Improving the Speed of Virtual Rear Projection: A GPU-Centric Architecture

Matthew Flagg, Jay Summet, and James M. Rehg
{mflagg,summetj,rehg}@cc.gatech.edu
GVU Center and College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

Projection is the only viable way to produce very large displays. Rear projection of large-scale upright displays is often preferred over front projection because of the lack of shadows that occlude the projected image. However, rear projection is not always a feasible option for space and cost reasons. Recent research suggests that many of the desirable features of rear projection, in particular lack of shadows, can be reproduced using active virtual rear projection (VRP). We present a new approach to shadow detection that addresses limitations with previous work. Furthermore, we demonstrate how to exploit the image processing capabilities of a GPU to shift the main performance bottleneck from image processing to camera capture and projector display rates. The improvements presented in this paper enable a speed increase in image processing from 15Hz to 110Hz in our new active VRP prototype.

1. Introduction

Front projection technology, in which projectors are located in the same space as the users, is currently the most economical way to create large flexible displays. However, front projection suffers from the drawback that the projected light can easily be occluded by a user or object which moves between the display and the projector. Rear projection solves this problem, but is much more expensive[1] and commonly results in immobile displays. Additionally, rear-projection is infeasible when attempting to co-opt existing surfaces for display. These observations have motivated us to develop Virtual Rear Projection (VRP) techniques for dealing with the problem of occluders when using front projection [15].

In Passive Virtual Rear Projection, dual projectors redundantly illuminate a display. Although they still cast light on users, they mostly eliminate full shadows, allowing users to interact with the display screen directly in front of them. We currently use passive VRP across a large 17.5×4.5 foot interactive whiteboard (see Figure 1). Additionally, we have installed a passive VRP display in a collaborative design lab at the School of Aerospace Engineering at the Georgia Institute of Technology.

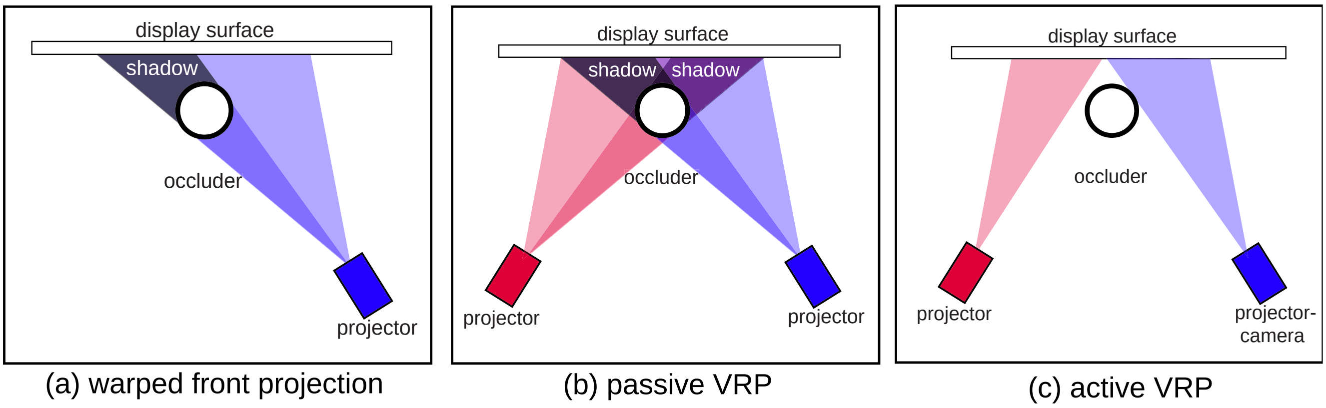
Figure 1: Players interacting with a game on the BigBoard projected using passive virtual rear projection.

Empirical studies of users working with front projection, passive virtual rear-projection, and rear-projection displays indicate that users prefer passive virtual rear projected displays over traditional front projection. Additionally, users working with front projected displays exhibited clear coping behavior when shadows would occlude the display. Participants using passive VRP displays did not exhibit these coping behaviors and behaved the same as participants using rear projected displays [13]. However, passive virtual rear projection is not a full solution, as users still disliked the blinding light projected onto their faces. These empirical studies of passive VRP displays support the need for active VRP [12].

Active virtual rear projection uses visual feedback with a camera to actively detect occluders and eliminate their cast shadows and the blinding light falling on them. Through shadow elimination and occluder light suppression, the illusion and feeling of a rear projection display is created.

---

[1] In addition to the cost of the display surface and its installation, the projection area behind the screen costs an average of $77 (USD) per square foot in an office building [18].



(a) warped front projection (b) passive VRP (c) active VRP

Figure 2: **3 Front Projection Displays**: (a) Warped front projection alleviates shadows by shifting them to the side of interaction. (b) Passive VRP reduces the effects of shadows, but forms two half-power shadows and increases the amount of light falling on the occluder (c) Our active VRP approach uses two projectors and a camera co-located with one projector to eliminate all shadows and occluder light in real-time.

There are two challenges that active VRP must overcome to effectively create the illusion of a rear projection display: (1) *maintaining sufficient display image quality* in the face of shadows and any visible artifacts caused by their elimination and (2) reacting quickly enough to occluder movements to *avoid shadow perception*. In this paper, we address the latter challenge by presenting a new technique based on occluder detection using an infrared camera, as opposed to shadow detection [12], and by exploiting the fast image processing capabilities of current programmable graphics cards (GPUs).

In order to meet the challenge of image quality in an active VRP display, a number of image processing steps must be taken to segment the shadowed regions of the display and treat the resulting masks to hide seams between regions projected by different projectors. In practice, the computational cost incurred by these steps is the primary bottleneck slowing the system's framerate, thereby hindering our ability to adequately address the second challenge of shadow perception avoidance.

In Section 3, we describe the details of these image processing steps and in Section 4 we present our solution for shifting the burden of this computation from the CPU to the GPU. As a result, the image processing framerate increases dramatically from 15Hz to 110Hz, the capture speed of our camera. The boost in performance solves the second challenge up to a limit of occluder movement speed and effectively creates the illusion of a rear projection display, allowing us to pursue user studies that compare the effectiveness of active VRP with traditional front projection and passive VRP displays (see Figure 2).

Although the techniques presented in this paper are developed in the context of improving the performance of active VRP, they are generalizable to many other projector-camera systems which require high speed performance. The contents of this paper and source code included online in the GVU Procams Toolkit [14] should serve as a reference for other researchers to capitalize on modern graphics hardware for their PROCAMS applications.

2. Related Work

Early work on shadow elimination [11, 5] and occluder light suppression [12] required that a camera have an unobstructed view of the display which, inconveniently, cannot be co-located with a projector and required a reference image to be captured before the display was occluded.

The Metaverse lab used a photometric model to eliminate the need for a captured reference image, but their system only eliminated shadows, worked at non-interactive speeds, and still required an unoccluded view of the display [5].

The system described in this paper is inspired by the work of Tan which used an infrared camera and infrared light to backlight occluders for accurate and reliable blinding light suppression [16]. In contrast to their work, however, we address the more challenging problem of shadow elimination and occluder light suppression at highly interactive framerates. By adapting Tan's strategy of occluder detection, our system no longer needs an unobstructed view of the display, which was difficult to obtain during display interaction, but conveniently positions the camera directly (ideally in a coaxial arrangement) with the projector as done in other projector-camera systems [7, 2]. iSkia, a commercial product by iMatte, uses specialized hardware to perform occluder light suppression for single front projector displays using active IR [17]. The techniques presented in this paper could be used to replicate the specialized functionality of iMatte using a standard GPU.

The use of the GPU for image processing and general computation outside the traditional lighting and geometry calculations for computer graphics is becoming commonplace [4, 8, 9, 3]. In particular, the use of pixel shaders with multiple render targets for morphological operations, order-statistics-based filtering, and fast convolution with separable kernels [1] have been applied in many recent videogames such as Tron 2.0 for glowing and blooming effects and video-editing applications like Apple's Motion for real-time previews. Our application of GPU-based image filters with the improved multihead adaptor resource-sharing capabilities provided by DirectX 9.0 should be applied by others in the projector-camera community needing high framerates to drive multiple projectors.

3. Active Virtual Rear Projection

In this section, we present a simple approach to active VRP that eliminates two problems with previous work. First, work by Cham et. al. required positioning a camera in a problematic location such that an occluder's shadow was visible but the occluder was not [12]. Under the assumption that all projected display frames had been captured, shadows were detected using background subtraction and corresponding pixels in each projector were modulated accordingly. A problem with this approach becomes evident when occluders are close to the projected display. When users interact with the display by touching it, for example, the camera detects them as shadows and fruitlessly attempts to compensate for them, triggering an annoying flickering effect.

Second, the use of visual feedback for shadow detection imposes an unnecessary delay in the feedback loop as a pause must be introduced while the projector's output updates before a new camera image is processed. Without this intentional delay, the camera will measure the display before it has been updated and trigger an unwanted update of projector masks under the assumption that shadows had not been eliminated in the previous compensation iteration. This results in a flickering effect similar to the one caused by falsely detecting occluders as shadows. By adopting a strategy of occluder detection instead of shadow detection, we can solve both of these problems.

3.1. System Setup

In our approach, we position a camera close to the projector lens so that detected occluder silhouettes align with corresponding projector mask silhouettes with little to no parallax effects caused by projector-camera disparity. If the optical axes of the projector and camera are aligned by means of a beam-splitter, parallax effects are eliminated [7]. To simplify the detection of occluders, the camera is filtered to pass near-infrared light and the display surface is illuminated with infrared lights. Background subtraction of the IR camera images is not affected by projected light and, as shown in Figure 6(b), the backlit silhouette of occluders creates a strong contrast between foreground and background. For our prototype, we placed two infrared floodlights on the floor and one foot from the display surface and positioned them such that their illumination covered most of the display surface.

Because we are detecting occluders (instead of shadows) we do not need to pre-shoot background plates for each expected frame [12] or predict the expected appearance of each image when projected onto the display surface [5].

3.2. Occluder Mask Construction

For each compensation step, the IR camera image must be processed to meet the challenge of preserving high image quality in the face of varying pixel-projector ownership. These steps are illustrated in Figure 6. First, the acquired image must be warped to align with the display surface using a camera-surface homography. Second, the image is segmented into occluder and non-occluder regions. Our implementation uses background subtraction. In some cases, median filtering is needed for noise removal, but in our experiments the backlit occluders were easily segmented without noise. Third, the occluder regions are dilated to allow a region of tolerance for occluder movement within each compensation step. Finally, the mask is blurred to blend seams between projectors. Figure 3 illustrates the necessity for blending to avoid distracting seams.

Figure 3: Boundary between regions of varying projector ownership. **Left**: before seam blending **Righ**t: after seam blending

3.3. Enforcing Photometric Uniformity

Following occluder mask creation, each projector's output is computed using appropriate projector-surface warps and luminance attenuation maps (LAMs) to correct for photometric non-uniformity. Each projector's output is warped in the normal way using pre-calibrated projector-surface homographies [10] to transform the projection matrix in Direct3D or OpenGL. One limitation of overlapping two obliquely positioned projectors to create a display is the variation in display resolution caused by projector keystoning and changes in display region ”ownership” from one projector to another. In practice, we found that this resolution variation is only noticeable when carefully viewed up close (less than 6 inches) and does not cause a distraction to the user.

To correct for photometric non-uniformity, a known problem for overlapping or obliquely positioned projectors [6], we employ LAMs to attenuate the light output of each projector pixel such that each projector projects a display that is spatially uniform in brightness and is consistent across all projectors. We use a simple feedback-based technique based on previous work in photometric compensation [7] to construct a LAM for each projector. The darkest intensity measured when projecting white from each projector independently is set as a target. All pixels are iteratively reduced in intensity one step at a time (to account for non-linear projector and camera responses) until the target intensity is uniform across the display. Figure 4 shows two example LAMs and the following pseudocode describes our simple algorithm for their creation:

```
CREATE-LAMS:

for each projector p
  1. project white for p and
     black for all other projectors
  2. capture image
  3. if darkest intensity d for projector p
     is darker than overall darkest intensity d*,
     d* = d
  4. initialize LAM(i,p) = white for all pixels i
end for

for each projector p
  initialize l = UPDATE_LIMIT
  project black for all other projectors
  while l > 0
    project LAM(*,p) and capture image
    for each pixel i
      if (intensity(i) > d*)
        LAM(i,p)--
      end if
    end for
    l--
  end while
  low-pass filter LAM(*,p)
end for
```

3.4. Projector Roles

Finally, each projector is assigned one of two roles: (1) shadow eliminator and (2) unoccluded light projector. The shadow eliminator uses the occluder mask to illuminate the regions where a shadow would be cast by the occluder. Note that for the shadow to be eliminated, the projector must have an unoccluded view of the display surface. This is achieved for most occluder configurations by placing the shadow eliminating projector at an oblique position far from the unoccluded light projector.

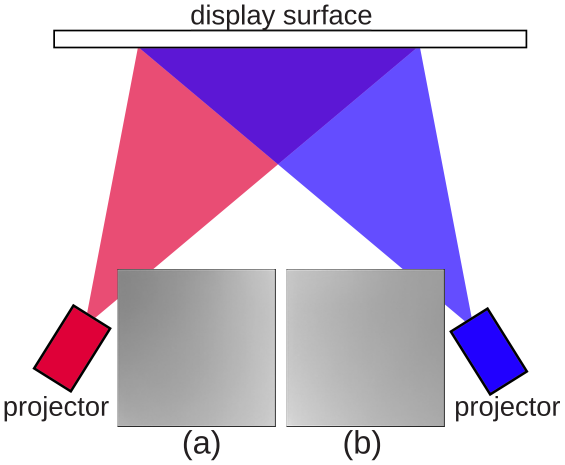


Figure 4: **Luminance Attenuation Maps (LAMs)**: (a) LAM for projector positioned to left of projection surface (b) LAM for projector positioned to the right of the projection surface. Note that the dark regions of each LAM correspond with the shortest projection distance to the display surface.

The unoccluded light projector has the camera attached to it and is in charge of projecting the pixels on the display that are not occluded. This is simply achieved by taking the inverse of the occluder mask. By adopting specific virtual rear projection roles for each projector, both goals of shadow elimination and occluder light suppression are met by each projector independently. Note that by removing the shadow eliminating projector, the projection system reduces to a simple unoccluded light projector with functionality that is similar to the iSkia product by iMatte [17] or Tan’s pre-emptive shadows [16], which use a single projector and camera.

4. Improving Performance Using the GPU

As users move in front of an active VRP display, they may cast new shadows by moving faster than the system can update the screen. This occurs when the users move outside of the region of tolerance created by the dilation operation before the display is updated. Increasing the system framerate and decreasing system latency enables users to make quick natural movements such as emphasizing a point with a fast hand gesture. The image processing steps described in Section 3.2 may be optimized by exploiting today’s programmable graphics cards (GPUs). Image processing on the GPU shifts the speed limit of active VRP away from computation on the CPU to capture and display rates of the camera and projector. Figure 6 illustrates our image processing pipeline using the GPU.

There are three capabilities of GPUs and DirectX 9.0 that we exercise in order to eliminate the bottleneck of image processing: (a) multiple render targets, (b) pixel shaders and (c) multihead resource sharing. First, the Multiple Render Targets (MRT) capability provided with Direct3D as of DX9.0 enables us to store the results of each image processing step in an off-screen rendering surface for succeeding filter operations to use as input. By setting the texture coordinates (u,v) of a screen-aligned quadrilateral to correspond with the camera image coordinates (x,y) of the projected display, the camera-surface warp may be performed by rendering the quadrilateral texture-mapped with the camera image. The warped texture is now available on an off-screen surface for subsequent filtering using pixel shaders.

The second capability provided by GPUs is fast image processing using pixel shaders. Background subtraction, dilation, median filtering and blurring may be implemented as pixel shader programs [1]. These pixel shaders were written in DirectX High-Level Shader Language (HLSL). Using two texture samples and a threshold, the result of a background subtraction shader is stored in the first of two off-screen render targets. Next, dilation is performed using two separate pixels shaders. The first shader dilates the result of background subtraction using 1D texture samples horizontally and the second dilates the resulting texture vertically. Separating dilation into two operations decreases the number of required texture samples and improves performance from $O(n^2)$ to $O(n)$. To further improve processing time, the two off-screen render textures were reduced to a resolution of 128×128 pixels (to be sub-sampled during compositing operations). Following dilation, blurring is performed in a similar manner using two separate shaders. Finally, the resulting occluder mask is composited with the display frame using one pixel shader. The interaction between each pixel shader and the input / output textures used by them is illustrated in Figure 5.

Finally, multihead resource sharing provided as of DX9.0 makes it possible to use one DirectX rendering device across multiple display heads. Previously, each head required its own device and therefore needed separate sets of textures and pixel shader computations. By using one device instead of two, some of the pixel shaders need only be executed once saving time and texture memory. A background subtraction and dilation pixel shader computation is removed. An initial dilation of n pixels is performed to permit sufficient occluder movement within frame updates. A second dilation of k pixels is needed to overlap projector masks before blending. Before multihead resource sharing, one display device performed $2n$ texture samples and the other sampled $2(n+k)$ pixels ($4n+2k$ total samples). After multihead sharing, a dilation using $2n$ texture samples is shared among both display heads and a remaining $2k$ pixels are sampled for the overlapping region ($2n+2k$ total samples), saving $2n$ texture samples per pixel. Following dilation, blurring and compositing operations must be performed for each display head separately due to differences between the occluder masks.

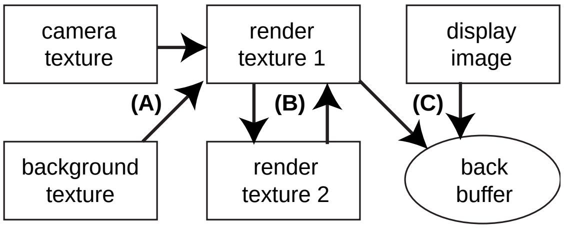


Figure 5: **Pixel Shader Pipeline**: Boxes represent textures and arrows denote texture sampling operations used in pixel shaders. **(a)** Background subtraction shader stores result in render texture 1 **(b)** Render textures 1 and 2 are used as sampling buffers for dilation and blurring operations, each of which require 2 independent shaders **(c)** The final occluder mask is composited with a display texture and rendered into the DirectX back buffer for display.

5. Results

We performed two experiments to test the framerate and latency of our active VRP system. The active VRP system components consist of: (a) Lumenera LU-070M USB2.0 camera with an IR-pass filter, capable of capturing at 110Hz, (b) dual P4-2.2GHz PC with AGP4x graphics bus, (c) NVidia GeForceFX 6800GT graphics card, and (d) two Hitachi CP-SX5600 LCOS projectors with a maximum vertical refresh rate of 85Hz. The first experiment measured the time to capture an image and render (including all pixel shading) 3000 frames. A framerate of 110Hz was measured for image capture and processing time which is the same framerate as the camera. This experiment measured the rate that the graphics card's back buffer was updated, which is faster than the actual projector display rate of 85Hz. Therefore, the overall capture, process and display framerate is only as fast as the 85Hz refresh rate of the projector. In previous experiments using the same Lumenera camera, a system framerate of only 15Hz was recorded when performing all image processing on the CPU using IPL2.5 and OpenCV3.1b.

The second experiment measured the overall system latency between image capture and display. The components of the system process that we measured were: (1) camera capture to PC memory, (2) texture copy from PC memory to GPU memory, (3) rendering including pixel shaders and (4) projector display. To verify the latencies of the first three components, 100 trials of 1000 measurements were

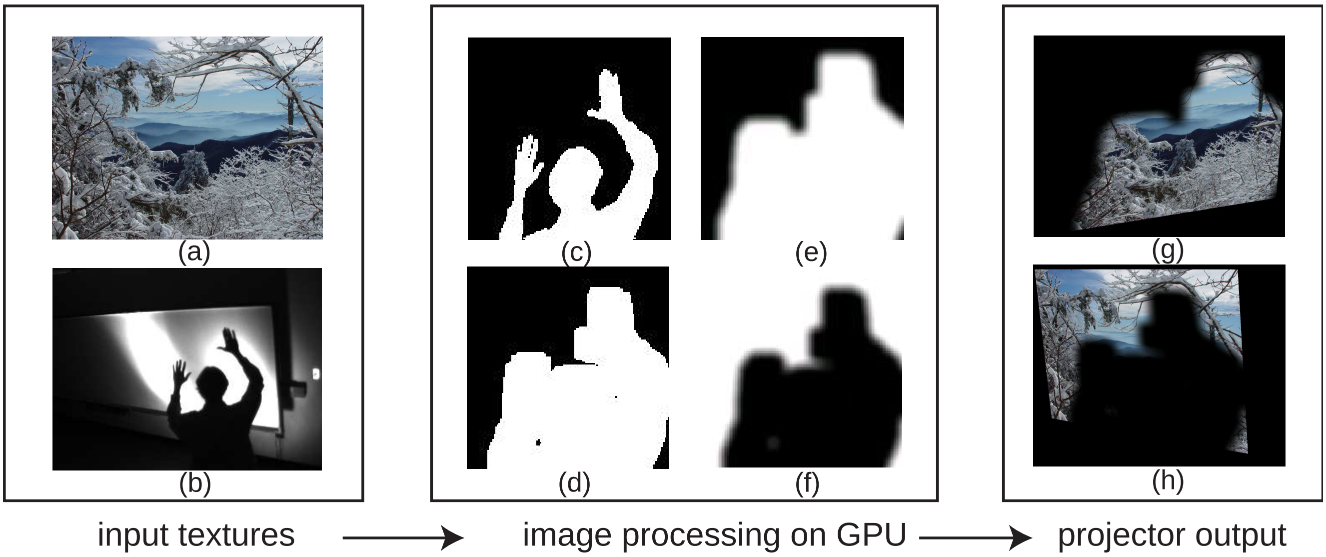


Figure 6: **GPU-centric architecture**: (a) display texture (b) IR camera frame (c) occluder mask texture (d) dilated mask to tolerate inter-frame occluder movement (e) blurred mask for projector 1 blending (f) blurred mask for projector 2 blending (g) keystone-corrected projector 1 output (h) keystone-corrected projector 2 output

recorded. Although the image processing was limited by the camera capture rate of 110Hz, our experiments showed a GPU processing time of only 2.14ms (467Hz) when measured without camera capture. The fourth component, projector latency, includes the time from frame buffer update on the graphics card to a change in lighting on the display surface. To measure projector latency, the IR filter was removed from the Lumenera camera. The number of camera frames recorded before a display change was detected following a request to change the display from black to white. Because the projector latency was measured using the camera, the actual projector latency could only be measured with 9ms of accuracy. To get a more accurate estimate of this latency, 100 trials were recorded. An average of 4.43 camera frames were recorded before a change in the display was detected. The following table lists the latency of each component:

System Component	Latency
Camera Capture to PC-Memory	9.09ms
PC-Memory to GPU-Memory	1.70ms
Pixel Shaders	2.14ms
Projector	40.27ms
Total Latency	**53.20ms**

Although the system is capable of capturing and processing camera images faster than the projector refresh rate, the projector latency accounts for 76% of total system latency and therefore serves as the main bottleneck to avoiding shadow perception. Assuming an occluder movement tolerance of 5cm, a typical distance measured in our experimental setup, a user will exceed the limit of projector latency upon moving a point on the edge of the occluder's silhouette 94cm/s or faster. While this limit is high enough for casual walking and moderately slow hand movements for display interaction, it can certainly be exceeded by flapping arms quickly or jumping up and down (see included video for example). These results suggest that the goal of *shadow perception avoidance* can be attained across all reasonable occluder movements (natural human motion) with a low enough projector latency.

6. Conclusions and Future Work

In this paper, we presented a new approach to active virtual rear projection that improves system framerate and latency such that shadows are imperceptible up to a limit of occluder movement. By adopting a strategy of occluder detection using active IR illumination, two problems with previous approaches to active VRP using shadow detection were eliminated: (a) inconvenient camera placement to avoid camera occlusions being mistakenly detected as shadows and (b) intentional lag in the feedback loop to account for high projector latency. Furthermore, we demonstrated how to take advantage of the GPU and DX9.0 capabilities to shift all image processing required to maintain image quality from the CPU to the GPU. These speed improvements place the latency limit on projector and camera hardware and eliminate the previous bottleneck of image processing.

In future work, we plan to take a closer look at projector latency and explore methods for decreasing it with a hardware solution. Before the presented speed improvements were made, active VRP was not ready for a user evaluation because of the visibility of large distracting shadows when users moved. These improvements make it possible to replicate our laboratory evaluation of passive virtual rear projection [13] with active VRP included.

We have installed a passive VRP display in a collaborative design lab at the School of Aerospace Engineering at the Georgia Institute of Technology. The speed improvements presented in this paper enable us to upgrade the display in the Aerospace Engineering design lab from passive VRP to active VRP so that we can observe it in real-world use. This will also help us to identify remaining problems and challenges for active VRP.

Acknowledgments

We would like to thank Ravi Ruddarraju and Charles Brubaker for their helpful discussions.

References

[1] Wolfgang F. Engel. *Shaderx2: Shader Programming Tips & Tricks With Directx 9*, chapter Advanced Image Processing with DirectX 9 Pixel Shaders. Wordware Publishing, 2003.

[2] R. Raskar et al. iLamps: Geometrically aware and self-configuring projectors. In *Proceedings of SIGGRAPH*, 2003.

[3] James Fung and Steve Mann. Computer vision signal processing on graphics processing units. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*.

[4] Mark J. Harris. Gpgpu - general purpose computation using graphics hardware. ⟨`http://www.gpgpu.org`⟩.

[5] Christopher Jaynes, Stephen Webb, R. Matt Steele, Michael Brown, and W. Brent Seales. Dynamic shadow removal from front projection displays. In *Proceedings of the conference on Visualization 2001*, pages 175–182. IEEE Press, 2001.

[6] Aditi Majumder and Rick Stevens. Lam: Luminance attenuation map for photometric uniformity in projection based displays. In *Proceedings of ACM Virtual Reality and Software Technology*, pages 147–154, 2002.

[7] S. Nayar, H. Peri, M. Grossberg, and P. Belhumeur. A projection system with radiometric compensation for screen imperfections. In *First IEEE International Workshop on Projector-Camera Systems (PROCAMS-2003)*, 2003.

[8] Robert Strzodka and Christoph Garbe. Real-time motion estimation and visualization on graphics cards. In *Proceedings of IEEE Visualization*, 2004.

[9] Robert Strzodka and Alexandru Telea. Generalized distance transforms and skeletons in graphics hardware. In *Proceedings of EG/IEEE TCVG Symposium on Visualization (VisSym '04)*, pages 221–230, 2004.

[10] R. Sukthankar, R. Stockton, and M. Mullin. Smarter presentations: Exploiting homography in camera-projector systems. In *Proceedings of International Conference on Computer Vision*, 2001.

[11] Rahul Sukthankar, Tat Jen Cham, Gita Sukthankar. Dynamic shadow elimination for multi-projector displays. In *Proceedings of CVPR*, pages 151–157, 2001.

[12] Tat Jen Cham, James M. Rehg, Rahul Sukthankar, Gita Sukthankar. Shadow elimination and occluder light suppression for multi-projector displays. In *Proceedings of Computer Vision and Pattern Recognition*, pages 513–520, 2003.

[13] Jay Summet, Gregory D. Abowd, Gregory M. Corso, and James M. Rehg. Virtual rear projection: Do shadows matter? In *CHI'05 Extended Abstracts*, 2005.

[14] Jay Summet, Matthew Flagg, and James M. Rehg. Gvu procams toolkit. ⟨`http://www.cc.gatech.edu/cpl/procams/`⟩.

[15] Jay Summet, Matthew Flagg, James M. Rehg, Gregory M. Corso, and Gregory D. Abowd. Increasing the usability of virtual rear projection. In *First IEEE International Workshop on Projector-Camera Systems (PROCAMS-2003)*, 2003.

[16] Desney S. Tan and Randy Pausch. Pre-emptive shadows: Eliminating the blinding light from projectors. In *CHI'02 extended abstracts*, pages 682–683, 2002.

[17] Paul Vlahos. Method and apparatus for inhibiting the projection of a shadow of a presenter onto a projection screen. United States Patent 6,860,604, 2005.

[18] Phillip R. Waier, editor. *Building Construction Cost Data 2002.* Robert S. Means Co., 2002.