Spatiotemporal Sensitivity and Visual Attention for Efficient Rendering of Dynamic Environments

HECTOR YEE, SUMANTA PATTANAIK and DONALD P. GREENBERG

Program of Computer Graphics, Cornell University

We present a method to accelerate global illumination computation in pre-rendered animations by taking advantage of limitations of the human visual system. A spatiotemporal error tolerance map, constructed from psychophysical data based on velocity dependent contrast sensitivity, is used to accelerate rendering. The error map is augmented by a model of visual attention in order to account for the tracking behavior of the eye. Perceptual acceleration combined with good sampling protocols provide a global illumination solution feasible for use in animation. Results indicate an order of magnitude improvement in computational speed.

Keywords: Animation, Computer Vision, Human Visual Perception, Illumination, Monte Carlo Techniques

1 INTRODUCTION

Global illumination is the physically accurate calculation of lighting in an environment. It is computationally expensive for static environments and even more so for dynamic environments. Not only are many images required for an animation, but the calculation involved increases with the presence of moving objects. In static environments, global illumination algorithms can precompute a lighting solution and reuse it whenever the viewpoint changes, but in dynamic environments, any moving object or light potentially affects the illumination of every other object in a scene. To guarantee accuracy, the algorithm has to recompute the entire lighting solution for each frame. This paper describes a perceptually-based technique that can dramatically reduce this computational load. The technique may also be used in image based rendering, geometry level of detail selection, realistic image synthesis, video telephony and video compression.

Reference Image (a)

Spatiotemporal error tolerance map (Aleph Map) (b)

Figure 1: Global Illumination of a Dynamic Environment (*see color plate*). Global illumination correctly simulates effects such as color bleeding (the green of the leaves on to the petals), motion blur (the pink flamingo), caustics (the reflection of the light by the golden ash tray on the wall), soft shadows, anti-aliasing, and area light sources (a). This expensive operation benefits greatly from our perceptual technique, which can be applied to animation as well as motion-blurred still images such as shown above. The spatiotemporal error tolerance map (which we call the Aleph Map) is shown on the right (b). Bright areas on the map indicate areas where less effort should be spent in computing the lighting solution. The map takes a few seconds to compute but will save many hours of calculation.

Perceptually-based rendering operates by applying models of the human visual system to images in order to determine the stopping condition for rendering. In doing so, perceptually assisted renderers attempt to expend the least amount of work to obtain an image that is perceptually indistinguishable from a fully converged solution. The technique described in this paper assists rendering algorithms by producing a spatiotemporal error tolerance map (Aleph Map) that can be used as a guide to optimize rendering. Figure 1 shows a scene containing moving objects (a) and its Aleph Map (b). The brighter areas in the map show regions where sensitivity to errors is low, permitting shortcuts in computation in those areas.

Two psychophysical concepts are harnessed in this paper: spatiotemporal sensitivity and visual attention. The former tells us how much error we can tolerate and the latter expresses where we look. Knowledge of error sensitivity is important because it allows us to save on computation in areas where the eye is less sensitive and visual attention is important because it allows us to use sensitivity information wisely. Areas where attention is focused must be rendered more accurately than less important regions.

Spatiotemporal sensitivity considers the reduced sensitivity of the human visual system to moving spatial patterns. This limitation of the human visual system makes us less sensitive to errors in regions where there are high spatial frequency patterns and movement. Movement is caused by the observer in motion or objects in motion. We exploit this reduced sensitivity to speed up the computation of global illumination in dynamic environments. This principle of reduced sensitivity cannot be applied naively, however, since the eye has an excellent ability to track objects in motion. The eye reduces the velocity of the objects/areas of interest with respect to the retina, nullifying the loss of sensitivity due to motion. By using a robust model of visual attention, we predict where viewers direct their attention, allowing us to accurately derive the viewer's spatiotemporal sensitivity to the scene.

The Aleph Map represents spatiotemporal error tolerance. Figure 2 shows an outline of our technique. To obtain the Aleph Map, we require

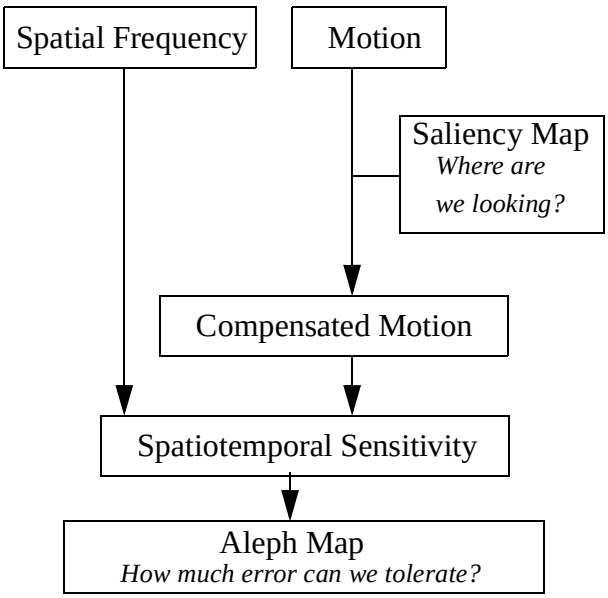


Figure 2: Flowchart outlining the computation of spatiotemporal error tolerance. The Aleph Map is a perceptual oracle derived from the spatial frequency, motion and visually important information in a scene. The saliency map is a measure of visual attention and is used to compensate for eye tracking movements in order to fully take advantage of perceptual sensitivity in dynamic scenes.

knowledge about the motion and spatial frequencies present in the scene. We also need to factor in visual attention, which tells us areas of importance in the scene. Image regions that receive visual attention are estimated by a saliency map. The saliency map is used to account for the tracking behavior of the eye in order to correctly compensate for eye motion before spatiotemporal sensitivity is calculated. It is built up from conspicuity associated with intensity, color, orientation changes and motion. One may think of conspicuity as the visual attractor due to a single channel such as motion and saliency as the visual attractor due to all the stimuli combined. The saliency map tells us where the eye is paying attention and the Aleph Map uses that information to tell us how much error we can tolerate in that region. The saliency map allows us to compensate for eye movements without the use of eye tracking devices. Although eye tracking hardware exists, such hardware is specialized and would be impractical for multiple viewers. Our technique yields significant gains in efficiency without incurring the costs and disadvantages of such hardware. Figure 3 shows speedup achieved by using our technique in global illumination computation.

In Section 2, we will discuss the previous work on which our algorithm is based. Section 3 discusses the advantages of our technique. In Section 4, we review current ideas about spatial sensitivity, spatiotemporal sensitivity, eye tracking, eye movements and visual attention. Section 5 covers the implementation details. We demonstrate the usefulness of our algorithms with a practical augmentation of the popular lighting simulator RADIANCE in Section 6 and present our conclusions in Section 7.

2 PREVIOUS WORK

Gibson and Hubbold [10] applied perceptual techniques to compute view independent global illumination by using a tone reproduction operator to guide the progress of a radiosity algorithm. Their approach differs from ours as we will be focusing on view dependent algorithms. Most view dependent perceptual techniques that are used to speed up rendering involve the use of a perceptual metric to inform the renderer

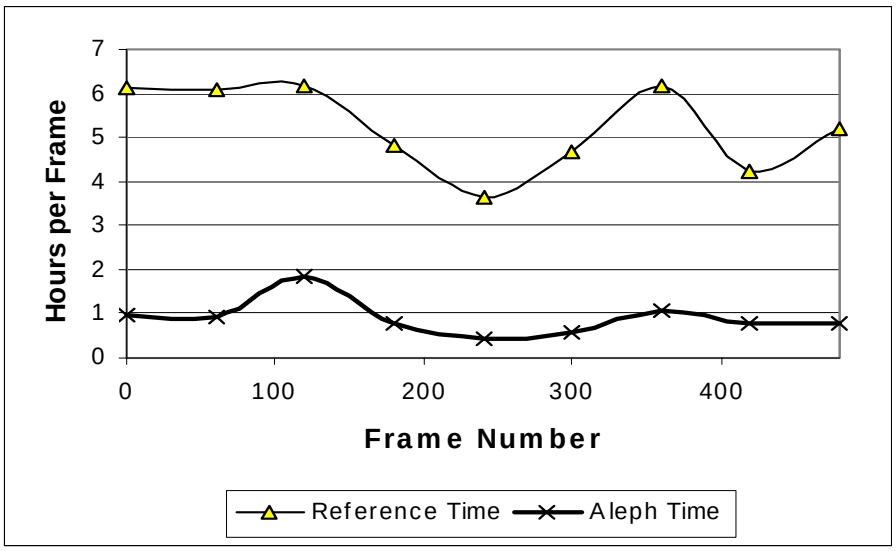


Figure 3: Timing comparison between a reference lighting solution of a complex environment generated using the irradiance caching technique and our Aleph Map enhanced irradiance cache. Interestingly, the time taken for the perceptual solution remains relatively flat, perhaps because as scene complexity increases, the tolerance for error also increases. The time for the perceptual solution includes the time for computing the Aleph Map, which is small. The irradiance cache is used in the lighting simulator RADIANCE. Calculations were done on a quad processor 500 Mhz Intel Pentium III machine.

to stop calculating well before the physical convergence is achieved; that is, whenever the rendered image is perceptually indistinguishable from a fully converged solution.

Bolin and Meyer [3], Meyer and Liu [22], and Myszkowski [23] relied on the use of sophisticated perceptual metrics to estimate perceptual differences between two images to determine the perceived quality at an intermediate stage of a lighting computation. Based on perceptual quality, they determined the perceptual convergence of the solution and used it as a stopping condition in their global illumination algorithm. These metrics perform signal processing on the two images to be compared, mimicking the response of the human visual system to spatial frequency patterns and calculating a perceptual distance between the two images. Myskowski uses the Daly Visible Differences Predictor [6] to determine the stopping condition of rendering by comparing two images at different stages of the lighting solution. Bolin and Meyer used a computationally efficient and simplified variant of the Sarnoff Visual Discrimination Model [20] on an upper bound and a lower bound pair of images, resulting in a bounded-error, perceptually-guided algorithm. Both algorithms required repeated applications of the perceptual error metric at intermediate stages of a lighting solution, adding substantial overhead to the rendering algorithm.

Ramasubramanian, et al., [26] reduced the cost of such metrics by decoupling the expensive spatial frequency component evaluation from the perceptual metric computation. They reasoned that the spatial frequency content of the scene does not change significantly during the global illumination computation step, and precomputed this information from a cheaper estimate of the scene image. They reused the spatial frequency information during the evaluation of the perceptual metric without having to recalculate it at every iteration of the global illumination computation. They carried out this precomputation from the direct illumination solution of the scene. Their technique does not take into account any sensitivity loss due to motion and is not well suited for use in dynamic environments. Furthermore, direct illumination evaluation is often expensive, especially when area light sources are present in a scene, and hence is not always suitable for precomputation.

Myskowski, et al., [24] addressed the perceptual issues relevant to rendering dynamic environments. They incorporated spatiotemporal sensitivity of the human visual system into the Daly VDP [6] to create a perceptually-based Animation Quality Metric (AQM) and used it in conjunction with image-based rendering techniques [21] to accelerate the rendering of a key-frame based animation sequence. Myskowski's framework assumed that the eye tracks all objects in a scene. The tracking ability of the eye is very important in the consideration of spatiotemporal sensitivity [7]. Perceptually-based rendering algorithms which ignore this ability of the eye can introduce perceptible error in visually salient areas of the scene. On the other hand, the most conservative approach of indiscriminate tracking of all the objects of a scene, as taken by Myskowski's algorithm, effectively reduces a dynamic scene to a static scene, thus reducing the benefits of spatiotemporally-based perceptual acceleration. The use of AQM during global illumination computation will also add substantial overhead to the rendering process.

3 OUR APPROACH

Our technique improves on existing algorithms by including not only spatial information but temporal as well. The scene's spatiotemporal error tolerances, held in an Aleph Map, are quickly precomputed from frame estimates of the animation that capture spatial frequency and

motion correctly. We make use of fast graphics hardware to obtain the Aleph Map quickly and efficiently. The map is better because it incorporates a model of visual attention in order to include effects due to ability of the visual system to locate regions of interest.

The Aleph Map can be adapted for use as a perceptually-based physical error metric, or as in our application, as an oracle that guides perceptual rendering without the use of an expensive comparison operator. By using a perceptual oracle instead of a metric, we incur negligible overhead while rendering. The next section introduces the background information required to understand the construction of the Aleph Map.

4 BACKGROUND

This section covers the background relevant to this paper. The first part reviews the spatiotemporal sensitivity of the human visual system and the second part addresses the attention mechanism of the visual system. For an in-depth discussion of perception in general, we refer readers to "Foundations of Vision" by Wandell [30].

4.1 Spatiotemporal Contrast Sensitivity

4.1.1 Contrast Sensitivity

The sensitivity of the human visual system changes with the spatial frequency content of the viewing scene. This sensitivity is psychophysically derived by measuring the threshold contrast for viewing sine wave gratings at various frequencies [5]. A sine wave grating is shown to viewers who are then asked if they can distinguish the grating from a background. The minimum contrast at which they can distinguish the grating from the background is the threshold contrast. The Contrast Sensitivity Function (CSF) is the inverse of this measured threshold contrast, and is a measure of the sensitivity of the human visual system towards static spatial frequency patterns. This CSF function peaks between 4-5 cycles per degree (cpd) and falls rapidly at higher frequencies. The reduced sensitivity of the human visual system to high frequency patterns allows the visual system to tolerate greater error in high frequency areas of rendered scenes and has been exploited extensively [2][3][23][24][26] in the rendering of scenes containing areas of high frequency texture patterns and geometric complexity.

4.1.2 Temporal Effects

The human visual system varies in sensitivity not only with spatial frequency but also with motion. Kelly [17] has studied this effect by measuring threshold contrast for viewing travelling sine waves. Kelly's experiment used a special technique to stabilize the retinal image during measurements and therefore his models use the retinal velocity, the velocity of the target stimulus with respect to the retina. Figure 4 summarizes these measurements.

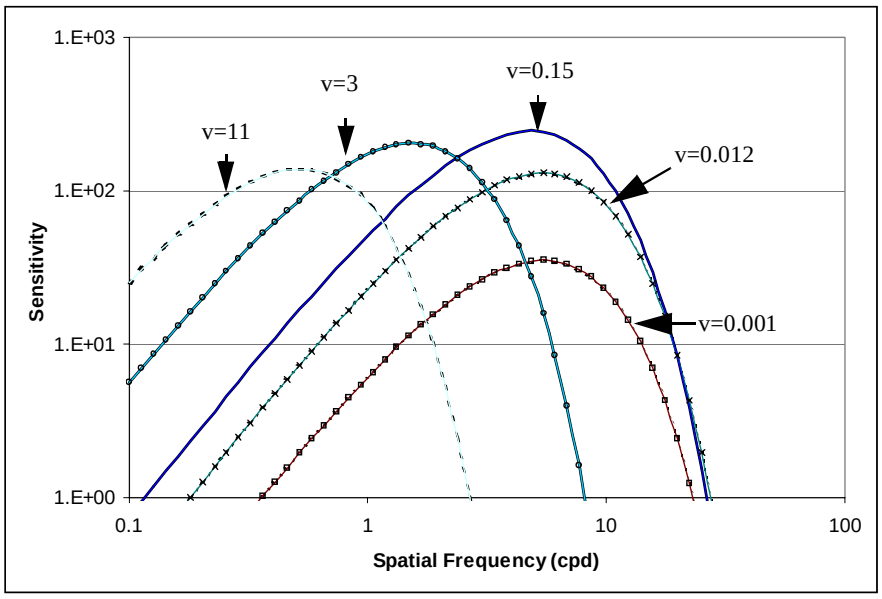


Figure 4: Velocity dependent CSF, plotted from an equation empirically derived from Kelly's sensitivity measurements [7]. The velocities v are measured in degrees/second.

From Figure 4, we can see that the contrast sensitivity changes significantly with the retinal velocity. Above the retinal velocity of 0.15 deg/sec, the peak sensitivity drops and the entire curve shifts to the left. This shift implies that waveforms of higher frequency become increasingly difficult to discern as the velocity increases. At retinal velocities below 0.15 deg/sec the whole sensitivity curve drops significantly.

Speeds below 0.15 deg/sec are artificial as the eye naturally moves about slightly even when it is in a steady fixed stare. The measurements also showed that the sensitivity function obtained at the retinal velocity of 0.15 deg/sec matched with the static CSF function described earlier. This agrees with the fact that the drift velocity of a fixated eye is about 0.15 deg/sec, and must be taken into account when using Kelly's measurement results in real world applications.

4.1.3 Eye Movements

The loss of sensitivity to high frequency spatial patterns in motion gives an opportunity to extend existing perceptually-based rendering techniques from static environments to dynamic environments. The eye, however, is able to track objects in motion to keep objects of interest in the foveal region where spatial sensitivity is at its highest. This tracking capability of the eye, also known as smooth pursuit, reduces the retinal velocity of the tracked objects and thus compensates for the loss of sensitivity due to motion.

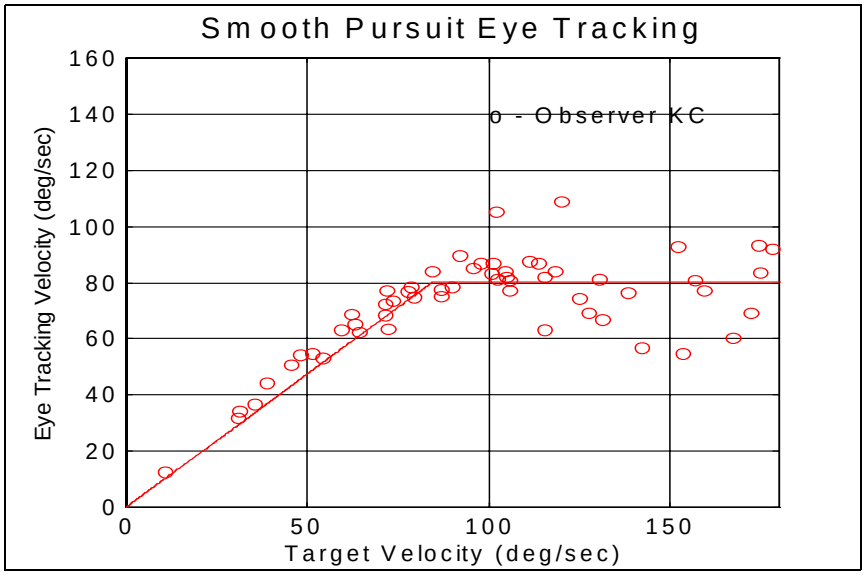


Figure 5: Smooth pursuit behavior of the eye. The eye can track targets reliably up to a speed of 80.0 deg/sec beyond which tracking is erratic. Reproduced from Daly [7].

Measurements by Daly [7] have shown that the eye can track targets cleanly at speeds up to 80 deg/sec. Beyond this speed, the eye is no longer able to track perfectly. The results of such measurements are shown in Figure 5. The open circles in Figure 5 show the velocity of the eye of an observer in a target tracking experiment. The measured tracking velocity is on the vertical axis while the actual target velocity is on the horizontal axis. The solid line in Figure 5 represents a model of the eye's smooth pursuit motion.

Evidently, it is crucial that we compensate for smooth pursuit movements of the eye when calculating spatiotemporal sensitivity. The following equation describes a motion compensation heuristic proposed by Daly [7]:

$$v_R = v_I - min(0.82v_I + v_{Min}, v_{Max}) \quad (1)$$

where v_R is the compensated retinal velocity, v_I is the physical velocity, v_{Min} is 0.15 deg/sec (the drift velocity of the eye), v_{Max} is 80 deg/sec (which is the maximum velocity that the eye can track efficiently). The value 0.82 accounts for Daly's data fitting that indicates the eye tracks all objects in the visual field with an efficiency of 82%. The solid line in Figure 5 was constructed using this fit. Use of this heuristic would imply only a marginal improvement of efficiency in extending perceptual rendering algorithms for dynamic environments, but our method offers an order of magnitude improvement.

4.2 Visual Attention and Saliency

Though the eye's smooth pursuit behavior can compensate for the motion of the moving objects in its focus of attention, not every moving object in the world is the object of one's attention. The pioneering work of Yarbus [36] shows that even under static viewing conditions not every object in the viewing field captures visual attention. If we can predict the focus of attention, then other less important areas may have much larger error tolerances, allowing us to save calculation time on those areas. To accomplish this, we need a model of visual attention which will correctly identify the possible areas of visual interest.

Visual attention is the process of selecting a portion of the available visual information for localization, identification and understanding of objects in an environment. It allows the visual system to process visual input preferentially by shifting attention about an image, giving more

attention to salient locations and less attention to unimportant regions. The scan path of the eye is thus strongly affected by visual attention. In recent years, considerable efforts have been devoted to understanding the mechanism driving visual attention. Contributors to the field include Yarbus [36], Yantis [35], Tsotsos, et al. [28], Koch and Ullman [18], Niebur & Koch [25], Horvitz & Lengyel [12].

Two general processes significantly influence visual attention, called bottom-up and top-down processes. The bottom-up process is purely stimulus driven. A few examples of such stimuli are: a candle burning in a dark room; a red ball among a large number of blue balls; or sudden motions. In all these cases, the conspicuous visual stimulus captures attention automatically without volitional control. The top-down process, on the other hand, is a directed volitional process of focusing attention on one or more objects which are relevant to the observer's goal. Such goals may include looking for street signs or searching for a target in a computer game. Though the attention drawn due to conspicuity may be deliberately ignored because of irrelevance to the goal at hand, in most cases, the bottom-up process is thought to provide the context over which the top-down process operates. Thus, the bottom-up process is fundamental to the visual attention.

We disregard the top-down component in favor of a more general and automated bottom-up approach. In doing so, we would be ignoring non-stimulus cues such as a "look over there" command given by the narrator of a scene or shifts of attention due to familiarity. Moreover, a task driven top-down regime can always be added later, if needed, with the use of supervised learning [14].

Itti, Koch and Niebur [13][14][15][16] have provided a computational model to this bottom up approach to visual attention. We chose this model because the integration of this model into our computational framework required minimal changes.The model is built on a biologically plausible architecture proposed by Koch and Ullman [18] and by Niebur and Koch [25]. Figure 6 graphically illustrates the model of visual attention.

The computational architecture of this model is largely a set of center-surround linear operations that mimic the biological functions of the retina, lateral geniculate nucleus and primary visual cortex [19]. These biological systems tend to have a receptive field that triggers in response to changes between the center of the field and its surroundings. The center-surround effect makes the visual system highly sensitive to features such as edges, abrupt changes in color and sudden movements. This model generates feature maps using center surround mechanisms for visually important channels such as intensity, color and orientation. A feature map can be considered to represent the conspicuity at different spatial scales. Each of these features for each of these channels is computed at multiple scales and then processed with an operator, N(.), that mimics the lateral inhibition effect. That is, features that are similar and near each other cancel each other out. Feature maps that have outstanding features are emphasized while feature maps which have competing features or no outstanding features are suppressed. For example, a single white square in a dark background would be emphasized, but a checkerboard pattern would be suppressed. The sum of the feature maps for each channel after they have been processed for lateral inhibition results in a conspicuity map. The conspicuity maps are processed themselves for lateral inhibition and then summed together to obtain a single saliency map that quantifies visual attention. The model of Itti, et al., has been tested with real world scenes and has been found to be effective [13].

The model of Itti, Koch and Niebur does not include motion as a conspicuity channel. We include motion as an additional conspicuity channel in our implementation. We added in the motion with minimal changes to the attention model. The next section describes the process of obtaining the spatiotemporal error tolerance map by building on the knowledge presented here. The two components necessary for spatiotemporal sensitivity calculation, motion and spatial frequency are computed, as is the saliency map necessary for quantifying visual attention.

5 IMPLEMENTATION

Our process begins with a rapid image estimate of the scene. This image estimate serves both to identify areas where spatiotemporal sensitivity is low and also to locate areas where an observer will be most likely to look. Such an image may be quickly generated using an OpenGL rendering, or a ray traced rendering of the scene with only direct lighting. We have typically used OpenGL to render estimates for our work and use the estimate only for the computation of the Aleph Map and the saliency map. Before they are used, the image estimates are converted from RGB into AC_1C_2 opponent color space, using the transformation matrices given in [2].

Our computation proceeds in four major steps: 1) motion estimation, 2) spatial frequency estimation, 3) saliency estimation and 4) computing the Aleph Map. We will discuss each of these steps in detail in the following section. We use the following notation in our description. A capital letter such as 'A' or 'C_1' or 'C_2' denotes a channel and a number in parenthesis denotes the level of scale. Thus, 'A(0)' would correspond to the finest scale of a multiscale decomposition of the achromatic channel of the AC_1C_2 color space. For conciseness, a per-pixel operation, e.g. A(x,y) is implied. Appendix A graphically depicts an overview of the process.

5.1 Motion Estimation

Velocity is one of the two components needed to estimate the spatiotemporal sensitivity of the human visual system. We implemented two different techniques to estimate image plane velocity. One makes use of the image estimate alone and the other makes use of additional information such as geometry and knowledge of the transformations used for movement. The latter model is appropriate for model-based image synthesis applications while the former can be used even when only the image is available, as in image-based rendering. In both of these techniques, the goal is first to estimate displacements of pixels $\Delta P(x,y)$ from one frame to another, and then to compute the image velocity from this pixel displacement, using frame rate and pixel density information.

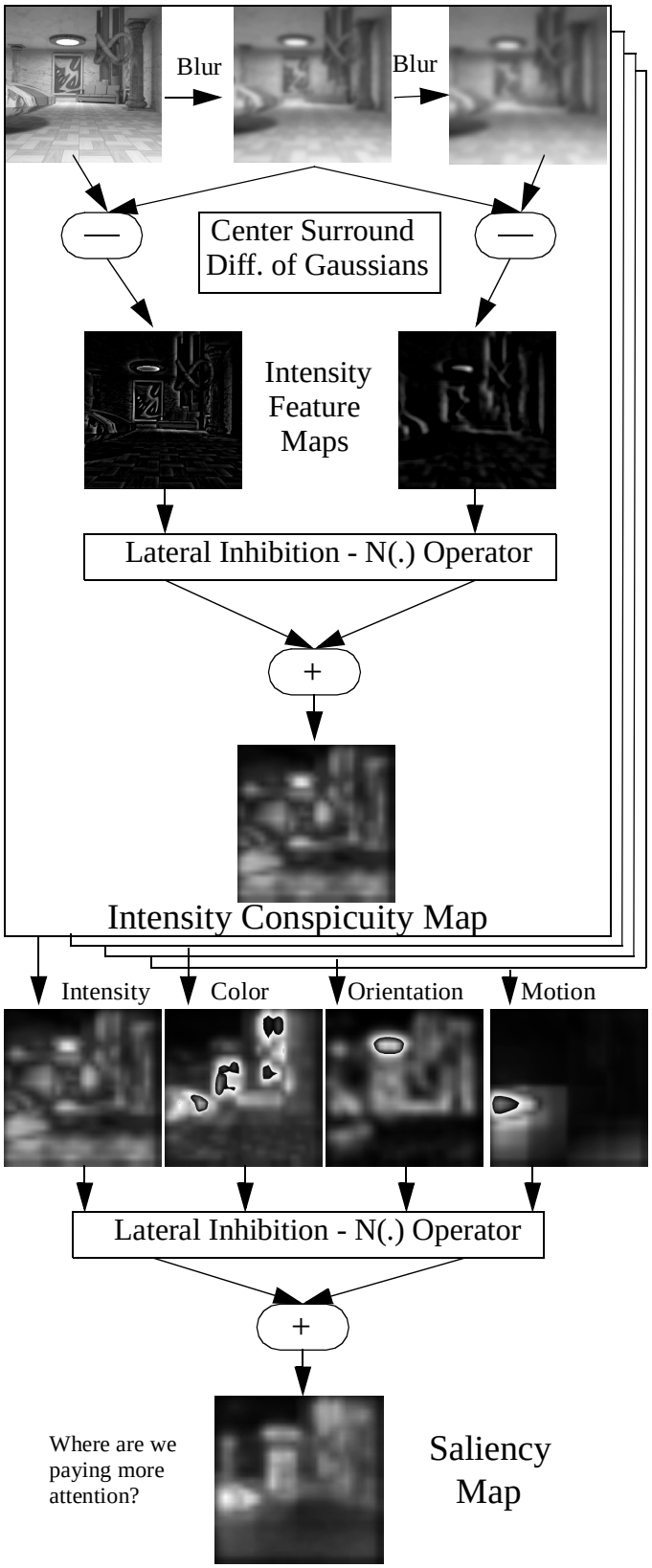


Figure 6: An outline of the computational model of visual attention. An abridged version of the process is shown for the achromatic intensity channel. The conspicuity maps of intensity, color, orientation and motion are combined to obtain the saliency map. Bright regions on the map denote areas of interest to the visual system.

5.1.1 Image Based Motion Estimation

Image-based motion estimation is useful only when consecutive images are available. In this technique, the achromatic 'A' channels of two consecutive image frames are decomposed into multiscale Gaussian pyramids using the filtering method proposed by Burt and Adelson [4]. The Gaussian pyramid created in this section may be reused later to estimate both saliency and spatial frequency.

We now briefly describe the *census transform* [37], a local transform that is used to improve the robustness of motion estimation. The census transform generates a bitstring for each pixel that is a summary of the local spatial structure around the pixel. The bits in the bitstring correspond to the neighboring pixels of the pixel under consideration. The bit is set to 0 if the neighboring pixel is of lower intensity than the pixel under consideration. Otherwise, it is set to 1. For example, in the 1D case, suppose we have a pixel '5' surrounded by other pixels {1,6,5,1,7}. The census transform for the pixel '5' would then be "0101." Performing the census transform allows us to find correspondences in the two images by capturing both intensity and local spatial structure. It also makes motion estimation effective against exposure variations between frames (if a real world photograph was used). Comparisons can then be made between regions of census transformed images by calculating the minimum Hamming distance between two bit strings being compared. The Hamming distance of two bit strings is defined as the number of bits that are different between the two strings and can be implemented efficiently with a simple XOR and bit counting.

The A(0,1,2) levels of the pyramid are passed through the census transform. The three levels were picked as a trade off between computational efficiency and accuracy. An exhaustive search would be most accurate but slow, and a hierarchical search would be fast but inaccurate. We perform an exhaustive search on the census transformed A(2), which is cheap due to its reduced size, to figure out how far pixels have moved between frames. Subsequently, the displacement information is propagated to level 1 and a three-step search heuristic (see page 104 of Tekalp [27]) is used to refine displacement positions iteratively. The three-step heuristic is a search pattern that begins with a large search radius that reduces up to three times until a likely match is found. The results of level 1 is propagated to level 0 and a three-step search again conducted to get our final pixel displacement value. Our implementation estimated motion for two consecutive 512x512 frames in the order of 10 seconds per frame on a 500 Mhz Pentium III machine.

5.1.2 Model Based Motion Estimation

Model-based motion estimation (Agrawala, et. al.[1]) is useful when geometry and transformations of each object in the scene are available. In this technique, we first obtain an object identifier and point of intersection on the object for every pixel in frame N, using either ray casting or using OpenGL hardware projection (Wallach et. al. [29]). We advance the frame to N+1, apply the dynamic transformation to the moving objects in the scene, and project each image point onto the viewing plane corresponding to the $(N+1)^{th}$ frame. The distance of pixel movement is the displacement needed for calculating the image velocity. Due to the discretization of the color buffer (256 values per color channel), the OpenGL based motion estimation had discretization artifacts. For simplicity's sake, we used the ray casting motion estimation data for motion estimation. Our implementation of the ray casting scheme ran in 6 seconds on a Pentium III 500 Mhz machine for a 512x512 image of a 70,000 polygon scene.

Figure 7 compares the two motion estimation techniques. One drawback of using an image-based technique is that the algorithms cannot calculate pixel disparities across regions of uniform color. The model-based motion estimation technique is unaffected by the lack of textures and is less noisy than image based techniques.

We convert the pixel displacements $\Delta P(x,y)$ computed by either of the two techniques into image plane velocities v_I using the following equation.

$$v_I(x,y) = \frac{\Delta P(x,y)}{\text{Pixels Per Degree}} \cdot \text{Frames per Second} \tag{2}$$

In our setup, our values were 30 frames per second on a display with a pixel density of 31 pixels per degree.

5.2 Spatial Frequency Estimation

The remaining component needed to calculate spatiotemporal error sensitivity is the spatial frequency content of the scene. We applied the Difference-of-Gaussians (Laplacian) Pyramid approach of Burt and Adelson [4] to estimate spatial frequency content. One may reuse the Gaussian pyramid of the achromatic channel if it was computed in the motion estimation step. Each level of the Gaussian pyramid is upsampled to the size of the original image and then the absolute difference of the levels is computed to obtain the seven level bandpass Laplacian pyramid, L(0..6).

$$L(i) = |A(i) - A(i+1)| \tag{3}$$

The Laplacian pyramid has peak spatial frequency responses at 16, 8, 4, 2, 1, 0.5 and 0.25 cpd (assuming a pixel density of around 31 pixels per degree). Using a method similar to that followed by Ramasubramanian, et al., [26], each level of the Laplacian pyramid is then normalized by summing all the levels and dividing each level by the sum to obtain the estimation of the spatial frequency content in each frequency band:

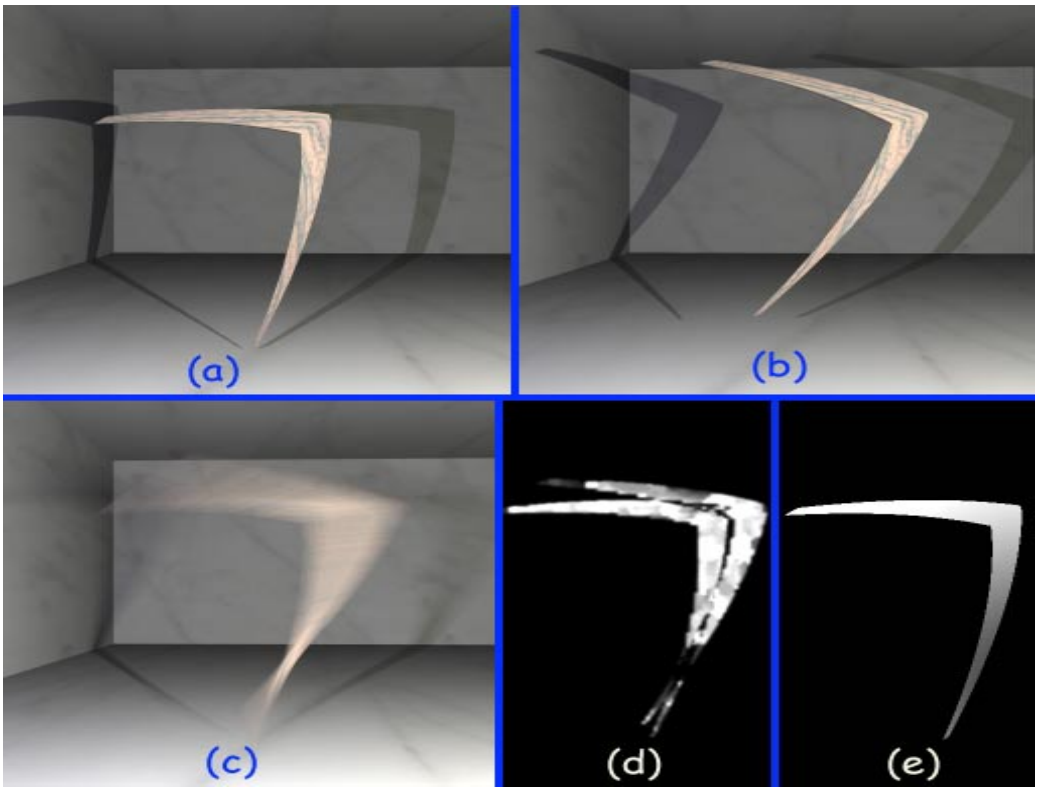


Figure 7: Comparison of Image-Based and Model-Based Motion Estimation. Two consecutive frames (a) and (b) are shown with the boomerang moving to the right from (a) to (b). Motion-blurred image in (c) shows the direction of motion. The results obtained using image-based motion estimation are shown in (d) and using model-based motion estimation is shown in (e). Model-based motion estimation (e) is less noisy and more accurate than image-based motion estimation (d), which explains why (e) has a smooth motion estimation and (d) has a splotchy motion estimation.

$$R_i = \frac{L(i)}{\sum_{\text{all levels j}} L(j)} \tag{4}$$

5.3 Saliency Estimation

The saliency estimation is carried out using an extension of the computational model developed by Itti, et al., [13][16]. Our extension incorporates motion as an additional feature channel. The saliency map tells us where attention is directed to and is computed via the combination of four conspicuity maps of intensity, color, orientation and motion. The conspicuity maps are in turn computed using feature maps at varying spatial scales. One may think of features as stimuli at varying scales, conspicuity as a summary of a specific stimulus at all the scale levels combined and saliency as a summary of all the conspicuity of all the stimuli combined together. Figure 6 illustrates the process visually.

Feature maps for the achromatic (A) and chromatic (C_1,C_2) channels are computed by constructing image pyramids similar to the Laplacian pyramid described in the previous section. A Gaussian pyramid is constructed for each channel and following Itti, et al., we obtain the feature maps in the following manner:

$$X(\text{center}, \text{surround}) = |X(\text{center}) - X(\text{surround})| \tag{5}$$

where X stands for A,C_1,C_2 and (center,surround) $\in$ {(2,5), (2,6), (3,6), (3,7), (4,7), (4,8)}. The numbers correspond to the levels in the Laplacian pyramid.

Motion feature maps are created by applying a similar decomposition to the velocity map generated in the motion estimation section. We perform the computation this manner in order to minimize the changes to the computational model of Itti, et al.

Orientation feature maps are obtained by creating four pyramids using Greenspan's [11] filter on the achromatic channel. Greenspan's filter was tuned to orientations of (0, 45, 90 and 135 degrees) and indicates what components of the image lie along those orientations. We generate a total of 48 feature maps, 6 for intensity at different spatial scales, 12 for color, 6 for motion, and 24 for orientation for determining the saliency map.

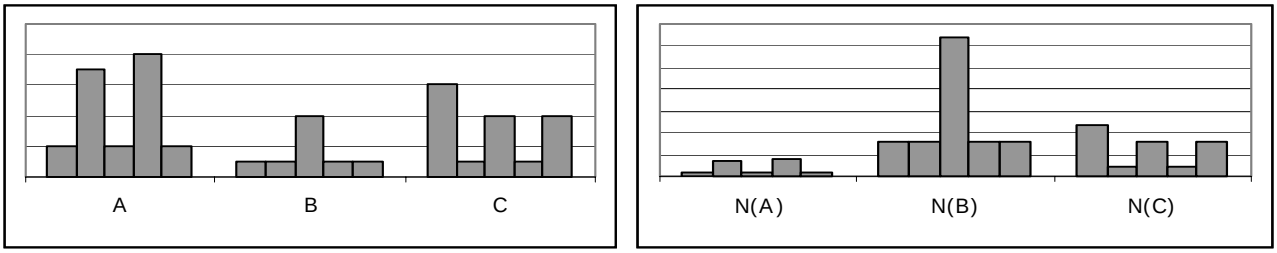
Next, we combine these feature maps to get the conspicuity maps and then combine the conspicuity maps to obtain a single saliency map for each image frame. We use a global non-linear normalization operator, N(.), described in [16] to simulate lateral inhibition and them sum the maps together to perform this combination. This operator carries out the following operations:

1. Normalize each map to the same dynamic range

2. Find the global maximum M and the average $\overline{m}$ of all other local maxima

3. Scale the entire map by $(M-\overline{m})^2$

The purpose of the N(.) operator is to promote maps with significantly conspicuous features while suppressing those that are non-conspicuous. Figure 8 illustrates the action of the N(.) operator on three generic maps.



Before N(.) Operator (a)

After N(.) Operator (b)

Figure 8: Action of the N(.) lateral inhibition operator on three generic maps A, B and C. The left half (a) shows the maps after step 1. The right half (b) shows the maps after steps 2 and 3. Map A and C have competing signals and are suppressed. Map B has a clear spike and is therefore promoted. In this way, the N(.) operator roughly simulates the lateral inhibition behavior of the visual system. When N(.) is applied to feature maps, A,B,C represent the levels of the corresponding Laplacian pyramid of the feature. When applied to conspicuity maps, A,B and C represent channels such as intensity or color.

We apply the N(.) operator to each feature map and combine the resulting maps of each channel's pyramid into a conspicuity map. We now get the four conspicuity maps of intensity, color, orientation and motion. We then compute the saliency map by applying N(.) to each of the four conspicuity maps and then sum them together. We will call the saliency map S(x,y) with the per pixel saliency normalized to a range of (0.0... 1.0) where 1.0 represents the most salient region and 0 represents the least salient region in the image. In our implementation, the saliency computation for a 512x512 image frame is completed in 4 seconds on a 500 Mhz Pentium III machine. Figure 9 shows the saliency map computed for one of the animation image frames

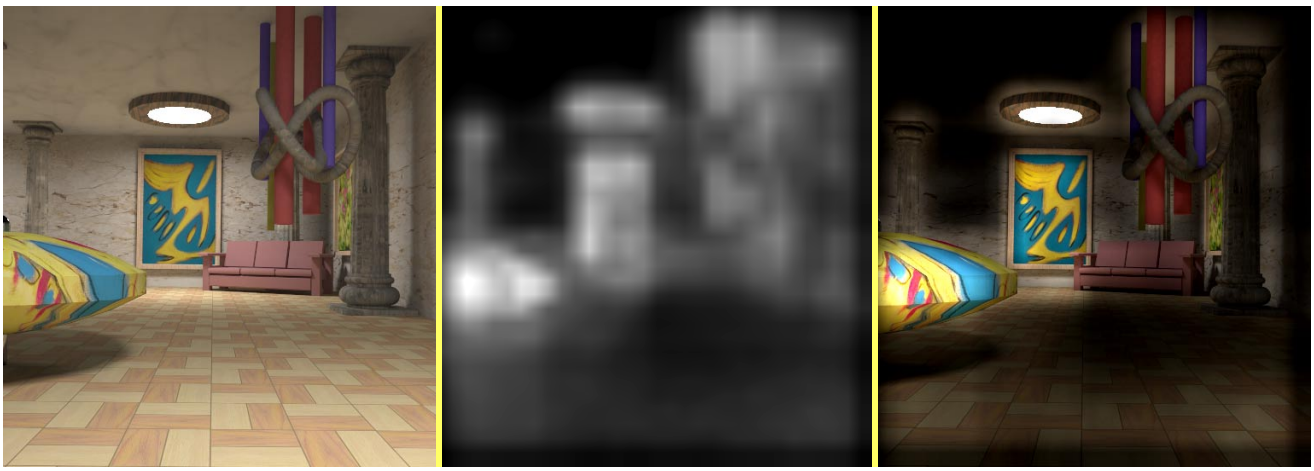


Image (a)

Saliency Map (b)

Superimposed (c)

Figure 9: Saliency map visualization (*see color plate*). In image (a) the yellow and blue top on the left is spinning rapidly. The entire image in motion due to changes in camera position. The computed saliency map is shown in (b) and (c) graphically depicts the modulation of the saliency map with the image. Brighter areas denote areas of greater saliency. Attention is drawn strongly to the spinning top, the paintings, the ceiling sculpture, the area light and the couch. These areas undergo strict motion compensation. The floor and ceiling are not as salient and undergo less compensation.

5.4 Aleph Map Computation

At this stage, we will have the weights for spatial frequency from the bandpass responses $R_i(x,y)$ (equation 4) with peak frequencies ρ_i = {16,8,4,2,1,0.5,0.25} cycles per degree, the image plane pixel velocities $v_I(x,y)$ (equation 2), and the saliency map S(x,y). We now have all the necessary ingredients to estimate the spatiotemporal sensitivity of the human visual system. The first step is to obtain the potential optimal retinal velocity v_R from the image plane velocity v_I with the use of the saliency map S(x,y):

$$v_R(x,y) = v_I(x,y) - min(S(x,y) \cdot v_I(x,y) + v_{Min}, v_{Max}) \tag{6}$$

where v_{Min} is the drift velocity of the eye (0.15 deg/sec [17]) and v_{Max} is the maximum velocity beyond which the eye cannot track moving objects efficiently (80 deg/sec [7]). It is a slight modification of equation (1), where we replace the 82% tracking efficiency with the saliency map. We assume here that the visual system's tracking efficiency is linearly proportional to the saliency. We use this velocity to compute the spatiotemporal sensitivities at each of the spatial frequency bands ρ_i. For this computation, we use Kelly's experimentally derived contrast sensitivity function (CSF):

$$CSF(\rho, v_R) = k \cdot c_0 \cdot c_2 \cdot v_R \cdot (2\pi\rho c_1)^2 \cdot e^{-(4\pi c_1 \rho)/\rho_{max}} \tag{7}$$

$$k = 6.1 + 7.3\left|\log((c_2 \cdot v_R)/3)\right|^3 \tag{8}$$

$$\rho_{max} = (45.9)/(c_2 \cdot v_R + 2) \tag{9}$$

Following the suggestions of Daly [7], we set c_0=1.14, c_1=0.67 and c_2=1.7. These parameters are tuned to CRT display luminance of 100 cd/m^2.

The inverse of the CSF intuitively gives us an elevation factor that increases our tolerance of error beyond the minimum discernible luminance threshold in optimal viewing conditions. We calculate this elevation factor for each of the peak spatial frequencies of our Laplacian pyramid $\rho_i \in$ {16,8,4,2,1,0.5,0.25}cpd:

$$f_i(\rho_i, v_R) = \begin{cases} \dfrac{CSF_{Max}(v_R)}{CSF(\rho_i, v_R)} & \text{if } (\rho_i > \rho_{max}) \\ 1.0 & \text{otherwise} \end{cases} \tag{10}$$

$$CSF_{Max}(v_R) = \frac{\rho_{Max}}{2\pi c_1} \tag{11}$$

where v_R is the retinal velocity, CSF is the spatiotemporal sensitivity function, $CSF_{Max}(v_R)$ is the maximum value of the CSF at velocity v_R, and ρ_{max} is the spatial frequency at which this maximum occurs.

Finally we compute the Aleph Map, the spatiotemporal error tolerance map, as a weighted sum of the elevation factors f_i, and the frequency responses R_i at each location (x,y):

$$\aleph(x,y) = \sum_i R_i \times f_i \tag{12}$$

The computation of equations (10) - (12) are similar to the computation of the threshold elevation map described in [26] with the difference that the CSF function used here is the spatiotemporal CSF instead of the spatial only CSF. Figure 10 shows the error tolerance map $\aleph$(x,y) for an image frame of a dynamic scene. This map captures the sensitivity of the human visual system to the spatiotemporal contents of a scene. $\aleph$(x,y) has values ranging from 1.0 (lowest tolerance to error) to at most 250.0 (most tolerance to error). The total time taken to compute the Aleph Map, including motion estimation, saliency estimation and error tolerance computation is approximately 15 seconds for a 512x512 image on a Pentium III 550 Mhz machine.

In the next section, we show the application of the Aleph Map to efficiently compute global illumination in a dynamic environment.

6 APPLICATION AND RESULTS

The Aleph Map developed in the previous sections is general. It operates on image estimates of any animation sequence to predict the relative error tolerance at every location of the image frame and can be used to efficiently render dynamic environments. Similar to earlier, perceptually-based acceleration techniques [2][3][23][26], we can use this map to adaptively stop computation in a progressive global

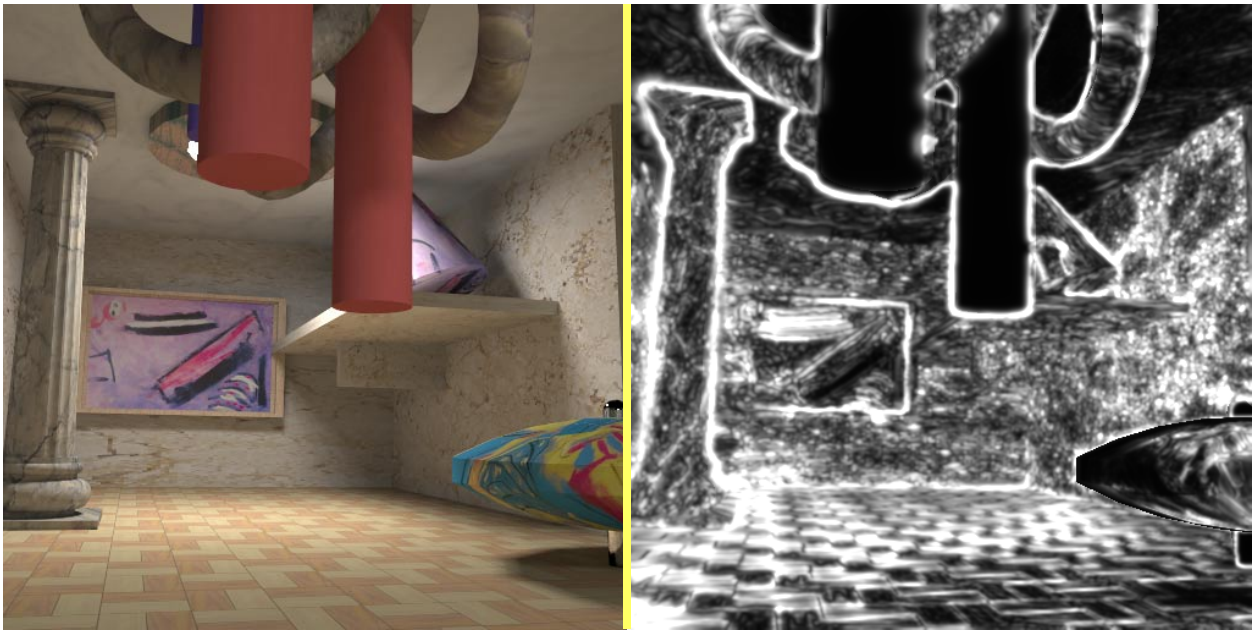


Image (a) Aleph Map (b)

Figure 10: Spatiotemporal sensitivity visualization (*see color plate*). Image (a) and its corresponding error tolerance map, the Aleph Map (b). Note that the spinning top in the bottom right has reduced tolerance to error although it has textures and is moving. This is due to the information introduced by the saliency map, telling the algorithm to be stricter on the top because the viewer will more likely focus attention there. The red beams are treated strictly because there are no high frequency details.

illumination algorithm. To demonstrate the wider usefulness of this map, we have applied the map to improve the computational efficiency of RADIANCE.

The irradiance caching algorithm is the core technique used by RADIANCE to accelerate global illumination and is well documented by Ward [31][33][34]. As suggested by its name, the irradiance caching technique works by caching the diffuse indirect illumination component of global illumination [34]. A global illumination lighting solution can be calculated as the sum of a direct illumination term and an indirect illumination term. Indirect illumination is by far the most computationally expensive portion of the calculation. Irradiance caching addresses this problem by reusing irradiance values from nearby locations in object space and interpolating them, provided the error that results from doing so is bounded by the evaluation of an *ambient accuracy* term. The ambient accuracy term α_{Acc} varies from 0.0 (no interpolation, purely Monte Carlo simulation) to 1.0 (maximum ambient error allowed). Hence, by reusing information, the irradiance caching algorithm is faster than the standard Monte Carlo simulation of the global illumination problem by several orders of magnitude, while at the same time providing a solution that has bounded error.

The ambient accuracy term is user supplied and gives a measure of the tolerated error. RADIANCE uses this term uniformly over the entire image, and thus does not take advantage of the variation of sensitivity of the human visual system over different parts of the image. Our application of the Aleph Map to the irradiance caching algorithm works by modulating the ambient accuracy term on a per pixel basis. Hence, if the Aleph Map allows for greater error for that pixel, a larger neighborhood is considered for interpolation and hence the irradiance cache is used more efficiently. In order to use the Aleph Map with the irradiance cache, we need to use a compression function to map the values of $\aleph(x,y)$ onto (α_{Acc}.. 1.0) for use as a perceptual ambient accuracy term. The following equation accomplishes this compression:

$$\aleph_\alpha = \frac{\aleph}{\aleph - 1 + \frac{1}{\alpha_{Acc}}} \tag{13}$$

where $\aleph_\alpha$ is the adapted map used in lieu of the original ambient accuracy term α_{Acc}. The $\aleph-1$ term merely accounts for the fact that $\aleph$ starts from 1.0 and increases from there. The equation ensures that $\aleph_\alpha$ is bounded between α_{Acc} and 1.0. Hence, in regions where attention is focused and where there are no high frequencies to mask errors, $\aleph_\alpha=\alpha_{Acc}$ and in areas where the errors will be masked, $\aleph_\alpha$ asymptotically approaches 1.0. Computation of $\aleph_\alpha$ is carried out only once, at the beginning of the global illumination computation of every frame. However, should a stricter bound be desired, one may opt to recompute $\aleph(x,y)$ and hence recompute $\aleph_\alpha$ at intermediate stages of computation.

We demonstrate the performance of our model using a test scene of a synthetic art gallery. The scene contains approximately 70,000 primitives and 8 area light sources. It contains many moving objects, including bouncing balls, a spinning top and a kinetic sculpture that demon-

strates color bleeding on a moving object. Figure 11 compares two still frames from the reference solution and the perceptually accelerated solution.

Reference Solution (a) Perceptual Solution (b)

Figure 11: Image comparison from frame 0 of the Art Gallery sequence (*see color plate*). The image on the left is the reference image and the image on the right is the image generated with the perceptually enhanced irradiance cache technique.

Figure 12 shows the root mean square error between two equal time solutions and the reference solution in Figure 11 (a). In Figure 12, the left image relaxes the ambient accuracy to 1.0 throughout the image uniformly and the right has a base ambient accuracy of 0.6 tolerable error modulated by the Aleph map. The Aleph Map guided solution has a lower base error tolerance, meaning that where it is important, the algorithm spends more time on calculating the solution.

Figure 13 shows the performance improvement resulting from the use of the Aleph Map. In most of the frames, we achieve a 6x to 8x speedup over standard irradiance caching. Using spatial factors only we achieve a 2x speedup. A marginal improvement over spatial sensitivity is obtained if the Daly motion compensation heuristic is used in conjunction with spatiotemporal sensitivity. Note that all these improvements are compared to the speed of the unaugmented irradiance caching technique, which is an order of magnitude more efficient than simple path tracing techniques. In addition, the speedup was found to be largely independent of the number of samples shot. Another video sequence, the pool sequence, was found to exhibit a similar speedup of 3x to 9x speedup depending on the amount of moving objects and textures in parts of the sequence. The images for the pool sequence are found in Figure 14.

In this demonstration, we maintained good sampling protocols. The sampling density for each irradiance value is left unchanged, but the irradiance cache usage is perceptually optimized. Figure 15 shows the locations in the image at which irradiance values were actually computed. Bright spots indicate that an irradiance value was calculated while dark regions are places where the cache was used to obtain an interpolated irradiance value. This also explains why the speedup is independent of the number of samples shot, because the spacing of the irradiance cache is optimized, not the number of samples per irradiance value.

In static scenes where only the camera moves, the irradiance cache can be maintained over consecutive frames. Our technique was found to perform well even when such interframe coherence is used. Our results from a proof of concept test (Figure 16) show that even under this situation the use of $\aleph_\alpha$ improves the computation speed.

In viewing the Art Gallery sequence, it was discovered that repeated viewings can cause the viewer to pay more attention to unimportant regions. In doing so, the viewer deliberately chose to ignore attention cues and focus on unimportant areas such as the ceiling. This introduces a top-down behavioral component to visual attention that is not accounted for in our model. The pool sequence had unambiguous salient features (the pool balls) and was not as susceptible to the replay effect.

Visual sensitivity falls rapidly as a function of foveal eccentricity [8]. An experiment incorporating foveal eccentricity into the model was performed, and significant speedup was achieved. However, the animations generated with the use of foveal eccentricity tended to be useful

Uniform error tolerance (a)

Aleph Map guided error tolerance (b)

Figure 12: Equal time error comparison. The image (a) on the left shows the root mean square error between the reference solution and an image with uniform ambient accuracy of 1.0. The image (b) on the right shows the root mean square error between the reference solution and an Aleph map guided solution with a base ambient accuracy of 0.6. Both solutions took approximately the same amount of time to render. White indicates a larger error and black indicates less error from the reference solution.

only in the first few runs of the animation, as viewers tended to look away from foveal regions once they had seen the animation a number of times.

An important point to note is that the Aleph Map is general and adaptable. For example, it can be converted to a physically based error metric [26] via a multiplication with the luminance threshold sensitivity function.

$$\Delta L = \aleph(x, y) \times \Delta L_{TVI}(L) \tag{14}$$

where ΔL is the luminance threshold, L is the adaptation luminance calculated as the average luminance in a 1 degree diameter solid angle centered around the fixating pixel and ΔL_{TVI} is the threshold vs. intensity function defined in Ward-Larson, et al. [32]. In video compression and telephony, it can be used to optimize compression simply by compressing more rigorously when $\aleph$(x,y) is high and less when it is low. In geometric level of detail selection, one may opt to use coarser models when spatiotemporal error tolerance is high and a detailed model where tolerance is low.

In this paper, a two part vision model is presented to the graphics community, the first part quantifying visual attention and the second part quantifying spatiotemporal error tolerance. Both parts were validated extensively by their authors as described in their respective papers [7][13]. In order to examine the effectiveness of our hybrid model, the Aleph map was tested by calculating the luminance threshold using equation (14) for each pixel in the reference image and multiplying the threshold with a unit random number. The resulting noise map was added to each frame of the reference solution to obtain the sub-threshold noisy image sequence which was viewed by a panel of observers for discrepancy. The noise was found to be visible when still frames are viewed but not when the images are in motion during a video sequence. Figure 17 outlines the process used to test the Aleph Map. The Aleph map assisted irradiance caching was also tested on other scenes with comparable results and speedups.

Our implementation does not include color and orientation in the sensitivity computation, although those factors are considered in the computational model of visual attention. We also do not implement contrast masking (Fewerda, et. al. [9]) as it is not well understood how motion affects it. Omitting it simplifies our model and makes it more conservative, but it is better to err on the safe side. We also have chosen to treat each component of the visual system as multiplicative with each other and the results have shown that it works, but the human visual system is nonlinear and has vagaries that would be hard to model.

7 CONCLUSIONS

A model of visual attention and spatiotemporal sensitivity was presented that exploited the limitations of the human visual system in perceiving moving spatial patterns. When applied to animation sequences, results indicated an order of magnitude improvement in computation speed. The technique has many applications and can be used in image based rendering, global illumination, video compression and video

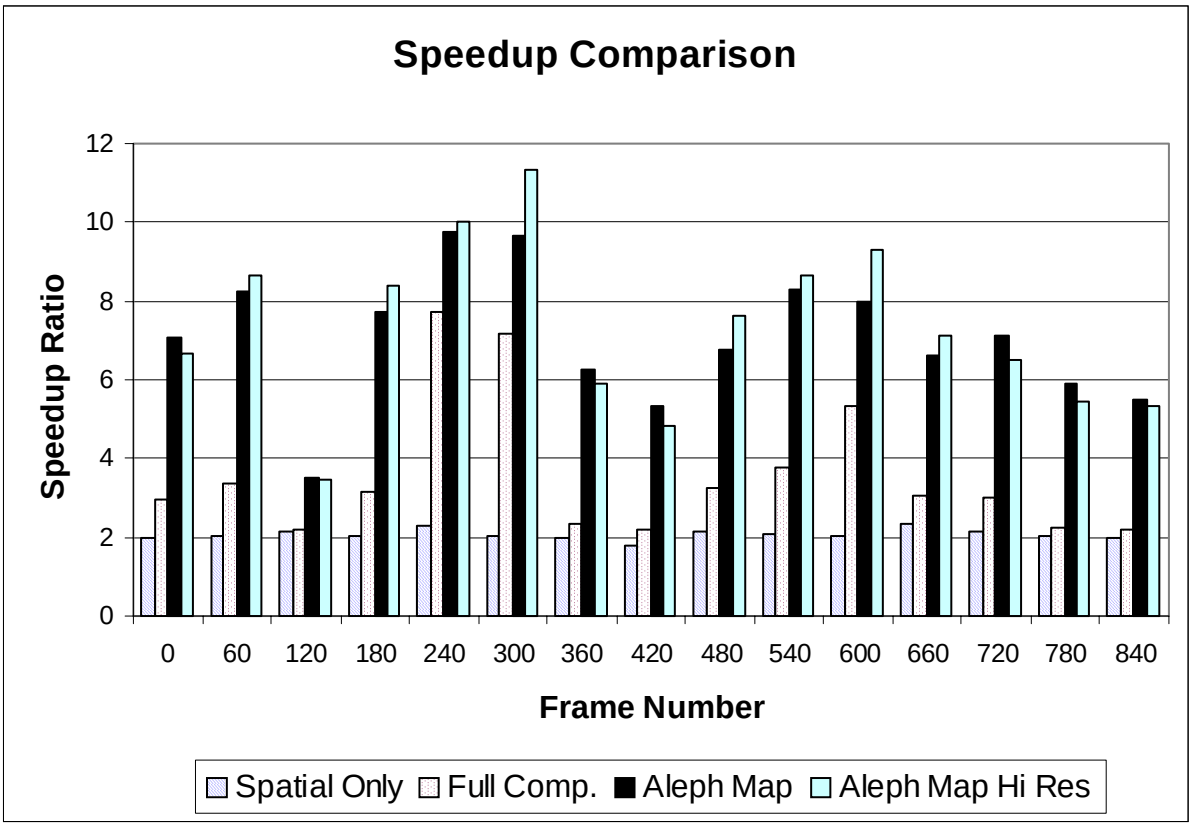


Figure 13: Speedup over Irradiance Cache for the Art Gallery sequence. The total number of ray triangle intersections per pixel are compared. The Aleph Map enhanced irradiance cache performs significantly better (6-8x) than the unaugmented irradiance cache. Spatial factors contribute to an average of 2x speedup while Daly (full) motion compensation gives marginally better results. The spatial only solution corresponds to applying the technique of Ramasubramanian et. al. [26] (less the masking term) to irradiance caching. These speedup factors are *multiplied* to that provided by irradiance caching, a technique far faster than straight Monte Carlo pathtracing. Image frames were computed using an ambient accuracy setting of 15% and an ambient sampling density of 2048 samples per irradiance value at a resolution of 512x512. Note that with these settings the reference solution is almost but not completely converged. For comparison purposes, a reference solution and a perceptually accelerated solution are rendered at a higher resolution (640x480) and a sampling density of 8192 samples per irradiance value (Aleph Map Hi Res). As seen on the graph (Aleph Map vs. Aleph Map Hi Res), the acceleration is largely independent of the number of samples shot, because the perceptual solution changes only the spacing of the samples but not the sampling density.

telephony. This work will be useful and beneficial in all areas of graphics research where spatiotemporal sensitivity and visual attention are used.

Acknowledgements. This work was supported by the NSF Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-8920219). The paintings in the Art Gallery sequence were done by Zehna Barros of Zehna Originals, with the exception of the Gnome painting by Nordica Raapana. Modeling software was provided by Autodesk, and free models were provided courtesy of Viewpoint Datalabs, 3D Cafe and Platinum Pictures. Computation for this work was performed on workstations and compute clusters donated by Intel Corporation. This research was conducted in part using the resources of the Cornell Theory Center, which receives funding from Cornell University, New York State, federal agencies, and corporate partners. Many thanks go to the anonymous reviewers, as well as to the staff and students of the Program of Computer Graphics for proofreading this paper, especially Stephen Westin, Jack Tumblin, Peggy Anderson and Jonathan Corson-Rikert. Thanks to Westwood Studios and ATI Technologies for contributing computational resources and graphics cards respectively for use in the revision of this paper.

Supplementary electronic material are available at http://www.acm.org/tog/yee01.

Reference

Full Compensation

Saliency Compensation

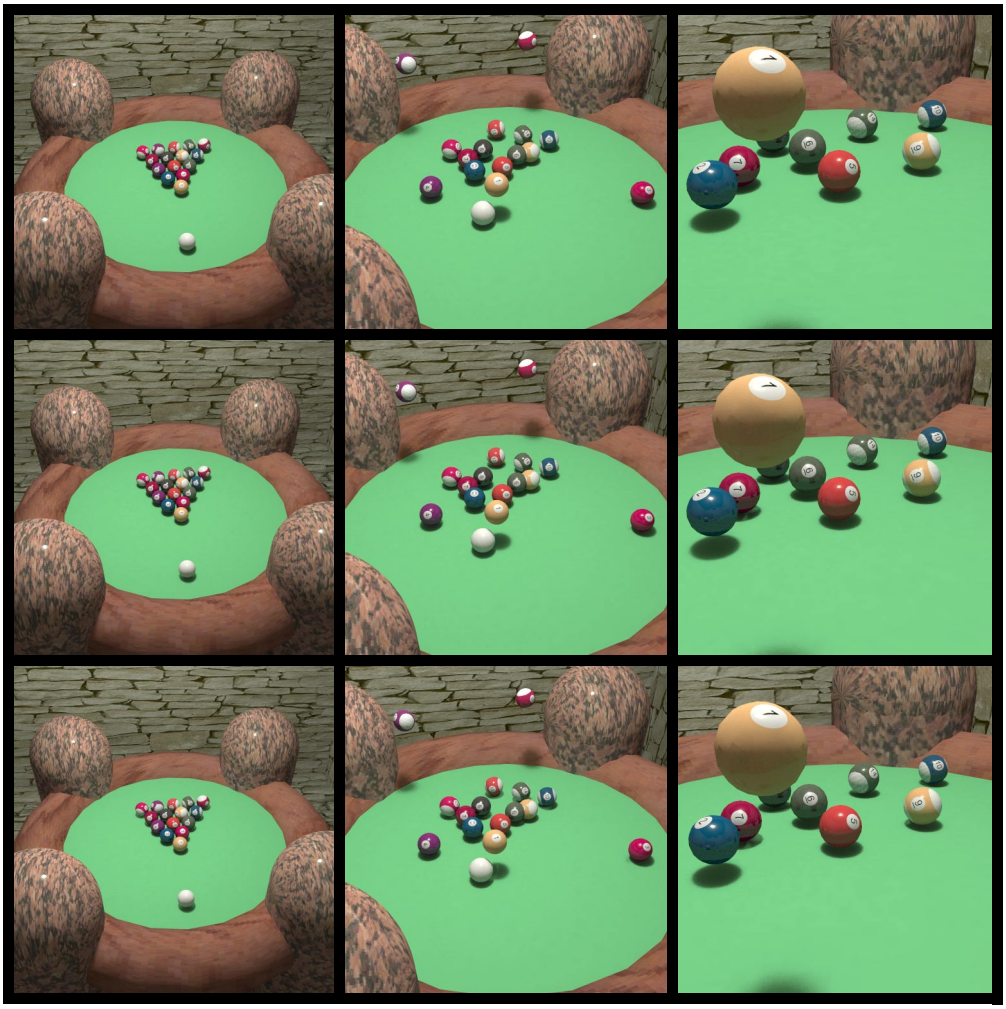
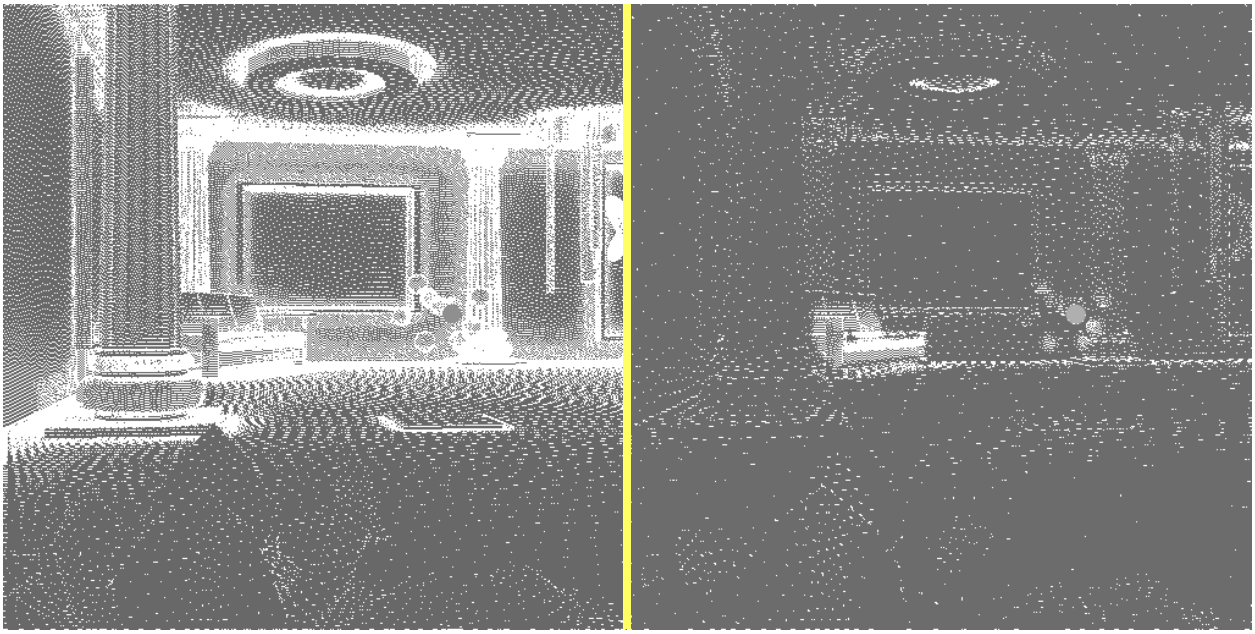


Figure 14: Pool Sequence Visual Comparison. The top row shows images from the pool sequence computed using plain vanilla irradiance caching. The middle row was rendered with Aleph Map enhanced irradiance caching, except that the retinal velocity computed using equation (1). The bottom row shows the images rendered using the Aleph Map as described in this paper, with the retinal velocity derived using the saliency map. The full compensation offers an average of a 3x speedup over the reference solution. The saliency compensation offers an average of 6x speedup over the reference solution.



Reference Sampling Density (a)

Perceptual Solution Sampling Density (b)

Figure 15: Sampling patterns for frame 0 of the Art Gallery sequence. The bright spots indicate where the irradiance value for the irradiance cache is generated and the dark spots indicate where an interpolated irradiance value is used.

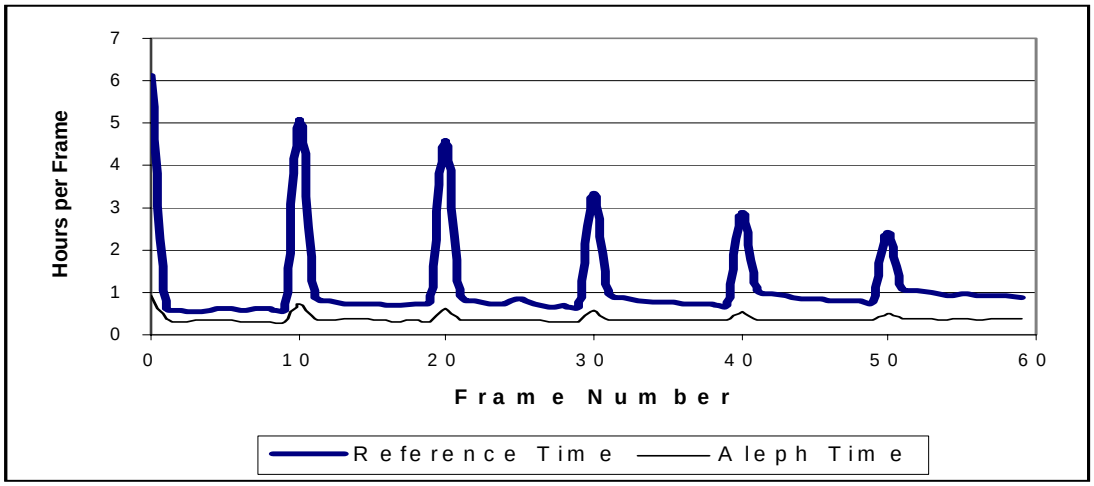
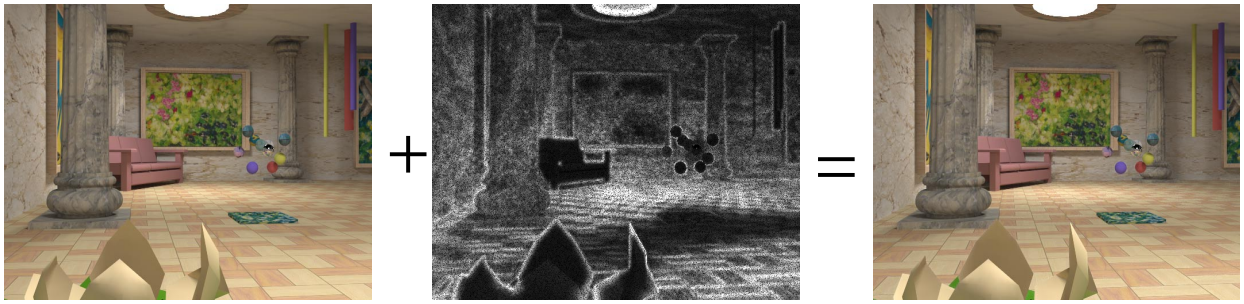


Figure 16: Timing comparison with interframe coherence. Coherence is achieved by flushing the irradiance cache when the age reaches 10 frames. The spikes denote when a cache filling operation is performed. The ℵ Map enhanced irradiance cache (even when no coherence is available, e.g. frame 0) performs better than the irradiance cache with interframe coherence.



Reference Solution (a) Sub-threshold noise Map (b) Noisy Reference Solution (c)

Enlarged version of Noisy Reference Solution (d)

Figure 17: The Aleph Map was tested by using the reference solution (a) to construct a sub-threshold noise map (b) and adding the two to obtain a 'noisy reference solution' (c) (*see color plate*). The process is repeated for all frames of the reference solution. The noise can be seen in (d) when the image is still but is difficult to discern during a video sequence of the noisy reference solution.

References

[1] Agrawala, M., Beers, A. C., and Chaddha, N. Model-Based Motion Estimation for Synthetic Animations. In *Proceedings of the Third International Conference on Multimedia '95*, pp. 477-488. 1995.

[2] Bolin, M. R., and Meyer, G. W. A Frequency Based Ray Tracer. In *SIGGRAPH 95 Conference Proceedings*, pp. 409-418. Los Angeles, CA, August 1995.

[3] Bolin, M. R., and Meyer, G. W. A Perceptually Based Adaptive Sampling Algorithm. In *SIGGRAPH 98 Conference Proceedings*, pp. 299-309, Orlando, Florida, July 1998.

[4] Burt, P.J. and Adelson, E.H. The Laplacian Pyramid as a Compact Image Code. In *IEEE Transactions on Communications*, Vol. Com-31, No. 4, pp. 532-540, April 1983.

[5] Campbell, F. W. and Robson, J. G. Application of Fourier Analysis to the Visibility of Gratings. In *Journal of Physiology* (London) 197, pp. 551-566. 1968.

[6] Daly, S. The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity. In Watson, A.B. editor, *Digital Images and Human Vision*, pp. 179-206, MIT Press, Cambridge, MA, 1993.

[7] Daly, S. Engineering Observations from Spatiovelocity and Spatiotemporal Visual Models. In *IS&T/SPIE Conference on Human Vision and Electronic Imaging III.*, SPIE Vol. 3299, pp. 180-191, January 1998.

[8] Daly, S., Matthews, K., and Ribas-Corbera, J. Visual Eccentricity Models in Face-based Video Compression. In *IS&T/SPIE Conference on Human Vision and Electronic Imaging IV.*, SPIE Vol. 3644, pp. 152-166, January 1999.

[9] Ferwerda, J.A., Pattanaik, S., Shirley, P., and Greenberg, D.P. A Model of Visual Masking for Computer Graphics. In *SIGGRAPH 1997 Conference Proceedings*, pp. 143-152, 1997.

[10] Gibson, S., and Hubbold, R. Perceptually-Driven Radiosity. In *Computer Graphics Forum*, 16(2), pp. 192-140. June, 1997.

[11] Greenspan, H., Belongie, S., Goodman, R., Perona, P., Rakshit, S., Anderson, C.H. Overcomplete Steerable Pyramid Filters and Rotation Invariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, Washington, pp. 222-228. June, 1994.

[12] Horvitz, E. and Lengyel, J. Perception, Attention, and Resources: A Decision-Theoretic Approach to Graphics Rendering. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pp. 238-249, Providence, RI, August 1997.

[13] Itti, L. and Koch, C. A Saliency-Based Search Mechanism for Overt and Covert Shifts of Visual Attention. In *Vision Research*, Vol. 40, No. 10-12, pp. 1489-1506, 2000.

[14] Itti, L. and Koch, C. A Comparison of Feature Combination Strategies for Saliency-Based Visual Attention Systems. In *IS&T/SPIE Conference on Human Vision and Electronic Imaging IV.*, SPIE Vol. 3644, pp. 373-382, January 1999.

[15] Itti, L. and Koch, C. Learning to Detect Salient Objects in Natural Scenes Using Visual Attention. In *Image Understanding Workshop*, 1999. (In press. A preprint version of this article is available from http://www.klab.caltech.edu/~itti/attention).

[16] Itti, L., Koch, C., and Niebur, E. A Model of Saliency-based Visual Attention for Rapid Scene Analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 20(11), pp. 1254-1259. 1998.

[17] Kelly, D.H. Motion and vision II. Stabilized Spatio-temporal Threshold Surface. In *Journal of the Optical Society of America*, Vol. 69, No. 10, pp. 1340-1349, October 1979.

[18] Koch, C. and Ullman, S. Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry. In *Human Neurobiology*, 4, pp. 219-227. 1985.

[19] Leventhal, A.G. The Neural Basis of Visual Function. In *Vision and Visual Dysfunction Vol 4.*, Boca Raton, FL: CRC Press, 1991.

[20] Lubin, J. A Visual Discrimination Model for Imaging System Design and Evaluation. In Peli E. editor, *Vision Models for Target Detection and Recognition*, pp. 245-283, World Scientific, New Jersey. 1995.

[21] McMillan, L. An Image-Based Approach to 3D Computer Graphics. Ph. D. thesis, 1997.

[22] Meyer, G.W. and Liu, A. Color Spatial Acuity Control of a Screen Subdivision Image Synthesis Algorithm. In Bernice E. Rogowitz, editor, *Human Vision, Visual Processing and Digital Display III*, vol 1666, pp. 387-399, Proc SPIE, 1992.

[23] Myszkowski, K. The Visible Differences Predictor: Applications to Global Illumination Problems. In *Proceedings of the Ninth Eurographics Workshop on Rendering*, pp. 223-236. Vienna, Austria, June 1998.

[24] Myskowski, K., Rokita, P., and Tawara, T. Perceptually-informed Accelerated Rendering of High Quality Walkthrough Sequences. In *Proceedings of the Tenth Eurographics Workshop on Rendering*, pp. 5-18. Grenada, Spain. June 1999.

[25] Niebur, E. and Koch, C. Computational Architectures for Attention. In Parasuraman, R. editor, *The Attentive Brain*, pp. 164-186, MIT Press, Cambridge, MA, 1998.

[26] Ramasubramanian M., Pattanaik, S.N., Greenberg, D.P. A Perceptually Based Physical Error Metric for Realistic Image Synthesis. In *SIGGRAPH 99 Proceedings*, pp. 73-82, Los Angeles, CA, 1999.

[27] Tekalp, A. M. Digital Video Processing, Prentice Hall, NJ. 1995.

[28] Tsotsos, J. K., Culhane, S.M., Wai, W.Y.K., Lai, Y., Davis, N., Nuflo, F. Modeling Visual Attention via Selective Tuning. In *Artificial Intelligence*, 78, pp. 507-545, Elsevier Science B.V., 1995.

[29] Wallach, D., Kunapalli, S., and Cohen, M. Accelerated MPEG Compression of Polygonal Dynamic Scenes. In *SIGGRAPH 1994 Proceedings*, pp. 193 - 196, ACM Press, 1994.

[30] Wandell, B. Foundations of Vision. Sinauer Associates, Inc. 1995.

[31] Ward-Larson, G., Shakespeare, R. Rendering with Radiance. Morgan Kaufmann, San Francisco, CA. 1998.

[32] Ward-Larson, G., Rushmeier, H., and Piatko, C. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. In *IEE Transactions on Visualization and Computer Graphics*, 3(4):291-306, October 1997.

[33] Ward, G., Heckbert, P.S. Irradiance Gradients. In *Third Annual Eurographics Workshop on Rendering, Springer-Verlag*, 1992.

[34] Ward, G. A Ray Tracing Solution for Diffuse Interreflection. In *SIGGRAPH 1988 Proceedings*, pp. 85-92, ACM Press. 1988.

[35] Yantis, S. Attentional Capture in Vision. In A. Kramer, M. Coles, & G. Logan (Eds.), Converging Operations in the Study of Selective Visual Attention, pp. 45-76, Washington, DC: American Psychological Association. 1996.

[36] Yarbus, A. L. Eye Movements and Vision, Plenum Press, New York NY, 1967.

[37] Zabih, R. and Woodfill, J. Non-parametric Local Transforms for Computing Visual Correspondence. In *Third European Conference on Computer Vision*, Stockholm, Sweden, May 1994.

Appendix A - Flowchart of Aleph Map Computation