

# Real-Time Adaptive Radiometric Compensation

Anselm Grundhöfer and Oliver Bimber, *Member, IEEE*,

**Abstract**—Recent radiometric compensation techniques make it possible to project images onto colored and textured surfaces. This is realized with projector-camera systems by scanning the projection surface on a per-pixel basis. With the captured information, a compensation image is calculated that neutralizes geometric distortions and color blending caused by the underlying surface. As a result, the brightness and the contrast of the input image is reduced compared to a conventional projection onto a white canvas. If the input image is not manipulated in its intensities, the compensation image can contain values that are outside the dynamic range of the projector. They will lead to clipping errors and to visible artifacts on the surface. In this article, we present a novel algorithm that dynamically adjusts the content of the input images before radiometric compensation is carried out. This reduces the perceived visual artifacts while simultaneously preserving a maximum of luminance and contrast. The algorithm is implemented entirely on the GPU and is the first of its kind to run in real-time.

**Index Terms**—Computer graphics, picture/image generation, display algorithms, image processing, computer vision, radiometry, reflectance, enhancement, color

## I. INTRODUCTION

Compact pocket projectors that are running from battery and communicate to laptops or cell phones via WiFi will support a maximum level of mobility in future. There is no doubt that LED technology will become bright enough to keep up with today's projector lamps. Yet one question still remains: On what to project on if not carrying around a projection canvas?

Projector-camera systems have been used together with radiometric compensation algorithms for projecting onto complex everyday surfaces, like papered walls or structured table tops. Most of the existing approaches consider only the characteristics of the surface - such as its reflectance or geometry - for the compensation of visual artifacts. The properties of the image to be displayed - like its brightness and contrast - are normally not taken into account. This can lead to clipping errors and to remaining visual artifacts at dark surface pigments that are due to the limited dynamic range and brightness of projectors. An adjustment of the image intensity to avoid this problem can only be carried out manually. Only recent approaches adapt the image content (sometimes based on the capabilities of the human visual system) for minimizing these artifacts [21] [1]. But because of their complex image transformations and numerical minimizations such algorithms are too complex to support real-time applications. Thus, animated or interactive content cannot be compensated in an optimized way. This, however, will be essential for future mobile projection systems.

We propose a novel algorithm that adjusts the image content to reduce visible artifacts in real-time. This is achieved by an analysis of the projection surface and the image content,

followed by a manipulation of the image's local and global intensity values before radiometric compensation is carried out. The result is a significant reduction of clipping errors and visible artifacts while preserving a high contrast ratio and brightness. The objective enhancement of the perceived visual quality for projected animated content is validated by an informal user study. The real-time capability of our algorithm enables an adaptive radiometric compensation for presentations of interactive and animated imagery onto non-optimized surfaces with consumer projectors.

### A. Outline of the Article

The remainder of the article is organized as follows: We start with a discussion on relevant related work in section II, followed by an overview of our approach in section III. The algorithm is then described in detail: While section IV outlines the image and surface analysis components, section V describes the real-time adaptation and compensation technique. Examples are shown in section VI and a performance analysis of our algorithm is carried out in section VII. The results of an informal user study are presented in section VIII. Finally, section IX concludes our work and gives indications for future enhancements.

## II. RELATED AND PREVIOUS WORK

This section gives an overview over existing radiometric compensation techniques. Basic methods are explained first, since they are partially used by our algorithm. Recent approaches that adapt the image content in a pre-process to achieve optimized results with one of the basic methods are described next. In contrast to our algorithm, they can only be applied to static images.

### A. Basic Radiometric Compensation Techniques

Most radiometric compensation approaches apply structured light projection and camera feedback for measuring surface and environment parameters, such as geometry, reflectance and environmental light, as well as for establishing a precise mapping between projector and camera pixels. This is usually carried out during a short off-line calibration step and assumes a static constellation afterward. The parameters are used during runtime for radiometric compensation and geometric warping computations. An exception is the work by Fujii et al. [9] that utilizes an co-axial alignment of projector and camera for dynamic compensation on non-static surfaces. A closed feedback loop is used in this case to re-adjust the compensations over time while either surface or projector-camera pair can be moved.

Several algorithms have been developed to project compensated images onto planar and non-planar diffuse surfaces.

Nayar et al. [18] use a 3x3 matrix for each pixel to encode the mixing between the color channels of projector and camera, as well as the surface reflectance. The values of the matrices are measured by projecting a series of uniform color patterns onto the surface and capturing the reflected images. Computing the compensation image is then equivalent to multiplying the inverse color mixing matrices onto the color vectors of the corresponding pixels in the input image. This method (or variations of it) has been applied in combination with planar [18], more complex [12], and non-static [9] surfaces.

Bimber et al. [4] describe a radiometric compensation technique that supports multiple projectors to increase the overall brightness and consequently reduces clipping artifacts on complex surfaces. For planar surfaces - such as paintings - clipping errors can also be minimized by coating the surface with a transparent film material that diffuses a certain amount of light directly while transmitting the remaining portion [2].

Wetzstein et al. [22] present a generalized approach to radiometric compensation by inverting the full light transport captured between projector and camera. This method is extended to support color mixing and the application of an arbitrary number of projectors and cameras. All local and global light modulations, as well as projector and camera defocus can be compensated in real-time with this approach. The calibration process, however, can take up to several hours.

All of these techniques support an image compensation in real-time, but suffer from the same problem: If the compensation image contains values above the maximal brightness or below the black level of the projector, clipping artifacts will occur. These artifacts let the underlying surface structure become visible. While applying multiple projectors [4] is one option that increases the brightness but also the complexity of the projection system, the application of an amplifying film material [2] is another option that restricts such an approach to simple surfaces.

### *B. Content Dependent Radiometric and Photometric Compensation*

Recent algorithms extend radiometric compensation by varying the input image first to achieve an optimized compensation quality with minimized clipping artifacts.

Wang et al. [21] presented the first technique that scales the overall intensity of the input image until clipping errors that result from radiometric compensation are below a perceivable threshold. Their computational intensive numerical minimization can only be applied to static monochrome images and surfaces.

Park et al. [19] describe a technique for increasing the contrast in the compensation image by applying a histogram equalization to the colored input image. This does not preserve the contrast ratio of the original image. Consequently, the image content is thus modified significantly. The problem of occurring clipping errors is not considered by this method.

A complex framework for computing an optimized photometric compensation is presented by Ashdown et al. [1]. The surface's reflectance is scanned with a color calibrated HDR camera. The captured data and the image content is

then transformed into the device-independent CIE  $L^*u^*v$  color space in which color distances are based on the human visual perception. The chrominance values are fitted into the gamut of each projector pixel. Finally a luminance fitting is applied with a relaxation method based on differential equations. The compensation algorithm presented in [18] is then used with the adjusted image rather than with the original image. This method can achieve optimized results even for projections onto surfaces with high varying reflectance properties or high saturation - but for static images only.

All of these techniques lead to reduced clipping artifacts and consequently to an increased visual quality compared to the basic compensation methods that do not pre-adapt the input images. However, due to their computational complexity that can mainly be contributed to numerical minimizations, a real-time compensation cannot be achieved. This constrains them to the presentation of still images. Animated content, such as movies, can only be displayed after a time consuming pre-correction. This, however, is impractical for most applications. It is particularly useless if surface and setup don't remain completely static, such as it is the case for mobile projectors that will require a flexible and frequent re-calibration on different surfaces. Furthermore, distributed content, such as DVDs or broadcasted media cannot be pre-corrected for multiple individual surface. Finally, it is clear that real-time dynamic content, such as interactive applications cannot be presented at all.

## III. OVERVIEW

Our algorithm performs content adaptation and radiometric compensation in real-time, and reduces visual artifacts while preserving a maximum brightness and contrast. Although we chose the basic compensation scheme presented in [4], only minor modifications to the adaptation algorithm are necessary to use any of the other techniques instead. The algorithm is implemented entirely on the GPU in five steps:

- 1) An analysis of the input image and of the projection surface is performed to gain sufficient parameters for adapting the input image.
- 2) The intensities of the input image are globally scaled depending on the parameters determined in step 1.
- 3) The scaled image that results from step 2 is analyzed for clipping errors.
- 4) The intensities of the image content is re-scaled globally and locally depending on the results of the error analysis in step 3.
- 5) The re-scaled image from step 4 is radiometrically compensated and projected.

Additional time-dependent adaptation factors are applied to avoid abrupt changes in the displayed brightness. These steps will be described in detail below.

## IV. ANALYSIS

Generating radiometrically compensated images without clipping errors while simultaneously preserving a high contrast ratio and brightness requires the analysis of both - projection surface and input image. While the captured surface

reflectance has to be analyzed only once<sup>1</sup>, the image content has to be analyzed continuously and in real-time. Our algorithm stores each input image in the texture memory of the graphics card. All following tasks are accomplished via fragment shading and multi-pass rendering entirely on the GPU. Framebuffer Objects (FBOs) are used for an efficient exchange of data.

### A. Surface Analysis

By applying the basic radiometric compensation technique described in [4], structured light projection and camera feedback delivers several surfaces properties after a one-time off-line calibration step. First, the projector-camera pixel correspondences are determined and stored in two-dimensional look-up tables that are passed as textures to fragment shaders. During run-time, the shaders carry out a per-pixel displacement mapping of all projector pixels to warp the images' geometry with respect to the surface geometry. The result is an undeformed appearance from the perspective of the calibration camera.

For radiometric calibration the same method requires two additional parameters for each pixel:

- The contribution of the (uncontrollable) environmental light which is reflected from the surface -  $EM$  (including the projector's black-level).
- The surface's reflectance and the projector-to-surface form-factors -  $FM$  (the fall-off of projected intensity, depending on the projector-to-surface distance and the projection angle).

The intensity range for which a radiometric compensation without clipping is possible can now be computed from the two parameters  $FM$  and  $EM$ . Figure 1 visualizes the

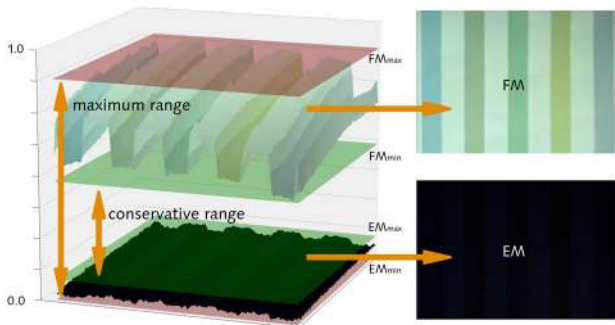


Fig. 1. Three-dimensional view of the intensity range reflected by a striped wall paper. The area between both green planes depict the range of a conservative compensation without clipping errors. The area between the red planes represents the maximum range in which a compensation is possible (potentially with clipping errors).

reflection properties for a sample surface. By analyzing the responses in both datasets, we can compute the range of intensities for a conservative compensation. Thus, any input pixels  $O(x, y)$  within this global range (bound by the two

<sup>1</sup>As long as we assume that the surface and the projector-camera system remains static.

green planes - from the maximum value  $EM_{max}$  to the minimum value  $FM_{min}$ ) can be compensated correctly for each point on the surface without causing clipping artifacts:

$$EM_{max} \leq O(x, y) \leq FM_{min} \quad (1)$$

The red planes in figure 1 define the maximum range of displayable intensities in which a compensation without clipping is only possible if the color values of any input pixel lie within their local range:

$$EM(x, y) \leq O(x, y) \leq FM(x, y) \quad (2)$$

The calculation of the extreme values is carried out once on the CPU. To avoid extreme values, intensities outside the threefold of the standard deviation are omitted.

### B. Content Analysis

Since pixels outside the displayable range cause clipping artifacts, the input image is analyzed to support subsequent global and local luminance adjustments that ensure an optimized compensation. These image processing steps have to be performed in real-time for each input image and are implemented on the GPU to benefit from the high memory bandwidth and parallel processing capabilities of modern graphics hardware.

1) *Average Image Luminance*: The arithmetic luminance mean  $L_{avg}$  of the input image can be used as an initial factor for an automatic scaling of the image content. This is described in section V-A. It is computed via multi-pass rendering on the GPU:

A fragment shader transforms the input image into its CIEXYZ representation according to the sRGB transformation matrix. The Y values storing the required luminance information [8] is extracted from this image and is directly rendered into a texture by using FBOs. Throughout multiple rendering passes, the luminance image is successively downscaled by the factor 2 while averaging the four neighboring pixels in each step. The results are also directly rendered into textures via FBOs and are forwarded to the next rendering pass. Like for mip-mapping, this is repeated until the remaining luminance image contains a single pixel that stores the average value. With a slight modification, the same technique can be applied to determine global maximum and minimum values of an image, as required for the error analysis described in section V-B. Using multiple render targets, all of these steps can be carried out in parallel.

2) *Threshold Map*: The perception of luminance variations in images depends on many different factors, such as the display brightness, the local image contrast and the spatial frequencies of the content.

In [20] the so-called threshold map is introduced that contains the maximum non-perceivable luminance threshold for every pixel of an image. This value contains the maximum luminance differences that can be varied in the original image without causing a visually perceivable difference. For computing the threshold map, the physical luminance values

of the displayed image, as well as the spatial frequencies and contrast relations of the image content are required. The luminance-dependent values are calculated separately by using the threshold-versus-intensity (TVI) function and are stored in the so-called TVI map. This function describes the peak luminance sensitivity of the human visual system which is only correct for environments with uniform background luminance. This, however, is not the case if an image with non-uniform content is projected. In this situation the sensitivity depends also on the local image content. These spatially varying factors are computed and stored in the so-called elevation factor map. The threshold map is computed by multiplying the corresponding entries of the TVI map and the elevation factor map.

Thus, varying the local luminance of the input image by an amount that is below the corresponding values in the threshold map will not lead to perceivable luminance differences. A compensated projection, however, can be enhanced if this is done for regions in which clipping occurs. This is described in section V-C.

The TVI map, the elevation factor map and the threshold map are computed on the GPU in real-time, as outlined in the following sub-sections.

*a) TVI Map:* As proposed in [20], we use the procedure presented in [14] to calculate the TVI map which stores the results of the TVI-function for each pixel. In this approximation, the adaptation luminance  $L_{adapt}$  at each pixel is used which is calculated by averaging the luminance values over  $1^\circ$  of the visual angle centered at the according pixel. As described in section IV-B.1, the RGB values of the input image are transformed into their luminance representation first. We use a photometer to measure the average minimum and maximum reflected physical luminance values of the surface by projecting a complete white and black image. These values are then used to transform the image's luminance representation into their physical luminance values. Furthermore, the size of the projection area and the distance to the observer has to be known.

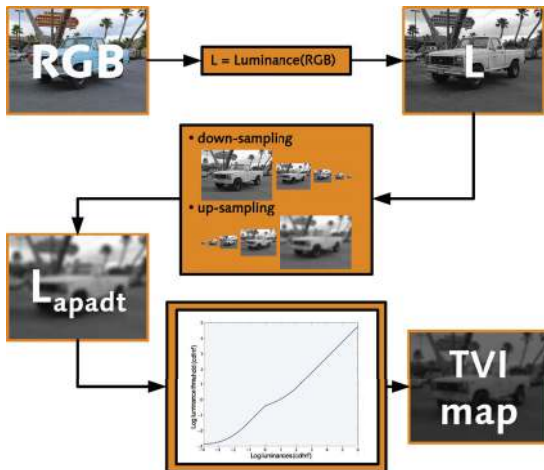


Fig. 2. Flow chart of the GPU-based TVI map generation.

Due to the fact that the TVI function requires the adaptation

luminance which can be reached by smoothing the luminance values over  $1^\circ$  visual angle, the calculated luminance  $L$  has to be convolved with a low-pass filter. This smoothing operation can be approximated efficiently via multi-pass rendering on the GPU: First, the luminance image is bilinearly filtered and down-sampled iteratively until a single pixel represents the size of  $1^\circ$  of the visual angle. The result is then iteratively up-sampled and bilinearly filtered until the original resolution is reached. Once again, FBOs are used to render directly into floating point textures. Finally the TVI function is calculated for each pixel, as described in [20], and stored in an additional floating point texture.

Figure 2 illustrates this process. The function in the lower box represents the values of the used TVI-function. Note, that orange boxes represent single or groups of fragment shaders.

*b) Elevation Factor Map:* The elevation factor map is used for adjusting the TVI map depending on the spatial frequencies and contrast ratios of the input image content. Therefore, a Laplacian pyramid is computed from the input images' Gaussian pyramid. The Laplacian pyramid is then converted into a contrast pyramid that can be used for calculating the elevation factor map as described in [20].

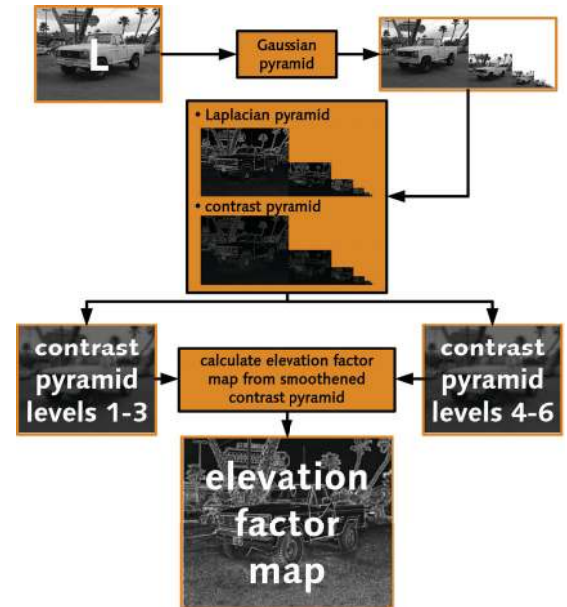


Fig. 3. Flow chart of the GPU-based elevation factor map generation.

Figure 3 outlines the single steps of the GPU-based generation of the elevation factor map.

The Gaussian pyramid is computed from the luminance image by iteratively down sampling and applying a Gaussian blur fragment shader. Another shader is then used for computing the levels of the Laplacian and contrast pyramids.

Since the levels of the contrast pyramid have to be locally averaged next (as suggested by Lubin [16]), they are efficiently stored in the RGB channels of two floating point textures. These computations can be carried out in parallel by using multiple texture targets. The final shader applies the average

operation to each level of the contrast pyramid and calculates the elevation factor map values as discussed in [20].

Corresponding values of the TVI map and the elevation factor map are finally multiplied with a fragment shader to compute the threshold map. Once more, FBOs are used for all of these operations to render directly into floating point textures.

3) *High Spatial Frequencies*: The amount of high spatial frequencies can be approximated by analyzing the Laplacian pyramid. This is a useful parameter for the local intensity variation, described in section V-C. Therefore, the lowest level of the Laplacian pyramid is binarized with a fragment shader based on a pre-defined threshold. Using hardware accelerated occlusion queries the pixels in the high frequency domain are counted. Dividing this by the number of all pixels in the image leads to the ratio of high-frequency to low-frequency pixels.

## V. ADAPTATION AND COMPENSATION

With the results of the analysis, the input image is adapted to minimize clipping errors while preserving a high overall brightness and contrast after radiometric compensation. The adaptation is realized in three steps:

- 1) Global scaling of the image's intensities.
- 2) Error analysis of the scaled images resulting from step 1.
- 3) Global and local intensity adjustments based on the errors determined in step 2.

After these adaptation steps, the radiometric compensation is applied and the result is projected. These steps are described in more detail below.

### A. Pre-Adaptation

In a first adaptation step the acquired information about the average image luminance and the surface properties (see sections IV-B.1 and IV-A) is used to apply an approximate global scaling of the image's intensity. A compensation image is then calculated from the scaled input image. This allows to analyze the resulting quality of the global intensity adjustments and to identify local clipping errors. The results are used for calculating the final global and local scaling parameters.

The pre-adaptation step is carried out in a reduced resolution to speed up the necessary calculations. The lower sampling rate leads to minimal clipping artefacts that can be tolerated<sup>2</sup>.

The intensities of the input image are scaled depending on the average image luminance and the maximum and minimum color values of the projection surface. While images with a low average luminance are up-scaled, too bright images are down-scaled in their intensity according to equations 3 and 4:

$$\begin{aligned} I_{max}(x, y) &= scale(I(x, y), FM_{max}, EM_{min}) \\ I_{min}(x, y) &= scale(I(x, y), FM_{min}, EM_{max}) \\ scale(in, max, min) &= min + (in \cdot (max - min)) \end{aligned} \quad (3)$$

<sup>2</sup>We used a resolution of 128<sup>2</sup> pixels which seems to be a good trade-off between performance and quality on current GPUs. By varying this resolution, the algorithm's performance can be adjusted to the capabilities of the applied graphics hardware.

$I_{min}(x, y)$  and  $I_{max}(x, y)$  are two scaled representations of the input image  $I(x, y)$ . While  $I_{min}(x, y)$  represents the image in the conservative compensation range,  $I_{max}(x, y)$  is scaled to the maximum range in which a compensation is possible (see section IV-A). The final image intensities  $I_{scale}(x, y)$  are a linear interpolation between of  $I_{minR}(x, y)$  and  $I_{max}(x, y)$ . Thereby, the interpolation weights depend on the derived luminance average of the input image  $L_{avg}$ :

$$\begin{aligned} I_{scale}(x, y) &= interp(I_{max}(x, y), I_{min}(x, y), L_{avg}) \\ interp(x, y, u) &= x \cdot (1.0 - u) + y \cdot u \end{aligned} \quad (4)$$

After the global adjustment, a radiometric compensation according to [4] is applied to  $I_{scale}$ <sup>3</sup>. The result is an initial compensation image  $I_c$ . A fragment shader analyzes each pixel of  $I_c$  for the maximum clipping error in each color channel. The result of this analysis is written into separate color channels of an FBO. Values that are above the highest displayable intensity (i.e., > 1.0) are written into the red color channel, and values that are below the projector's black level are written into the green color channel. If no clipping appears at a pixel, the blue channel is used to store the minimum distance of the pixel's color values to 1.0. Otherwise zero value are stored in this channel. Consequently, the error definitions for each pixel are given with:

$$\begin{aligned} Err_r &= \begin{cases} 1.0 - \max(I_{c,r}, I_{c,g}, I_{c,b}); (I_{c,r} \vee I_{c,g} \vee I_{c,b} < 0) \\ 0; else \end{cases} \\ Err_g &= \begin{cases} |\min(I_{c,r}, I_{c,g}, I_{c,b})|; (I_{c,r} \vee I_{c,g} \vee I_{c,b} < 0) \\ 0; else \end{cases} \\ Err_b &= \begin{cases} 1.0 - \max(I_{c,r}, I_{c,g}, I_{c,b}); (I_{c,r} \vee I_{c,g} \vee I_{c,b} < 0) \\ 0; else \end{cases} \end{aligned} \quad (5)$$

As mentioned above, this step is realized by a direct render-to-texture operation via frame buffer objects. In the next step, the generated error texture  $Err$  is analyzed to re-scale the image locally and globally for achieving optimized compensation results.

### B. Error Analysis

Clipping errors lead to abrupt alternations in luminance and chrominance within the displayed image.

A conservative global luminance reduction leads to a full elimination of clipping errors, but also to a significant reduction in contrast and brightness. Therefore, our algorithm varies the image intensities locally in addition to a neutralization of remaining clipping errors while preserving a high overall image brightness and contrast.

Studies of human visual perception indicate that abrupt changes in luminance are perceived more intense than smooth and low frequent modifications [17] [10] [15]. Consequently, we blur the calculated clipping errors with a Gaussian smoothing kernel  $G$ :

$$\begin{aligned} Err_{FM}(x, y) &= Err_r(x, y) \otimes G(\sigma) \\ Err_{EM}(x, y) &= Err_g(x, y) \otimes G(\sigma) \end{aligned} \quad (6)$$

Attention has to be paid when applying the smoothing operator to the clipping errors: On the one hand, a smooth

<sup>3</sup>See equations 11 and 12.



local modification is required for avoiding abrupt intensity variations. On the other hand the image content should not be alternated more than necessary. We adjust the  $\sigma$  parameter of the filter kernel inverse proportionally to the amount of high spatial frequencies of the input image (see section IV-B.3). Thus, local luminance reductions affect a larger region in the image if the image content stores mostly low spatial frequencies. The affected area is decreased for a larger amount of high spatial frequencies. For this purpose a series of GPU-based Gaussian filters with varying  $\sigma$  values (ranging from 6 to 16) are applied. The factor that is used for global intensity scaling<sup>4</sup> equals the average image luminance  $L_{avg}$ . It can now be adjusted more precisely with respect to the largest detected clipping value within the image:

$$S' = \min(1, L_{avg} + \max(\max(Err_r) - Err_{max})) \quad (7)$$

If  $Err_r$  stores values above a maximum tolerated clipping error  $Err_{max}$ , the new global scaling factor  $S'$  is increased. This leads to a larger reduction in brightness according to equation 4. Empirical experiments confirmed that an  $Err_{max}$  of 0.55 delivers adequate results, while  $S'$  is constrained to an upper limit of 1.0 since no clipping can occur in this range ( $I(x, y)$  is scaled to the conservative range).

If there is no clipping at all within the entire image,  $S'$  will be adjusted with respect to the smallest value in  $Err_b$ , which leads to an increase in brightness:

$$S' = \max(0, L_{avg} - 1 + \max(Err_b)) \quad (8)$$

To avoid a perceivable flickering of the projection due to its continuous adjustment, the scaling factor is smoothed over time (see section V-D).

Figure 4 summarizes all calculation steps described above: From the result of the pre-adaptation (a) the computed clipping values are stored in the color channels of an auxiliary texture (b). The red ( $b_1$ ) and green ( $b_2$ ) channels store the clipping errors for values above and below the displayable range. Similar to the calculation of the average image luminance described in section IV-B.1, the maximum values in both channels are determined. Depending on these extrema, the global scaling factor is adjusted. Both error maps ( $b_1$  and  $b_2$ ) are blurred depending on the amount of high spatial frequencies within the input images ( $c_1$  and  $c_2$ ).

### C. Final Adaptation and Compensation

In the remaining adaptation step, the image content is globally re-adjusted as well as locally adapted before computing the compensation image which is finally being projected. For the global re-adjustment, equation 4 is applied together with the adjusted scaling factor  $S'$ . The result is the image  $I'_{scale}$ . Finally,  $I'_{scale}$  is adapted locally, depending on the blurred clipping errors. Therefore,  $Err_{FM}$  and  $Err_{EM}$  have to be up-scaled to projector resolution. To avoid an unnecessary re-scaling in areas in which no clipping occurs and to keep the perceivable image manipulation at a minimum, the luminance values  $L(x, y)$  and the threshold map  $TM(x, y)$  are also

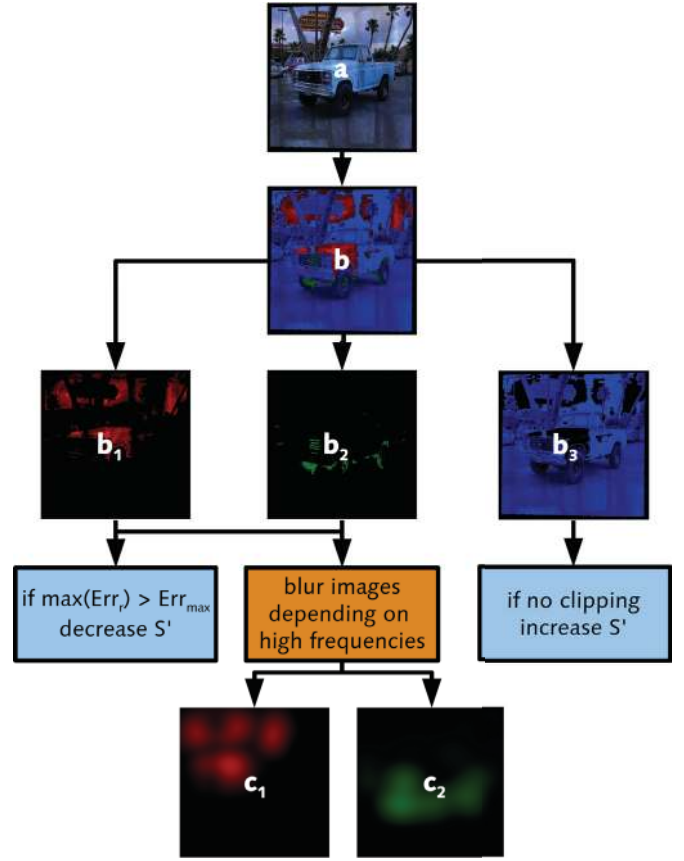


Fig. 4. Flow chart describing the steps of the clipping error analysis.

considered. The local adaptation in areas with clipping values above 1.0 is achieved with:

$$l = L(x, y) \cdot Err_{FM}(x, y) \cdot f_1$$

$$I''_{scale}(x, y) = \begin{cases} I'_{scale}(x, y) - l, & l < TM(x, y) \cdot f_2 \\ I'_{scale}(x, y) - TM(x, y) \cdot f_2, & else \end{cases} \quad (9)$$

In this case local intensities are decreased in the globally scaled image content. Clipping errors that occur in the reverse case are due to values below the black-level of the projector. Consequently, the local image intensities have to be increased:

$$l = (1.0 - L(x, y)) \cdot Err_{EM}(x, y) \cdot f_1$$

$$I''_{scale}(x, y) = \begin{cases} I'_{scale}(x, y) + l, & l < TM(x, y) \cdot f_2 \\ I'_{scale}(x, y) + TM(x, y) \cdot f_2, & else \end{cases} \quad (10)$$

$I''_{scale}(x, y)$  stores the results of the final adaptation stage. The scaling factor  $f_1$  can be used for varying the local adaptation manually. An optimal value of  $f_1 = 1.5$  was found empirically. It represent a good trade-off between quality and performance.

Blurring the error maps causes a spatial distribution of clipping error values. This may lead to unnecessary large intensity modifications in the original image. To adapt the amount of modification to the actual brightness of a pixel, the clipping errors  $Err_{FM}$  and  $Err_{EM}$  are weighted by the corresponding luminance values  $L(x, y)$ .

The threshold map can also be adjusted manually with a second scaling factor  $f_2$ . We found that  $f_2 = 2.0$  was a good compromise between a perceivable local reduction and

<sup>4</sup>Equation 4.

an efficient minimization of visible clipping. If clipping entries are stored in  $Err_{FM}(x, y)$  and in  $Err_{EM}(x, y)$ , the multiplied luminance values  $L(x, y)$  guarantee that both adjustments do not cancel each other out. This is because values in  $Err_{FM}(x, y)$  affect only input pixels with high intensities, while values in  $Err_{EM}(x, y)$  have only an impact on pixels with low intensities.

The adapted input image can now be radiometrically compensated in such a way that the resulting image is displayed with maximal brightness and contrast while disturbing clipping artifacts are minimized. As mentioned earlier we use the compensation equation presented in [4] for a one projector setup:

$$I'_c = \frac{I''_{scale} - EM}{FM} \quad (11)$$

The same adaptation algorithm can be applied, if  $n$  overlaying projectors ( $p = 1..n$ ) are used to produce a brighter image at the surface. In this case the surface analysis has to be carried out under simultaneous illumination of all projectors. The radiometric compensation equation then extends to (details can be found in [4]):

$$I'_{c,p} = \frac{I''_{scale} - \sum_{p=1}^n E_p M}{\sum_{p=1}^n F_p M} \quad (12)$$

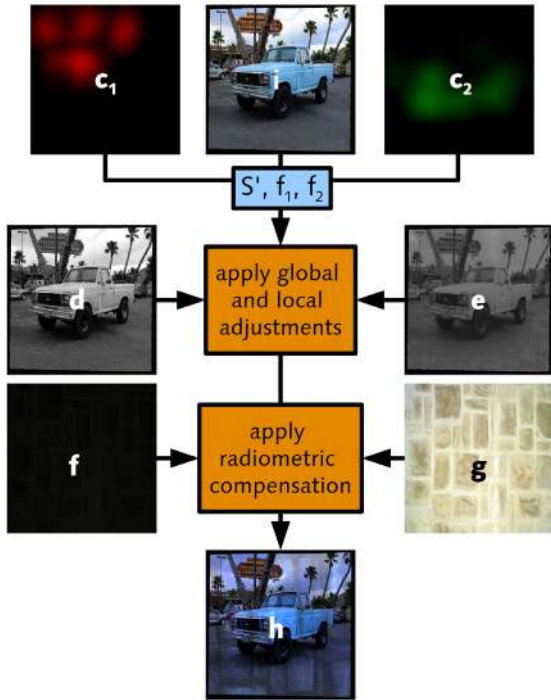


Fig. 5. Flow chart of the final image adaptation and radiometric compensation steps.

Figure 5 summarizes the steps of the final image adaptation and the subsequent radiometric compensation.

The adjusted scaling factor  $S'$  that results from the error analysis is used for re-scaling the input image globally. The smoothed error textures ( $c_1$  and  $c_2$ ) are applied together with the threshold map ( $e$ ) and the local luminance information ( $d$ ) to perform local intensity adjustments in the input image ( $i$ ).

The threshold map constraints the local intensity variations with respect to the largest non-perceivable luminance. In the last step, the surface reflectance ( $g$ ) and the environment light contribution ( $f$ ) are required to compute the adapted compensation image ( $h$ ) that is projected onto the surface.

#### D. Temporal Adaptation

Our algorithm adjusts each input image individually. In case of animated content, such as videos or interactive applications, this might lead to abrupt changes in brightness and contrast and to visible flickering.

To avoid these disturbing effect, we adapt the global scaling factors over time - depending on factors used for scaling previous images. We apply a temporal adaptation model that was initially developed for interactive tone mapping [7] [11] [13]:

$$S''_i = S'_{i-1} + (S'_i - S''_{i-1}) \cdot \left(1 - e^{-\frac{T}{\tau}}\right) \quad (13)$$

The temporally adapted scaling factor  $S''_i$  depends on the factor  $S'_i$  computed for the current frame, and on the factor  $S''_{i-1}$  used for the previous frame. In addition, an exponential attenuation function that is determined based on the actual frame rate  $T$ , as well as a constant  $\tau$  that describes the rate of human luminance adaptation are taken into account. Projection displays usually operate with luminance values in the range of photopic vision. Thus, a value of  $\tau = 0.1$  for rods is used [13]. With this temporal adaptation, abrupt global luminance variations are converted to smooth intensity changes over time. In some cases clipping might happen shortly. However, it can hardly be perceived since the adjustments are carried out very fast. The blurred clipping errors ( $Err_{FM}$  and  $Err_{EM}$ ) are smoothed over time to decrease the visibility of the local intensity adjustments.

Instead of using the computed error textures  $Err_i$  at time instance  $i$ , the values are averaged with the error values used for the previous image  $Err'_{i-1}$ :

$$Err'_i(x, y) = \frac{Err'_{i-1}(x, y) + Err_i(x, y)}{2} \quad (14)$$

This is computationally less expensive than applying equation 13 to each individual error pixel.

## VI. EXAMPLES

In this section, we want to provide several visual examples of our algorithm's outcome<sup>5</sup>.

Figure 6 illustrates a projection ( $e$ ) onto a striped wall paper ( $a$ ). The projection surfaces contains gray scales in this first example. An uncompensated projection ( $b$ ) leads to a clear visibility of the underlying surface. In ( $c$ ) an intermediate result of our adaptive algorithm without local reductions is shown. While it appears to be very similar to the original image ( $e$ ), clipping artifacts are visible (especially in the bright area of the upper left corner). The final result is shown in ( $d$ ). It includes automatic global and local adaptations. Visible clipping errors are reduced without decreasing the image's overall contrast and brightness much.

<sup>5</sup>See video for dynamic examples.

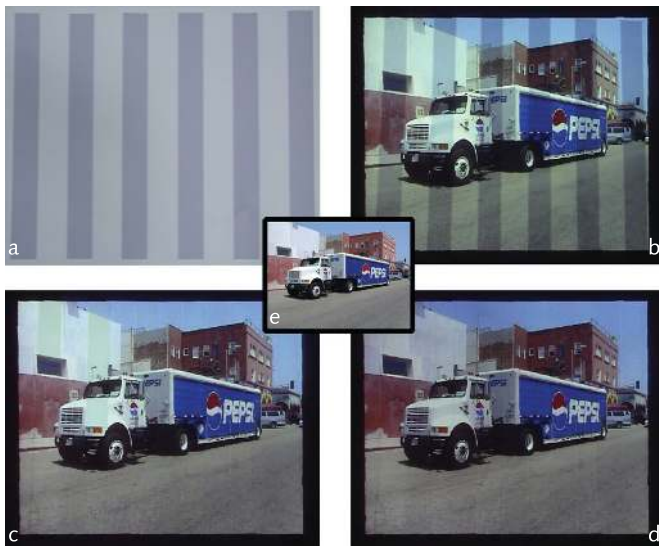


Fig. 6. Adaptive radiometric compensated projection with global (c) and additional local luminance adjustments (d).



Fig. 7. Projection onto a striped wallpaper with saturated colors, without (c) and with (d) adaptive radiometric compensation.

Figure 7 demonstrates a similar example as in figure 6. This time, however, an image (a) is projected onto a surface with saturated colors (b). Although slight color mismatches compared to the original image are detectable, our algorithm produces an acceptable result (d) - especially when comparing it with an uncompensated projection (c).

Figure 8 demonstrates another example. This time an image (e) is projected onto a wooden surface (a). With the local intensity adjustments, our algorithm produces a brighter result (d) compared to a manually adjusted compensation (c).

Note, that in all examples presented in this section, the original images are always illustrated in their native digital format, while the compensation results are shown as photographs of the projections onto the individual surfaces. Consequently, differences between original images and projected compensations can be contributed to camera parameters, such as response,



Fig. 8. Projection onto a wooden surface, without (b), with manually adjusted (c) and with adaptive radiometric compensation (d).

resolution, angle, distance, and field of view.

The main advantage of our approach over basic radiometric compensation algorithms can be shown with animated or interactive content. The continuous adaptation of the input images leads to a constant improvement of image quality compared to a compensations with manual adjustments that do not adapt to the displayed content.

Figure 9 illustrates two different frames from the movie *Shrek 2*<sup>6</sup>, projected onto a natural stone wall (a). While (b) contains bright scenes, a dark scene is shown in (e). The two frames are approximately one second apart in the original video. As demonstrated in (c,f), a basic compensation algorithm (e.g., [4]) will fail in this situation. On the one hand, visible clipping errors occur in image areas with bright intensities (c) due to the physical limitations of the projector. On the other hand, the displayed image becomes too dark (g) due to static adjustment parameters of the basic method. Similar results will be received with all other un-adaptive radiometric compensation methods. As illustrated in (d,g), our adaptive approach responds to these situations automatically and reduces the visibility of the underlying surface while widely preserving brightness and contrast of the original video.

## VII. PERFORMANCE ANALYSIS

Related algorithms [21] [1] implement an adapted radiometric compensation by applying numerical minimizations and relaxation algorithms. In contrast to our method, they are not capable of achieving interactive or real-time frame rates. This, however, is essential for future applications of radiometric compensation in general. This sections provides an analysis of our algorithm's performance.

On our test platform<sup>7</sup> a PAL-resolution video can be compensated with approximately 35 frames per second.

<sup>6</sup>©DreamWorks Animation SKG.

<sup>7</sup>Intel Pentium 4, 2.8 GHz, 1 GB RAM, NVidia GeForce 7900 GTX, XGA projector resolution.





Fig. 9. Two frames of an animation (b, e) projected onto a natural stone wall with a static radiometric compensation (c,f) and with our adaptive algorithm (d,g).

Since the resolution that is chosen for the pre-adaptation step has a significant impact on the overall performance, it can be adjusted according to the desired frame-rate. Figure 10 illustrates a diagram of the measured performance that can be achieved for different pre-adaptation resolutions. As mentioned earlier, we chose a pre-adaptation resolution of  $128 \times 128$  pixels, since it proved to be a good trade-off between performance and image quality for our hardware. As explained in section V-C the results of the pre-adaptation step are linearly interpolated and up-scaled to the resolution of the projector before the final adaptation step is carried out.

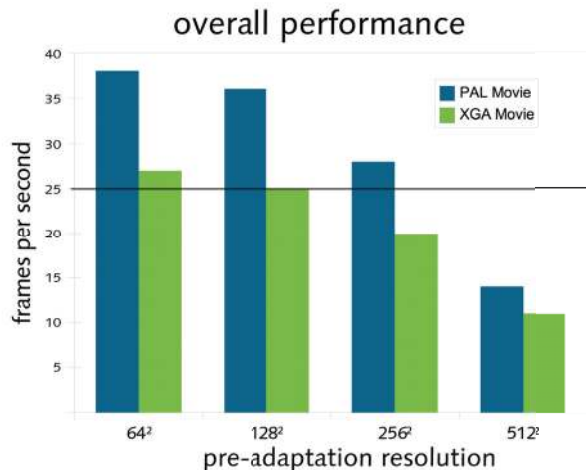


Fig. 10. Measured performance for different pre-adaptation resolutions.

Table I outlines the advantage of a GPU implementation compared to an optimized CPU implementation. It compares four of the necessary image processing operations<sup>8</sup> in our shader implementations with corresponding CPU realizations of Intel's OpenCV image processing library.

In all four tasks the shader computations on the GPU outnumber the CPU implementations. On our test platform we archived speed-up factors of 1.7 to 38.7. Furthermore, additional memory transfers between GPU and CPU are necessary for a CPU implementation. They have not been considered

<sup>8</sup>Gaussian pyramid, average calculation, Gaussian blur and high frequency calculation.

TABLE I  
COMPARISON BETWEEN CPU AND GPU IMPLEMENTATIONS OF FOUR NECESSARY IMAGE PROCESSING TASKS (IN FPS).

Task	CPU	GPU	Gain factor
Gaussian pyramid	38	166	4.3
Gaussian blur (Kernel: $77^2$ pixels)	4	115	38.7
Average luminance	175	303	1.7
High frequency analysis	41	397	9.6

at this point. We believe that GPUs will become a regular component of modern video projectors in the near future.

### VIII. INFORMAL USER STUDY

An informal user study was carried out to validate the increase of perceived visual improvement that can be gained by an adaptive approach compared to a basic radiometric compensation. Therefore participants were asked to compare and to judge the visual appearance of projected sample sequences. A preference for our adaptive radiometric compensation over basic methods was expected.

The user study was performed in a dark room without environment light<sup>9</sup>. Each subject had to adapt to the lighting conditions for five minutes before the test sequences were presented to them.

Two LCD-projectors<sup>10</sup> were used for the user study. They were color calibrated with a photometer for fitting the gamuts of both devices. While the first projector was used to project radiometrically compensated images onto a natural stone wall, the second one projected the original image onto a white canvas.

The user study was separated into three parts: In the first part a series of still images was projected sequentially onto the stone wall with a duration of 15 seconds. Each image was shown next to each other at the same time<sup>11</sup> - one compensated with the adaptive algorithm, the other one compensated with a constant basic method [4]. For the latter case, the same intensity scaling factor was used for all images. It was chosen in such a way that it equals the average scaling factor generated by the adaptive algorithm for all presented test images. Whether the adaptive algorithm was used on the left or the right side was selected at random to avoid an influence of the slightly different underlying surface. The images were labeled to avoid confusion. The subjects did not know which image was generated by which algorithm. They were asked to compare both and decide which one is preferred over the other one.

In the second part of the user study, four video sequences were projected one after another. Again, both compensations (adaptive and static) were presented next to each other, and their positions were randomly switched. For this experiment, two different scaling factors were used for the static compensation method: The first factor was chosen to avoid visible clipping errors completely. This leads to dim, but clipping-free

<sup>9</sup>Except the black level of the projector.

<sup>10</sup>A Sony VPL-CX80.

<sup>11</sup>One on the left side, the other one on the right side.

projections. Videos number 1 and number 2 were presented this way. The second factor was selected to be equal the average scaling factor of the adaptive algorithm over all presented video frames. This was used to display videos number 3 and number 4.

In the third part of the user study, the adaptive and static compensations of the still images presented in the first part were compared with a conventional projection onto a white planar canvas. For this task the participants had to decide which of the compensated images appears more like the reference projection.

For all three parts the subjects had to rate their preferences within five scales, ranging from "left image much more convenient" over "no difference" to "right image much more convenient".

For evaluation purposes, these scales were converted to numerical values ranging from  $-2$  to  $2$ . While positive numbers represent a preference for the adaptive algorithm, negative values indicate a preference for a constant radiometric compensation. Altogether 32 subjects<sup>12</sup> participated in the user study.

While the subjects indicated only a small preference for the adaptive algorithm when still images were presented, it was significantly favored for dynamic content. Especially videos with varying contrast and brightness levels were perceived as enhanced. The diagram in figure 11 illustrates this: A significant preference of the adaptive method was indicated for all four sample videos. It was confirmed that the adaptive approach delivers results that appear more like an ordinary projection than the static compensation method.

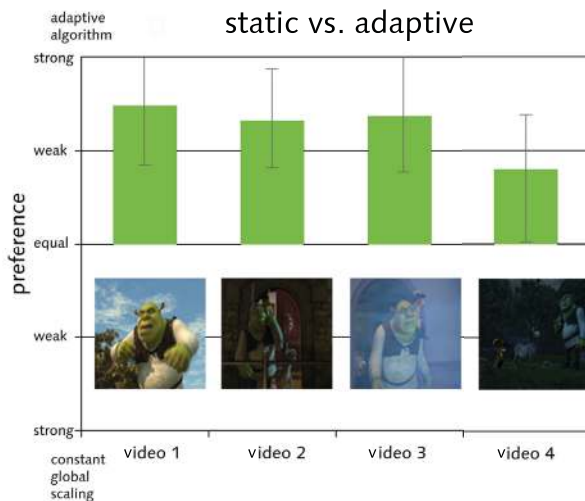


Fig. 11. Comparison of four compensated (static and adapted) videos projected onto a natural stone wall were compared.

## IX. CONCLUSION AND FUTURE WORK

In this article we presented a real-time capable adaptive radiometric compensation algorithm to enable enhanced projections onto everyday surfaces. It considers the surface's reflectance and geometry, the image content and the capabilities

of the human visual system to reduce visible artifacts that are due to the limited dynamic range and brightness of projectors. We believe that such techniques will be essential to support the portability of pocket projectors and projection systems that are integrated into mobile devices, such as cell-phones, PDAs and digital cameras.

Our algorithm analyses the projection surface as well as the image content and adapts the input images globally as well as locally in two steps before a radiometric compensation is applied. This leads to a minimization of clipping errors and corresponding chrominance shifts while preserving a maximum of brightness and contrast. The adaptation parameters are temporally adjusted to ensure smooth intensity transitions and to avoid visible flickering. While a GPU implementation of our algorithm enables real-time frame rates, a user study has confirmed that an automatic adaptation leads to higher visual image quality compared to basic compensation methods.

Besides a luminance adaptation, chrominance adjustments are future extensions to our algorithm that enable brighter projections on highly saturated surfaces. Currently all clipping errors are blurred with the same Gaussian filter kernel. Image segmentation will allow to compute and apply individual kernel parameters for different clipping areas. Both of these enhancements will lead to an improved image quality - but on the cost of performance. An acceptable balance has to be found. As in many other visualization areas, a corpus of performance bottlenecks will be solved by upcoming graphics hardware.

## REFERENCES

- [1] M. Ashdown, T. Okabe, I. Sato and Y. Sato, "Robust Content-Dependent Photometric Projector Compensation," *Proc. IEEE International Workshop on Projector-Camera Systems*, pp. 60-68, 2006
- [2] O. Bimber, F. Coriand, A. Kleppe, E. Bruns, S. Zollmann and T. Langlotz, "Superimposing Pictorial Artwork with Projected Imagery," *IEEE MultiMedia volume 12*, pp. 16-26, 2005
- [3] O. Bimber, G. Wetzstein, A. Emmerling and C. Nitschke, "Enabling View-Dependent Stereoscopic Projection in Real Environments," *Proc. of International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 14-23, 2005
- [4] O. Bimber, A. Emmerling and T. Klemmer, "Embedded entertainment with smart projectors," *IEEE Computer 38*, pp. 56-63, 2005
- [5] N. Boukala, P. Colatini and J.D. Runga, "Fast and accurate color image processing using 3d graphics cards," *Proc. of 8th International Fall Workshop: Vision Modeling and Visualization*, pp. ,2003
- [6] R. Dumont, F. Pellacini and J.A. Ferwerda, "A Perceptually-Based Texture Caching Algorithm for Hardware-Based Rendering," *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pp. 249-256, 2001
- [7] F. Durand and J. Dorsey, "Interactive Tone Mapping," *Proceedings of the Eurographics Workshop on Rendering Techniques*, pp. 219-230, 2000
- [8] J.D. Foley, A. v.Dam., S.K. Feiner, J.F. Hughes, *Computer Graphics: Principles and Practice*, Addison Wesley, Reading, MA, second edition, 1990
- [9] K. Fujii, M.D. Grossberg, S.K. Nayar, "A Projector-Camera System with Real-Time Photometric Adaptation for Dynamic Environments," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'05, Volume 01*, pp. 814-821, 2005
- [10] E.B. Goldstein, *Sensation and Perception*, Wadsworth Publishing Company, 2001
- [11] N. Goodnight, R. Wang, C. Woolley and G. Humphreys, "Interactive time-dependent tone mapping using programmable graphics hardware," *Proceedings of the 14th Eurographics workshop on Rendering*, pp. 26-37, 2003
- [12] M.D. Grossberg, P. Harish and S.K. Nayar, "Making One Object Look Like Another: Controlling Appearance Using a Projector-Camera System," *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 452-459, 2004

<sup>12</sup>Mixed female and male, between 21 and 36 years of age.

- [13] K. Grzegorz, K. Myszkowski and H.-P. Seidel, "Perceptual Effects in Real-Time Tone Mapping," *Proceedings of the 21st spring conference on Computer graphics*, pp. 195-202, 2005
- [14] G. W. Larson, H. Rushmeier and C. Piatko, "A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 3, No. 4, pp. 291-306, 1997
- [15] C. J. Lloyd, "Quantifying edge blend quality: Correlation with observed judgements," *Proceedings of Image Conference*, 2002
- [16] J. Lubin, "A Visual Discrimination Model for Imaging System Design and Evaluation," *Vision Models for Target Detection and Recognition*, World Scientific, pp. 245-283, 1995
- [17] A. Majumder, "A Practical Framework To Achieve Perceptually Seamless Multi-Projector Displays," *Ph. D. Thesis, Department of Computer Science, UNC-Chapel Hill*, 2003
- [18] S. K. Nayar, H. Peri, M. D. Grossberg and P. N. Belhumeur, "A Projection System with Radiometric Compensation for Screen Imperfections," *Proc. IEEE International Workshop on Projector-Camera Systems*, pp. , 2003
- [19] H. Park, M.-H. Lee, S.-J. Kim and J.-II Park, "Contrast Enhancement in Direct-Projected Augmented Reality," *ICME'06: International conference on Multimedia and Expo, 2006*
- [20] M. Ramasubramanian, S. Pattanaik and D.P. Greenberg, "A perceptually based physical error metric for realistic image synthesis," *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 7382, 1999)
- [21] D. Wang, I Sato, O. Takahiro and Y. Sato, "Radiometric Compensation in a Projector-Camera System Based Properties of Human Vision System," *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, Volume 03, page 100, 2005*
- [22] G. Wetzstein and O. Bimber, "Radiometric Compensation of Global Illumination Effects with Projector-Camera Systems," *SIGGRAPH '06: Proceedings of the 33th annual conference on Computer graphics and interactive techniques, Poster*, 2006