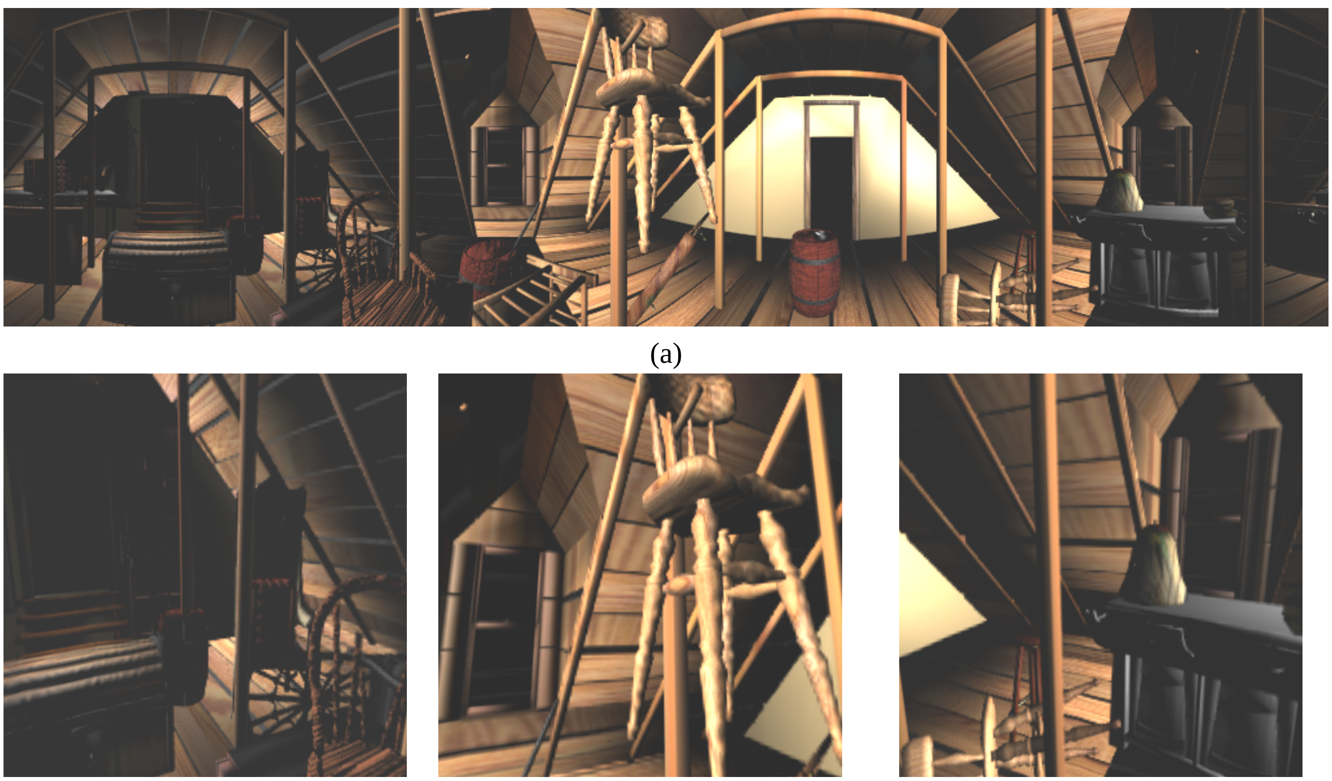


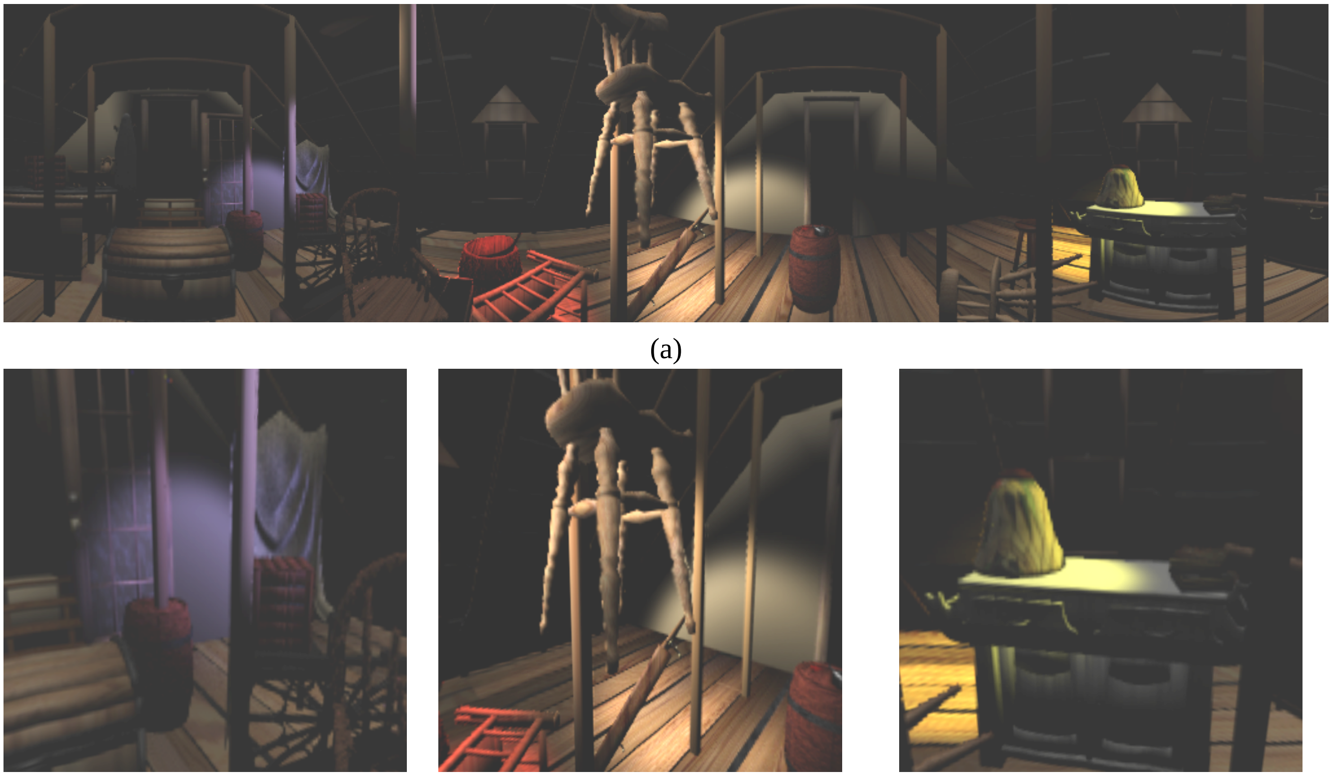
Fig. 11. Lazy re-lighting.



(a)

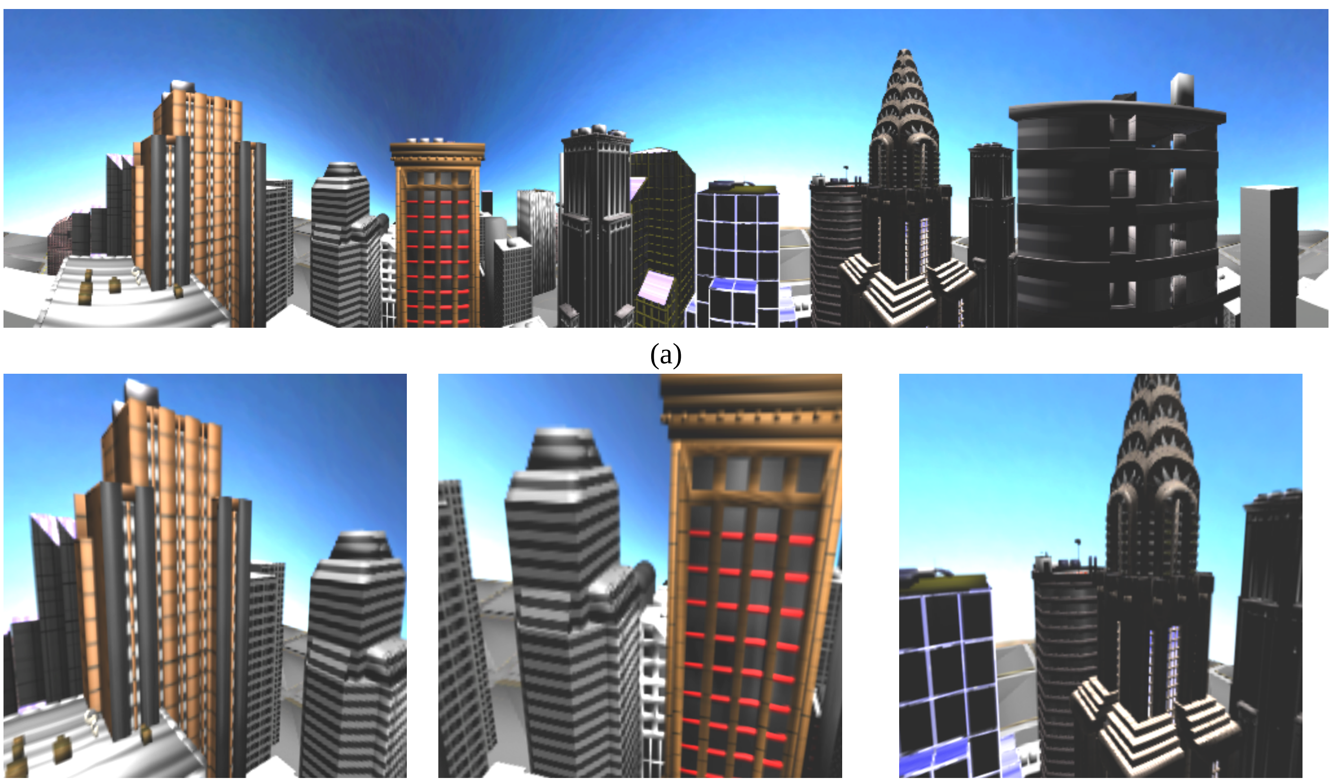
(b) (c) (d)

Fig. 12. Attic. Time for re-lighting: 0.661 sec.



(a) (b) (c) (d)

Fig. 13. Attic illuminated by five spotlights. Time for re-lighting: 10.305 sec.



(a) (b) (c) (d)

Fig. 14. City scene under clear skylight. Time for re-lighting: 0.661 sec.

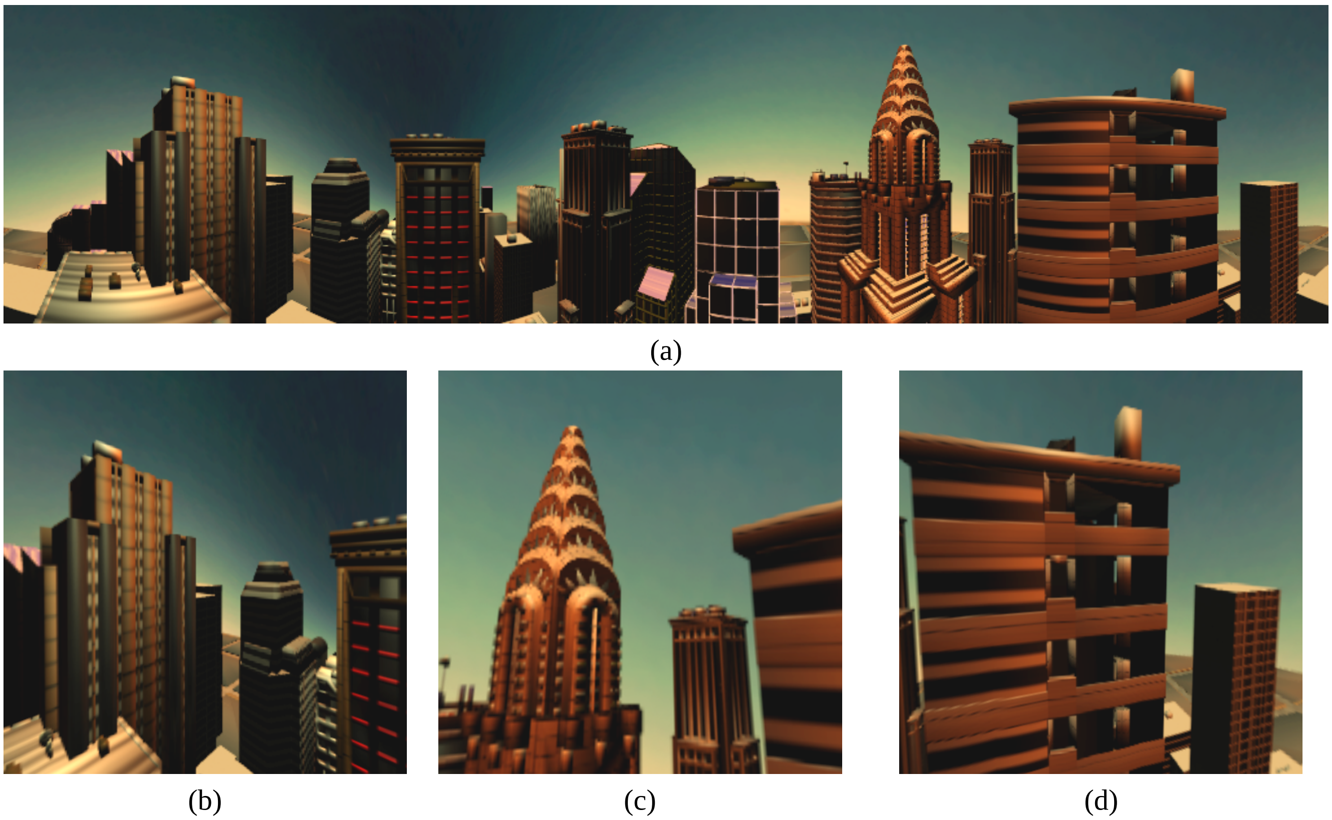


Fig. 15. Sunset city. Time for re-lighting: 0.661 sec.