Perception-Based Fast Rendering and Antialiasing of Walkthrough Sequences

Karol Myszkowski, *Member*, *IEEE Computer Society*,
Przemyslaw Rokita, *Member*, *IEEE*, and Takehiro Tawara

Abstract—In this paper, we consider accelerated rendering of high quality walkthrough animation sequences along predefined paths. To improve rendering performance, we use a combination of a hybrid ray tracing and Image-Based Rendering (IBR) technique and a novel perception-based antialiasing technique. In our rendering solution, we derive as many pixels as possible using inexpensive IBR techniques without affecting the animation quality. A perception-based spatiotemporal Animation Quality Metric (AQM) is used to automatically guide such a hybrid rendering. The Image Flow (IF) obtained as a byproduct of the IBR computation is an integral part of the AQM. The final animation quality is enhanced by an efficient spatiotemporal antialiasing which utilizes the IF to perform a motion-compensated filtering. The filter parameters have been tuned using the AQM predictions of animation quality as perceived by the human observer. These parameters adapt locally to the visual pattern velocity.

Index Terms—Walkthrough animation, human perception, video quality metrics, motion-compensated filtering.

1 Introduction

Rendering of animated sequences proves to be a very computation intensive task which, in professional production, involves specialized rendering farms designed specifically for this purpose. While progress in the efficiency of rendering algorithms and increasing processor power is very impressive, the requirements imposed by the complexity of rendered scenes has also increased at a similar pace. Effectively, rendering times reported for the final antialiased frames are still counted in tens of minutes or hours.

It is well-known in the video community that the human eye is less sensitive to higher spatial frequencies than to lower frequencies and this knowledge was used in designing video equipment [11]. It is also conventional wisdom that the requirements imposed on the quality of still images must be higher than for images used in an animated sequence. Another intuitive point is that the quality of rendering can usually be relaxed as the velocity of the moving object (visual pattern) increases. These observations are confirmed by systematic psychophysical experiments investigating the sensitivity of the human eye for various spatiotemporal patterns [18], [38]. For example, the perceived sharpness of moving low resolution (or blurred) patterns increases with velocity, which is attributed to the higher level processing in the visual system [43]. This means that all techniques attempting to speed up the rendering of every single frame separately cannot account for the eye sensitivity variations resulting from temporal considerations. Effectively, computational efforts can be easily wasted on processing image details which cannot be perceived in the animated sequence. In this context, a global approach involving both spatial and temporal dimensions appears promising [28] and is a relatively unexplored research direction.

This research is an attempt to develop a framework for the perceptually-based accelerated rendering of antialiased animated sequences. In our approach, computation is focused on those selected frames (keyframes) and frame fragments (in-between frames) which strongly affect the whole animation appearance by depicting image details readily perceivable by the human observer. All pixels related to these frames and frame fragments are computed using a costly rendering method (we use ray tracing as the final pass of our global illumination solution), which provides images of high quality. The remaining pixels are derived using an inexpensive method (we use IBR techniques [25], [24], [32]). Ideally, the differences between pixels computed using the slower and faster methods should not be perceived in animated sequences, even though such differences can be readily seen when the corresponding frames are observed as still images. The spatiotemporal perception-based quality metric for animated sequences is used to guide frame computation in a fully automatic and recursive manner. Special care is taken for efficient reduction of spatial and especially annoying temporal artifacts, which occasionally can be observed even in professionally produced animated sequences.

In our approach, we use the image flow (IF) [17], which is computed as the velocity vector field in the image plane due to the motion of the camera along the animation path. The velocity distribution is provided for all pixels and all frames in the animation sequence. The IF is the key point in our overall animated sequence processing. It is computed using IBR techniques, which guarantees very good accuracy

- *K. Myszkowski and T. Tawara are with the Max Planck Institute for Computer Science, Im Stadtwald, 66-123 Saarbrucken, Germany. E-mail: {karol, tawara}@mpi-sb.mpg.de.*
- *P. Rokita is with the Computer Science Department, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland. E-mail: pro@ii.pw.edu.pl.*

Manuscript received 23 Aug. 1999; revised 1 June 2000; accepted 12 July 2000.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number 110479.

and high speed of processing for the synthetic images.[1] The IF is used in our technique in four ways:

- to support nearer to optimal keyframe selection along the predefined animation path,
- to reproject pixels from the ray traced keyframes to the image-based in-betweens,
- to improve the temporal considerations of our perception-based animation quality metric,
- to enhance the animation quality by performing antialiasing based on motion-compensated filtering.

Obviously, the best cost-performance is achieved when the IF is used in all four processing steps. However, since these steps are only loosely coupled and the cost of computing IF is very low, other scenarios are also possible, e.g., fully ray traced animation can be filtered with motion compensation.

In this paper, we narrow our discussion to the production of high-quality walkthrough animations (only camera animation is considered), although some of the solutions proposed can be used in a more general animation framework (refer to [28] for discussion of problems with global illumination in this more general case). We assume that walkthrough animation is of high quality, involving complex geometry and global illumination solutions, and, thus, a single frame rendering incurs significant costs (e.g., about 170 minutes in the ATRIUM example chosen as one of the case studies in this research [1]). We also make other reasonable assumptions such as: The animation path and all camera positions are known in advance, ray tracing (or other high quality rendering method) for selected pixels is available, depth (range) data for each pixel is inexpensive to derive for every frame (e.g., using z-buffer), and the object identifiers for each pixel can be easily accessed for every frame (e.g., using item buffer [42]).

The material in this paper consolidates and expands on the results presented in [27]. In the following section, we discuss previous work on perception-based video quality metrics and improving performance of animation rendering. In Section 3, we present our animation quality metric. Then, we describe efficient methods of in-between frames computation we have used in our research. Section 5 describes our 3D antialiasing technique based on motion-compensated filtering. Section 6 and the accompanying Web page [1] show results obtained using our approach. Finally, we conclude this work.

2 Previous Work

In this research, our objective is the reduction of the time required for rendering in-between frames, which can be derived from the high-quality keyframes. To our knowledge, a method that automatically selects keyframes while minimizing distortions visible by human observers has not been presented yet. We review the perceptually-based video quality metrics which could be used to guide rendering of in-between frames. Next, we discuss the problem of keyframes selection, which improves the performance of in-between frames computation using IBR techniques. Finally, we review the IF applications in animation rendering.

2.1 Video Quality Metrics

Assessment of video quality in terms of artifacts visible to the human observer is becoming very important in various applications dealing with digital video technology. Subjective video quality measurement usually is costly and time-consuming and requires many human viewers to obtain statistically meaningful results [36]. Also, it is impractical or even impossible to involve human viewers in some routine applications such as continuous monitoring of the quality of a digital television broadcast. In recent years, a number of automatic video quality metrics based on the computational models of human vision have been proposed. Some of these metrics were designed specifically for video [11], [44], while others were extensions of some well-established still image quality metrics into the time domain [22], [39], [36]. While the majority of these objective metrics have had the same general purpose, one of the main motivations driving their development was the need to evaluate the performance of digital video encoding, transmission, and compression techniques. Because of these particular applications, some metrics [45] are specifically tuned for the assessment of perceivability of typical distortions arising in lossy video compression such as blocking artifacts, blurring, color shifts, and fragmentation. In this study, we deal exclusively with synthetic images and we are looking for a metric well-tuned to our application, even at the expense of some loss of its generality.

The existing video quality metrics account for the following important characteristics of the Human Visual System (HVS):

- temporal and spatial channels (mechanisms), which are used to represent the visual information at various scales and orientations in a similar way as it is believed that the primary visual cortex does [10], [37],
- spatio-temporal sensitivity to contrast, which varies with the spatial and temporal frequencies. The sensitivity is characterized by so-called spatiotemporal Contrast Sensitivity Function (CSF), which defines the detection threshold for a stimulus as a function of its spatial and temporal frequencies [18],
- visual masking accounts for the modification of the detection threshold of a stimulus as a function of the interfering background stimulus which is closely coupled in space or time [20]. The background stimulus located within the same frame causes the spatial masking, while the background stimulus considered along the time axis (which effectively corresponds to the previous and subsequent frames in respect to a given frame) causes the temporal masking [4].

The differences between the existing metrics rely mostly on the complexity of the human vision models that attempt to fit some psychophysical data derived for various experimental conditions and settings.

Spatial frequency and orientation channels are modeled by filter banks such as the steerable pyramid transform [44],

[1] For the natural image sequences, the optical flow can be derived [34], but is more costly and usually far less accurate.

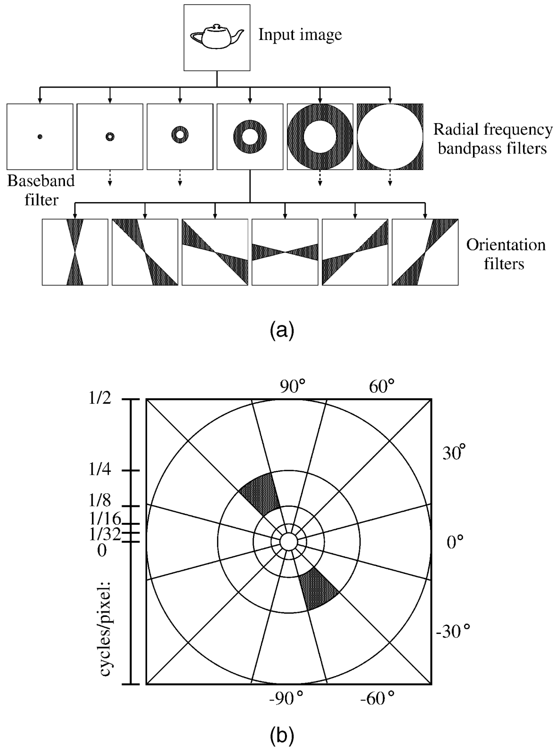


Fig. 1. Cortex transform: (a) organization of the filter bank and (b) decomposition of the image frequency plane into the radial and orientation selectivity channels. The filled regions show an example of the spatial frequencies allocated to a single channel. At the left side of (b), the spatial frequency scale in cycles per pixel is shown, which makes possible estimation of the bandpass frequencies of every radial channel. Also, the orientation of the band center in degrees is shown for every orientation channel.

Discrete Cosine Transform (DCT) [39], Differences of Gaussians (Laplacian) pyramid [22], and others. The choice of a particular filter bank seems not to be critical as long as it roughly approximates the point-spread function of the visual cortex neurons [23] and the frequency and orientation bandwidths are not too broad [35]. For all the above channel models, usually four to six bands are used, with each exhibiting three to six different orientations. In practical applications, the computational efficiency of a particular filter bank is a factor of primary importance. For example, Watson [39] promotes, in digital video applications, a metric based on the DCT because the channel decomposition can be obtained as a byproduct of MPEG compression and, on some platforms, the DCT computation can be supported by dedicated hardware. In this research, we use the Cortex transform developed by Daly [6], which is a pyramid-style, invertible, and computationally efficient image representation. In Fig. 1a, we show organization of the filter bank in the Cortex transform, which models the combined radial frequency and orientational selectivity of cortical neurons. After decomposing the input image into six frequency bands, each of these bands (except the lowest-frequency baseband) undergoes identical orientational selectivity processing. The resulting decomposition of the image frequency plane into 31 radial frequency and orientation channels is shown in Fig. 1b.

Temporal channels are usually modeled using just two channels [11], [22], [39] to account for transient (low pass) and sustained (band pass with a peak frequency around 8 Hz) channels [38]. A practical problem is computational cost and memory requirements involved in processing in the time domain (a number of consecutive frames must be considered). A support of about 150-400 milliseconds (5-13 frames) is usually assumed for temporal filters [39], [36], [45]. This choice is consistent with the experimental results reported by Watson and Ahumada [40] which show that the motion sensors in the human brain integrate over only a brief interval of time (less than 400 milliseconds) and further increase of the exposure duration has almost no effect on discrimination improvement.

There is general agreement on using the spatiotemporal CSF, which approximates the data from Kelly [18]. Fig. 2a shows changes of the human spatial CSF for stimuli of various temporal frequencies. The spatial sensitivity decreases at high spatial frequency for all temporal frequencies. Also, the spatial sensitivity falls at low spatial frequencies for a low temporal frequency (1 Hz), which is typical for the steady patterns as well. Such a loss of sensitivity cannot be observed for increasing temporal frequencies, in which case the shape of the spatial CSF curves changes. This results in the limited separability of the spatiotemporal CSF [18], which is separable (has the same shape up to a scale factor) only at high spatial and temporal frequencies [38]. In practice, spatial and temporal channels are modeled separately by a filterbank and the spatiotemporal interaction is then modeled at the level of respective gains of the filters [11], [39], [44], [36].

The spatial CSF shown in Fig. 2a was obtained for foveal vision. However, with the increase of retinal eccentricity, the contrast sensitivity falls rapidly (refer to Fig. 2b). The predictions of video quality metrics are tuned for foveal vision and are obviously too conservative for the whole frame since the eye can fixate only at its selected region. A practical problem is that it is difficult to predict in advance which frame region will be chosen by a viewer. Some consistency of viewers' gazes while watching the same video sequences has been reported [16], which suggests that there is some potential in considering the likely eye movements to reduce the requirements concerning local frame quality.

The vast majority of the existing video quality metrics limit visual masking considerations to spatial masking (refer to [14] for a comprehensive discussion of the spatial masking and related computational models). Yeh et al. [45] proposed a very simple model for the temporal masking involving just the global difference of the average luminances between two consecutive frames. It seems that the temporal masking is the most important at scene cuts [45] accompanied by dark-to-bright or bright-to-dark transitions. In such a case, the visibility thresholds are usually elevated for less than 100 milliseconds [4]. In this research, we ignore the temporal masking because, in our application (walkthroughs), the number of scene cuts is usually very limited. Moreover, since temporary losses of eye sensitivity usually affect at most two to three frames following the cut,

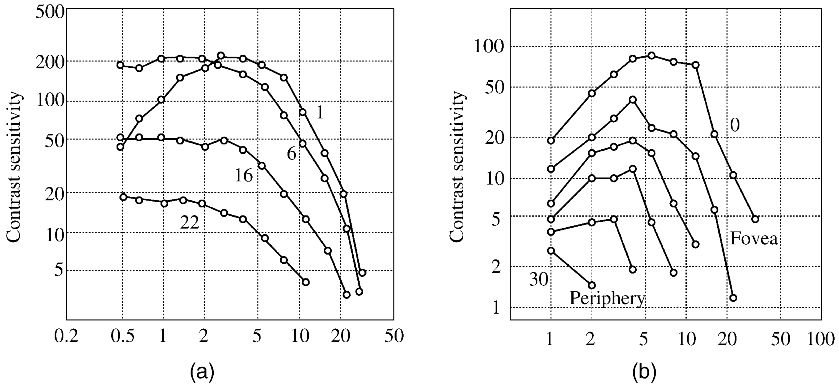


Fig. 2. (a) Spatial CSF measured using flickering grating stimuli of temporal frequencies 1, 6, 16, and 22 Hz. Reprinted with permission from [30] copyright (1966) Optical Society of America. (b) Spatial CSF measured using steady grating stimuli of small size for the following retinal eccentricities 0, 1.5, 4, 7.5, 14, and 30 degrees. The spatial frequency is expressed in cycles per degree (cpd) units. Reprinted with permission from *Nature* [31] copyright (1978) Macmillan Magazines Ltd.

only a very small reduction of the computation cost can be expected by exploiting temporal masking.

Lack of comparative studies makes it difficult to evaluate the actual performance of the discussed metrics. It seems that Sarnoff's Just-Noticeable Difference (JND) Model [22] is the most developed, while a DCT-based model proposed by Watson [39] is computationally efficient and retains many basic characteristics of the Sarnoff model [41]. In this research, we decided to use our own metric of animated sequence quality which takes advantage of the IF that is readily available in our application. For this purpose, we extended a static image quality metric proposed by Eriksson et al. [13], which we selected because of its suitable architecture and good performance for static images.

A majority of the discussed video quality metrics perform some form of color processing. Usually three separable pattern-color visual mechanisms are considered [11], [39], [44], [36] and the spatiotemporal filters are applied independently to each mechanism, which can effectively triple computational efforts. To reduce computation in the temporal domain, usually only the low pass channel is considered [11], [44] since the temporal sensitivity drops very quickly for chrominance [19]. Some computation savings can be made in spatial processing as well [11] since the spatial sensitivity for chrominance is very low at above eight cpd (cycles per degree) [19]. Color considerations, although still very costly, are an important part of a quality metric, especially in digital video applications in which lossy-compression of pattern-color components is performed independently. In our application, which involves IBR techniques, the pattern-color information is never separated and always undergoes the same processing (i.e., image warping and resampling). Taking into account much poorer acuity of color vision than pattern vision [19], we believe that, in our application, most of the animation impairments can be captured by achromatic processing. In this research, we ignore color processing for efficiency reasons and we leave for future work a more rigorous performance comparison of chromatic and achromatic video quality metrics in our application involving the perception-based guiding of in-between frame computation.

2.2 In-Between Frame Generation

Frame-to-frame coherence has been widely used in camera animation to speedup rendering (refer to [21] for discussion of the existing solutions). Early research focused mostly on speeding up ray tracing by interpolating images for views similar to a given keyframe [3], [2]. These algorithms involved costly procedures for cleaning up image artifacts such as gaps between pixels (resulting from stretching samples reprojected from keyframes to in-between frames) and occlusion (visibility) errors. Recently developed IBR techniques solve these problems more efficiently and are the usual choice in applications requiring free camera motion within an environment [32], [21], [26]. In our walkthrough applications, having a predefined animation path, even less general solutions are required because the camera motion is limited [24].

In this work, we apply well-known off-the-shelf IBR solutions suitable for in-between frame computations which are based on simple data structures and do not require intensive preparatory computations. We use a combination of the following standard techniques:

- To account for proper IF computation and occlusion relations, we select 3D warping and warp ordering algorithms developed by McMillan [25], which require just the reference image and the corresponding range data.
- To reduce gaps between stretched samples during image reprojection, we use the adaptive "splatting" technique proposed by Shade et al. [32].
- To remove holes resulting from occluded objects, we composite the two warped keyframes as proposed by Mark et al. [24]. Pixels depicting objects occluded in the two warped keyframes are computed using ray tracing.

This choice is the result of extensive analysis of the suitability of existing IBR techniques for walkthrough applications which we presented in [27]. Fig. 3 summarizes the processing flow for the in-between frame derivation using the techniques we selected.

In this paper, we focus on the important problem of keyframe selection along the animation path, which has not

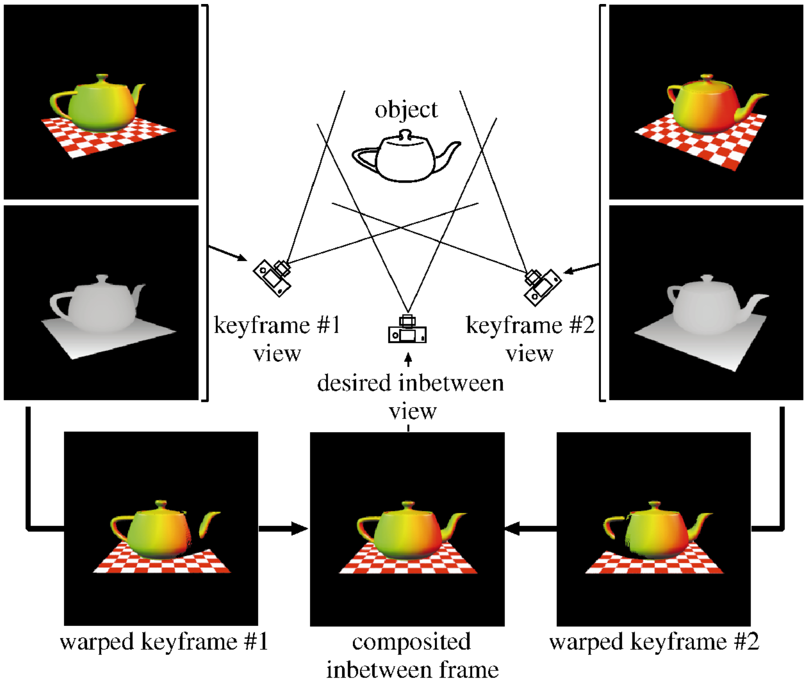


Fig. 3. Derivation of an in-between frame based on two keyframes and the corresponding range data (the distance is shown in grayscale). At first, the keyframes are 3D warped to the in-between frame viewpoint. Then, compositing of the keyframe warps is performed, accounting for the proper solution of occlusion problems.

attracted much attention in the IBR literature so far. Usually, keyframes are placed manually [9], uniformly distributed along the animation time code [24], or distributed in space at the nodes of some 2D or 3D lattice [5]. A notable exception is contained in the work done by Nimeroff et al. [28], who introduced a quality criterion which guides keyframe placement in space. At the initial step, the keyframe cameras were placed at the corners of the "view-space" and the corresponding images were computed. The quality criterion relied on computing the percentage of pixels whose item buffer identifiers were identical for all these images. If the percentage was below a predefined threshold value, the space was subdivided (the 2D case was investigated) and additional keyframe cameras were inserted. The quality criterion proposed by Nimeroff et al. seems not to be very reliable in the context of high quality rendering involving global illumination effects. For example, for scenes with mirrors and transparent objects, the distortion of reflected/refracted patterns is not estimated, while it can be quite significant in the in-between frames due to interpolation. From the standpoint of our application, rendering keyframes covering the whole view-space might not pay off in terms of the gains obtained during in-between frame computation along a predefined walkthrough path. Also, there is no guarantee that the quality of the resulting frames will be adequate for the human observer since the quality criterion does not take into account even the basic properties of human perception.

In this research, we investigate an automatic solution for placement of keyframes which improves both the IBR rendering performance and the quality of the animation as perceived by the human observer.

2.3 Image Flow Applications in Animation Rendering

The IF (also called the pixel flow [34], pixel tracing [33], or motion prediction [15]) found many successful applications in video signal processing [34] and animated sequence compression [15]. Also, some applications in computer animation have been shown. Zeghers et al. [46] used the linear interpolation between densely placed keyframes, which was performed along the IF trajectories. To avoid visible image distortions, only a limited number of in-between frames could be derived (the authors showed examples for one or three consecutive in-betweens only). Shinya [33] proposed the motion-compensated filtering as an antialiasing tool for animation sequences. Shinya derived the subpixel information improving the efficiency of the antialiasing for the image sequences by tracking a given sample point location along the IF trajectories. In his approach, Shinya emphasized temporal filtering (his filter has ideal antialiasing properties when its size is infinite), which lead to costly filters because of their very wide support. In practice, Shinya acquired temporal samples from 16-128 subsequent frames of animation. Shinya did not take into account perceptual considerations for moving visual patterns [43], which creates a possible trade-off between reducing the support of filters in the temporal domain and collecting the samples required for proper antialiasing in the spatial domain.

Zeghers et al. and Shinya used animation information to compute the IF between images, while visibility computations were performed explicitly for every pixel. Using IBR techniques, the IF computation is greatly simplified for walkthrough sequences and visibility is handled automatically.

3 Quality Metric for Animated Sequences

In this section, we propose a novel animation quality metric which is particularly suitable for synthetic image sequences. Before we move on to the description of our metric, we recall the well-known relationship between sensitivity to temporal fluctuations and moving visual patterns [38]. This relationship is used to justify the replacement of the central part of all state-of-the-art video quality metrics—the spatiotemporal CSF, with the spatiovelocity CSF, which is far more convenient to use in our application. Also, we discuss extensions to the spatiovelocity CSF derived in experimental settings with the stabilized retina toward more reliable sensitivity estimation for natural observation conditions.

3.1 Spatiovelocity vs. Spatiotemporal Considerations

Let $f(x, y, t)$ denote the space-time distribution of an intensity function (image) f, and v_x and v_y denote the horizontal and vertical components of the velocity vector $\vec{\mathbf{v}}$, which is defined in the xy plane of f. For simplicity, we assume that the whole image f moves with constant velocity $\vec{\mathbf{v}}$ and the same reasoning can be applied separately to any finite region of f that moves with a homogeneous, constant velocity [46]. The intensity distribution function $f_{\vec{\mathbf{v}}}$ of the image moving with speed $\vec{\mathbf{v}}$ can be expressed as:

$$f_{\vec{\mathbf{v}}}(x, y, t) = f(x - v_x t, y - v_y t, 0). \tag{1}$$

Let $F(\rho_1, \rho_2, \omega)$ denote the 3D Fourier transform of $f(x, y, t)$, where ρ_1 and ρ_2 are spatial frequencies and ω is temporal frequency. Then, the Fourier transform $F_{\vec{\mathbf{v}}}$ of the image moving with speed $\vec{\mathbf{v}}$ can be expressed as:

$$F_{\vec{\mathbf{v}}}(\rho_1, \rho_2, \omega) = F(\rho_1, \rho_2)\delta(v_x\rho_1 + v_y\rho_2 + \omega). \tag{2}$$

This equation shows the relation between the spatial frequencies and the temporal frequencies, resulting from the movement of the image along the image plane. We can see that a given flickering pattern, characterized by the spatial frequencies ρ_1 and ρ_2, and the temporal fluctuation ω, is equivalent to the steady visual pattern of the same spatial frequencies, but moving along the image plane with speed $\vec{\mathbf{v}}$ such that

$$\omega = v_x\rho_1 + v_y\rho_2 = \vec{\mathbf{v}} \cdot \vec{\rho}. \tag{3}$$

Equation (2) defines the relationship between temporal fluctuations and moving visual patterns, which is instrumental in understanding of the visual system sensitivity issues for these kinds of stimuli.

3.2 Spatiovelocity CSF Model

The spatiotemporal CSF of the visual system is widely used in multiple applications such as digital imaging systems dealing with motion imagery. One of the most commonly used analytical approximations of the spatiotemporal CSF is the formulas derived experimentally by Kelly [18]. Kelly measured contrast sensitivity at several fixed velocities for traveling waves of various spatial frequencies. Kelly found that the constant velocity CSF curves have a very regular shape at any velocity greater than about 0.1 degree/second. This made it easy to fit an analytical approximation to the contrast sensitivity data derived by Kelly in the psychophysical experiment. As a result, Kelly obtained the spatiovelocity CSF, which he was able to convert into the spatiotemporal CSF using (3).

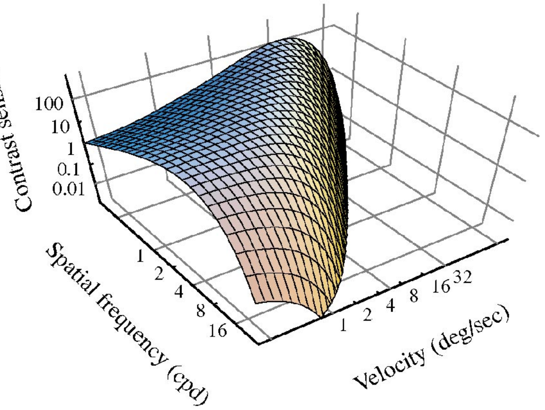


Fig. 4. Spatiovelocity Contrast Sensitivity Function.

Kelly originally designed his spatiovelocity CSF for displays of relatively low luminance levels (less that $20\,cd/m^2$). Daly [7] extended Kelly's model to accomodate the requirements of current displays (luminance levels of $100\,cd/m^2$ were assumed) and obtained the following formula:

$$CSF(\rho, v) = \\ c_0\left(6.1 + 7.3|\log\left(\frac{c_2 v}{3}\right)|^3\right)c_2 v(2\pi c_1\rho)^2 \exp\left(-\frac{4\pi c_1\rho(c_2 v + 2)}{45.9}\right), \tag{4}$$

where ρ is spatial frequency in cycles per degree, v is retinal velocity in degrees per second, and $c_0 = 1.14$, $c_1 = 0.67$, $c_2 = 1.7$ are coefficients introduced by Daly. Fig. 4 depicts the spatiovelocity CSF model specified in (4).

3.3 Eye Movements

Kelly performed his psychophysical experiments with stabilization of the retinal image to eliminate eye movements. Effectively, the retinal image velocity depended exclusively on the velocity of the visual pattern. However, in natural observation conditions, the spatial acuity of a visual system is affected also by eye movements of three types: smooth pursuit, saccadic, and natural drift. Tracking moving image patterns with smooth-pursuit eye movements makes it possible to compensate for the motion of the object of interest, which leads to a reduction of the retinal velocity and improving acuity. Smooth pursuit movements also make it possible to keep the retinal image of the object of interest in the foveal region in which the ability for resolving spatial details is the best. The smooth-pursuit eye movement is affected by saccades, which shift the eye's focus of attention and may occur every 100-500 milliseconds [29]. The saccadic eye movements are of very high velocity (160-300 deg/sec) and, during this motion, the eye sensitivity is effectively zero [7]. During intentional gaze fixation, drift eye movements are present and their velocity can be estimated at 0.15 deg/sec based on the good match between the CSF curve obtained by

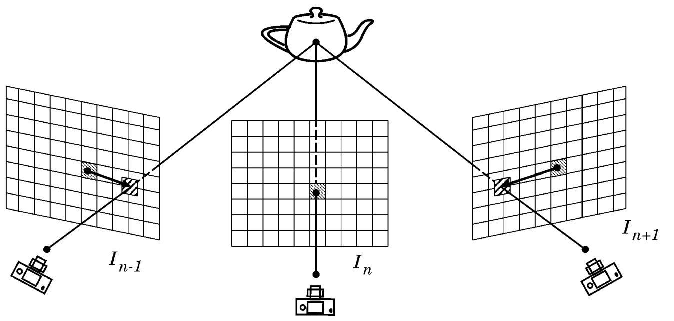


Fig. 5. The displacement vectors between positions of the corresponding pixels which represent the same scene detail in the subsequent animation frames I_{n-1}, I_n, and I_{n+1}. The planar image-warping equation [25] is used to derive the displacement vectors that are computed in respect to the pixel positions in I_n. The computation requires the range data for I_n and the camera parameters of all three involved frames. Note that the RGB values are not needed to compute the displacement vectors.

Kelly for stabilized retinal image and the normal, unstabilized CSF curves [18], [7].

Daly [7] pointed out that a direct use of spatiotemporal CSF as developed by Kelly usually leads to underestimating human vision sensitivity because target tracking by the eye movements is ignored. It is relatively easy to extend Kelly's spatiovelocity CSF model (4) to account for eye movements (it is far more difficult to perform such an extension directly for the spatiotemporal CSF). The retinal velocity v in (4) can be estimated as the difference between the image velocity v_I and the eye movement velocity v_E [7]:

$$v = v_I - v_E = v_I - \min(0.82 v_I + v_{Min}, v_{Max}), \qquad (5)$$

where $v_{Min} = 0.15$ deg/sec is the estimated eye drift velocity, $v_{Max} = 80$ deg/sec the maximum velocity of smooth eye pursuit, and the coefficient 0.82 is the experimentally derived efficiency of eye tracking for a simple stimulus on the CRT display [7].

In general, the estimate of retinal velocity given by (5) is very conservative because it assumes that the eye is tracking all moving image elements at the same time. However, it cannot be considered as the upper bound of eye sensitivity because, for lower spatial frequencies, the sensitivity may increase with increasing retinal velocity (refer to Fig. 4). In such a case, the best detection can occur when the eye is moving in the opposite direction as the moving visual pattern so that it effectively boosts the retinal velocity. Since it is difficult to predict actual eye movements, in practice, eye movement is completely ignored by a vast majority of existing video quality metrics [11], [22], [39].

To improve reliability of the eye sensitivity measurement, two retinal velocity estimates can be considered: 1) including an estimate of the smooth eye pursuit velocity using (5) and 2) ignoring the eye motion at all ($v_E = 0$), in which case $v = v_I$. The maximum value of the sensitivity for these two retinal velocity estimates should then be chosen.

The practical question that arises then is how to estimate local image velocity v_I. In our framework, the computation of v_I is equivalent to the IF derivation between neighboring frames in the animation sequence. For a given frame I_n, we compute v_I in respect to the previous I_{n-1} and subsequent I_{n+1} frames and, based on the obtained v_I values, we compute their average value to improve the accuracy of retinal velocity estimate. For every pixel of I_n, we apply the planar image-warping equation (refer to Section 3.3 in [25] for more details on this equation) to derive the corresponding image-space coordinates in I_{n-1} and I_{n+1}. This makes it possible to compute the displacement vectors between pixels in I_n and their new locations in I_{n-1} and I_{n+1} (refer to Fig. 5). Based on the displacement vectors, and knowing the time span between the subsequent animation frames (e.g., in the NTSC composite, video standard 30 frames per second are displayed), it is easy to compute the corresponding velocity vectors. Finally, the obtained v_I values, which are expressed in pixels per second, are converted into the visual degrees per second as required by (4) and (5).

In the following section, we describe the animation quality metric developed in this research. This metric requires the above-discussed estimates of retinal velocity in order to predict the eye sensitivity for moving visual patterns.

3.4 Animation Quality Metric

Before we move on to a description of our quality metric, let us justify an important design decision that we have made. As we discussed in Section 2.1, the spatiotemporal CSF is one of the most important components in virtually all state-of-the-art video quality metrics. However, we found that, in our application, it is more convenient to include the spatiovelocity CSF directly in our animation quality metric. The following reasons may justify our approach:

- The widely used spatiotemporal CSF was in fact derived from Kelly's spatiovelocity CSF, which was measured for moving stimuli (traveling waves).
- As Daly has shown [7], accounting for eye movements is more straightforward for a spatiovelocity CSF than for a spatiotemporal CSF.
- It is not clear whether vision channels are better described as spatiotemporal or spatiovelocity [18], [8]. It is an unresolved issue whether or not the aggregates of cell behavior are best described as spatiotemporal or spatiovelocity.
- The IF provides us directly with local velocity estimates for every frame.

As the framework of our Animation Quality Metric (AQM), we decided to expand the perception-based visible differences predictor for static images proposed by Eriksson et al. [13]. The architecture of this predictor was validated by Eriksson et al. through psychophysical experiments and its integrity was shown for various contrast and visual masking models [13]. Also, we found that the responses of this predictor are very robust and its architecture was suitable for incorporation into the spatiovelocity CSF.

Fig. 6 illustrates the processing flow of the AQM. Two comparison animation sequences are provided as input. For every pair of input frames I' and I'', a map of probability values is generated as output, which characterizes the difference perceivability. Also, the percentage of pixels with the predicted differences over the Just Noticeable Differences (JND) unit [22], [6] is reported. Each of the compared animation frames I' and I'' undergoes the identical initial processing. At first, the original pixel intensities are

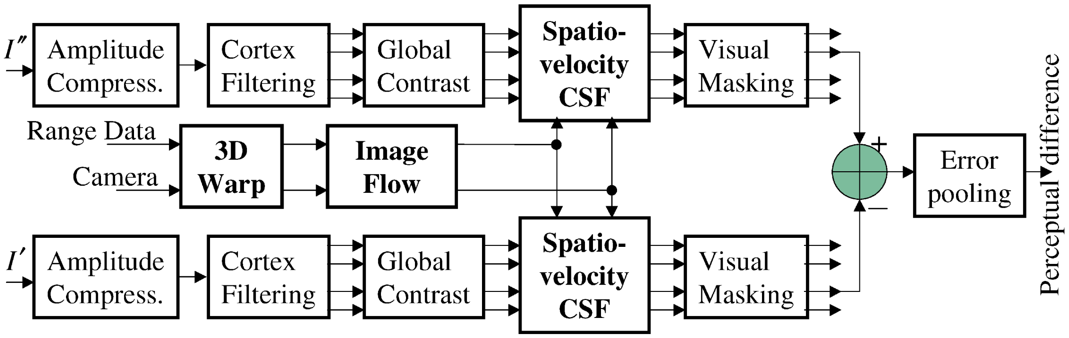


Fig. 6. Animation Quality Metric. The spatiovelocity CSF requires the velocity information for every pixel, which involves 3D warping and the IF computation (refer to Fig. 5 for details on the velocity derivation).

compressed by amplitude nonlinearity and normalized to the luminance levels of the CRT display. Then, the resulting images are converted into the frequency domain and decomposition into radial frequency and orientation channels is performed using the Cortex transform proposed by Daly [6] (refer to Section 2.1 for more details). Then, the individual channels are transformed back to the spatial domain and contrast in every channel is computed (the global contrast definition [13] with respect to the mean luminance value of the whole image was assumed). In the next stage, the spatiovelocity CSF is computed according to the Kelly model. The contrast sensitivity values are calculated using (4) for the center frequency ρ of each Cortex frequency band. The visual pattern velocity is estimated based on the average IF magnitude between the currently considered frame and the previous/subsequent frames (refer to Section 3.3 for details). Since the visual pattern is maximally blurred in the direction of retinal motion and spatial acuity is retained in the direction orthogonal to the retinal motion direction [12], we project the retinal velocity vector onto the direction of the filter band orientation. The contrast sensitivity values resulting from such processing are used to normalize the contrasts in every frequency-orientation channel into the JND units. Next, the visual masking is modeled using the threshold elevation approach [13]. The final stage is error pooling across all channels.

In this research, we apply the AQM to guide in-between frame computation and to adjust the parameters of our spatiotemporal animation quality enhancement technique. We discuss these AQM applications in the following two sections.

4 Rendering of the Animation

For animation techniques relying on keyframing, the rendering cost depends heavily upon the efficiency of in-between frame computation because the in-between frames usually significantly outnumber the keyframes. We use IBR techniques [25], [24] discussed in Section 2.2 to derive the in-between frames. However, the quality of pixels computed using these techniques can be deteriorated occasionally due to such reasons as occlusions in the keyframes of the scene regions that are visible in the in-between frames, specular properties of depicted objects, and so on. In the following section, we discuss our solutions to modifying bad pixels which could affect the animation quality as perceived by the human observer. One of the important factors toward reducing the number of bad pixels is selection of keyframes along the walkthrough trajectory. In Section 4.2, we propose an efficient method for initial keyframe selection which is specifically tuned for deriving in-between frames using IBR techniques. Finally, we describe our algorithm for adaptive keyframe selection, which is guided by the AQM predictions.

4.1 Quality Problems with In-Between Frames

The goal of our animation rendering solution is to maximize the number of pixels computed using the IBR approach without deteriorating the animation quality. However, the quality of pixels derived using IBR techniques is usually lower than ray-traced pixels, e.g., in the regions of in-between frames which are expanded in respect to the keyframe frames.

Human vision is especially sensitive to distortions in image regions with low IF velocities. As a part of our antialiasing solution (which we describe in more detail in Section 5), we replace IBR-derived pixels in such regions with ray-traced pixels. The replacement is performed when the IF velocity is below a specified threshold value, which we estimated in subjective and objective (using the AQM) experiments. In typical animations, usually only a few percent of the pixels are replaced, unless the camera motion is very slow.

Since specular effects are usually of high contrast and they attract the viewer's attention when looking at a video sequence [29], special care is taken to process them properly. In existing IBR methods, handling of nondiffuse reflectance functions requires very costly preprocessing to derive images of good quality. For example, a huge number of images is needed to obtain crisp mirror reflections [26], [21]. Because of these problems, we decided to use ray tracing for pixels depicting objects with strong specular or transparent properties. We use our AQM to decide for which objects with glossy reflectance properties such computations are required.

Pixels appearing in the in-between frames that are not visible in the keyframes cannot be properly derived using the IBR techniques and we apply ray tracing to fill in the resulting holes. An appropriate selection of keyframes, which we discuss in the following section, is an important factor in reducing the number of pixels which must be ray traced.

4.2 Initial Keyframe Placement

The selection of keyframes should be considered in the context of the actual technique used for in-between frame computation. Our goal is to find an inexpensive and automatic solution for the initial placement of keyframes which improves the IBR rendering performance. We assume a fixed number of initial keyframes and we want to minimize the number of pixels which cannot be properly derived from the keyframes due to visibility problems [25]. In this section, we focus on the initial keyframe placement, which is driven by the above objective. In Section 4.3, we discuss a further refinement of keyframe placement which is performed taking into account perceptual considerations and is guided by AQM predictions.

The occlusion problems in 3D warping usually are somehow dependent on the differences between the virtual camera parameters for the keyframes and in-between frames. A good and compact measure of such differences which does not involve any explicit camera parameter is the IF between these two frames. When the keyframe camera parameters are changed in such a way that pixel dislocations (which are proportional to the IF magnitude) are reduced, then the number of pixels which cannot be processed properly due to occlusion problems is also usually reduced. We used this observation in our approach to initial keyframe placement in which we reduce the variance in the average IF magnitude between keyframes. In our first attempt, we accumulated the average IF for all frames along the animation path and, by splitting the total accumulated IF value into equal intervals, we obtained the keyframe placement. This approach resulted in a well-balanced per frame distribution of pixels, which cannot be properly derived using IBR techniques. However, in the tests that we performed, the total number of such pixels was usually bigger than in the case of uniform keyframe placement along the time axis. We found that, for sequences with little difference in the IF magnitude between frames, the uniform keyframe placement is a good choice. However, when the IF variance within a sequence is high, better results can be obtained when limited dislocation of keyframes from uniform spacing is allowed to accommodate the IF variations. Let us assume that, in the initial phase, the spacing between keyframes is constant and equal to Δ. It is also assumed that maximal possible change of keyframe spacing is expressed by $\pm A\Delta$, where A is a fixed coefficient which usually takes values of $0.25 < A < 0.75$. Let $\bar{q}_i$ denote the actual accumulated IF magnitude between the currently considered pair of keyframes and let $\bar{Q}$ denote the average accumulated IF magnitude for all pairs of keyframes in the sequence. The fixed spacing Δ is assumed for $\bar{q}_i$ and $\bar{Q}$ computation. Then, the actual spacing δ_i for the currently considered animation segment can be computed as follows:

$$\begin{aligned} \alpha_i &= \frac{\bar{Q} - \bar{q}_i}{\bar{Q}} \\ \alpha_i &= \begin{cases} -A & \text{if} \quad \alpha_i < -A \\ \alpha_i & \text{if} \quad -A \le \alpha_i \le A \\ A & \text{if} \quad \alpha_i > A \end{cases} \\ \delta_i &= (1 + \alpha_i)\Delta. \end{aligned} \tag{6}$$

A new length δ_i is assigned to the currently processed segment and the procedure is repeated until the whole sequence is processed. According to our experience using this procedure, we are able to significantly reduce the percentage of pixels which cannot be properly generated by IBR techniques due to the occlusion problem. The cost of the procedure is negligible and the IF information used by this procedure is required nonetheless in the subsequent stages of our animation rendering, such as AQM processing and motion-compensated filtering.

4.3 Adaptive Refinement of Keyframe Placement

After selecting the initial frame placement, every resulting segment S of length $\delta = N + 1$ is processed separately through application of the following recursive procedure:

1. Generate the first frame I_0 and the last frame I_N in S using ray tracing. The keyframes that are shared by two neighboring segments are computed only once.
2. Derive two instances of the central in-between frame $I'_{[N/2]}$ and $I''_{[N/2]}$ for segment S by 3D warping [25] the keyframes:
   - I_0: $I'_{[N/2]} = 3DWarp(I_0)$, and
   - I_N: $I''_{[N/2]} = 3DWarp(I_N)$.
3. Use the AQM to compute the probability map P_{Map} with perceivable differences between $I'_{[N/2]}$ and $I''_{[N/2]}$.
4. Mask out from P_{Map} all pixels that must be ray traced because of the IBR deficiencies (discussed in Section 4.1). The following order for masking out pixels is taken:
   a. Mask out from P_{Map} pixels with low IF values (in Section 5.2, we discuss experimental derivation of the IF threshold value used for such masking).
   b. Mask out from P_{Map} pixels depicting objects with strong specular properties (i.e., mirrors, transparent and glossy objects). The item buffer [42] of frame $I_{[N/2]}$ is used to identify pixels representing objects with such properties. Only those specular objects are masked out for which the differences between $I'_{[N/2]}$ and $I''_{[N/2]}$, as reported in P_{Map}, can be readily perceived by the human observer. In Section 6, we provide details on setting the thresholds of the AQM response which are used by us to discriminate between the perceivable and imperceivable differences.
   c. Mask out from P_{Map} holes composed of pixels that could not be derived from keyframes I_0 and I_N using 3D warping.
5. If masked-out P_{Map} shows the differences between $I'_{[N/2]}$ and $I''_{[N/2]}$ for a bigger percentage of pixels than the assumed threshold value:
   a. Split S at frame $I_{[N/2]}$ into two subsegments S_1 $(I_0, \ldots, I_{[N/2]})$ and S_2 $(I_{[N/2]}, \ldots, I_N)$.

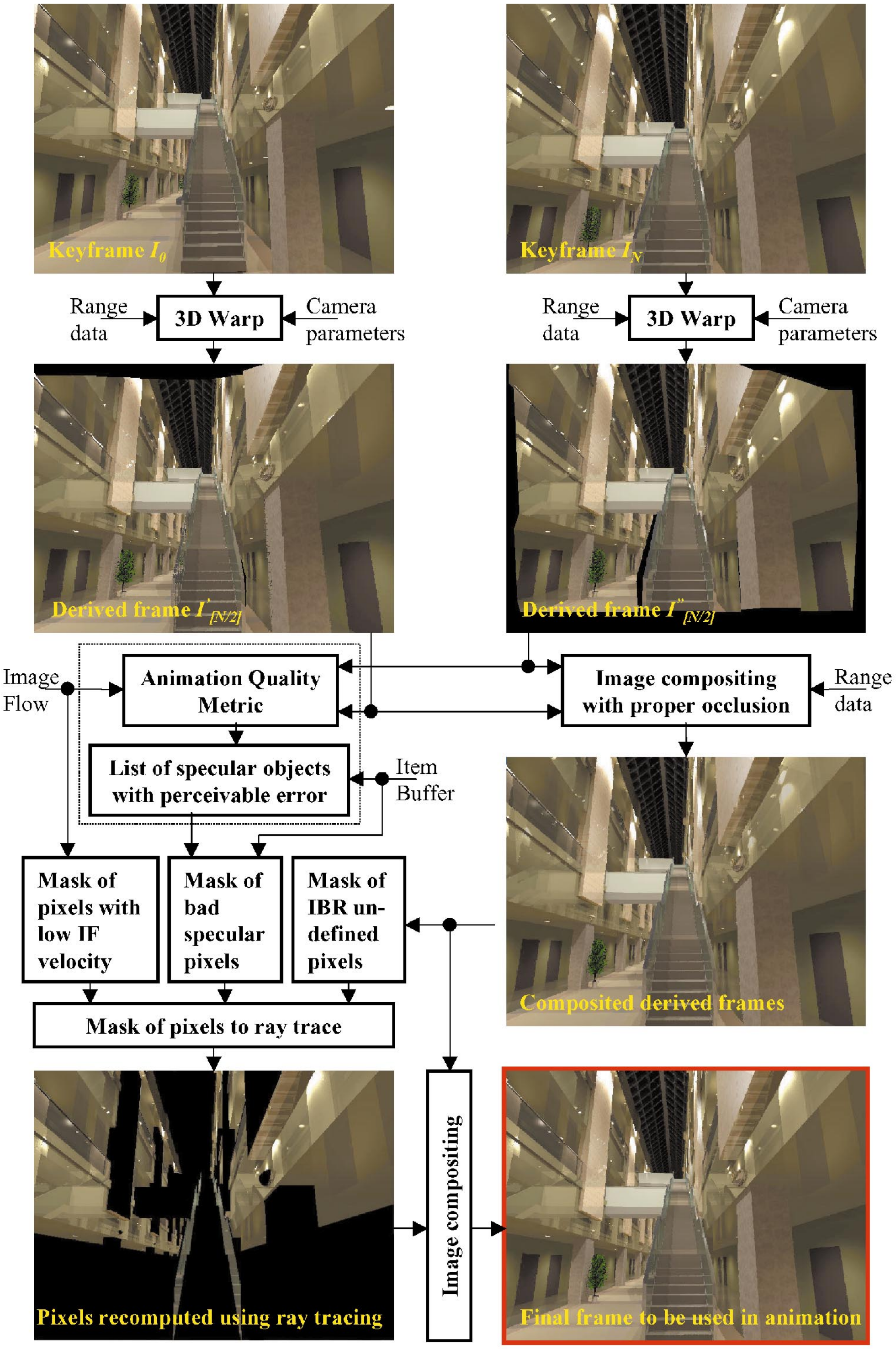

Keyframe I_0
Keyframe I_N
Range data
3D Warp
Camera parameters
Range data
3D Warp
Camera parameters
Derived frame $I'_{[N/2]}$
Derived frame $I''_{[N/2]}$
Image Flow
Animation Quality Metric
Image compositing with proper occlusion
Range data
List of specular objects with perceivable error
Item Buffer
Mask of pixels with low IF velocity
Mask of bad specular pixels
Mask of IBR un-defined pixels
Composited derived frames
Mask of pixels to ray trace
Image compositing
Pixels recomputed using ray tracing
Final frame to be used in animation

Fig. 7. The processing flow for in-between frames computation.

b. Process recursively S_1 and S_2, starting this procedure from the beginning for each of them.

Else

a. Composite $I'_{[N/2]}$ and $I''_{[N/2]}$ with correct processing of object occlusions [24], [32] to derive $I_{[N/2]}$.

b. Ray trace all pixels which were masked out in Step 4 of this procedure and composite these pixels with $I_{[N/2]}$.

c. Repeat two latter steps for all remaining in-between frames, i.e., $I_1, \ldots, I_{[N/2]-1}$ and $I_{[N/2]+1}, \ldots, I_{N-1}$ in S.

To avoid image quality degradation resulting from multiple resamplings, the fully ray-traced keyframes I_0 and I_N are always warped in Step 5c to obtain all in-between frames in S. Pixels to be ray traced, i.e., pixels with low IF values, pixels depicting specular objects with visible differences (such objects are selected once for the whole S in step 4b), and pixels with holes resulting from the IBR processing, must be identified for every in-between frame separately.

We evaluate the AQM response only for frame $I_{[N/2]}$. We assume that derivation of $I_{[N/2]}$ applying the IBR techniques is the most error-prone in the whole segment S because its arclength distance along the animation path to either the I_0 or I_N frames is the longest one. This assumption is a trade-off between the time spent for rendering and for the control of its quality (we discuss the costs of AQM in Section 6), but, in practice, it holds well for typical animation paths.

Fig. 7 summarizes the computation and compositing of an in-between frame. We used a dotted line to mark those processing stages that are performed only once for segment S. All other processing stages are repeated for all in-between frames.

As a final step, we perform our spatiotemporal antialiasing. To speed up the rendering phase, all pixels (including those that have been ray traced) are not antialiased until the last stage of processing. The following section discusses our antialiasing solution in more detail.

5 Animation Quality Enhancement

Composing still images of high quality into an animated sequence might not result in an equally high quality of animation because of possible temporal artifacts. On the other hand, proper temporal processing of the sequence makes it possible to relax the quality of frames without a perceivable degradation in the animation quality, which effectively means that simpler and faster rendering methods can be applied. In this section, we propose an efficient spatiotemporal antialiasing technique which makes it possible to replace the traditionally used supersampled pixels by raw pixels derived using IBR techniques or ray tracing (one sample per pixel only) without perceivable losses in the animation quality. Our antialiasing technique makes intensive use of the IF. In this context, we discuss some technical problems with IF validity for objects that occlude each other and for objects with specular reflectance properties.

5.1 Spatiotemporal Antialiasing

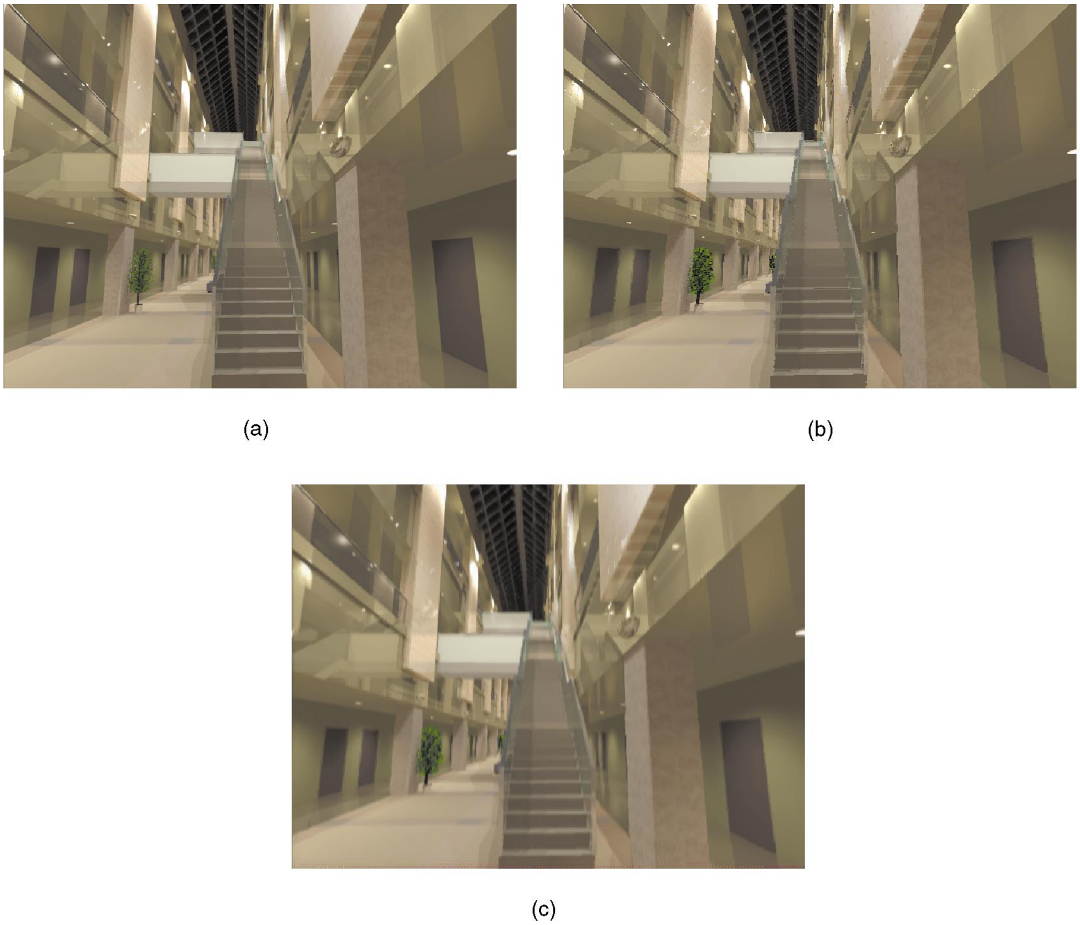
It is well-known that aliasing affects the quality of images generated using rendering techniques. This also concerns images obtained using IBR methods which may additionally exhibit various kind of discontinuities (such as holes resulting from visibility problems). These discontinuities can be significantly reduced using techniques like splatting and the image compositing introduced above, but, nonetheless, in many places in the resulting images, instead of smooth transitions, jagged unwanted edges and contours will be easily perceivable (refer to animation samples posted at [1]).

Aliasing is also inherent in all raster images with significant content. Images obtained in computer graphics or, in general, all digital images, are the sampled versions of their synthetic or real world continuous counterparts. Sampling theory states that a signal can be properly reconstructed from its samples if the original signal is sampled at the Nyquist rate. Due to limited resolution of output devices, such as printers and, especially, CRTs, the Nyquist rate criterion in computer graphics is rarely met—and the image signal cannot be represented properly with a restricted number of samples.

From the point of view of signal processing theory, the discontinuities and aliasing artifacts described above are high frequency distortions. This suggests the possibility of replacing the traditional, computationally expensive antialiasing techniques, like unweighted and weighted area sampling and supersampling, by an appropriate image processing method. Shinya [33] noticed that the subpixel information required for antialiasing can be derived in the time domain by tracking a given sample point location along the IF trajectories. This approach is based on the fundamental principle of ideal motion compensation, which requires that the intensity of a pixel remains unchanged along a well-defined motion trajectory [34]. Shinya used temporal filters of very wide and fixed support. In our research, we have found that by treating both aspects, spatial and temporal, in a balanced way and by adapting the filter support size as a function of IF, we were able to improve both the quality and the efficiency of antialiasing. We have obtained a very efficient antialiasing and image quality enhancement method based on low pass filtering using spatial convolution. Spatial convolution is a neighborhood operation, i.e., the result at each output pixel is calculated using the corresponding input pixel and its neighboring pixels. For the convolution, this result is the sum of the products of pixel intensities and their corresponding weights from the convolution mask. The values in the convolution mask determine the effect of the convolution by defining the filter to be applied. Those values are derived from the point spread function of the particular filter (in the case of low pass filtering, typically it will be the Gaussian function). In our case (i.e., the case of a sequence of images composing an animation), we have to consider not only the spatial but also the temporal aspect of aliasing and discontinuities. The proper way of solving the problem is to filter the three-dimensional intensity function (composed of a sequence of frames) not along the time axis, but along the IF introduced earlier in this paper (results and differences between those approaches can be seen on animation samples posted at [1]). Such a filtering approach, known also as the motion compensated filtering, was used earlier in video signal processing [34], image interpolation [46], and image compression [15].

5.2 Selection of Filter Parameters

It is well-known that low pass filtering as a side effect causes blurring and, in fact, a loss of information in the processed signal or image. In our case, we have to consider that the content and the final quality of the resulting animation is to be judged by a human observer. We were quite fortunate to find that, with the pixels velocity increase, there is an increase in perceived sharpness (see also [43]). For example, an animation perceived as sharp and of good quality can be composed of relatively highly blurred frames, while the same frames observed as still images would be judged as blurred and unacceptable by the human observer. Our antialiasing approach is based on this



(a)

(b)

(c)

Fig. 8. Selected animation frame computed using: (a) ray tracing with antialiasing, (b) compositing of IBR-derived and ray-traced pixels as described in Section 4, and (c) as in (b), but processed by our 3D antialiasing solution. Note that frames in (a) and (c) are visually indistinguishable when observed within the animation context.

perceptual phenomena and takes advantage of compensation by the visual system of excessive blurring introduced by low pass filtering in animation frames. Fig. 8a and Fig. 8b show the result for a single frame from an animation sequence that was obtained using ray tracing with antialiasing and that from our technique of in-between frame computation described in Section 4. Fig. 8c shows the frame depicted in Fig. 8b, which was processed using our spatiotemporal antialiasing. Although the frames in Fig. 8a and Fig. 8c exhibit many perceivable differences when observed as still images, they are visually indistinguishable when observed within animation sequences.

A practical problem that arises is how to select the filter support size to avoid perceivable image blurring and, at the same time, efficiently remove the spatial and temporal aliasing. To solve this problem properly, the ability to resolve spatial details by the human observer for objects moving in the image plane should be taken into account. The object's velocity may vary from zero for still frames or their fragments, under which conditions the eye sensitivity is very high, to tens of visual degrees per second when the eye is hardly able to resolve any spatial detail. Moreover, in walkthrough applications, the velocity of different objects' motions can be quite different within a single frame; thus, a local adaptive approach to the selection of filter support size should be considered. In summary, a filtering solution taking into account all these characteristics of human perception for still and moving visual patterns requires a kind of *continuum* approach between traditional 2D and spatiotemporal 3D antialiasing techniques.

The problem of antialiasing for still images is well-elaborated in computer graphics and image processing literature. However, it is not clear how to adapt antialiasing parameters when the visual pattern starts to move and changes its velocity. Our intention was to investigate this problem experimentally, taking into account basic characteristics of human vision and using AQM for objective predictions of the resulting animation quality.

Before we move on to a more detailed description of our antialiasing technique, let us make some simple observations which lay the foundation of our approach. Filtering of still visual patterns by processing neighboring pixels may lead to excessively blurry images that are objectionable for the human observer. Subpixel information is usually required to accommodate the high sensitivity of the eye when observing still images. Since such subpixel information, required for proper antialiasing, cannot be derived in the time domain, some traditional approaches such as adaptive supersampling or jittered sampling must be considered. Fortunately, computation of high quality still frames in animation using the traditional techniques is not a problem, especially if the same image can be duplicated and used for multiple animation frames. The eye is highly sensitive to very slowly moving patterns as well, but there is a chance that adequate samples can be collected in the temporal domain. Thus, for slow IF velocities, the temporal filter support should change adaptively. A practical problem here is that due to the limited accuracy of IF and intensity values, excessive expansion of the filter support in the time domain results in an accumulation of errors, which may cause image artifacts perceivable by the human observer. When the velocity of moving patterns increases, the eye sensitivity for high spatial frequencies decreases and samples required for antialiasing can be collected in the spatial domain as well. When the spatial filter support is selected properly as a function of the pattern velocity, the image blurriness resulting from filtering neighboring pixels within a frame cannot be perceived. In such conditions, the temporal domain is not the only source of samples and temporal filter support can be limited to the size which results in proper antialiasing. As the visual pattern velocity further increases, expansion of the spatial filter support becomes possible. For quickly moving objects, instead of expanding filter support, more sparse image sampling could be performed and then, at the filtering stage, intensity of nonsampled pixels could be interpolated. This possibility could be especially attractive for rendering techniques involving the high cost of image sample computation (e.g., ray tracing). In this research, we do not explore reduction of sample density for quickly moving visual patterns because, in this case, deriving samples using the IBR approach is inexpensive and more reliable.

A practical problem that arises is how to tune the size of spatial and temporal filter supports, taking into account the above observations, in order to get a reliable antialiasing technique which performs well for various animations. We feel that the continuum approach to spatial and spatiotemporal filtering as we outlined in the previous paragraph, is a research topic in itself which requires separate further in-depth treatment. In this research, we attempt to provide a practical antialiasing solution which works well for typical animation sequences.

Before we move on to the discussion of our filter settings, we want to address the accuracy problem of the IF and pixel intensity information. Obviously, the density of the IF and the intensity samples is limited by the frame resolution. Effectively, the exact IF values are known for the centers of pixels only. This is also the case when intensity is computed using ray tracing (one sample per pixel). When the IBR technique is used, the exact intensity value corresponds to some point within the selected pixel on the splat center, so, in fact, the reliability of the intensity information is much worse. Since, for temporal processing, subpixel accuracy is required, the IBR-pixels accuracy might not be sufficient for slowly moving visual patterns when eye sensitivity is extremely high. In such image regions, replacement of the IBR-derived pixels by ray-traced pixels should be considered.

Another issue is the computation of sample point positions along the IF trajectory with subpixel accuracy. We assume that the IF information which is available for every frame and every pixel is derived with respect to the next/previous frame and that it is stored as a floating point value. To derive the IF between a pair of nonneighboring frames, the IF for all intermediate frames must be accumulated along the IF trajectory. Since the exact IF values are stored only for pixel centers, to derive the IF value for an arbitrary point within a frame, we perform a bilinear interpolation of the IF using the centers of four pixels in the proximity of this point. When the position of the sample point in the next/previous frame is established based on the interpolated IF, the intensity at this point is also bilinearly interpolated. Such an intensity value is used for temporal filtering.

The following problems should be addressed to make our spatiotemporal antialiasing technique workable: 1) controlling the temporal and spatial filter support as a function of moving visual pattern velocity v_{pf}, which is derived from the IF, and 2) selecting the upper velocity threshold v_t for slowly moving visual patterns which always require ray-traced pixels. All filter support settings given below are expressed in pixels, assuming that the CRT observation distance was 0.5 meter, the image diagonal was 0.25 meter, and the image resolution is 640×480 (i.e., 1 visual degree corresponds to about 28 pixels).

The size of the temporal filter support is decided, keeping in mind the trade-off between excessive animation blurring and the reduction of temporal aliasing. Obviously, the reduction of the filter support size improves computation efficiency as well. Subjectively, we found that a size of 11 frames is a good trade-off which results in visually pleasant animations. The only exceptions are frame regions with slowly moving visual patterns. In such regions, expanding the support size is the only way to increase the number of samples used for antialiasing. We adaptively expand the support size over 11 frames when the IF velocity is below 0.15 degrees/second, which, for our observation conditions specified above, is equivalent to 0.14 pixel/frame (assuming 30 frames/second). For low IF velocities, we estimate the size of the temporal filter support by processing the subsequent frames until the accumulated IF magnitude reaches the size of half a pixel, which means that the collected samples roughly cover the pixel area. However, due to excessive accumulation of IF errors which affect the accuracy of motion compensation and result in blurry images, we limit the maximum size of our temporal filter to 15 frames.

We decided to tune the spatial filter support manually, keeping in mind the trade-off between image aliasing and blurring. Subjectively, we obtained good visual results with the following settings:

- For $0 < v_{pf} \leq 16$ degrees/second, the filter support size should be linearly changed from 3×3 to 9×9 pixels.
- For $v_{pf} > 16$ degrees/second, a filter size of 11×11 pixels was used because the wider filter support did not introduce perceivable changes in the animation quality. As we discussed above, for such quickly moving visual patterns, the sampling density during rendering can be relaxed, but, in our application, we did not expect significant gains because the IBR

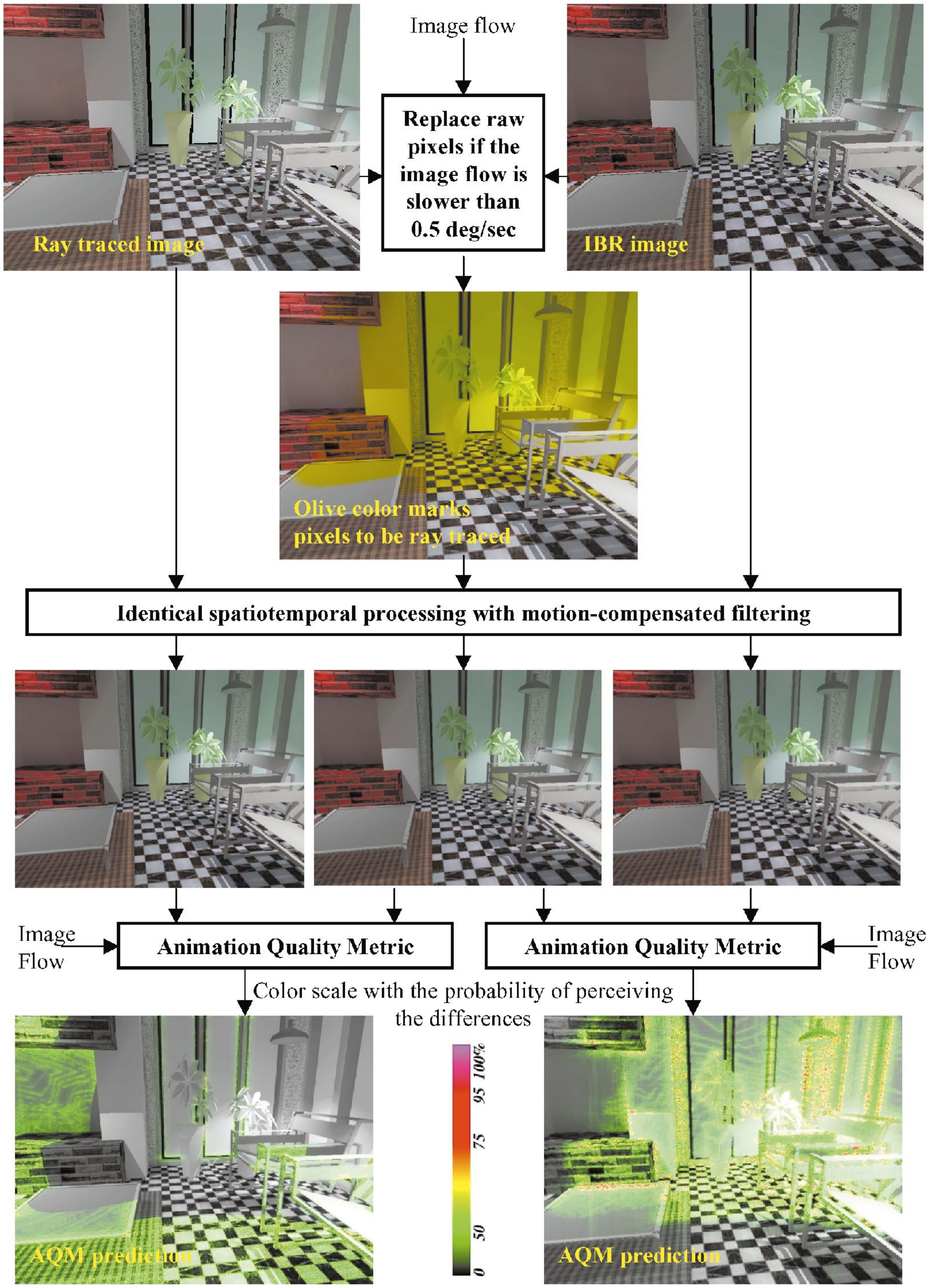


Image flow
Replace raw pixels if the image flow is slower than 0.5 deg/sec
Ray traced image
IBR image
Olive color marks pixels to be ray traced
Identical spatiotemporal processing with motion-compensated filtering
Image Flow
Animation Quality Metric
Animation Quality Metric
Image Flow
Color scale with the probability of perceiving the differences
0
50
75
95
100%
AQM prediction
AQM prediction

Fig. 9. Experimental settings for estimating the upper threshold of IF velocity which is used to identify image regions that require ray-traced pixels to avoid deterioration of the animation quality as perceived by the human observer.

technique we used generates samples at very low cost.

We validated these settings objectively as well as by using the AQM. We computed the perceivable differences between 1) the fully ray traced animation (one sample per pixel) and 2) the IBR-based animation with occlusion problems fixed using ray traced samples. The resulting two sequences were spatiotemporally processed using the above filter settings. The only perceivable differences that were predicted by the AQM were located in the image regions with very small v_{pf} values. The differences in these image regions can be eliminated by replacing IBR-derived pixels by ray-traced pixels. To make such a replacement automatic, the velocity threshold v_t should be appropriately adjusted and pixels with $v_{pf} < v_t$ should be replaced.

We used the experimental setting shown in Fig. 9 to estimate a reliable value of v_t. Two input images were considered: I_{rt} with all pixels ray-traced and I_{ibr} with IBR-

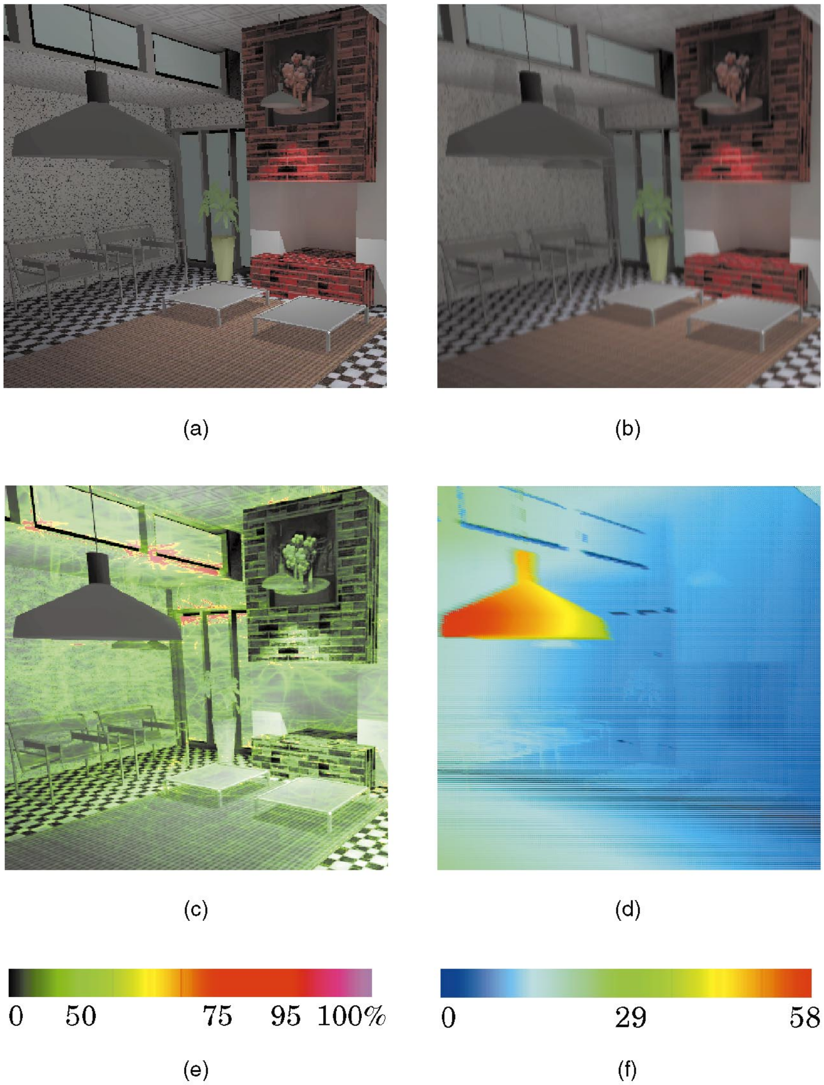


Fig. 10. Image flow separation problem for occluding objects: (a) ray-traced frame (one sample per pixel) without temporal processing, (b) ray-traced frame resulting from our spatiotemporal processing, (c) the AQM prediction of visible differences between the two frames in the context of animation, (d) the corresponding IF velocity, (e) the color scale used to encode the probability of detecting the differences, and (f) the color scale used to encode the velocity in degree/second units.

based pixels (the IBR occlusion problems were fixed with ray traced pixels). Next, based on the IF and for a given value of v_t, we composed a new image I_c with pixels taken from I_{rt} when $v_{pf} < v_t$ and with pixels acquired from I_{ibr} otherwise. All three images undergo identical spatiotemporal processing using motion-compensated filtering. The goal of this processing is to prepare these frames in the same way as the final frames in our animation were prepared. The AQM is used to compare I_{rt} and I_c and to check whether the visible differences do not appear in the regions which are not ray-traced. When such differences are predicted, v_t must be increased to include more ray-traced pixels in I_c. The AQM is also used to find the visible differences between I_{ibr} and I_c. If the differences are only reported for velocity values v_{pf} significantly lower than the velocity limit v_t, then this limit can be reduced without impairing the animation quality. This means that the rendering cost can be reduced as well. The described procedure was performed repeatedly for various settings of v_t and for various scenes with various v_{pf} distributions. We found that $v_t \approx 0.5$ degree/second favorably accommodates the required AQM predictions for all scenes we tested. For some scenes, this threshold is too conservative; however, we decided to use this value for the sake of robustness of our animation rendering system. In our discussion, we assumed that v_t is the same for the whole animation. Important savings could be expected when the v_t value could be adaptively changed across every frame as a function of visual pattern complexity. We leave such investigations as future work.

5.3 Image Flow Separation

There are some practical problems with collecting the proper samples along the IF trajectory due to occlusions between objects. Effectively, samples representing different objects can be considered during temporal filtering, which leads to improper sample blending. On the other hand, when samples representing different objects will be separated (e.g., using the item buffer and by comparing their z-values), the temporal filter support can be reduced to single samples which might compromise the antialiasing quality.

To illustrate this problem, let us consider the relevant fragment of the animation frame shown in Fig. 10a in which the foreground lamp occludes the background portions of the wall and ceiling. In this animation sequence, the virtual camera moves quite fast (Fig. 10d depicts the corresponding IF velocity) and it is located close to the lamp. In this particular configuration of the camera, foreground occluder, and background objects, the motion parallax effect is very strong, which results in significant occluding of different portions of the wall and ceiling by the lamp in neighboring frames. Now, if some portion of the wall that is visible in the current frame is obscured by the lamp in the previous/subsequent frames, the temporal processing along the IF trajectories causes an improper blending of pixels. Fig. 10b shows results of such motion-compensated filtering which ignores the IF separation of different objects. The "ghosting" effect is clearly visible on the wall in the foreground lamp proximity. To prevent such improper blending, Shinya [33] applied a quite involved and costly process to separate the IF for different objects that occasionally "cross" their paths at some pixels. Shinya used temporal filters with extremely wide support, so the involvement of the IF separation procedure was quite likely for many pixels.

In the animation examples we investigate in this study, we were not able to perceive animation artifacts caused by ignoring the IF separation, even though we specifically knew in advance in which image regions they should be expected based on the preview of still frames. These subjective observations were also confirmed by objective measurements using our AQM. We compared the animation quality between the two sequences (we used ray-traced sequences to eliminate the possibility of artifacts resulting from IBR-derived pixels): 1) ray traced images (one sample per pixel) without any temporal processing and 2) ray traced images with our spatiotemporal filtering, but without the IF separation. The corresponding frames are presented in Fig. 10a and Fig. 10b. The AQM response which is shown in Fig. 10c does not reveal any differences between the two sequences in the region of the ghost appearance. The results of our subjective and objective experiments can be explained more intuitively as follows: The ghosting effect in the temporally processed frames always appears along the IF trajectories and can be considered as a form of motion blur. Because of such temporal coherence, the ghosts are not so objectionable to the viewer as they could be if appearing unexpectedly in the image space (refer to the example of improper mirror reflection pattern described in Section 5.4, which can be readily perceived for animation). For a quickly moving camera, the motion parallax effect can be strong for some object configurations; however, eye sensitivity is reduced and the image artifacts, which can be readily seen in the still frames, cannot be perceived in the animation. On the other hand, for a slowly moving camera when the eye is more sensitive, the motion parallax effect between subsequent frames is weak and the "ghosting" effect appears as a slightly fuzzy boundary on occluding objects. Again, these fuzzy boundaries can usually be perceived for the still images, but the apparent boundary sharpness "improves" in the context of animation [43]. Obviously, the fuzziness of boundaries is a function of the temporal extent of the filter support which, in our solution, is fortunately quite narrow.

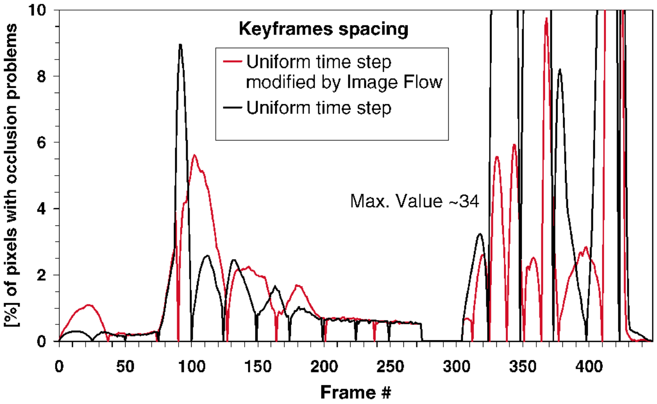


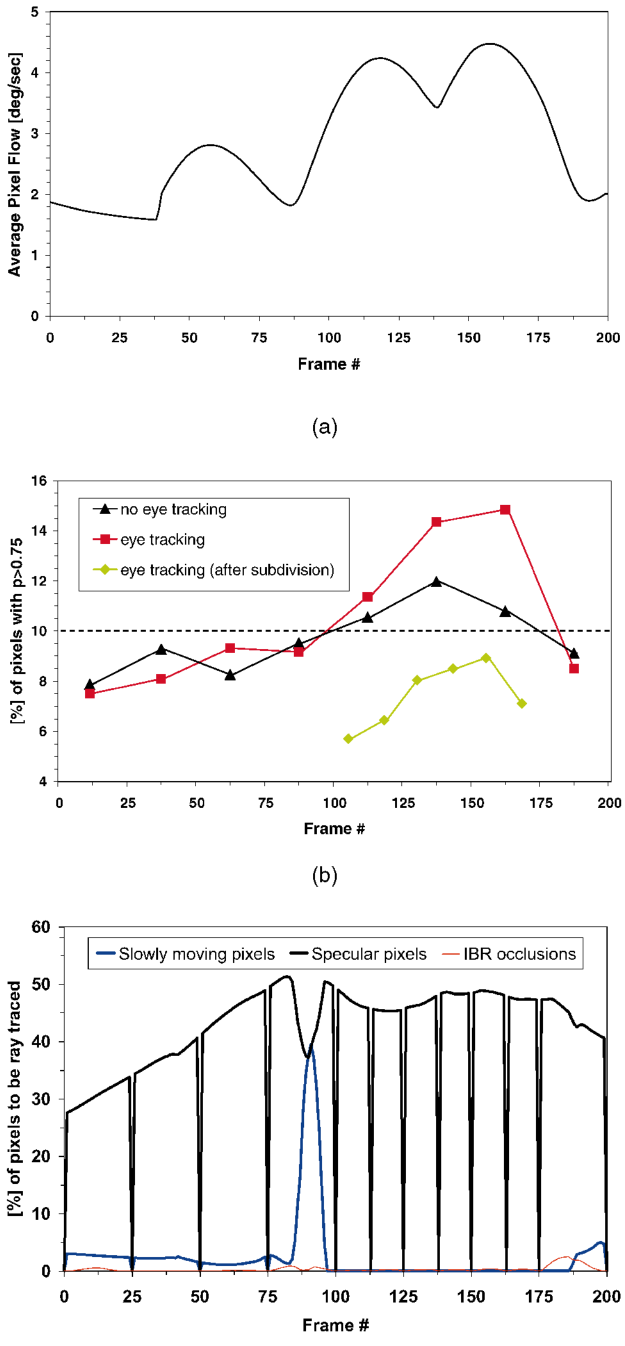
Fig. 11. Selection of the initial keyframes for the ROOM walkthrough.

5.4 Image Flow for Specular Surfaces

The drawback of our current motion-compensated filtering (as well as the other solutions [33], [46], [34]) is the incorrect processing of directional lighting effects, which are especially objectionable for crisp mirror reflections. The objective measure which compares animation frames with and without motion-compensated filtering using our AQM also confirmed these subjective results. Indeed, the motion of the reflected/refracted patterns over the specular surfaces as a function of camera motion does not correspond to the motion of these surfaces in the image plane which is described by the IF. Since the estimation of the optical flow for reflections and refractions is quite involved, we used a simple heuristic relying on the reduction of the size of temporal filter support for objects with strong directional reflectance/refraction properties. While, in the still frame, some reflection artifacts can be seen, in the context of animation they become imperceivable, which was confirmed both by the objective and subjective measures. We found that the heuristic worked well in the walkthroughs that we tested. However, more systematic investigation of the optical flow for specular surfaces would be required, which we leave as a topic for future work.

6 Results

As the case study in this research we selected a walkthrough animation for two different scenes: the ATRIUM of the University of Aizu [1] and a ROOM. Selected frames from the ATRIUM scene are shown in Fig. 7 and Fig. 8 and from the ROOM scene in Fig. 9 and Fig. 10. The main motivation for this choice was the interesting occlusion relationships between objects which are challenging for IBR. In the case of the ATRIUM scene, a vast majority of the surfaces exhibit some view-dependent reflection properties, including the mirror-like and transparent surfaces, which made in-between frames calculation more difficult. Under

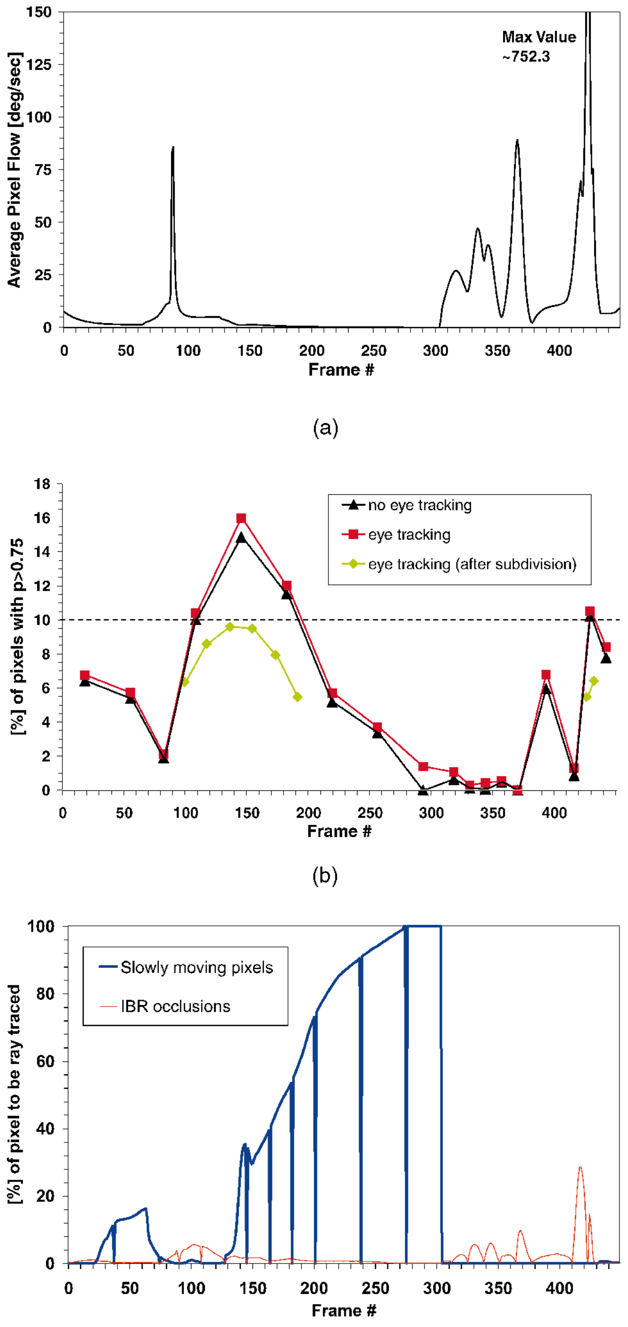


(a)

(b)

(c)

Fig. 12. ATRIUM walkthrough statistics: (a) the average IF velocity expressed in degree/second units, (b) the AQM prediction of the perceived differences between the warped images of two neighboring keyframes (taking into account various retinal image velocity), (c) the percentage of pixels to be recalculated by ray tracing. In (b), lines connecting the symbols were added for figure readability; they do not have any meaning for the unmarked frames.



(a)

(b)

(c)

Fig. 13. ROOM walkthrough statistics: (a) the average IF velocity expressed in degree/second units, (b) the AQM prediction of the perceived differences between the warped images of two neighboring keyframes (taking into account various retinal image velocity), (c) the percentage of pixels to be recalculated by ray tracing. In (b), lines connected the symbols were added for figure readability; they do not have any meaning for unmarked frames.

such conditions, the AQM guided selection of keyframes and glossy objects within in-between frames to be recomputed was more critical and wrong decisions concerning these issues could be easy to perceive. For the ROOM scene, we disabled specular properties and we designed an animation path which causes great variation in the IF velocity. Our goal was to investigate the performance of our animation rendering solution for the conditions in which eye sensitivity changes dramatically.

For our experiments, we selected a walkthrough sequence of 200 frames for the ATRIUM and 448 frames for the ROOM. The resolution of each frame was 640×480 (to accommodate for the NTSC standard). At the initial keyframe selection step (refer to Section 4.2), we assumed an animation segment length of $\Delta = 25$ frames. For the

TABLE 1
Statistics of the Ray Traced Pixels in the ATRIUM and ROOM Walkthroughs

Scene	Slow motion [%]	Specular objects [%]	IBR occlusions [%]	Keyframes [%]	Total [%]
ATRIUM	2.4	40.8	0.3	6.0	49.5
ROOM	28.1	0.0	1.9	5.1	35.1

ATRIUM walkthrough, we kept the length of every segment fixed, i.e., $\delta_i = \Delta$, because, as can be seen in Fig. 12a, changes in the average IF velocity computed for every frame are relatively small. For the ROOM scene, the average IF velocity variates significantly (refer to Fig. 13a), so we adjusted the length of every segment δ_i using the algorithm presented in Section 4.2. The goal of such adjustment was reduction in the percentage of pixels with occlusion problems which arise in IBR techniques. Fig. 11 shows distribution of the percentage of improperly IBR-derived pixels per frame (these pixels have to be recomputed using ray tracing) for two keyframe placement methods: the uniform time step and the uniform time step modified by IF with the coefficient $A = 0.5$ in (6). The average percentage per frame of recomputed pixels was, respectively, 4.04 percent and 2.00 percent. The keyframe placement obtained in the latter case (in which we obtained over 50 percent reduction of recomputed pixels compared with the other solutions) was used in further experiments with the ROOM sequence.

As described in Section 4.3, for every segment S, we run the AQM once to decide upon the specular objects which require recomputation. The AQM is calibrated in such a way that 1 JND unit corresponds to a 75 percent probability that an observer can perceive the difference between the corresponding image regions (such a probability value is the standard threshold value for discrimination tasks [6]). If a group of connected pixels representing an object (or a part of an object) exhibits differences greater than 2 JND (93.75 percent probability of discrimination), we select such an object for recalculation. If for an object the differences below 2 JND are reported by the AQM, then we estimate the ratio of pixels exhibiting such differences to all pixels depicting this object. If the ratio is bigger than 25 percent, we select such an object for recomputation—25 percent is an experimentally selected trade-off value which makes possible a reduction in the number of specular objects requiring recomputation at the expense of some potentially perceivable image artifacts. These artifacts are usually hard to notice unless the observer's attention is specifically directed to the given image region. Visual sensitivity is high only in the region of approximately 5.2 visual degrees[2] due to foveal vision [37], while the sensitivity decreases significantly for the remaining image regions which are perceived by means of peripheral vision (refer to Fig. 2b illustrating the eccentricity effect). This means that the AQM predictions which are tuned for foveal vision might be too conservative for many image regions.

After masking out the pixels to be recomputed, the decision for further splitting S is made using AQM predictions for the remaining pixels. The predictions are expressed by the percentage of unmasked pixels for which the probability p of detecting the differences is greater than 0.75. Based on experiments that we conducted, we decided to split every segment S when the percentage of such pixels is bigger than 10 percent. When computing the AQM predictions that we used to decide upon segment splitting, we assumed good tracking of moving visual patterns with smooth-pursuit eye movements (the retinal velocity is computed using (5)). The filled squares in Fig. 12b and Fig. 13b show such predictions for the in-between frames located in the middle of every initial segment S. Segments with AQM predictions over 10 percent were split and the filled diamonds show the corresponding reduction of the predicted perceivable differences. We also performed experiments assuming higher levels of retinal velocity for our walkthrough animation. The filled triangles in Fig. 12b and Fig. 13b show the AQM predictions when the retinal velocity is equal to the IF (eye movements are ignored). For all segments selected for splitting based on smooth-pursuit eye movement assumption, the AQM predictions also exceeded the threshold of 10 percent when the eye movements were ignored. As we discussed in Section 3.4, although, in general, eye sensitivity improves when eye tracking is considered, for some visual patterns, eye sensitivity can be better when eye tracking is ignored (e.g., refer to the AQM predictions in Fig. 12b for the in-between frame #38).

The overall costs of in-between frame computations are strongly affected by the average number of pixels that must be ray traced. The graph in Fig. 12c shows the percentage of pixels depicting specular objects that are replaced by ray traced pixels in the ATRIUM walkthrough sequence. This graph also shows the percentage of replaced pixels due to IBR occlusion problems and the high sensitivity of the

TABLE 2
Average Computation Time per Frame
for Various Animation Rendering Solutions

Scene	ART [minutes]	RT [minutes]	IBR+RT [minutes]
ATRIUM	170.0	40.0	20.5
ROOM	6.9	1.5	1.1

All timings were measured on the MIPS 195 MHz processor.

[2] Assuming that the viewer is located 50 centimeters away from the display, 5.2 visual degrees corresponds to the image region of diameter approximately 4.5 centimeters.

visual system for image patterns moving with low velocity (the velocity threshold of 0.5 degree/second was assumed as explained in Section 5.2). Obviously, a given pixel was replaced only once and we assumed the following processing order of pixels replacement: 1) pixels depicting slowly moving patterns, 2) pixels with possible reflection/refraction artifacts, and 3) pixels with occlusion problems. Fig. 13c shows the equivalent results for the scene ROOM. Table 1 presents the average percentage of pixels to be ray traced per frame.

To evaluate the efficiency of our animation rendering system we compared the average time required for a single frame of our test walkthroughs using the following rendering methods: ART—fully ray traced frames with antialiasing (using adaptive supersampling) which are commonly applied in the traditional rendering animation approach, RT—fully ray traced frames (one sample per pixel), and IBR+RT frames generated using our approach with mixed ray traced and IBR-derived pixels. Table 2 summarizes the obtained results for the ATRIUM and ROOM walkthroughs. In the case IBR+RT, we included the computation involved in IBR rendering (which requires about 12 seconds to warp and composite two keyframes frames), motion-compensated 3D filtering which added an overhead of 10 seconds per frame, and AQM processing which takes 243 seconds to process a pair of frames. The AQM computations are so costly mainly because of the software implemented Fast Fourier Transform (FFT). Since our frames are of resolution 640×480, we had to consider images of resolution $1,024 \times 512$ for the FFT processing.

The most significant speedup was achieved by using our spatiotemporal antialiasing technique and avoiding the traditional adaptive supersampling. Our in-between frames rendering technique added a further 25-50 percent of speedup with respect to the RT approach. The tested scenes were hard for our algorithm because of the strong specular reflectance properties exhibited by many of the surfaces (ATRIUM) and the slow motion of the camera, in which case, eye sensitivity is high (ROOM). Also, the chessboard-like pattern of textures in the ROOM scene made it quite challenging in terms of proper antialiasing. Even better performance can be expected for environments in which specular objects are depicted by a moderate percentage of pixels, and camera motion is faster.

7 Conclusions

In this work, we proposed an efficient approach for rendering of high quality walkthrough animation sequences. Our contribution is in developing a fully automatic, perception-based guidance of in-between frame computation which minimizes the number of pixels computed using costly ray tracing and seamlessly (in terms of the perception of animated sequences) replaces them by pixels derived using inexpensive IBR techniques. Also, we have shown three very important applications of the image flow obtained as a by-product of IBR processing. It was applied to: 1) place keyframes along the animation path, which improved in-between frame computation performance, 2) estimate the spatio-velocity Contrast Sensitivity Function, which made it possible to incorporate temporal factors into our perceptually-based image quality metric, 3) perform the spatiotemporal antialiasing with motion-compensated filtering based on image processing principles (in contrast to traditional antialiasing techniques used in computer graphics). We integrated all these techniques into a balanced animation rendering system.

As future work, we plan to conduct validation studies of our AQM in psychophysical experiments. Also, we believe that our approach has some potential for the automatic selection of keyframes used in IBR systems.

Acknowledgments

Special thanks to Scott Daly for his stimulating comments on video quality metrics and to Eric Fausett for proofreading of the manuscript.

References

[1] http://www.mpi-sb.mpg.de/resources/aqm/, the Web page accompanying to this paper.

[2] S.J. Adelson and L.F. Hodges, "Generating Exact Ray-Traced Animation Frames by Reprojection," *IEEE Computer Graphics and Applications,* vol. 15, no. 3, pp. 43-52, 1995.

[3] S. Badt Jr., "Two Algorithms for Taking Advantage of Temporal Coherence in Ray Tracing," *The Visual Computer,* vol. 4, no. 3, pp. 123-132, 1988.

[4] B. Girod, "The Information Theoretical Significance of Spatial and Temporal Masking in Video Signals," *Proc. SPIE,* vol. 1,077, pp. 178-187, 1989.

[5] S.E. Chen, "Quicktime VR—An Image-Based Approach to Virtual Environment Navigation," *SIGGRAPH '95 Proc.,* pp. 29-38, 1995.

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

[7] S. Daly, "Engineering Observations from Spatiovelocity and Spatiotemporal Visual Models," *Human Vision and Electronic Imaging III,* vol. 3,299, pp. 180-191, SPIE, 1998.

[8] S. Daly personal communication, 1999.

[9] L. Darsa, B.C. Silva, and A. Varshney, "Navigating Static Environments Using Image-Space Simplification and Morphing," *Proc. 1997 Symp. Interactive 3D Graphics,* pp. 25-34, 1997.

[10] R.L. De Valois and K.K. De Valois, *Spatial Vision.* Oxford Univ. Press, 1990.

[11] C.J. van den Branden Lambrecht, "Perceptual Models and Architectures for Video Coding Applications," PhD thesis, Ecole Polytechnique Federale de Lausanne, 1996.

[12] M.P. Eckert and G. Buchsbaum, "The Significance of Eye Movements and Image Acceleration for Coding Television Image Sequences," *Digital Image and Human Vision,* A.B. Watson, ed., pp. 89-98, Cambridge, Mass.: MIT Press, 1993.

[13] R. Eriksson, B. Andren, and K. Brunnstrom, "Modelling of Perception of Digital Images: A Performance Study," *Human Vision and Electronic Imaging III,* pp. 88-97, SPIE, 1998.

[14] J.A. Ferwerda, S. Pattanaik, P. Shirley, and D.P. Greenberg, "A Model of Visual Masking for Computer Graphics," *ACM SIGGRAPH '97 Conf. Proc.,* pp. 143-152, 1997.

[15] B.K. Guenter, H.C. Yun, and R.M. Mersereau, "Motion Compensated Compression of Computer Animation Frames," *SIGGRAPH '93 Proc.,* vol. 27, pp. 297-304, 1993.

[16] P.J. Hearty, "Achieving and Confirming Optimum Image Quality," *Digital Image and Human Vision,* A.B. Watson, ed., pp. 149-162, MIT Press, 1993.

[17] R. Jain, R. Kasturi, and B.G. Schunck, *Machine Vision.* New York: McGraw-Hill, 1995.

[18] D.H. Kelly, "Motion and Vision 2. Stabilized Spatio-Temporal Threshold Surface," *J. Optical Soc. Am.,* vol. 69, no. 10, pp. 1,340-1,349, 1979.

[19] D.H. Kelly, "Spatiotemporal Variation of Chromatic and Achromatic Contrast Thresholds," *J. Optical Soc. Am.,* vol. 73, no. 6, pp. 742-750, 1983.

[20] G.E. Legge and J.M. Foley, "Contrast Masking in Human Vision," *J. Optical Soc. Am.,* vol. 70, no. 12, pp. 1,458-1,471, 1980.

[21] D. Lischinski and A. Rappoport, "Image-Based Rendering for Non-Diffuse Synthetic Scenes," *Rendering Techniques '98 (Proc. Eurographics Rendering Workshop '98),* pp. 301-314, 1998.

[22] J. Lubin, "A Human Vision Model for Objective Picture Quality Measurements," *Conf. Publication No. 447,* pp. 498-503, IEE Int'l Broadcasting Convention, 1997.

[23] J. Malik and P. Perona, "Preattentive Texture Discrimination with Early Vision Mechanisms," *J. Optical Soc. Am.,* vol. 7, no. 5, pp. 923-932, 1990.

[24] W.R. Mark, L. McMillan, and G. Bishop, "Post-Rendering 3D Warping," *Proc. 1997 Symp. Interactive 3D Graphics,* pp. 7-16, 1997.

[25] L. McMillan, "An Image-Based Approach to 3D Computer Graphics," PhD thesis, Univ. of North Carolina, Chapel Hill, 1997.

[26] G. Miller, S. Rubin, and D. Poncelen, "Lazy Decompression of Surface Light Fields for Precomputed Global Illumination," *Rendering Techniques '98 (Proc. Eurographics Rendering Workshop '98),* pp. 281-292, 1998.

[27] K. Myszkowski, P. Rokita, and T. Tawara, "Perceptually-Informed Accelerated Rendering of High quality Walkthrough Sequences," *Rendering Techniques '99 (Proc. 10th Eurographics Workshop Rendering),* pp. 13-26, 1999.

[28] J. Nimeroff, J. Dorsey, and H. Rushmeier, "Implementation and Analysis of an Image-Based Global Illumination Framework for Animated Environments," *IEEE Trans. Visualization and Computer Graphics,* vol. 2, no. 4, pp. 283-298, Dec. 1996.

[29] W. Osberger, A.J. Maeder, and N. Bergmann, "A Perceptually Based Quantization Technique for MPEG Encoding," *Human Vision and Electronic Imaging III,* pp. 148-159, SPIE, 1998.

[30] J.G. Robson, "Spatial and Temporal Contrast Sensitivity Functions of the Visual System," *J. Optical Soc. Am.,* vol. 56, pp. 583-601, 1966.

[31] J. Rovamo, V. Virsu, and R. Nasaren, "Cortical Magnification Factor Predicts the Photopic Contrast Senstivity of Peripherial Vision," *Nature,* vol. 271, pp. 54-56, 1978.

[32] J.W. Shade, S.J. Gortler, L. He, and R. Szeliski, "Layered Depth Images," *SIGGRAPH 98 Conference Proc.,* pp. 231-242, 1998.

[33] M. Shinya, "Spatial Anti-Aliasing for Animation Sequences with Spatio-Temporal Filtering," *SIGGRAPH '93 Proc.,* vol. 27, pp. 289-296, 1993.

[34] A. Murat Tekalp, *Digital Video Processing.* Prentice Hall, 1995.

[35] P.C. Teo and D.J. Heeger, "Perceptual Image Distortion," *Proc. SPIE,* vol. 2,179, pp. 127-141, 1994.

[36] X. Tong, D. Heeger, C. van den Branden Lambrecht, "Video Quality Evaluation Using ST-CIELAB," *Human Vision and Electronic Imaging IV,* pp. 185-196, SPIE, 1999.

[37] B.A. Wandell, *Foundations of Vision,* Sunderland, Mass.: Sinauer Associates, 1995.

[38] A.B. Watson, "Temporal Sensitivity," *Handbook of Perception and Human Performance,* chapter 6, New York: John Wiley, 1986.

[39] A.B. Watson, "Toward a Perceptual Video Quality Metric," *Human Vision and Electronic Imaging III,* pp. 139-147, SPIE, 1998.

[40] A.B. Watson and A.J. Ahumada, "Model of Human Visual-Motion Sensing," *J. Optical Soc. Am.,* vol. 2, no. 2, pp. 322-342, 1985.

[41] A.B. Watson, J. Hu, J.F. McGowan III, and J.B. Mulligan, "Design and Performance of a Digital Video Quality Metric," *Human Vision and Electronic Imaging IV,* pp. 168-174, SPIE, 1999.

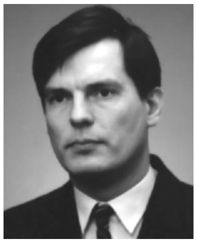
[42] H. Weghorst, G. Hooper, and D.P. Greenberg, "Improved Computational Methods for Ray Tracing," *ACM Trans. Graphics,* vol. 3, no. 1, pp. 52-69, Jan. 1984.

[43] J.H.D.M. Westerink and C. Teunissen, "Perceived Sharpness in Moving Images," *Proc. SPIE,* vol. 1,249, pp. 78-87, 1990.

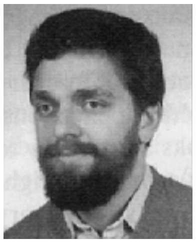
[44] S. Winkler, "A Perceptual Distortion Metric for Digital Color Video," *Human Vision and Electronic Imaging IV,* pp. 175-184, SPIE, 1999.

[45] E.M. Yeh, A.C. Kokaram, and N.G. Kingsbury, "A Perceptual Distortion Measure for Edge-Like Artifacts in Image Sequences," *Human Vision and Electronic Imaging III,* pp. 160-172, SPIE, 1998.

[46] E. Zeghers, K. Bouatouch, E. Maisel, and C. Bouville, "Faster Image Rendering in Animation through Motion Compensated Interpolation," *Graphics, Design, and Visualization,* pp. 49-62, 1993.



Karol Myszkowski received his PhD degree in computer science from Warsaw University of Technology (Poland) in 1991. He is a senior visiting researcher at Max-Planck-Institute for Computer Science, Germany. Until recently, he served as an associate professor in the Department of Computer Software at the University of Aizu, Japan. His current research is investigating the role of human perception to improving the performance of photo-realistic rendering and animation techniques. He is a member of the IEEE Computer Society.



Przemyslaw Rokita received his MSc and PhD degrees in computer science in 1985 and 1993, respectively, from Warsaw University of Technology. He is an associate professor at the Institute of Computer Science at Warsaw University of Technology. His research interests include computer graphics, image processing, and perceptual aspects of human vision. He is a member of the IEEE, ACM, and SPIE.

Takehiro Tawara is a PhD student at the Max-Planck-Institute for Computer Science, Germany. He received the MSc degree in computer science from the University of Aizu in 2000. His research interests include computer graphics, virtual reality, and artificial intelligence.