Projectibles: Optimizing Surface Color For Projection

Brett R. Jones, Rajinder Sodhi, Pulkit Budhiraja, Kevin Karsch, Brian Bailey, David Forsyth
University of Illinois at Urbana-Champaign
{brjones2, rsodhi2, budhirj2, karsch1, bpbailey, daf}@illinois.edu

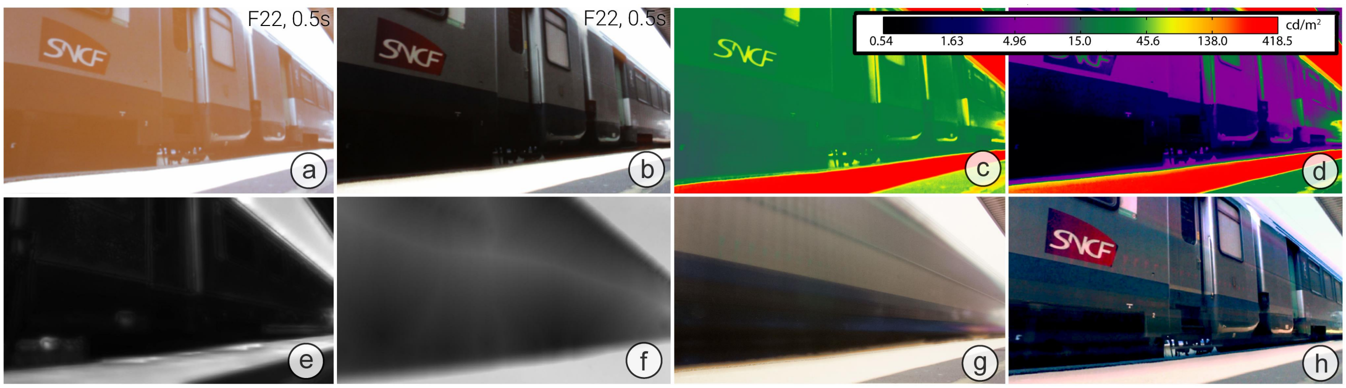

Figure 1. Results for a looped video sequence of a train passing by a station (see Figure 2a for source video). (a, b) Shows real world captured images (F22, 0.5s, 800 ISO) of (a) a normal projected image on a white screen, and (b) the Projectible combining projected light with a printed display surface. (c, d) Log plots of the real-world measured luminance in cd/m^2 for (c) the projector only, and (d) the Projectible display. We compute a perceptual error metric (e) for each frame of the sequence, and use it to smoothly vary (f) the contrast across the display. We then separate out the printed image (g) and the projected image (h), which combine to form the Projectible. Please see the video.

ABSTRACT
Typically video projectors display images onto white screens, which can result in a washed out image. *Projectibles* algorithmically control the display surface color to increase the contrast and resolution. By combining a printed image with projected light, we can create animated, high resolution, high dynamic range visual experiences for video sequences. We present two algorithms for separating an input video sequence into a printed component and a projected component, maximizing the combined contrast and resolution while minimizing any visual artifacts introduced from the decomposition. We present empirical measurements of real-world results of six example video sequences, subjective viewer feedback ratings, and we discuss the benefits and limitations of Projectibles. This is the first approach to combine a static display with a dynamic display for video, and the first to optimize static surface color for projection of video.

Author Keywords
projection mapping; radiometric compensation

ACM Classification Keywords
H.5.1 Multimedia Information Systems: Artificial, augmented, and virtual realities

INTRODUCTION
Typically video projectors are used with white screens (e.g. in conference rooms). Movie theaters frequently use gray screens for darker black levels. Making the screen darker makes the content darker, effectively multiplying the appearance of the display by a constant. For darker video sequences like a city in twilight, a darker screen would look better because of darker black levels, but brighter scenes like a sunny sky would look better with a lighter screen. Thus, there is no optimal screen color for displaying every video sequence.

In this work, we present *Projectibles* which use a spatially varying display surface color for video projection, greatly increasing the contrast and resolution of the combined display. For a particular video sequence, we algorithmically determine a static printed image, then overlay video projection onto the printed image, creating a high dynamic range (high contrast), high resolution display, that can be larger than flat panel displays, have any contour (non-rectangular), and can even be 3D (non-flat).

Projectibles combine two low dynamic range (LDR) displays, a printer and projector, to create a high dynamic range (HDR) display. High dynamic range is important because the dynamic range of many real-world scenes exceeds the capabilities of current display technology by several orders of magnitude [25]. For instance, the sun is 8 orders of magnitude brighter than twilight. Traditional displays simply cannot reproduce the feeling of seeing a sunset in real life, or the twinkling lights of a city at night.

Projectibles have the most effect on videos with some spatial and temporal consistency. These types of videos are often

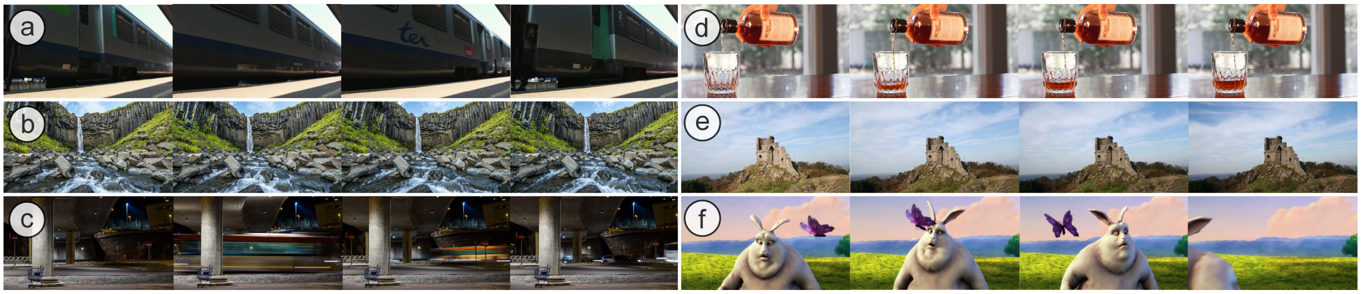

Figure 2. Example frames of the looped video sequences (see the supplemental video). (a) *Train*: an exposure bracketed HDR video shot with a DSLR of a train passing a station. (b) *Svartifoss*: an ultra resolution [7360,4140] RAW time-lapse of the Svartifoss waterfall (credit: Stian Rekdal, http://www.stianrekdal.com). (c) *Underpass*: a high res time-lapse video of an underpass at night (credit: Eric Solheim, http://eirikso.com). (d) *Whiskey*: A video of pouring whiskey made by combining a high res still image with 1080p video. (e) *Castle*: A high res time-lapse of Mow Cop Castle (credit: NatureClip, http://natureclip.co.nr). (f) *Bunny*: An animated sequence from Big Buck Bunny (credit: Blender Foundation, https://peach.blender.org).

called *Cinemagraphs* or *Cliplets* [16, 21], which are a recent trend in art and advertising that combine still images with video sequences, and are often distributed as '.gif' sequences or looped videos (for examples, see Figure 2 and the supplementary video). These videos are frequently used for advertising, as the subtle motion is great at capturing attention.

With Projectibles, for each pixel in the video, if the pixel has similar colors through time, we can alter the screen color at this location to make the darks darker or the brights brighter. For instance, Figure 1 shows a looped video sequence of a train passing by a station. Parts of the image are consistently brighter (the sky), and others are consistently darker (the bottom of the train). We can determine a static printed image to serve as the display surface for video projection that increases the contrast of the overall video. The process of determining the printed image is non-trivial as it is limited by the dynamic range of the printer and projector, and the amount of motion in the input video. An incorrect choice of the printed image could lead to incorrect colors in the combined display.

Projectibles can be much higher resolution than traditional digital displays. Printers have a much higher effective resolution than projectors. Projectibles use the printed image to increase the resolution of the display in regions with temporally consistent high resolution details. For instance, in the waterfall sequence (Figure 2b) while the water and sky are moving, the cliffs and moss are stationary with only lighting changes from passing clouds. By printing high resolution details of the cliff and moss, the resolution of the combined display can be increased in these regions (see Figure 6). Projectibles have spatially varying contrast and resolution, with large improvements in contrast and resolution in areas with temporal consistency throughout the video.

The core problem of Projectibles is to decompose an input video into a static (printed) component and a dynamic (projected) component. Naive algorithms for decomposition result in distracting visual artifacts, or poor contrast/resolution. We present two algorithms for splitting a video into a static and dynamic component, based on models of human perception, that maximize the contrast ratio and resolution, while minimizing visual artifacts. We present an offline constrained optimization method, and a faster heuristic solution which enables the content author to adjust the parameters for the solution in near real-time. Finally, we present empirical measurements of real-world results, along with subjective ratings of viewer feedback, and discuss the benefits and limitations of Projectibles. Overall, this is the first approach to combine a static display and a dynamic display for video, and the first to optimize static surface color for the projection of video.

BACKGROUND

Most closely related, Bimber el al. [6] demonstrated a static HDR display by combining a static printed image with a static projected image. For static images, splitting into printed and projected is as simple taking a square root of the input after linearizing each device. As we will describe later, this only works for static images and will lead to visual artifacts for video sequences.

Previous work has also combined two dynamic LDR displays to create an HDR display, e.g. by stacking two LCD panels on top of each other [12], combining LCD panels and LED back-lights [28, 25, 26], or LCD panels with projectors [25]. Other work combined E Ink with synchronized lighting [18], and UV LEDs with special paint [15], but yields low resolution. We also derive inspiration from work in HDR/super-resolution displays using stacked LCD panels and tensor factorization methods [14, 19]. All of these displays are capable of creating HDR scenes for arbitrary input and therefore show great promise, but require custom hardware solutions. Projectibles use commodity hardware, and unlike previous work, Projectibles can be made into any shape or contour.

Projectibles optimize the surface color for projection, but a wide range of previous work has established methods to compensate the projected image for an existing surface color. Using a process called radiometric compensation [7], previous work has demonstrated turning a curtain [5], or even a painting [4] into a display. If we want to achieve a desired appearance, we can model the surface reflectance and the incoming light in the scene. Then given an existing surface color, we can solve for the required projected light. The lighting model can be as simple as a Lambertian model [11, 22, 2, 29, 20], include color mixing between projector channels [23], or be as complex as a global illumination model [30, 27].

Usually, radiometric compensation is accomplished by using a camera to establish a geometric transfer function between camera pixels and projector pixels, often using structured light [20, 3, 7]. In this work, we focus on the core problem of optimizing the surface color and we use a homography for geometric registration. Future work could extend our approach to support more complex geometric registration.

Unfortunately, radiometric compensation is not perfect and is limited by the existing surface color and the dynamic range of the projector. This results in distracting visual artifacts. One solution is to use multiple projectors [1], increasing the brightness and decreasing artifacts, but also increasing setup complexity and system cost.

In this paper, we combine a wide background of previous work in radiometric compensation, building on previous methods that optimize the black/white level based on the surface color, the projector dynamic range, and mathematical models of human perception [8, 22]. We also build on content-dependent methods that optimize over the particular video being shown, particularly previous work in offline optimizations [29, 2, 20] and real-time systems [11]. This previous work all assumes an existing fixed display surface color. We take parts and pieces from each method, and present two novel algorithms for optimizing the surface color for projection of a video sequence.

LIGHTING MODEL

In this work we use a Lambertian lighting model, as the display surface is diffuse professional quality matte paper. We use a homography to register the projected content onto the printed page. Initially, we will assume that the printer and projector are equal in resolution.

We adapt the lighting model for a projector-camera system from [17]. At each pixel on the display surface (x, y), and for each frame t of the video sequence, the final appearance $L(x, y, t)$ can be modeled as follows,

$$L(x, y, t) = k(s(x, y))\Big(i(d(x, y, t)) f(x, y) + a(x, y)\Big). \quad (1)$$

The reflectivity coefficient of the surface (i.e. surface color) is defined by k, the printer's spatially invariant transfer function. It is a function of the normalized input *static* image $s(x, y)$, sent to the printer. The contribution from incoming light can be broken down into three components. First, $i(d)$ is a spatially invariant transfer function that describes the color response of the projector, where $d(x, y, t)$ is the normalized input image for each frame sent to the projector. Second, $f(x, y)$ is a spatially variant function incorporating the vignetting in the optics of the projector, the form factor between the projector and the surface normal, and light falloff. Finally, the contribution of ambient light, $a(x, y)$, is a spatially variant function over the display surface, incorporating all incoming ambient light and display surface form factors.

To measure the above terms, we use a ColorMunki spectrophotometer and HDR photographs from a DSLR camera. We measure k and i directly using the spectrophotometer. We measure f by taking an HDR exposure bracketed image sequence of a projected white image without ambient light. We measure a with an HDR image of the ambient light only.

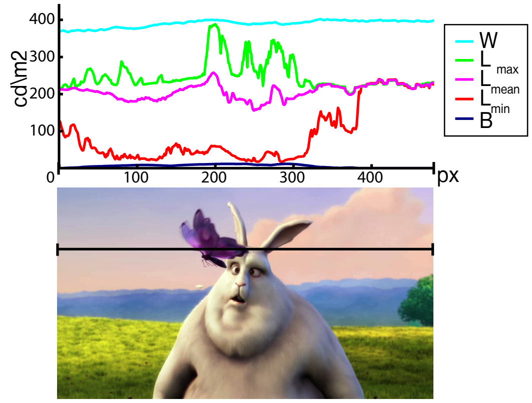

Figure 3. Statistics of the luminance of a single row (noted in black) from the Bunny sequence. The butterfly flies across the screen to the left, and the bunny exits to the left. The right quarter of the video is static. We plot the white level $W(x, y)$, black level $B(x, y)$, maximum luminance $L_{max} = \max_t(L_g(x, y, t))$, minimum luminance $L_{min} = min_t(L_g(x, y, t))$ and mean luminance $L_{mean} = mean_t(L_g(x, y, t))$.

After calibration, we simplify the lighting equation by creating two new terms representing the overall contributions from the printed component $K(x, y) = k(s(x, y))$ and the projected component $D(x, y, t) = i(d(x, y, t)) f(x, y) + a(x, y)$. The lighting equation becomes

$$L(x, y, t) = K(x, y) D(x, y, t). \quad (2)$$

PROBLEM FORMULATION

The core challenge of Projectibles to decompose a dynamic input video into a static printed component and a dynamic projected component, while maximizing the display's contrast ratio and minimizing any visual artifacts from the separation process. We can break this down into two main steps, solving for the black/white level of the display, and solving for the printed/projected component.

Black/White Level

The maximal value a display can show at each pixel is called the white level of the display, and the minimal value is the black level. We start with an input video sequence $I(x, y, t)$ in CIE XYZ coordinates with the luminance values (Y coordinate) normalized. The desired goal appearance of the Projectible $L_g(x, y, t)$ should be a linear interpolation between the black level $B(x, y)$ and the white level $W(x, y)$.

$$L_g(x, y, t) = I(x, y, t)\Big(W(x, y) - B(x, y)\Big) + B(x, y) \quad (3)$$

We are combining two displays and have a black/white level for each display. For the printed component $K(x, y)$, the white level (paper white) is a measured constant K_w, and the black level (paper white with black ink) is a measured constant K_b. For the projected component $D(x, y, t)$, the white level (projected white and ambient light) is a spatially varying measured value $D_w(x, y)$, and the black level is a spatially varying measured value $D_b(x, y)$. The projected black/white levels are spatially varying because projected displays are brighter at the center of projection [22]. The ideal combined display would utilize the maximal possible contrast ratio, and have a

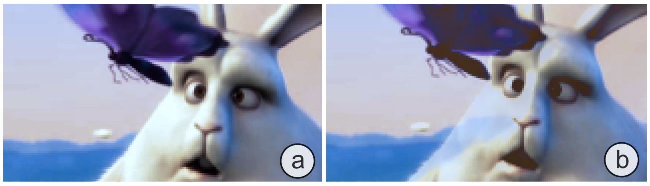

Figure 4. Artifacts result when $L(x,y,t) \neq L_g(x,y,t)$. (a) The desired appearance $L_g(x,y,t)$ is determined by the black and white level of the display. (b) The actual physical appearance $L(x,y,t)$ is determined by the printed and projected components. Here we have a sub-optimal solution (printing the square root of the temporal mean with $B = B_{min}$ and $W = W_{max}$), resulting in visual artifacts (b) on the left side of the bunny's face.

maximal white level $W_{max}(x,y) = K_w D_w(x,y)$ and a minimal black level $B_{min}(x,y) = K_b D_b(x,y)$.

Printed/Projected Components

The actual physical appearance of each pixel of the display $L(x,y,t)$ is the combination of the printed and projected components. We want the actual appearance of each pixel to be the same as the desired appearance, $L(x,y,t) = L_g(x,y,t)$. Thus, given a printed component $K(x,y)$, we can solve for the required projected component for each frame, $D(x,y,t) = L_g(x,y,t)/K(x,y)$.

We will have visual artifacts if $L(x,y,t) \neq L_g(x,y,t)$ (see Figure 4). For a given printed component $K(x,y)$, we will have visual artifacts if the solution for the projected component exceeds the dynamic range of the projector, which is defined by the maximum/minimum luminance values.

$$\frac{L_g(x,y,t)}{K(x,y)} > D_w(x,y), \quad \frac{L_g(x,y,t)}{K(x,y)} < D_b(x,y) \tag{4}$$

Overall, we will have clipping artifacts if the temporal dynamic range of our target appearance exceeds the projector's temporal dynamic range.

$$\frac{\max_t\left(L_g(x,y,t)\right)}{\min_t\left(L_g(x,y,t)\right)} > \frac{D_w(x,y)}{D_b(x,y)}. \tag{5}$$

We need to determine what our printed image should be given our input looped video sequence. As with previous static/static [6] and dynamic/dynamic [12, 28, 25] HDR displays, the logical place to start is to separate the input video into each display equally, which places equal bits of information into each display, and maximizes their combined contrast ratio. In a static/static display [6], we could choose $s(x,y) = d(x,y) = \sqrt{I(x,y)}$, where I is the input image. But this doesn't work for video sequences as we have to choose a single printed image for our entire video sequence. Instead, we could use the square root of the average value of each pixel, or $s(x,y) = \sqrt{\text{mean}_t(I(x,y,t))}$. However, this will result in visual artifacts whenever Equation 5 is true, thus clipping pixels with high temporal dynamic range (see Figure 4).

Instead, we could choose to accurately reproduce the brightest colors, by setting $K(x,y) = \max_t\left(L_g(x,y,t)\right)/D_w(x,y)$, which concentrates the reproduction errors into darker content. As Figure 5 shows, always giving preference to brighter content could result in prominent visual artifacts in darker content that last the entire video sequence.

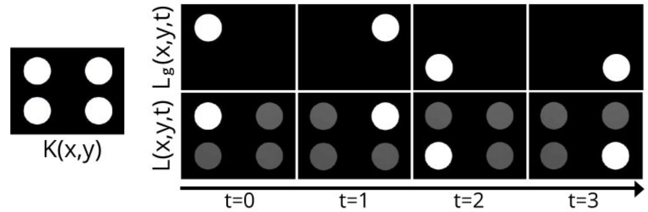

Figure 5. An example 4 frame input video. A naive solution for the printed image $K(x,y)$ that uses the maximal contrast ratio and gives preference to brighter content would concentrate the error into darker content in a non-spatially uniform way, leading to permanent gray circles in the background of the entire video sequence. $L_g(x,y,t)$ is the goal appearance, and $L(x,y,t)$ is the actual appearance.

Adjusting the Black/White Level

To reduce visual artifacts we can reduce the contrast ratio of the display, by artificially raising the black level or lowering the white level. For each pixel in the display, we solve for $B(x,y)$ and $W(x,y)$ that minimizes our reproduction error $(L(x,y,t) - L_g(x,y,t))$, subject to the constraints that $B(x,y) > B_{min}(x,y)$ and $W(x,y) < W_{max}(x,y)$. Additionally, we want the black/white levels to be smooth. Consider two pixels with equal values in the source image $I(x_0,y_0,t_0) = I(x_1,y_1,t_1)$, ideally the final display values will also be equal, $L(x_0,y_0,t_0) = L(x_1,y_1,t_1)$, but due to spatially varying black/white levels the values may not be equal. However, if the change in black/white levels is smooth enough the human visual system will not notice it [22].

In Projectibles, we allow the content author to choose between altering the white level or black level of the display. For content that is predominately brighter (through space and time), it would be very noticeable to lower the white level, and less noticeable to raise the black level, and vice versa for predominately darker content. Previous work in radiometric compensation focused on projection onto existing surfaces, and could not control the surface color and only solved for the white level of a display. However, as our display surface is completely controllable, we can either make the entire display brighter or darker based on the content author's goals.

Perceptual Error

So far we have only considered the physical error $((L_g(x,y,t) - L(x,y,t))$ in (cd/m^2), and have not taken into account how perceptually noticeable these errors are to the viewer. Similar to [11, 29] we use the perceptual physical error metric developed in [24] which leverages three known properties of human visual perception to approximate perceptual error instead of physical error. First, our visual system is more sensitive at lower luminance levels than at higher luminance levels. Second, our visual sensitivity is dependent on spatial frequencies, being most sensitive at frequencies around 2 to 4 cycles per degree. Third, complex images contain visual patterns at different frequencies which nonlinearly mix to influence our perception [10]. These factors combine into a perceptual physical error metric, $M(x,y,t)$, that predicts for a given input frame the amount each pixel can change in physical units (cd/m^2) without being noticed. We pre-compute $M(x,y,t)$ over the entire video sequence, using $B = B_{min}$, $W = W_{max}$ as a proxy for the final luminance, and we use a viewer position of 3 feet away from the display. Figure 1e demonstrates the perceptual physical error metric for the train sequence. Overall, the error metric will tolerate more error on edges and in bright areas. (Note: the perceptual LAB color space could be used, but by itself would not incorporate spatial sensitivities.)

SPLITTING STATIC AND DYNAMIC

We present two algorithms for splitting into a printed and projected component: an optimal solution based on a constrained optimization, and a heuristic method that leverages approximations to generate results in near real-time. Overall, we would like to develop methods that globally minimize the visual errors through time of the display, while maximizing the contrast ratio and maintaining a smooth black/white level.

Constrained Optimization

We can formulate our problem as a constrained optimization, similar to the approach taken in [2]. We minimize an error function E over the entire video sequence, solving for $B(x,y)$, $W(x,y)$ and $K(x,y)$, given the constraints $K_b \leq K(x,y) \leq K_w$, $B(x,y) \geq B_{min}(x,y)$, and $W(x,y) \leq W_{max}(x,y)$. Our error function E consists of seven terms, each with a corresponding weighting constant α_i.

$$E = \sum_t \sum_{xy} (\alpha_0 f_0 + \alpha_1 f_1) + \sum_{xy} \sum_{i=2}^{6} \alpha_i f_i \tag{6}$$

The first term f_0, accounts for the total error through time in reproducing our target image, $(L_g(x,y,t) - L(x,y,t))^2$. This can be re-written as a piecewise function, with errors when the projector clips at its maximal/minimal values.

$$f_0 = \begin{cases} (L_g(x,y,t) - K(x,y)D_w(x,y))^2, & (i) \\ (L_g(x,y,t) - K(x,y)D_b(x,y))^2, & (ii) \\ 0, \ otherwise \end{cases}$$
$$\text{where } (i)\ L_g(x,y,t)/K(x,y) > D_w(x,y), \quad (ii)\ L_g(x,y,t)/K(x,y) < D_b(x,y) \tag{7}$$

The above term measures error in physical units, but we would like to minimize perceptual error. We use the perceptual error metric $M(x,y,t)$ in two ways. First, we can increase the tolerance for an acceptable solution (similar to [11]). Secondly, we divide the absolute error by this perceptual error threshold to get an approximate perceptual error.

$$f_0 = \begin{cases} \dfrac{(L_g(x,y,t) - M(x,y,t) - K(x,y)D_w(x,y))^2}{M(x,y,t)^2}, & (i) \\ \dfrac{(L_g(x,y,t) + M(x,y,t) - K(x,y)D_b(x,y))^2}{M(x,y,t)^2}, & (ii) \\ 0, \ otherwise \end{cases}$$
$$\text{where } (i)\ (L_g(x,y,t) - M(x,y,t))/K(x,y) > D_w(x,y), \quad (ii)\ (L_g(x,y,t) - M(x,y,t))/K(x,y) < D_b(x,y) \tag{8}$$

This leaves the printed image underconstrained. In these situations, we would like to place equal amounts of information into the printer and projector [6, 12, 25]. We add a term that encourages the normalized printed s image and normalized projected d images to be equal.

$$f_1 = \left(s(x,y) - d(x,y,t)\right)^2 \tag{9}$$

We want the white/black levels to be smooth so the changes are not noticeable. Similar to [2], we form two terms that minimize the sum of the gradients of B and W, $f_2 = (\nabla B(x,y))^2$, and $f_3 = (\nabla W(x,y))^2$. Because we are optimizing our display for a particular video sequence, we can smooth the gradients further. Using the perceptual error metric $M(x,y,t)$, we can tolerate a faster change in our black/white level in regions where the error tolerance is higher, and a slower change where the error tolerance is lower. For instance, in Figure 1f the white level of the train sequence varies greatly, and changes most rapidly in bright regions and on edges where the changes will be least noticeable. We divide our previous terms by the minimum error tolerance through time.

$$f_2 = \frac{(\nabla B(x,y))^2}{\min_t(M(x,y,t))^2}, \quad f_3 = \frac{(\nabla W(x,y))^2}{\min_t(M(x,y,t))^2} \tag{10}$$

To ensure we use the maximum contrast ratio possible, we want the white level and black level to be as close as possible to their ideal values. The content author can prefer brighter content or prefer darker content by adjusting the α_4 and α_5 parameters, which weight f_4 and f_5.

$$f_4 = (B - B_{min})^2, \quad f_5 = (W - W_{max})^2 \tag{11}$$

Until this point we have assumed exact alignment between projected pixels and printed 'pixels'. In fact projector pixels are not rectangles but small ellipsis of light [1], and printers do not have pixels at all, but use overlapping dots of ink of different colors. It is possible to achieve sub-pixel alignment between projected content and a display surface [1], however in practice it is extremely difficult and any slight movement (someone walking by) could cause misalignment. In this work, we do not assume precise geometric alignment, so we add an additional term that slightly smooths the printed image to ensure mis-registrations will not yield visual artifacts, $f_6 = (\nabla s(x,y))^2$. In practice, we do not need to smooth the entire printed image. Tight spatial registration is most crucial when compensation is necessary, i.e. when the projector and printer have drastically different values. We create a masking term ω which is large for pixels with high temporal dynamic range and with a small error tolerance.

$$f_6 = \omega(\nabla s(x,y))^2, \quad \text{where } \omega = \frac{(I_{max}(x,y,t) - I_{min}(x,y,t))^2}{\min_t(M(x,y,t))^2} \tag{12}$$

We solve this constrained optimization problem in Matlab using a projected quasi-Newton method, minConf_PQN[1]. We tried other constrained solvers, and found the best performance with minConf_PQN. On the Bunny sequence at [270, 480] with 150 frames, when initialized with random input it took 3 hours, 44 minutes to converge (on a laptop with an i7 2.3GHz processor). Performance could be improved with multi-grid solvers [13] and using native code, however,

[1]http://www.cs.ubc.ca/ schmidtm/Software/minConf.html

simply evaluating the function once can take seconds to loop through every pixel and frame.

Heuristic Method

The above approach yields an optimal solution given the seven input parameters, α_i. Due to the computational complexity, this approach does not allow the content author to explore the parameter space. We would like a method which can be evaluated fast enough for the content author to drive the parameters in real-time. In our heuristic method, we split the problem into two main parts, solving for the black/white level and solving for the printed image.

Solving for the White/Black Level

To ensure there are no visual artifacts, we can reduce the contrast of the desired image until there are no errors, by lowering the white level and/or raising the black level of the display. To simplify the problem, let us start by fixing the black level at its maximal value $W(x, y) = W_{max}(x, y)$, and solve for the black level of the display. At pixels that would result in visual artifacts, Equation 5 become an equality, and we can solve for $B(x, y)$, and at pixels with no visual artifacts we can set $B(x, y) = B_{min}(x, y)$.

This results in a non-smooth black level. Smoothing with a simple Gaussian filter would yield pixels that violate the constraints [13], so we borrow a technique from Mujumder et al. [22]. We solve a constrained gradient blurring problem, finding a new black level $B'(x, y)$ such that it is minimized across the image subject to the constraints that $B'(x, y) \geq B(x, y)$ and $\|\nabla B'(x, y)\| \leq B'(x, y)/\delta_0$, where δ_0 is a constant. This can be solved using dynamic programming in real-time. However, the approach from [22] results in a very slowly changing black/white level, because the approach is not content dependent. We are optimizing our display for a single video sequence, so we can have a faster changing white/black level by hiding the changes in places where it will be least noticeable (where M is the highest). Similar to Equation 8, we modify the heuristic approach to use the perceptual physical error metric, where δ_0, δ_1 are constants.

$$\|\nabla B'(x, y)\| \leq \frac{B'(x, y)}{\delta_0} + \delta_1 min_t(M(x, y, t)), \qquad (13)$$

To control whether we adjust the black level or white level, we expose a single parameter to the content author which we call *prefer black/white*, $\rho_{bw} = [0, 1]$. We first solve for the adjusted B' via the process described above. If we prefer white ($\rho_{bw} = 1$) then we want to use the adjusted black level B', however if we prefer black ($\rho_{bw} = 0$) then we want to use the original minimal black level B_{min} and lower the white level instead. Thus, our adjusted black level B'' is

$$B''(x, y) = B_{min}(x, y)(1 - \rho_{bw}) + B'(x, y)(\rho_{bw}). \qquad (14)$$

We then solve for the white level given B'', and smooth it using the constrained gradient blur. The end result is a smooth black level and smooth white level that are guaranteed to not introduce any visual artifacts.

Allow Artifacts

Not allowing artifacts can actually be a bad thing. The offline constrained optimization solution balances visual artifacts and contrast ratio through the α_i parameters. However, the heuristic method described above will ensure zero artifacts, maximally clipping the display contrast. Thus, we provide the content author with an additional parameter which we call *allow artifacts*, $\rho_{aa} = [0, 1]$. Similar to ρ_{bw}, this parameter is a simple linear interpolation between the naive maximal contrast black/white levels and the conservative black/white levels that introduce no artifacts. We take the black level above and solve for an adjusted black level B''', and repeat a similar process for the white level.

$$B'''(x, y) = B''(x, y)(1 - \rho_{aa}) + B_{min}(x, y)(\rho_{aa}) \qquad (15)$$

Solving for the Printed Image

Now we need to solve for the printed image given the black/white level. For static pixels with no motion, we would like the normalized printed and projected images to be identical. We can solve the equation below for the printed image in static regions $K_s(x, y)$ by substituting $D(x, y, t) = L_g(x, y)/K_s(x, y)$.

$$\frac{K_s(x, y) - K_b}{K_w - K_b} = \frac{D(x, y, t) - D_b(x, y)}{D_w(x, y) - D_b(x, y)} \qquad (16)$$

For dynamic pixels, to allow zero artifacts the printed image will be constrained by the brightest desired image, and the darkest desired image. The darkest the printed image can be is determined by the brightest desired image, divided by the brightest the projector can display, $K_{min}(x, y) = max_t(L_g(x, y, t))/D_w(x, y)$, and the brightest the printed image can be is determined by the darkest desired image divided by the darkest the projector can display, $K_{max}(x, y) = min_t(L_g(x, y, t))/D_b(x, y)$. Thus we have the constraints $K_{min}(x, y) \leq K(x, y) \leq K_{max}(x, y)$. If $\rho_{aa} = 0$, then we should have already adjusted our black/white levels such that there can be no artifacts, and we can just ensure the above constraints are met. However, if $\rho_{aa} > 0$, then we may have artifacts and we need to choose whether to introduce artifacts in darker or brighter content.

The constrained optimization solution chooses a printed image that minimizes errors over time, and the above constraints do not incorporate time. For pixels that are mostly bright over time, we want to introduce artifacts into the darker frames for these pixels. And for pixels that are mostly dark over time, we want to introduce artifacts into the brighter frames for these pixels. To accomplish this we use the mean $I_{mean} = mean_t(I(x, y, t))$ to interpolate between K_{min} and K_{max},

$$K = K_{max}(1 - I_{mean}{}^{\gamma}) + K_{min}(I_{mean}{}^{\gamma}). \qquad (17)$$

Due to the multiplicative nature of light and surface color, we want to error on the side of a brighter surface color. The one exception to this is if we have a very dark scene (like Figure 5), as we don't want to introduce visually significant artifacts into the background throughout the whole sequence. Therefore, we apply a $\gamma = 0.1$, to give preference to a brighter printed image except at primarily dark pixels.

Authoring Interface

To enable the content author to guide the solution in real-time, we created a GUI with sliders for each parameter of

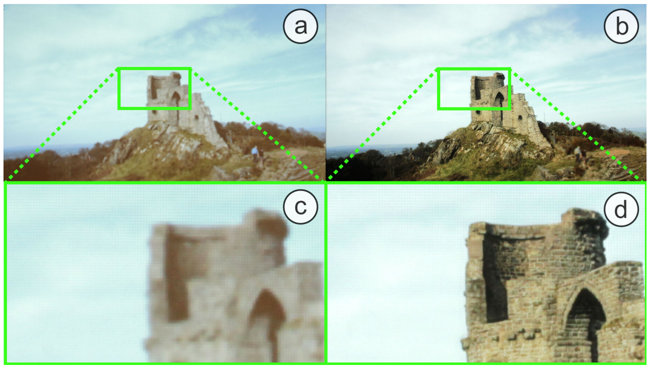

Figure 6. Projectibles have much higher resolution in areas where fine details are consistent throughout the video sequence. Despite the lighting changes, the high res details in the rocks are consistent in the sequence. (a) Projector only, (b) Projectible. (c-d) Close ups.

the heuristic algorithm. We also created a series of visualizations in two sets of tabs that the content author can view side by side while editing the Projectible. The content author can scrub through the video sequence via a time-line and play/pause controls. In the tabs, the content author can view the source content, the target image, the simulated Projectible result, the printed image, the projected image, the black level, the white level, and the perceptual physical error tolerance. These results can be viewed through a variety of tone mapping operators[2].

Temporal Adaption

So far we have not incorporated a perceptual model of how our visual system responds to input over time. This can lead to excessive clipping. For instance, in the Castle time-lapse sequence there are visitors walking around the castle, which are only visible in a single frame each. We don't want to reduce the contrast of the scene to properly display them for such a short period of time. We use an exponential decay function [11] as our temporal adaption model, which decreases the perceptual error for ephemeral content and is based on the temporal response for photopic vision. (This model could be extended using [9]). We pre-process the input video sequence before running the constrained optimization or heuristic methods.

Up-Res Printed

Until now we have assumed that the projector and printer are equal resolution. However, printers are much higher resolution than projectors, especially for larger format displays. The ideal approach would incorporate the respective resolutions (and pixel modeling) into the splitting algorithms, solving for a solution at the printer's resolution. As the printer can be easily >10X higher resolution than the projector, this would impractically increase the running time. We compute the splitting at the projector's resolution, then up-res the answer where possible to the printer's resolution.

As the resolution improvement is only from the printed image, we can only increase the resolution of the separation if the high frequency information at the printer's resolution remains constant throughout the video sequence. For instance, in the Castle sequence (Figure 6), the colors of the stone change significantly due to passing clouds, but the high resolution details on the stone remain the same, so we can increase the resolution. To separate out the high contrast details, we run a Gaussian high pass filter over each frame of the video sequence and compute the temporal mean and standard deviation of this high pass signal. We can up-res the printed image at the lower projector resolution K^d, to the higher printed resolution K^s by transferring high resolution details from the temporal mean image $L^s_{mean} = mean_t(L^s_g(x, y, t))$, in areas β_h where the temporal standard deviation of the Gaussian high pass filter is low.

$$K^s = (1 - \beta_h)\, K^d + (\beta_h)\, K^d \frac{L^s_{mean}}{G(L^s_{mean})} \tag{18}$$

AUTHORING CONTENT

In order to make use of the resolution and contrast improvements that Projectibles can provide, the source content should ideally be high resolution (4K or greater) and high dynamic range (12 bits per channel or greater). To acquire such content, the simplest option is to use a 4K HDR camera, however, these cameras are expensive and not ubiquitous yet.

Time-lapse photography is frequently used to created ultra-high resolution video from standard DSLR cameras. Time-lapse videos composite full resolution RAW images taken at regular intervals into a video sequence. For instance, the Svartifoss waterfall sequence is a [7360, 4140], 16 bit TIFF image sequence. Another content source for Projectibles are virtual animated sequences which can be easily rendered out to high resolution, HDR images (e.g. the Bunny sequence). Standard DLSR video cameras can also shoot HDR video using exposure bracketing with custom firmware[3] (e.g. the Train sequence), however only at standard resolutions $\leq$ [1920, 1080]. To create high resolution Projectibles with standard DSLR cameras, we can composite a high resolution still image with lower resolution video. For instance, in the Whiskey sequence, we shot 1080p video of the sequence, and captured a 15 megapixel still image, then composited them together in Adobe After Effects.

RESULTS

We demonstrate real world results for a variety of video sequences, with empirical measurements of the display's contrast ratios and subjective viewer feedback. All the results in this paper were captured in a light controlled environment, with high ambient light to better demonstrate the splitting algorithm and the robustness of the methods used. The ambient light was a $60W$ compact fluorescent bulb which resulted in a reddish white level of [20.72, 18.80, 6.15] in CIE XYZ.

We used low cost commodity hardware. The projector (InFocus IN126ST) cost \$719 and had [1280, 800] resolution, and we tiled 2 Projectibles vertically resulting in [480, 270] pixels each. The projector was 3000 lumens with a bluish white level [405.0, 447.8, 563.1], without ambient light the dynamic range was [1.9, 447.7] with a contrast ratio of 232:1, with ambient light the dynamic range was [20.7, 466.6] with a contrast ratio of 29:1. The printer (Epson R2000) cost \$499, has

[2] http://advancedhdrbook.com

[3] Magic Lantern (http://www.magiclantern.fm)

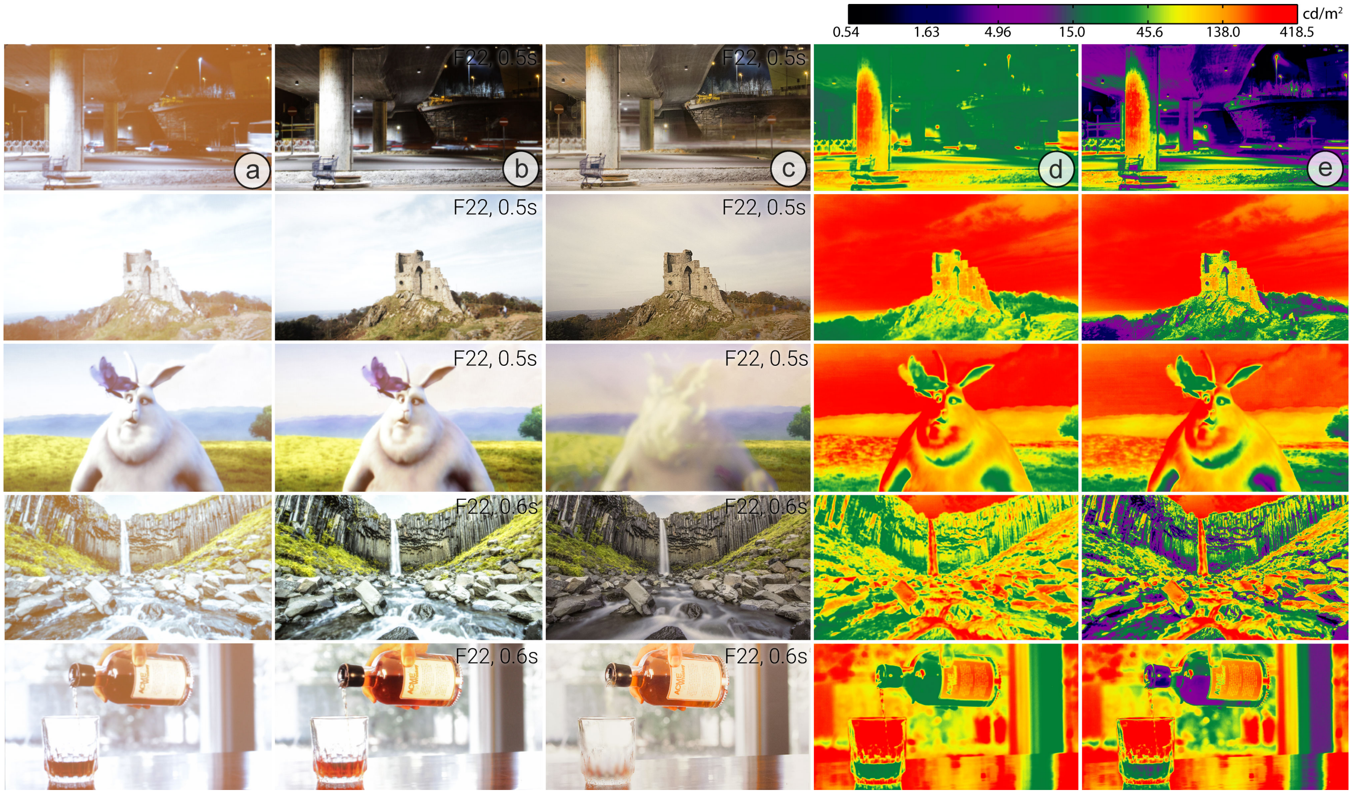

Figure 7. (a, b) Captured images ($F22, 0.5s, 800\ ISO$) of the (a) projector only and (b) combined Projectible. (Zoom in to see the resolution differences.) (c) Printed image. (d, e) Log plots of the luminance in cd/m^2 captured with an HDR image: (d) projector only, (e) combined Projectible.

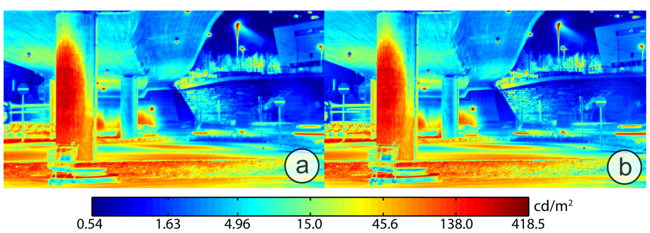

Figure 8. Verification of our lighting model and calibration process. Log plots of luminance values in cd/m^2 of (a) real world result captured with an HDR photograph of the Projectibles display and (b) the simulated result from our lighting model.

a DPI of [5760, 1440], and when used with matte paper (Epson Ultra Premium Presentation Paper Matte) the dynamic range was [0.026, 0.897] with a 34:1 contrast ratio. Under ambient light, the ideal combined display would have a dynamic range of [0.58, 418.5], with a contrast ratio of 721:1. Using more expensive projectors and printers, with no ambient light, can drastically increase the combined contrast [6].

In order to measure our example results we captured calibrated exposure bracketed HDR image sequences. In Figures 1 and 7, we show log plots of the real-world measured luminance values for the projector only image on white paper, compared to the combined Projectible display. These results demonstrate the significant contrast and resolution enhancement that Projectibles can provide. We also evaluate our calibration and lighting model, Figure 8 compares the predicted combined display $L(x, y, t)$ with the empirical real-world results captured via an HDR image.

Figure 9. The constrained optimization (left) and heuristic solution (right) yield similar solutions. (a) printed image, (b) white level, (c) black level, and (d) real world image of combined Projectible.

Similar to other image editing applications (e.g. in Photoshop), the processing time is highly dependent on the resolution of the projector/printer. As an example, for the Bunny sequence at [480, 270] projector resolution, calculating the black/white level via the heuristic method takes 0.98 seconds, and calculating the printed image (at projector resolution) takes 0.44 seconds. While not real-time, the content author can easily tune the parameters to achieve an ideal result. For the constrained optimization method, solving for the same resolution from random input took 3 hours, 44 minutes.

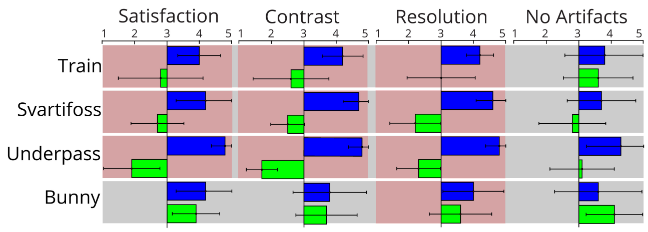

Figure 10. Results of Likert data from a user study (N=10) for four sequences (Train, Svartifoss, Underpass, Bunny), (blue) Projectible, (green) Projector only. Viewers clearly preferred the Projectible display for overall satisfaction, resolution and brightness. Viewers were asked to rate visual artifacts (if something didn't look right), and the results show that viewers couldn't clearly differentiate artifacts. (Results with a pink background were statistically significant $p < 0.05$ via t-tests).

It is not possible to empirically compare the results between the constrained optimization and heuristic solutions as the mathematical formulations are different, so the two methods *should not* converge to the same answer. However, Figure 9 shows that after manually tuning the parameters on each method, the results can be quite similar. Future work, could explore the impact of each parameter in a leave one out design.

Viewer Feedback

We also gathered feedback from 10 participants comparing a heuristic Projectible display to a traditional projected display on white paper (the control). The displays were stacked vertically and viewed simultaneously by participants for four example scenes (Train, Svartifoss, Underpass, Bunny). We divided the participants into two groups, and randomized the position of the displays along with the ordering of the scenes for each group. Users were instructed to view the displays from multiple positions and angles, and then rate on a Likert scale (1-5) their overall satisfaction, and judge the contrast, resolution and visual artifacts for each scene.

Figure 10 shows the results from the user study, demonstrating that the ratings for overall satisfaction, contrast and resolution were much higher for Projectibles. Via t-tests, the results for overall satisfaction, contrast and resolution were statistically significant for the Train, Svartifoss and Underpass sequences, but not for the Bunny sequence. Due to the motion of the dark butterfly and the white bunny across the scene in the Bunny sequence, the Projectible process has a lower contrast boost than the other examples. Viewers were also asked to rate the visual artifacts in the display (if something didn't look right). The results were only statistically significant for the Underpass sequence, and viewers rated the Projectible as having less artifacts. Thus, overall visual artifacts were not clearly noticeable and didn't effect overall satisfaction rates.

DISCUSSION

In this paper, we focus on the core algorithms for splitting printed and projected. Extensions to the approach could enable all kinds of unique visual experiences (see Figure 11). We could optimize the surface color of 3D objects, either constructed from paper cutouts, multi-material 3D printers, or industrial manufacturing processes. We could extend our lighting model to support transparency printing for rear projection surfaces (e.g. for store-front windows), or for other non-matte printed surfaces. We can easily create much larger Projectibles for truly immersive experiences, e.g. Figure 11d shows a large 5x3ft 3D version of the Svartifoss waterfall.

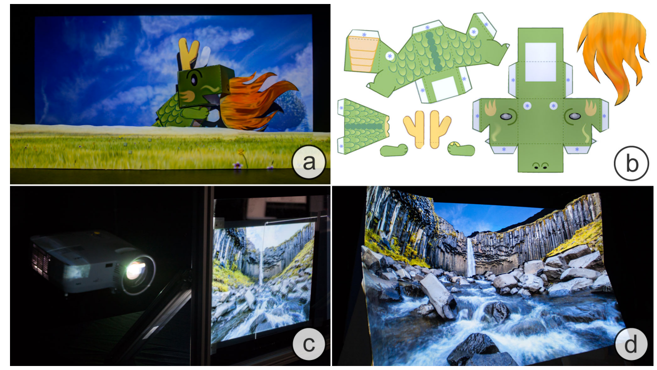

Figure 11. Projectibles could be extended in a variety of ways including: (a,b) optimizing the surface of animated 3D objects using paper cut-outs, (c) using rear projection surfaces, or (d) large scale 3D projection mapping installations (shown here at 5x3 ft).

Overall, Projectibles work well for looped video sequences with a combination of static and dynamic elements. For sequences that are mostly static (e.g. Whiskey sequence), Projectibles can achieve ideal contrast ratios and significant resolution improvements. For sequences that have significant motion (e.g. Train sequence), resolution improvements are impossible but contrast improvements can still be achieved. The worst case scenario is a video sequence that goes from uniform white to uniform black, in which case the printed component cannot improve resolution or contrast. Thus, Projectibles work best with looped video content, and we showcase a range of contrast and resolution improvements on six different sequences.

In environments with fixed high ambient light, normal projected displays on a white display surface are too washed out for practical use. Projectibles could be used in these locations to create projected displays that are non-rectangular or non-flat with greatly improved contrast ratios and resolution. In environments with low ambient light, Projectibles are capable of achieving HDR display contrast ratios beyond those of normal displays [6]. However, environments with variable lighting like skylights and windows don't work with the existing implementation. Projectibles could be extended with a real-time color calibrated camera that could adjust the projected image in real-time, but the printed image would still have to be optimized for a certain range of ambient light.

Another limitation is that the perceptual physical error tolerance is based on a fixed viewer location. In the user study, the viewers were asked to view the display from a variety of distance and angles, and did not report noticing any differences in perceived appearance. Furthermore, through the authoring interface, we provide the content author with an accurate simulation of the combined result, which can be viewed from multiple angles and distances. To commercially deploy Projectibles, the system would need an automated geometric registration process (e.g. [6]). If precise sub millimeter physical alignment of the printed image with the projected image were possible, sub-pixel super-resolution could also be achieved [1], and the printed image could compensate for the size and shape of each projector pixel. Also, for use in medical imaging, the JNDs of the display could be optimized [6].

CONCLUSION

Compared to a traditional display, Projectibles can be higher resolution, higher contrast (in low ambient light), and can be made into large format, non-rectangular and even non-flat displays. This work is the first to combine a static display with a dynamic display for showing high dynamic range, high resolution video. It is also the first to optimize static surface color for the projection of video. We presented an offline constrained optimization method, and a near real-time heuristic method for splitting a dynamic video into a printed and projected component. We showed real world results for six example sequences, demonstrating Projectibles' significant contrast and resolution improvements, and presented viewer feedback confirming these results. For advertising and entertainment applications, Projectibles combine the benefits of printed signage with traditional digital displays, to create high resolution, high dynamic range viewing experiences.

ACKNOWLEDGEMENTS

We would like to thank Stian Rekdal (http://stianrekdal.com) for the Svartifoss waterfall sequence, Dan at NatureClip (http://natureclip.co.nr) for the Castle sequence, and Eric Solheim (http://eirikso.com) for the Underpass sequence.

REFERENCES

1. Aliaga, D. G., Yeung, Y. Y. H., Law, A., Sajadi, B., and Majumder, A. Fast high-resolution appearance editing using superimposed projections. *ACM Transactions on Graphics 31*, 2 (Apr. 2012), 1–13.
2. Ashdown, M., Okabe, T., Sato, I., and Sato, Y. Robust Content-Dependent Photometric Projector Compensation. In *IEEE CVPR* (2006).
3. Batlle, J., Mouaddib, E., and Salvi, J. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern recognition 31*, 7 (1998), 963–982.
4. Bimber, O., Coriand, F., Kleppe, A., Bruns, E., Zollmann, S., and Langlotz, T. Superimposing pictorial artwork with projected imagery. *IEEE Multimedia 12*, 1 (2005), 16–26.
5. Bimber, O., Emmerling, A., and Klemmer, T. Embedded Entertainment with Smart Projectors. *IEEE Computer 38*, 1 (Jan. 2005), 16–26.
6. Bimber, O., and Iwai, D. Superimposing dynamic range. *ACM Transactions on Graphics 27*, 5 (Dec. 2008), 1.
7. Bimber, O., Iwai, D., Wetzstein, G., and Grundhöfer, A. The visual computing of projector-camera systems. *Computer Graphics Forum 27* (2008), 2219–2245.
8. Chen, X., Yang, X., Xiao, S., and Li, M. A practical radiometric compensation method for projector-based augmentation. In *IEEE ISMAR* (2008).
9. Didyk, P., Eisemann, E., Ritschel, T., Myszkowski, K., and Seidel, H.-P. Apparent display resolution enhancement for moving images. In *ACM Transactions on Graphics (TOG)*, vol. 29, ACM (2010), 113.
10. Ferwerda, J. a., Shirley, P., Pattanaik, S. N., and Greenberg, D. P. A model of visual masking for computer graphics. In *ACM SIGGRAPH* (1997).
11. Grundh, A., Grundhöfer, A., and Bimber, O. Real-time adaptive radiometric compensation. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 97–108.
12. Guarnieri, G., Albani, L., and Ramponi, G. Image-splitting techniques for a dual-layer high dynamic range LCD display. *Journal of Electronic Imaging 17*, 4 (Oct. 2008).
13. Guarnieri, G., Albani, L., and Ramponi, G. Minimum-Error Splitting Algorithm for a Dual Layer LCD Display. *Journal of Display Technology 4*, 4 (2008), 383–390.
14. Hirsch, M., Wetzstein, G., and Raskar, R. A compressive light field projection system. *ACM Transactions on Graphics (TOG) 33*, 4 (2014).
15. Iwai, D., Takeda, S., Hino, N., and Sato, K. Projection screen reflectance control for high contrast display using photochromic compounds and uv leds. *Optics express 22*, 11 (2014).
16. Joshi, N., Mehta, S., Drucker, S., Stollnitz, E., Hoppe, H., Uyttendaele, M., and Cohen, M. Cliplets: juxtaposing still and dynamic imagery. In *ACM UIST* (2012).
17. Juang, R., and Majumder, A. Photometric Self-Calibration of a Projector-Camera System. In *IEEE CVPR* (2007).
18. Kinjo, T., Saito, N., and Omodani, M. Vivid image projection system using e-paper active screen. *Journal of the Society for Information Display 20*, 10 (2012), 559–565.
19. Lanman, D., Heide, F., Reddy, D., Kautz, J., Pulli, K., and Luebke, D. Cascaded displays: spatiotemporal superresolution using offset pixel layers. In *ACM SIGGRAPH 2014 Emerging Technologies*, ACM (2014).
20. Law, A. J., Aliaga, D. G., Sajadi, B., Majumder, A., and Pizlo, Z. Perceptually Based Appearance Modification for Compliant Appearance Editing. *Computer Graphics Forum 30*, 8 (Dec. 2011), 2288–2300.
21. Liao, Z., Joshi, N., and Hoppe, H. Automated video looping with progressive dynamism. *ACM Transactions on Graphics 32*, 4 (July 2013), 1.
22. Majumder, A., and Stevens, R. Perceptual photometric seamlessness in projection-based tiled displays. *ACM Transactions on Graphics 24*, 1 (Jan. 2005), 118–139.
23. Nayar, S. K., Peri, H., Grossberg, M. D., and Belhumeur, P. N. A projection system with radiometric compensation for screen imperfections. In *ICCV PROCAMS* (2003).
24. Ramasubramanian, M., Pattanaik, S., and Greenberg, D. A perceptually based physical error metric for realistic image synthesis. In *ACM SIGGRAPH* (1999).
25. Seetzen, H., Heidrich, W., Stuerzlinger, W., Ward, G., Whitehead, L., Trentacoste, M., Ghosh, A., and Vorozcovs, A. High dynamic range display systems. *ACM Transactions on Graphics 23*, 3 (Aug. 2004), 760.
26. Seetzen, H., Whitehead, L., and Ward, G. A High Dynamic Range Display Using Low and High Resolution Modulators. In *SID* (2003), 1450–1453.
27. Sheng, Y., Yapo, T. C., and Cutler, B. Global Illumination Compensation for Spatially Augmented Reality. *Computer Graphics Forum 29*, 2 (May 2010), 387–396.
28. Trentacoste, M., Heidrich, W., Whitehead, L., Seetzen, H., and Ward, G. Photometric image processing for high dynamic range displays. *Journal of Visual Communication and Image Representation 18*, 5 (Oct. 2007), 439–451.
29. Wang, D., Sato, I., Okabe, T., and Sato, Y. Radiometric Compensation in a Projector-Camera System Based Properties of Human Vision System. In *IEEE CVPR Workshops (CVPRW)* (2005).
30. Wetzstein, G. Radiometric Compensation through Inverse Light Transport University of British Columbia. In *Environment* (2007).