

# A Context-Aware Light Source

Oliver Wang\*    Martin Fuchs<sup>§ †</sup>    Christian Fuchs<sup>◇</sup>    James Davis\*    Hans-Peter Seidel<sup>†</sup>  
Hendrik P. A. Lensch<sup>◇</sup>

\*University of California, Santa Cruz    <sup>§</sup>Princeton University    <sup>◇</sup>Universität Ulm    <sup>†</sup>MPI Informatik

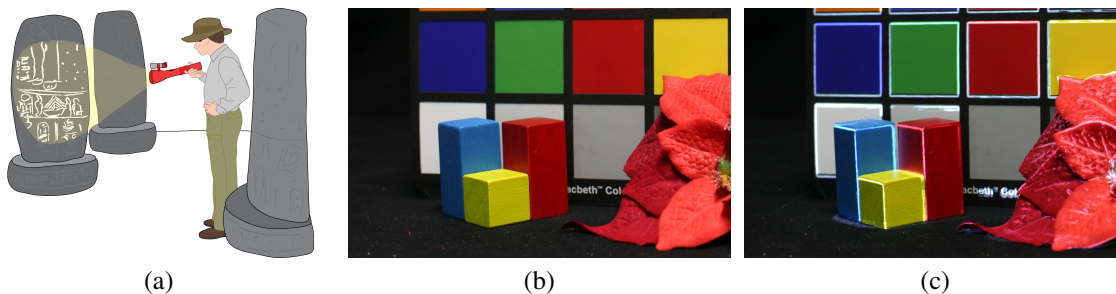


Figure 1. (a) Our vision of a context-aware light source. In contrast to a conventional light source (b), a context-aware light source (c) casts illumination that is dependent on the scene. This can be used for effects such as emphasizing edges.

## Abstract

*We present a technique that combines the visual benefits of virtual enhancement with the intuitive interaction of the real world. We accomplish this by introducing the concept of a context-aware light source. This light source provides illumination based on scene context in real-time. This allows us to project feature enhancement in-place onto an object while it is being manipulated by the user. A separate proxy light source can be employed to enable freely programmable shading responses for interactive scene analysis. We created a prototype hardware setup and have implemented several applications that demonstrate the approach, such as a sharpening light, an edge highlighting light, an accumulation light, and a light with a programmable, non-linear shading response.*

## 1. Introduction

Enhanced visualization and NPR 3D rendering techniques are proven tools for analyzing real world objects on computers.

Observation of enhanced imagery has so far been limited to viewing virtual objects on a computer display, and interacting with them with a mouse. However, physical interaction with real objects is far more intuitive as the object can be directly manipulated. In addition, with real world objects, multiple observers can freely choose their individual view

point. Many commercial fields are aware of this: film studios frequently design their models first in clay and then use a 3D scanner to digitize them. Our goal is to bring the utility of enhanced visualization into the intuitive interface of the real world, thereby allowing the user to interactively control the scene, the viewpoint and even the features that will be enhanced.

In order to achieve this, we define the concept of a context-aware light source. This is a light source that can modify its illumination dependent on sensor information obtained from the scene. A key feature of the context-aware light source is that all necessary information for the enhancements is acquired, processed and projected in real-time, operating without any prior knowledge about the scene. We can therefore freely move and rotate both objects in the scene and the context-aware light source itself while observing the enhanced effects directly on the objects.

We also introduce the notion of a proxy light as a light source that provides lighting information that is processed by the context-aware light source, but is not directly visible to the user. This distinction allows us to redefine the way that the illumination from the proxy light appears to interact with the scene, for example by highlighting specific features. Thus, we achieve a useful real world scene exploration tool for users.

We propose the concept of a context-aware light with a hardware implementation in mind. Using existing technology, it would be possible to incorporate all necessary hardware components into a small single-lens hand held flash-

light body with a detachable proxy light. This device would be easily transportable, allowing it to be used in the field to create context-aware illumination on arbitrary surfaces. We see this concept as a useful tool for areas such as forensic analysis or archaeological study. For the purposes of our experiments, we have constructed a simple prototype implementation of this device using commercially available hardware.

In a context-aware light source, scene features, such as edges or normals, are computed from a camera input. The images are then used to derive feature enhancements which are displayed on the objects using a projector. The necessary pixel-precise mapping between the camera and the projector is established independently of scene geometry by using a coaxial setup and a one-time pixel mapping calibration.

Observing the scene while it is being illuminated by a context-aware light would cause a feedback loop that prevents reasonable scene analysis. We decouple the camera's observations from the context-aware light by separating their wavelengths.

Our proposed system is flexible enough to enable the enhancement of a wide variety of scene properties. We examined several possible applications, such as contrast or edge enhancement, real-time acquisition of surface normals to modify illumination and shading, accumulation of features over time, and a programmable flashlight that creates the illusion of a physically impossible light source with non-linear shading profiles.

The primary contribution of this work is the generalized concept of a context-aware light source. In support of this primary contribution, we show a prototype implementation with a suite of simple applications.

## 2. Related Work

Our research is motivated by prior work in image based enhancements of object features.

**Scene Enhancement** There has been recent interest in processing photographs to create images that convey information better than the original input photographs. For example, this can be done by combining regions from photographs taken under different lighting conditions [1, 4]. Another technique that has been widely adopted for object analysis are relightable models, such as polynomial texture maps [13] that enable the user to modify reflectance properties to make features clearer. This technique has been shown to be real-time capable [14]. Normal information has further been used to create non-photorealistic renderings that are easier to understand than photographs [22]. These methods provide useful tools for object analysis and appearance

modification but are restricted to 2D displays. We extend these concepts by projecting our enhancements onto the object itself.

**Projection Systems** Projectors have been used to recreate large scale virtual environments that can be walked around in [11] and as head mounted displays to add computer graphics to real world objects, allowing for virtual painting [7].

Camera-projector systems have been used to interactively compensate for the geometry and color of the surface that they are projecting onto [18] [8] or to interface with external signals to visibly identify boxes with certain properties in a warehouse type setting [16]. In addition, they have been used to photometrically adapt to dynamic backgrounds [6], detect a moving background for projection [10], project messages onto multiple surfaces in a room [15], or to detect and remove shadows projected onto a screen [9]. In addition to these approaches, there is a large body of projector research on how to project on surfaces with different colors, shapes or reflectance properties. In contrast, we analyze the camera input to discover and enhance scene features, rather than enabling material independent projection.

**Augmented Reality** Enhancing real scenes directly has been studied since the advent of projector technology. Until now, these techniques have largely attempted to augment real scenes with virtual objects or information. Underkoffler et al. [23] describe a room where arbitrary surfaces can display information. They propose the idea of an I/O bulb which is capable of both receiving and emitting light, implemented as a camera/projector system.

Shader lamps [19] describe a system where artificial textures are projected onto white diffuse objects of known geometry in order to accurately recreate computer graphics effects on real world objects. The work of Flagg and Rehg [5] projected guides directly on artist's canvases to assist in real world painting. Instead of using 3D knowledge of the scene and tracking objects with fiducial markers, or using a fixed projection surface, we perform our appearance modification operations on the image domain, and can therefore work directly on the input images. As a result, we do not need to restrict our functionality to a priori known objects.

Most closely related is the work by Bimber et al. [3]. They increase the apparent dynamic range of real world textures by projecting extra contrast information onto surfaces. To avoid the projection interfering with their measurements of the scene, they carefully calibrate the geometry and color of the object, which implies a static scene. While the possibility of projecting onto 3D surfaces is mentioned briefly in the paper, the calibration and the presented results only focus on nearly-planar continuous surfaces. In our work, we are projecting onto objects of unknown shape and ap-

pearance in real time, and we are able to exchange objects arbitrarily.

There is an existing commercial system developed by Luminetx called VeinViewer [12] that embodies a subset of the principles that we explore. This product has fixed functionality that projects patients' veins onto their skin to allow physicians to more clearly see the vasculature. We propose a more general, programmable context aware light source that makes a broader range of applications possible.

Amano et al. [2] describe a method for feedback compensation by computing the projected image such that the camera-projector loop will converge at the desired image. In contrast to that, we use a method to robustly separate the projection from the camera's observations and thereby avoiding the feedback loop altogether.

### 3. Implementation

In order to implement a context-aware light source it is necessary to tightly couple the following components: the sensing camera, real-time processing, the proxy light source and the projector.

Our conceived device is portable, has the option of a fixed or detachable proxy light and can capture normals via multilight illumination.

To demonstrate this portability, we constructed a prototype movable hand held context aware light. This is shown in Figure 2. We fixed a small projector (Dell M109S), camera (Basler A504k) and beam splitter used for optical alignment onto a board of approximately  $35 \times 35$  cm. The captured images are streamed to a computer for analysis, and enhancements derived from them are then displayed on the projector in real time. This device allows the user to move around a room and interactively project enhancements on any surface. For desktop scenes, we use a second setup consisting of a higher resolution DLP projector (a Mitsubishi XD490U).

The proxy-light source, used as an interactive tool for exploring scenes, can be either fixed to the device, in which case it illuminates the scene similar to a standard flashlight, or detached from the device, allowing the user to move it independently from the rest of the setup. In this case, the processed scene now depends on the shading induced by the proxy light, i.e. the incident angle to the surface normal, the distance fall-off and the light's radial distribution. As the system reacts interactively to the proxy light changes, this creates the immediate impression of holding a regular flashlight, but one which projects previously unachievable shading effects.

We also experimented with a multi-flash light source fixed

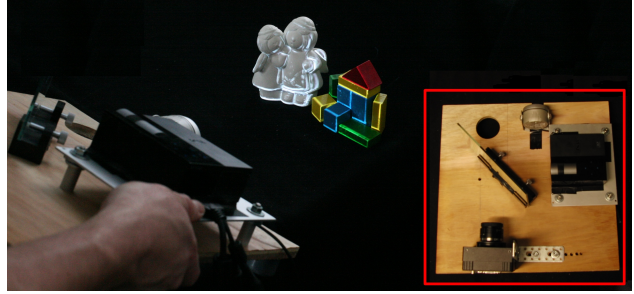


Figure 2. The portable version of our setup allows the user to walk around the room with our content-aware light source. The projector, camera, beam splitter and detachable proxy light are all shown.

to the camera projector system. For this, we incorporate a set of IR LEDs that are triggered in a round-robin fashion by the flash sync signal of the camera, so that each frame gets illuminated by a different LED. This controlled illumination exposes other scene features such as reflectance, surface normals and depth discontinuities, as shown in the work by Raskar et. al [17].

#### 3.1. Spatial Alignment

In order to project context-aware illumination, it is necessary to know which camera pixels are influenced by which projector pixels. For dynamic scene geometry, this information would have to be recomputed with each frame. We avoid this by establishing a scene-independent mapping. We use a beam splitter to arrange the camera and projector in a coaxial setup so that light is emitted and sensed along the same rays independent of the distance of the object to the sensors. With this setup, we only need to acquire the pixel-to-pixel correspondence once. We note that even with perfect coaxial alignment, the calibration step is still required to account for the differences in camera and projector resolution and pixel aspect ratios.

**Pixel Mapping** To establish the mapping from the projector to the camera pixels, we use a structured light technique based on sinusoidal patterns with different periods and frequencies [20]. The reverse mapping from camera to projector pixels is then obtained by warping these points back into projector space using scattered data interpolation to fill in holes.

Depending on the scene content during calibration the estimated mapping can be noisy, e.g. due to surface orientation or low albedo. We perform outlier rejection before interpolation to fill the missing areas. With the correspondence between projector and camera pixels available, we can perform all computations in camera space and warp the result to projector space in real-time.

### 3.2. Feedback Prevention

Using this computed camera-projector mapping, we can project scene enhancements directly onto the scene at the appropriate locations. However, what we project will then become part of the scene and processed in the next frame, causing a feedback loop. We experimented with two methods for preventing this feedback loop. First, we temporally separate the signals by projecting a reference frame between each enhancement frame. This method causes slight visible flickering and reduces the contrast of our projected enhancements. We therefore decided to separate our illumination and our scene observations into different wavelength bands. To do this, we only allow visible-spectrum light to be emitted by the projector, and only allow infrared light to be seen by the camera. The infrared light can come from natural sources or from an IR only proxy-light.

We found that this approach gave preferable results for the users as it does not induce any temporal flickering, so we used this method in all of our result figures. We note that while we use IR in our prototype, any wavelength separation could eliminate the problem with feedback and the exact filters used can be selected appropriately for different applications.

## 4. Applications

We now demonstrate different examples of our context-aware light source as a computational illumination device. In each application, we provide a brief algorithmic description of the functionality. We use the notation  $f(p)$  to refer to the context-aware light emitted at pixel  $p$ , on the range  $[0,1]$ .  $I_p$  corresponds to the intensity of pixel  $p$  as observed in the camera image and  $\nabla I_p$  is the image gradient as computed by a Sobel operator.

We write programmable shading transfer functions in the following format:

$$f(p) = \text{transferFunction}(I_p)$$

Here, *transferFunction* is an arbitrary routine that makes use of pixel intensities and produces a context-aware light output in real time. The user observes the output of the transfer function multiplied by the surface reflectance of the scene. While we do not correct for this, it is possible to partially compensate for the effect when normal information is available.

We begin with a simple demonstration of a programmable shading transfer function, shown in Figure 3. Figure 3a shows a direct mapping of the intensity from the proxy light ( $\text{transferFunction}(I_p) = I_p$ ). Figure 3b then inverts the intensity cast by the proxy light ( $\text{transferFunction}(I_p) =$

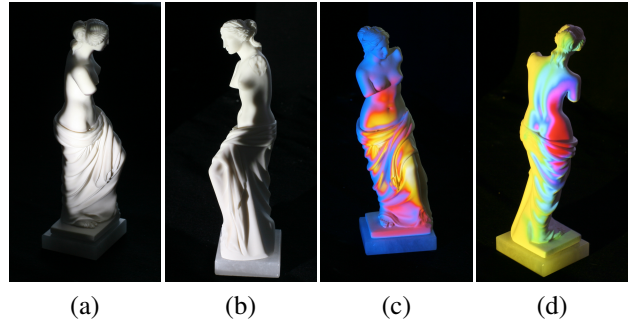


Figure 3. Programmable shading transfer: visualizing IR (a), creating the appearance of negative light (b), and false coloring (c),(d).

$1 - I_p$ ). Figure 3c and 3d show the proxy light intensity mapped to different RGB colormaps.

In many cases it is convenient to leave the proxy light attached to the hand held context aware light source. However in some cases it is desirable to separate the actual direction of projected light from the proxy direction. We have found that the interactive movement of the proxy light source is a strong cue, and contributes to the user’s perception that the perceived effect is actually being cast by the proxy light and not the projector. We have found both modes of interaction to be useful. Please refer to the supplemental video to see results of our system in action.

### Sharpening Light

$$f(p) = (\nabla_x I_p + \nabla_y I_p) \cdot w + .5$$

The sharpening light increases the visual clarity of details in the scene. We first compute local image gradients at each pixel and then project gray light into the scene with intensity influenced by the observed image gradients at these pixels. Here,  $w$  is a user-controlled scale factor determining the amount of enhancement performed. This light increases the perceived contrast of existing gradients in the scene without changing the overall color or brightness.

If the user moves the proxy light around, different features of the objects are revealed similar to how a regular light would behave. For example, edges that are apparent under grazing illumination from the proxy light are further clarified. We found the interaction with of the proxy light with the sharpening mode to be very intuitive for users. Figure 4 shows photographs of object illuminated with a sharpening context-aware light.

### Edge Highlighting

$$f(p) = \|\nabla I_p\| \cdot w$$

A similar, but more drastic effect is to illuminate the scene





Figure 4. An example for sharpening image features in a scene. The fine ridges on the stone and the lines on the plant leaves become more visible in the photographs with sharpening enabled.

only in edge regions. Again,  $w$  is a scale factor that controls the sensitivity of the edges shown. This mode highlights areas of interest. In our example with the building blocks (Figure 5) the structure is clarified as the borders between different blocks become more apparent and the crinkles in the skin of the hand are emphasized.

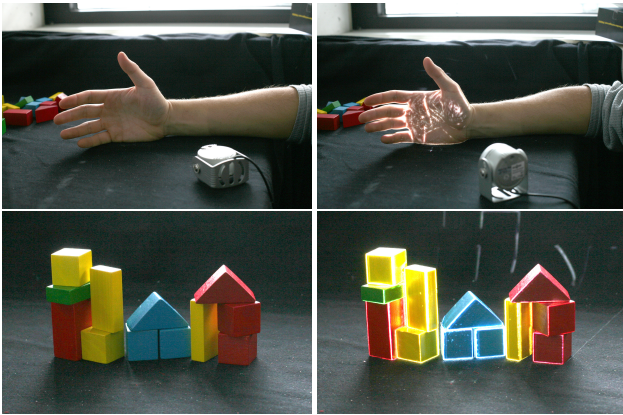


Figure 5. An example for highlighting edges in a scene. Edges are computed from local gradients and illuminated using the context aware light source.

### Object Highlighting

$$f(p) = \begin{cases} \text{red} & \text{if } p \in \text{floodfillSegmentation}() \\ \text{black} & \text{otherwise} \end{cases}$$

This mode highlights a specific object in the scene. We segment the image using a simple flood-fill algorithm seeded on the center pixel. Then, we highlight only this object, guiding the attention of the user. Figure 6 shows this mode being applied to two interlocking swirls. This kind of highlighting could be used to draw attention only to objects with specific properties in a cluttered scene for easy selection or analysis by human viewers.



Figure 6. Object highlighting. With standard illumination, the structure of the two spirals are hard to observe, however with the highlighting, the individual spirals can be easily distinguished.

### Accumulation Light

$$A_p = \max(\text{computeFeature}(I_p), A_p) \\ f(p) = A_p$$

In some cases, certain features of objects are apparent only under illumination from a specific angle. Using a standard visible light source, these features are forgotten when the light moves, making it hard to perceive the entire object at the same time. Using our context-aware light source in conjunction with a detached proxy light, we can store these features in an accumulation buffer ( $A$ ) and display them all together. We fix the context aware light source and move the proxy light by hand, revealing more information than would be visible at any given time with a standard light source. Figure 7 shows two different lighting directions illuminating different features. The words are still hard to read as there is no point in time in which all the features are simultaneously visible. However, with the accumulation buffer, the user is required only to sweep the proxy light over the object, and the most prominent features are remembered, rendering the text readable.

### Style Transfer

$$f(x) = \text{sphereLookup}(N_p)$$

Using our multi-flash proxy light, we can employ photometric stereo [24] to recover surface normals and albedo in real time. The position and angular variation of the LEDs are



Figure 7. Accumulation of features. Two frames are shown from different grazing light angles with standard illumination. The features of the letters is only faintly visible. In the accumulation mode, features are remembered as the user sweeps the proxy light. This allows us to show all the features at once, making the text visible.

calibrated using shiny spheres and a diffuse planar target, and remain fixed to and calibrated with the context aware light source during use.

Once we compute per-pixel normals,  $N_p$ , we can use a technique for style transfer which maps a color to each normal direction by lookup into a pre-drawn sphere [21]. In the equation above, this transfer is called *sphereLookup()*. This approach lets the user not only visualize normals, but also transfer artistic appearances to objects.

Figure 8(a) shows a photograph of a pear illuminated with a surface normal quantization style. In this case, each color corresponds to the closest of the four cardinal directions. Similarly, we can use the surface normals to change the appearance of the Venus figurine shown in Figure 8(b-d). We also note that we can compensate for the cosine shading term of Lambertian objects when we re-project styles to remove their shading effects.

While we use 8 lighting directions and thus 8 times as many images for each output, we compute an estimate of surface normals at every frame using the most recent window of data. It would also be possible to incorporate a high-speed camera and light system into our setup (such as the used in Malzbender et al. [14]) to acquire normals at the same processing speeds as the rest of our applications.

## 5. Limitations

We observed several limitations of our setup, due to its prototype level of construction, that can be fixed in a straightforward manner by additional engineering.

For example, small errors in the geometric calibration of the coaxial setup can cause artifacts, especially when projecting outside the calibration plane. This is occasionally visible when an edge “misses” the border of an object and is projected on another.

as is the case for most consumer-oriented projectors, our projector has a very large aperture so as to maximize brightness. This limits the possible operating scene depth, as objects will quickly become out of focus.

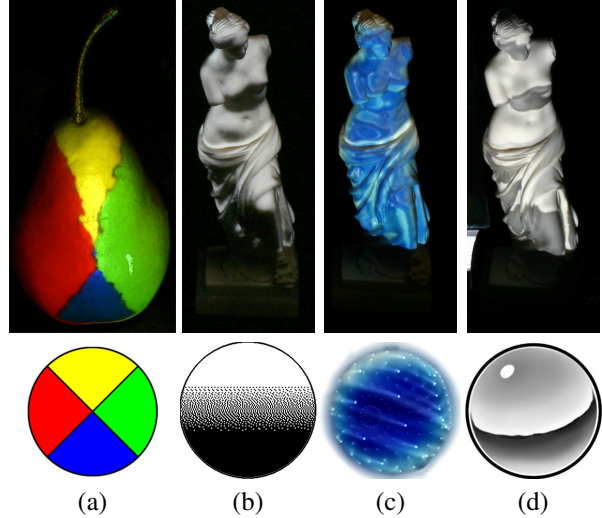


Figure 8. Style transfer result. Mapping the style input images (bottom) on points with corresponding normal direction in the scene can convey information on the geometry orientation (left) or create novel shading styles (right).

Another limitation of our prototype is in the lag between acquisition and projection. While the processing itself takes significantly less time than a single frame’s display time (16.6 ms at 60hz), the projected image in our current system still has a lag of 3-4 frames behind its observations. This is because of the camera’s exposure time (on the order of one frame), the time for transmitting the frame from the camera to the GPU and back to the projector, and the time the projector needs to switch to the next pattern. This lag could be addressed by customized hardware and processing with a FPGA.

There are also some inherent limitations to projector based systems. For example, some materials are difficult to project fine scale illumination on. These are objects with low albedo, high reflectance, subsurface scattering, or micro-geometry. Figure 9 shows an example in which we try to enhance edges on a stuffed plush toy. Although we can compute the image edges in high detail, the projection onto the toy itself is corrupted by the complex light transport through the fur.

Lastly, in cases where the proxy light is used as an interaction tool by the user, there is an additional set of occlusions since both the context aware light source itself and the proxy light must be visible in each illuminated pixel. We found that people are used to working with occlusions and intuitively move the objects to reduce this particular limitation.

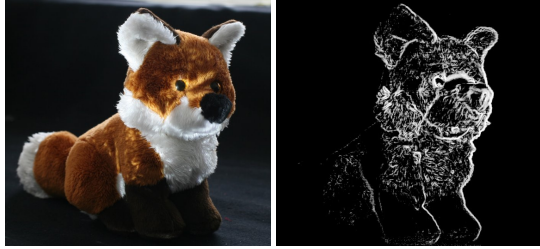


Figure 9. Projecting onto a surface with complex geometry featuring inter-reflections and subsurface scattering usually does not produce desirable results. When projecting on this plush toy (left), most of the edges are either misplaced or absorbed (right).

## 6. Conclusion

A context-aware light source provides an easy way to enhance real objects with feature-driven illumination. These effects are more compelling and more convenient when viewed in the real world than they are on a computer screen. Furthermore, a context-aware light provides a natural and multi-modal way to interact with the scene: users can move objects around by hand, move their heads to adjust the viewpoint, or move the proxy light to intuitively apply unconventional lighting effects to the scene. Finally, the light allows the visualization of information, such as scene normals, that is not directly perceivable to the users.

Because our system processes per-pixel image information without having any knowledge of scene semantics, it can be used on unknown and deformable objects. A purely image-based system allows robust and flexible operation. The brightness, contrast, and focal depth of our setup are limited only by current camera and projector technology.

We have introduced a new lighting device conceptualized as a hand-held smart flashlight. We presented a prototype implementation and have shown some applications, using the light to enhance the appearance of real world objects. Our method is robust, real-time, requires minimal calibration, and our prototype can be easily reproduced with standard hardware. With our current system, we can achieve frame rates between 50 and 60 fps using unoptimized CUDA code. We found that our approach is intuitive to use and convincing to users of our system, who were able to immediately pick up on its operation. With the help of context-aware light sources, we can now create new possibilities for object interaction through computational illumination.

## Acknowledgments

This work has been partially funded by the DFG Emmy Noether fellowship (Le 1341/1-1) and the Max Planck Center for Visual Computing and Communication (BMBF-FKZ01IMC01)

## References

- [1] D. Akers, F. Losasso, J. Klingner, M. Agrawala, J. Rick, and P. Hanrahan. Conveying shape and features with image-based relighting. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 46, 2003.
- [2] T. Amano and H. Kato. Real world dynamic appearance enhancement with procam feedback. In *PRO-CAMS '08: Proceedings of the 5th ACM/IEEE International Workshop on Projector camera systems*, pages 1–2, 2008.
- [3] O. Bimber and I. Daisuke. Superimposing dynamic range. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*, pages 1–8, New York, NY, USA, 2008. ACM.
- [4] R. Fattal, M. Agrawala, and S. Rusinkiewicz. Multi-scale shape and detail enhancement from multi-light image collections. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 51, 2007.
- [5] M. Flagg and J. M. Rehg. Projector-guided painting. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 235–244, New York, NY, USA, 2006. ACM.
- [6] K. Fujii, M. D. Grossberg, and S. K. Nayar. A projector-camera system with real-time photometric adaptation for dynamic environments. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, page 1180, 2005.
- [7] R. Grasset, L. Boissieux, J. D. Gascuel, and D. Schmalstieg. Interactive mediated reality. In *AUIC '05: Proceedings of the Sixth Australasian conference on User interface*, pages 21–29, 2005.
- [8] M. D. Grossberg, H. Peri, S. K. Nayar, and P. N. Belhumeur. Making one object look like another: Controlling appearance using a projector-camera system. In *2004 Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, pages 452–459, Jun 2004.
- [9] C. Jaynes, S. Webb, and R. M. Steele. Camera-based detection and removal of shadows from interactive multiprojector displays. *IEEE Transactions on Visualization and Computer Graphics*, 10(3):290–301, 2004.
- [10] J. C. Lee, S. E. Hudson, J. W. Summet, and P. H. Dietz. Moveable interactive projected displays using projector based tracking. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, 2005.



- [11] K.-L. Low, G. Welch, A. Lastra, and H. Fuchs. Life-sized projector-based dioramas. In *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 93–101, 2001.
- [12] Luminetx. VeinViewer (<http://www.luminetx.com>).
- [13] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 519–528, 2001.
- [14] T. Malzbender, B. Wilburn, D. Gelb, and B. Ambrisco. Surface enhancement using real-time photometric stereo and reflectance transformation. In *Rendering Techniques 2006: 17th Eurographics Workshop on Rendering*, pages 245–250, jun 2006.
- [15] C. Pinhanez. Using a steerable projector and a camera to transform surfaces into interactive displays. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 369–370, 2001.
- [16] R. Raskar, P. Beardsley, J. van Baar, Y. Wang, P. Dietz, J. Lee, D. Leigh, and T. Willwacher. RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 406–415, 2004.
- [17] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 679–688, 2004.
- [18] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. iLamps: geometrically aware and self-configuring projectors. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 7, 2006.
- [19] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with image-based illumination. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 89–102, 2001.
- [20] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR '03: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03) - Volume 1*, pages 195–202, 2003.
- [21] P.-P. J. Sloan, W. Martin, A. Gooch, and B. Gooch. The lit sphere: a model for capturing npr shading from art. In *GRIN'01: No description on Graphics interface 2001*, pages 143–150, Toronto, Ont., Canada, Canada, 2001. Canadian Information Processing Society.
- [22] C. Toler-Franklin, A. Finkelstein, and S. Rusinkiewicz. Illustration of complex real-world objects using images with normals. In *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 111–119, 2007.
- [23] J. Underkoffler, B. Ullmer, and H. Ishii. Emancipated pixels: real-world graphics in the luminous room. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 385–392, 1999.
- [24] R. J. Woodham. *Photometric method for determining surface orientation from multiple images*, pages 513–531. MIT Press, Cambridge, MA, USA, 1989.