

A Unified Paradigm for Scalable Multi-Projector Displays

Niranjan Damera-Venkata, *Member, IEEE*, Nelson L. Chang, *Member, IEEE*, and Jeffrey M. DiCarlo, *Member, IEEE*

Abstract—We present a general framework for the modeling and optimization of scalable multi-projector displays. Based on this framework, we derive algorithms that can robustly optimize the visual quality of an arbitrary combination of projectors without manual adjustment. When the projectors are tiled, we show that our framework automatically produces blending maps that outperform state-of-the-art projector blending methods. When all the projectors are superimposed, the framework can produce high-resolution images beyond the Nyquist resolution limits of component projectors. When a combination of tiled and superimposed projectors are deployed, the same framework harnesses the best features of both tiled and superimposed multi-projector projection paradigms. The framework creates for the first time a new unified paradigm that is agnostic to a particular configuration of projectors yet robustly optimizes for the brightness, contrast, and resolution of that configuration. In addition, we demonstrate that our algorithms support high resolution video at real-time interactive frame rates achieved on commodity graphics platforms. This work allows for inexpensive, compelling, flexible, and robust large scale visualization systems to be built and deployed very efficiently.

Index Terms—Multi-projector displays, tiled displays, large format displays, blending, stitching, automatic geometric alignment, photometric correction, super-resolution, superimposed projection.



1 INTRODUCTION

Large format displays (e.g. greater than 100" diagonal) create a compelling environment for the visualization of life-size design, immersive simulation, and multi-modal interaction. However, today's display technologies are prohibitively expensive to scale for large visualization applications. An approach that has gained popularity is to combine multiple projectors into a scalable display for visualization.

Previous attempts at seamless "scalable" projector displays have been confined to edge-blended tiled projectors, where multiple projectors are configured to minimize the inter-projector overlap and maximize the overall size of the displayed image [20, 18, 16, 1, 9]. These tiled approaches offer scalability of resolution and the ability to have arbitrary aspect ratios. However, tiled projection does not scale well along other image quality attributes. For instance, luminance discontinuities at the overlap regions prevent full brightness utilization. Here, the performance of a tiled array is limited by the quality of the worst projector in the array. If a single projector in a tiled array drifts from color or luminance calibration, or worse yet fails, the entire projector array needs to be changed to accommodate that projector or else the projector needs to be replaced. The dilemma that faces large scalable displays is how to enhance all image quality attributes and yet ensure a reliable system that is not limited by the performance or quality of the worst projector.

In contrast with tiled projection, superimposed projection is a paradigm where the goal is to maximize projector overlap instead of minimizing it. Superimposed projection has been used in event projection to maximize brightness while also offering robustness to projector failure via redundancy. Superimposed projection often results in a fundamentally more reliable system that can better compensate for common intra- and inter-projector variations (e.g. temporal variations such as color and luminance drift) that plague conventional tiled display walls. At first glance however, the benefits of superimposition seem severely mitigated by an apparent loss of resolution due to the superimposition [14]. We previously showed both in theory [5, 6] and in practice [7] that superimposed projection can significantly enhance resolution beyond the Nyquist limit of any component projector. We showed that these resolution gains are achievable if we generate the

component projector images intelligently. However unlike tiled projection, there are fundamental limits to the resolution gain using superimposed projection alone [6].

The preceding discussion motivates the need for a rigorous framework to support projector configurations that have the benefits of both tiled and superimposed projection. Likewise, there is a need for new algorithms that are capable of robustly optimizing the image quality of an arbitrary combination of projectors without any manual tweaking of blend maps or alignment. Furthermore, the rendering algorithms must also be suitable for video rates on today's high resolution (HD or better) content.

In this paper, we present a unified paradigm to automatically and efficiently deliver the best image quality for arbitrary multi-projector configurations. The paradigm leads to a multi-projector system that is

- *Flexible*: The paradigm can directly handle arbitrary projector configurations including tiled, superimposed, and even novel hybrid configurations;
- *Scalable*: The paradigm scales not only in the number of projectors, but also along a variety of dimensions (e.g. brightness, resolution, aspect ratio, fault tolerance);
- *Automatic*: The proposed camera-based calibration, estimation, and modeling techniques are fully automated;
- *Real-time*: The algorithms can be efficiently implemented on commodity GPUs to create practical real-time display systems;
- *Vivid*: The paradigm delivers seamless and superb image quality regardless of the multi-projector configuration, even delivering super-resolution where multiple projectors overlap.

Previous work in multi-projector displays has focused separately on only tiled configurations or only superimposed configurations. Techniques appropriate for one configuration will produce sub-optimal results when used for the other. In contrast, this paper proposes a unified paradigm that, for the first time, delivers high quality results from *any* multi-projector configuration. Examples of prototype multi-projector systems are shown in Figure 1.

The rest of the paper is outlined as follows. Section 2 describes the model to accurately account for geometry, color, resolution, and spatial luminance characteristics of the entire multi-projector system. Section 3 maps the input image into a target space that is a subspace of the physically realizable outputs. Section 4 derives an efficient rendering algorithm that generates the projector images (a.k.a. subframes) in real-time, such that the overall system is capable of accurately and robustly creating a desired target image. Section 5 shows examples of the framework in action for different projector configurations and demonstrates its advantages over the prior state of the art. Finally,

• The authors are all with Hewlett-Packard Laboratories in Palo Alto, CA. E-mail: niranjan.damera-venkata@hp.com, nelson.chang@hp.com, jeffrey.dicarlo@hp.com. URL: <http://hpl.hp.com/research/pluribus>

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007. Published 14 September 2007.

For information on obtaining reprints of this article, please send e-mail to: tcvg@computer.org.

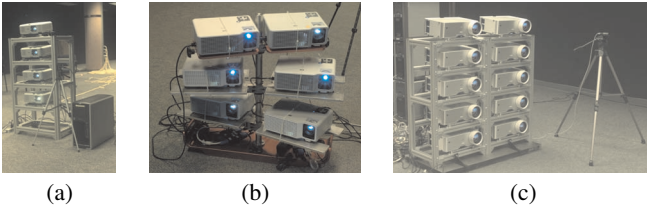


Fig. 1. Prototype multi-projector displays featuring (a) four, (b) six, and (c) ten projectors.

Section 6 concludes the paper by summarizing the key contributions and motivating future work.

2 MODELING MULTI-PROJECTOR SYSTEMS

2.1 Notation

In this paper, we choose to model images as stacked vectors instead of 2-D arrays. This allows linear operations on images to be represented compactly with matrix multiplications. Matrices and vectors are represented with boldface font, and matrices are capitalized. This approach is convenient in deriving and expressing computational algorithms. Consider a three-color (RGB) digital image $y(n_1, n_2)$ of size $N_1 \times N_2$. We arrange the R, G, and B components of each pixel in row scan order into a vector to obtain the $3N_1N_2 \times 1$ vector \mathbf{y} . Linear operations on this image can then be represented by matrix multiplication with $3N_1N_2 \times 3N_1N_2$ matrices.

Let us consider a couple of image processing operations that will be important in this paper. First, a 3×3 color transformation $\tilde{\mathbf{C}}$ applied to each pixel in the image may be represented as

$$\mathbf{C}\mathbf{y} = \begin{pmatrix} \tilde{\mathbf{C}} & & & \\ & \tilde{\mathbf{C}} & & \\ & & \ddots & \\ & & & \tilde{\mathbf{C}} \end{pmatrix} \mathbf{y} = (\mathbf{I} \otimes \tilde{\mathbf{C}}) \mathbf{y}$$

where \otimes represents the Kronecker product operator [10] and \mathbf{I} is the $N_1N_2 \times N_1N_2$ identity matrix.

Second, consider operations that are performed on a color-plane independent basis. For example, a spatial filtering operation using the same filter to process the red, green, and blue color planes of an image, respectively. In this case, the operation may be expressed in matrix form by the equation

$$\mathbf{A}\mathbf{y} = \begin{pmatrix} \tilde{\mathbf{A}}(0,0)\mathbf{I} & \cdots & \tilde{\mathbf{A}}(0,N_1-1)\mathbf{I} \\ \tilde{\mathbf{A}}(1,0)\mathbf{I} & \cdots & \tilde{\mathbf{A}}(1,N_1-1)\mathbf{I} \\ & \ddots & \\ \tilde{\mathbf{A}}(N_1-1,0)\mathbf{I} & \cdots & \tilde{\mathbf{A}}(N_1-1,N_2-1)\mathbf{I} \end{pmatrix} \mathbf{y} = (\tilde{\mathbf{A}} \otimes \mathbf{I}) \mathbf{y}$$

where here \mathbf{I} is the 3×3 identity matrix. The $N_1N_2 \times N_1N_2$ matrix $\tilde{\mathbf{A}}$ represents the filtering operation on a single $N_1 \times N_2$ color plane. In the above examples and in the paper, $\tilde{}$ (tilde) is used to denote an elementary matrix transformation that generates a full matrix operator via the Kronecker product. We use the $\hat{}$ (hat) accent to denote a predicted matrix or vector.

In the interest of brevity, the dimension of the matrices and vectors are not explicitly indicated since they should be clear from context. The notation does not distinguish between operators operating on images of different sizes. For example, a color transform applied to an $N_1 \times N_2$ image \mathbf{y} is denoted by $\mathbf{C}\mathbf{y}$ while the same color transform applied to an $M_1 \times M_2$ image \mathbf{x} is also denoted by $\mathbf{C}\mathbf{x}$. Note that the dimensions of the matrix \mathbf{C} is different in each case.

2.2 Mathematical model of multi-projector systems

With the above notation in mind, we can proceed to model the image formation process of a multi-projector system. Because light from

multiple sources adds linearly, an estimate of the final projected image is given by the summation of the individual projector images.¹ The key then is to model the imaging process from input image to the frame buffer of each component projector; let the resolution of the frame buffer of a component projector be $N_1 \times N_2$.

We next choose a target display "canvas" with respect to which each projector image is computed. The target display canvas is assumed to correspond to the largest rectangular region of specified aspect ratio where light from at least $1 \leq \kappa \leq K$ projectors hits the display surface. The resolution of the target display canvas is the desired resolution at which the final image is computed; we assume it to be $M_1 \times M_2$, typically larger than any one projector.

Mathematically, the model for a K projector system may then be represented as:

$$\hat{\mathbf{x}} = \left(\sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{C}_k \mathbf{y}_k \right) + \mathbf{b} \quad (1)$$

where \mathbf{y}_k represents the $N_1 \times N_2$ input color image to projector k and $\hat{\mathbf{x}}$ represents the $M_1 \times M_2$ predicted final color image.

The matrix $\mathbf{C}_k = \mathbf{I} \otimes \tilde{\mathbf{C}}_k$ represents the k^{th} projector's color matrix that transforms the projector-dependent RGB color space into a common projector independent reference space (such as XYZ) or to the RGB space of a reference camera. As defined, the matrix \mathbf{C}_k allows color mixing to be modeled. Several previous approaches to projector tiling treated the red, green, and blue color planes independently [18, 16, 9], ignoring the fact that the same digital value of a primary color displayed by any individual projector potentially excites all of the R, G, and B camera sensors. Not accounting for this mixing results in color blotches and spatially varying color shifts when displaying uniform colors [19, 15, 13].

The diagonal matrix $\mathbf{L}_k = \tilde{\mathbf{L}}_k \otimes \mathbf{I}$ represents the k^{th} projector's spatial luminance rolloff due to lens vignetting in the projector-camera system. This models the fact that the camera image of a flat field of digital values projected by a projector exhibits spatially varying attenuation or vignetting. Each element along the diagonal scales the RGB pixel values at a given pixel location equally. However, the scaling across pixel locations is likely to be different.

The matrix $\mathbf{A}_k = \tilde{\mathbf{A}}_k \otimes \mathbf{I}$ represents the resampling operation from the k^{th} $N_1 \times N_2$ low-resolution projector frame buffer to a high resolution display canvas of size $M_1 \times M_2$. It encapsulates the geometric warps, pixel reconstruction point spread function, and resample filtering operations. Additional details on its derivation may be found in [7].

Finally, the vector \mathbf{b} in equation (1) represents the total black offset image of the system. It may be derived from the image the camera captures when all projectors are projecting images of all zeros due to light leakage from the projectors. This camera image, when warped and resampled to the reference display canvas coordinate system, gives the black offset \mathbf{b} of the overall system.

It should be clear in this exposition that we have not assumed a particular configuration of projectors (e.g. tiled, superimposed) and have instead treated all configurations by the summation of the light contributions to a target display canvas by each component projector. Also, while the paper demonstrates results on planar surfaces, this general formulation handles arbitrary display surfaces as well.

2.3 Model parameter estimation

To accurately estimate the parameters of the above model, we introduce a camera into the setup to automatically solve for the above projector parameters in an offline calibration procedure. We leverage the techniques described in [7, 4].

- \mathbf{A}_k : For each projector in sequence, an efficient structured-light coding technique computes the dense projector-camera mapping

¹Without loss of generality, the display surface is assumed to be uniformly Lambertian. Also, all image operations are performed in linear color space; nonlinear gamma effects are assumed to be undone with an inverse gamma function.

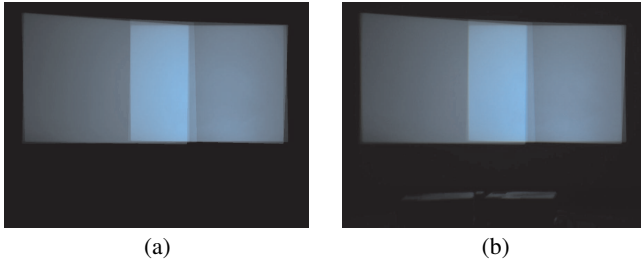


Fig. 2. Comparison between (a) the predicted output and (b) the corresponding actual output for a multi-projector system (six projectors arranged in a “two-wide, three-deep” configuration).

to cover every subframe location [2, 3]. These projector-camera mappings are subsequently remapped to the target display canvas coordinate system to finally obtain the resampling filters \mathbf{A}_k from each projector to the target display canvas [7].

- \mathbf{L}_k and \mathbf{C}_k : The system also measures the luminance profile and color transform for each projector. Because of cross talk, the camera RGB response to the maximum red, green, and blue of each projector is recorded and remapped to projector coordinates. These responses are then factorized using the technique described in [7] to obtain \mathbf{L}_k and \mathbf{C}_k .
- **b**: The system measures the overall black offset by first capturing an image when the inputs to all projector frame buffers are zero and then remapping it to the target display canvas. As seen from equation (1), it is sufficient and easier to estimate the overall black offset rather than the individual projector black offsets.

Figure 2 helps to validate the previous modeling and measurement sections. Six projectors are arranged so that three overlapping projectors cover each of half of the screen, the so-called “two-wide, three-deep” configuration. Figure 2(a) is the predicted output according to equation (1) for this multi-projector system where all projectors are projecting white. Figure 2(b) shows the actual camera-captured output of the physical system. It should be clear that the proposed paradigm accurately predicts the output. Since we do not model anything outside the target display canvas, the top of the projectors is visible at the bottom of the camera captured image (Figure 2(b)), and not in the model-predicted output (Figure 2(a)).

3 THE TARGET SPACE

Our objective is to make the output of a system of projectors appear as the output of a single hypothetical projector, whose projected output coincides with the target display canvas defined in Section 2.2. The so-called *target space* describes the desired color and luminance properties of such a virtual projector. It is preferable to construct the target space to ensure a consistent color and smoothly varying luminance.

It is worth noting that the target space does not include certain colors or intensity values that can be physically projected by the projector system. In fact, the target space is a subspace of the so-called *feasible space*, the space of outputs achievable by the system of projectors for any valid frame buffer inputs in the range [0,1]. The color and luminance characteristics of feasible space itself may be highly non-uniform. To see this, consider the case when all projectors project full white. The intersection region where projectors overlap will have a much higher intensity than regions where there is no overlap or a smaller degree of overlap; an example can be seen in Figure 2.

From the preceding discussion, we may define the target space as being generated by a single hypothetical virtual “super projector” of resolution $M_1 \times M_2$. An $M_1 \times M_2$ image \mathbf{x} may be represented in the target rendering space using the following mapping:

$$\bar{\mathbf{x}} = \bar{\mathbf{L}} \bar{\mathbf{C}} \mathbf{x} + \bar{\mathbf{b}} \quad (2)$$

The matrix $\bar{\mathbf{C}} = \mathbf{I} \otimes \tilde{\mathbf{C}}$ represents a color transform that converts from the color space of the source image to be rendered into the target color

space. The matrix $\bar{\mathbf{L}} = \tilde{\tilde{\mathbf{L}}} \otimes \mathbf{I}$ represents the spatial luminance characteristics of the virtual projector. The luminance surface $\bar{\mathbf{L}}$ simply scales gray input pixel luminance along the white point of the system so that no color shift is introduced about the neutral axis. The vector $\bar{\mathbf{b}}$ represents the desired target black offset. Clearly, there are constraints in choosing $\bar{\mathbf{C}}$, $\bar{\mathbf{L}}$, and $\bar{\mathbf{b}}$ imposed by the fact that the target space must be completely contained within the feasible space. Further, it may be desirable to maximize gamut, achieve a certain white point, or achieve a custom gamut. There is no one optimal value for $\bar{\mathbf{C}}$. It is highly dependent on user preferences, the application, and the viewing conditions, but it must be chosen to be contained in the feasible space. A simple way to choose a $\bar{\mathbf{C}}$ that is in the feasible space when the projectors have similar color characteristics (typically satisfied by using projectors of the same model) is to define it based on the intersection of the individual gamuts of \mathbf{C}_k [19]. This is represented by the equation:

$$\bar{\mathbf{C}} \subseteq \cap_k \mathbf{C}_k$$

Once the color matrix $\bar{\mathbf{C}}$ has been determined, we can design $\bar{\mathbf{L}}$ to maximize brightness subject to feasibility constraints. It is also desirable to ensure that the variations of the target luminance surface are visually imperceptible. We solve a constrained optimization problem using linear programming to achieve this. It is convenient to illustrate this surface fitting process in one dimension on image rows. Let $l(x; r)$ represent the desired luminance profile for row r . x is the distance along the profile in pixels. Using a basis of N basis functions,² we may expand $l(x; r)$ as

$$l(x; r) = \sum_{j=1}^N B_j(x; r) \theta_{jr}$$

where θ_{jr} are the coefficients in the basis expansion for row r . If this signal is sampled uniformly at M grid locations, we may represent the resulting vector \mathbf{l}_r in discrete form by the equation

$$\mathbf{l}_r = \mathbf{B}_r \boldsymbol{\theta}_r$$

where the $M \times N$ matrix \mathbf{B}_r is defined by $\mathbf{B}(i, j; r) = B_j(x_i; r)$. The brightness maximizing curve parameters θ_r^* may be obtained via linear programming. Specifically,

$$\begin{aligned} \theta_r^* &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \quad \mathbf{1}^T \mathbf{B}_r \boldsymbol{\theta} \\ &\text{s.t.} \\ \mathbf{B}_r \boldsymbol{\theta} &\leq \mathbf{l}_r^{\max} \\ \check{\mathbf{B}}_r \boldsymbol{\theta} &\leq +\delta \mathbf{1} \\ \check{\mathbf{B}}_r \boldsymbol{\theta} &\geq -\delta \mathbf{1} \end{aligned}$$

where $\check{\mathbf{B}}_r(i, j) = \frac{\partial^2 B_j(x_i; r)}{\partial x^2}$, \mathbf{l}_r^{\max} is the maximum feasible luminance, and $\mathbf{1}$ is simply a column vector of all ones. The derivative constraint assures that the magnitude of the second derivative of any luminance curve is just small enough to be imperceptible by humans. The parameter δ may be set empirically or derived from psychovisual experiments. To get the luminance surface, the above linear program is solved for each row using the dual-simplex algorithm. The dual simplex method is preferred over the simplex method since there are relatively few basis functions in comparison with grid points, so the dual problem has far fewer constraints and can be solved efficiently. Once the curves for each row are obtained, a maximum luminance surface $\bar{\mathbf{L}}^{\max}$ is obtained. The above algorithm is run for each column with \mathbf{l}_c^{\max} for the column problem equal to the columns of $\bar{\mathbf{L}}^{\max}$ after the row problem has been solved. This separable fitting procedure is both memory efficient and quick. It may be iterated a few times to obtain $\bar{\mathbf{L}} = \bar{\mathbf{L}}^{\max}$.

A virtually identical procedure to the one outlined above is used to obtain the target black surface $\bar{\mathbf{b}}$ by incorporating the constraint that the target black level should lie above the minimum feasible black

²In this work, we use a cubic B-spline basis.

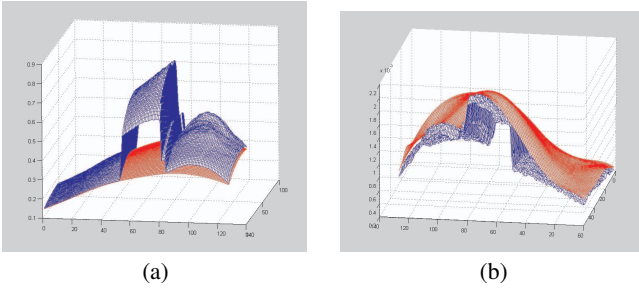


Fig. 3. Comparison of 3-D plots between feasible and target quantities for the six-projector example in Figure 2: (a) Luminance; (b) Black offset.

level. While the luminance surface $\bar{\mathbf{L}}$ maximizes brightness subject to smoothness and feasibility constraints, the black surface $\bar{\mathbf{b}}$ darkens the black level of the system as much as possible also subject to smoothness and feasibility constraints.

Figure 3 is an example for the same “two-wide, three-deep” configuration as Figure 2. The top surface (drawn in blue) in Figure 3(a) is the feasible 3-D luminance surface of this configuration while the bottom surface (in red) is the automatically computed target luminance surface. Likewise, the bottom (blue) surface in Figure 3(b) is the feasible black offset whereas the top (red) surface is the target black offset. It should be clear that the automatically computed target surfaces in both cases are smooth and offer maximal contrast.

4 OPTIMAL SUBFRAME GENERATION

Once model parameters are estimated and the target space is defined, the optimal rendering task reduces to finding subframe images \mathbf{y}_k such that the model prediction $\hat{\mathbf{x}}$ in equation (1) is as close as possible in the MSE sense to $\bar{\mathbf{x}}$ in equation (2). The subframe images \mathbf{y}_k may be assumed to be generated from the following *correction* process.

$$\mathbf{y}_k = \mathbf{C}_k^{-1} \bar{\mathbf{C}} \mathbf{y}_k^+ + \mathbf{y}_k^0 \quad (3)$$

This form is particularly convenient since it allows us to split the full optimization task of finding the optimal \mathbf{y}_k into the simpler tasks of finding an optimal correction \mathbf{y}_k^+ that encapsulates luminance correction and resolution enhancement (these may be implemented with color-plane independent filtering, as we shall see in the next subsection) and a black offset correction \mathbf{y}_k^0 (that is, a fixed correction for a given projector configuration, independent of the input image). To see how the optimization task is split, we may simply substitute (3) into (1). This yields

$$\begin{aligned} \hat{\mathbf{x}} &= \left(\sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{C}_k \left\{ \mathbf{C}_k^{-1} \bar{\mathbf{C}} \mathbf{y}_k^+ + \mathbf{y}_k^0 \right\} \right) + \mathbf{b} \\ &= \bar{\mathbf{C}} \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{y}_k^+ + \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{C}_k \mathbf{y}_k^0 + \mathbf{b} \end{aligned} \quad (4)$$

Commuting $\bar{\mathbf{C}}$ with \mathbf{L}_k and \mathbf{A}_k in (4) is allowed since applying the same color transform to all pixels in an image and then warping produces an identical result to warping an image and then applying the same color transform to all pixels.

We may rewrite the target mapping by commuting $\bar{\mathbf{C}}$ and $\bar{\mathbf{L}}$ as

$$\bar{\mathbf{x}} = \bar{\mathbf{C}} \bar{\mathbf{L}} \mathbf{x} + \bar{\mathbf{b}} \quad (5)$$

Comparing equations (4) and (5), we see that it suffices to choose \mathbf{y}_k^+ and \mathbf{y}_k^0 such that

$$\begin{aligned} \{\mathbf{y}_k^+\} &= \operatorname{argmin}_{\{\mathbf{y}_k^+\}} \left\| \bar{\mathbf{L}} \mathbf{x} - \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{y}_k^+ \right\|^2 \\ &\text{subject to: } \mathbf{0} \leq \mathbf{y}_k^+(i) \leq \mathbf{1}, \forall i, k \end{aligned} \quad (6)$$

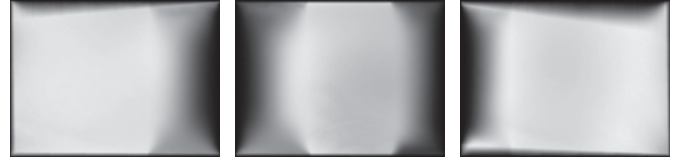


Fig. 4. Automatically computed blend maps for three tiled projectors. These smooth blend maps combine to form exactly the target surface.

and

$$\{\mathbf{y}_k^0\} = \operatorname{argmin}_{\{\mathbf{y}_k^0\}} \left\| \underbrace{(\bar{\mathbf{b}} - \mathbf{b})}_{\mathbf{r}} - \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{C}_k \mathbf{y}_k^0 \right\|^2 \quad (7)$$

$$\text{subject to: } \mathbf{0} \leq \mathbf{y}_k^0(i) \leq \mathbf{1}, \forall i, k$$

where \mathbf{r} is called the black residual image. We can treat the above two optimization problems separately, where \mathbf{x} appears only in equation (6).

4.1 Luminance and resolution optimization

Since the objective function in the optimization problem of equation (6) is convex on a convex constraint set, the optimal subframe images for a given \mathbf{x} may be directly obtained by iterative gradient descent. However, the first term $\bar{\mathbf{L}} \mathbf{x}$ in the objective function of equation (6) is smooth by design, while the second term could have sharp jumps at the projector boundaries (for example, this is observed when the projectors are tiled) for some general initial guess for the subframes \mathbf{y}_k . Since iterative gradient descent uses local updates, it would require several hundred or even thousands of iterations to propagate error corrections from the discontinuities to all subframe regions. Hence, the algorithm’s convergence speed is unacceptable. To greatly speed up convergence, we further split this optimization task by making the substitution

$$\mathbf{y}_k^+ = \mathbf{L}_k^+ \check{\mathbf{y}}_k^+ \quad (8)$$

\mathbf{L}_k^+ are *smooth luminance blend maps* that are designed to eliminate discontinuities while accurately matching desired luminance. We first pre-process (offline) the luminance target surface to derive an optimal set of blend maps described in Section 4.1.1. This enables a huge reduction in the required per image optimization iterations required to determine $\check{\mathbf{y}}_k^+$ that optimizes resolution. Only a few localized iterations are needed to optimize resolution for each input image. Resolution optimization is discussed in Section 4.1.2. Splitting the optimization task for \mathbf{y}_k^+ does not compromise the optimality of the solution.

4.1.1 Optimal blend maps

We define the desired blend maps \mathbf{L}_k^+ as diagonal matrices. For convenience in optimization, we represent them as vectors by choosing the diagonal elements only. This is accomplished by the transformation $\mathbf{I}_k^+ = \mathbf{L}_k^+ \mathbf{1}$. We then seek optimal vectors \mathbf{I}_k^+ as solutions to the following energy functional minimization problem:

$$\begin{aligned} \{\mathbf{I}_k^+\} &= \operatorname{argmin}_{\{\mathbf{I}_k^+\}} \left\{ \alpha \left\| \bar{\mathbf{L}} \mathbf{1} - \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{I}_k^+ \right\|^2 \right. \\ &\quad \left. + \beta \left\| \nabla \bar{\mathbf{L}} \mathbf{1} - \nabla \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{I}_k^+ \right\|^2 + \eta \sum_{k=1}^K \|\nabla \mathbf{I}_k^+\|^2 \right\} \\ &\text{subject to: } \mathbf{0} \leq \mathbf{I}_k^+(i) \leq \mathbf{1}, \forall i, k \end{aligned} \quad (9)$$

The first term is simply obtained by a substitution of $\mathbf{1}$ instead of \mathbf{x} in equation (6), which essentially selects the diagonal elements of $\bar{\mathbf{L}}$ that completely define the target luminance surface as the desired target image. The second term seeks to match at each pixel the gradients of the luminance surface with the gradients of the simulated luminance

image from the multi-projector display. This is consistent with recent results in the area of image editing and seamless photo blending [8, 17, 12, 22] that showed that gradient domain fusion resulted in the high quality seamless images when illumination varied considerably in the images to be blended. The third term is a regularization term critical in guaranteeing robustness to calibration drift. It enforces a smoothness of the resulting blend maps \mathbf{I}_k^+ by minimizing gradient variation. The factors α, β, η trade off the various factors. We choose $\alpha \gg \beta \sim \eta$ ensuring that the solution ensures accurate reproduction of the target luminance surface that is also as smooth as possible. Taking derivatives and setting them to zero yields the first-order necessary and sufficient conditions for optima. Thus, the optimal blend maps are solutions to the following set of Euler-Lagrange partial differential equations.

$$\alpha \mathbf{L}_j^T \mathbf{A}_j^T \left\{ \alpha \mathbf{e} + \beta \nabla^2 \mathbf{e} \right\} = \eta \nabla^2 \mathbf{I}_j^+, \quad j = 1 \dots K \quad (10)$$

where

$$\mathbf{e} = \bar{\mathbf{L}}\mathbf{1} - \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{I}_k^+$$

and ∇^2 is the discrete Laplacian operator.

These equations may be solved very efficiently for \mathbf{I}_j^+ with fast convergence using coarse-to-fine multigrid methods [21, 8, 17] with Neumann boundary conditions. These methods solve the above system of PDEs hierarchically by first transforming the problem to a coarse resolution. This is done by subsampling the luminance images. An iterative gradient descent algorithm can be used to find the optimal solution for the coarse level. Once the solution is obtained at the coarsest levels, it is mapped via interpolation to the next higher resolution level where it forms the initial guess at the solution for that level. The same iterative algorithm is applied at the new level to refine the initial guess into an optimal solution for that level. This process is repeated at successively finer resolutions until the finest resolution (which corresponds to the resolution of the original problem) is reached. The multi-resolution approach propagates error corrections very rapidly, allowing for fast convergence with only a few iterations needed at each level. The iterative gradient descent algorithm at each level may be expressed compactly as:

$$\mathbf{I}_j^{+(n+1)} = \psi \left(\mathbf{I}_j^{+(n)} + \alpha \mathbf{L}_j^T \mathbf{A}_j^T \left\{ \alpha \mathbf{e}^{(n)} + \beta \nabla^2 \mathbf{e}^{(n)} \right\} - \eta \nabla^2 \mathbf{I}_j^{+(n)} \right)$$

where n indicates the iteration number and where ψ is the clipping function that clips all elements of its vector argument to the $[0, 1]$ range. Thus, we can obtain the smooth luminance blend maps \mathbf{L}_k^+ in equation (8).

Figure 4 shows the luminance blend maps for an example of three tiled projectors. The blend maps have been automatically generated by the above process converging in only ten iterations performed at each of seven levels. Notice that they are smoothly varying.

4.1.2 Resolution enhancement

Incorporating equation (8), the subframe generation problem of equation (6) becomes for a given \mathbf{x}

$$\{\check{\mathbf{y}}_k^+\} = \underset{\{\check{\mathbf{y}}_k^+\}}{\operatorname{argmin}} \left\| \bar{\mathbf{L}}\mathbf{x} - \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{L}_k^+ \check{\mathbf{y}}_k^+ \right\|^2 \quad (11)$$

$$\text{subject to: } \mathbf{0} \leq \check{\mathbf{y}}_k^+(i) \leq \mathbf{1}, \quad \forall i, k$$

For an arbitrary \mathbf{x} , both terms in the objective function of equation (11) are smooth and global luminance variations have been eliminated. This implies that in generating the optimal subframes, local corrections to subframe pixels suffice and only a few iterations will be needed to optimize image resolution by cancelling residual aliasing and eliminating blur. A straightforward gradient descent algorithm

will yield the optimal estimates for the subframes in only a few iterations. This algorithm is presented below:

$$\check{\mathbf{y}}_k^{+(0)} = \mathbf{L}_k^{+T} \mathbf{L}_k^T \mathbf{A}_k^T \mathbf{x} \quad (\text{initial guess}) \quad (12)$$

$$\hat{\mathbf{f}}^{(n)} = \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{L}_k^+ \check{\mathbf{y}}_k^{+(n)} \quad (\text{modeling}) \quad (13)$$

$$\frac{\partial J}{\partial \check{\mathbf{y}}_k^{+(n)}} = -\mathbf{L}_k^{+T} \mathbf{L}_k^T \mathbf{A}_k^T \left(\bar{\mathbf{L}}\mathbf{x} - \hat{\mathbf{f}}^{(n)} \right) \quad (\text{gradient}) \quad (14)$$

$$\check{\mathbf{y}}_k^{+(n+1)} = \psi \left(\check{\mathbf{y}}_k^{+(n)} - \mu \frac{\partial J}{\partial \check{\mathbf{y}}_k^{+(n)}} \right) \quad (\text{correction}) \quad (15)$$

$$\{\check{\mathbf{y}}_k^+\} = \lim_{n \rightarrow \infty} \{\check{\mathbf{y}}_k^{+(n)}\} \quad (16)$$

where μ is a parameter indicating the fraction of error to be incorporated at each iteration. The algorithm consists of two passes: In the modeling pass (13), we compute $\hat{\mathbf{f}}^{(n)}$ from the current guesses of the subframes $\check{\mathbf{y}}_k^{+(n)}$. Then, a correction pass (15) updates the subframes based on prediction errors. This algorithm may be intuitively understood as an iterative process of computing an error in the reference high resolution coordinate system and projecting a filtered version of it back onto the subframe data to form better estimates of the subframes. The operator \mathbf{A}_k^T is simply another compactly supported resampling filter [7]. Since the problem is convex with convex constraints, this process is guaranteed to converge to the optimal solution $\check{\mathbf{y}}_k^+$ for the subframes, and hence one can obtain the optimal \mathbf{y}_k^+ given by equation (8).

4.2 Black offset optimization

Once \mathbf{L}_k^+ has been determined, we can solve the optimization problem of equation (7) by the following iteration:

$$\begin{aligned} \mathbf{y}_k^{0(0)} &= \mathbf{L}_k^{+T} \mathbf{L}_k^T \mathbf{A}_k^T \mathbf{r} \\ \mathbf{e}^{0(n)} &= \mathbf{r} - \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{C}_k \mathbf{y}_k^{0(n)} \\ \mathbf{y}_k^{0(n+1)} &= \psi \left(\mathbf{y}_k^{0(n)} + \lambda \mathbf{C}_k^T \mathbf{L}_k^T \mathbf{A}_k^T \mathbf{e}^{0(n)} \right) \\ \{\mathbf{y}_k^0\} &= \lim_{n \rightarrow \infty} \{\mathbf{y}_k^{0(n)}\} \end{aligned}$$

where λ is a parameter indicating the fraction of error to be incorporated at each iteration. This optimization yields the optimal black corrections.

4.3 Overall optimal solution

The results of the previous subsections may be combined to form the overall optimal subframe that optimizes for color, luminance, black levels, and resolution. Specifically, the solutions for \mathbf{y}_k^+ and \mathbf{y}_k^0 may be substituted into equation (3) to compute the overall optimal solution.

The iterative resolution enhancement algorithm of Section 4.1.2 produces optimal subframes. However, this algorithm precludes real-time rendering due to its computational complexity. To address this issue, we consider an efficient non-iterative algorithm for subframe generation that may be implemented as a bank of filters operating directly on the input image to be displayed. As proved in [5], we derive near-optimal filters by directly finding the impulse response (possibly space varying) of a linear approximation to the non-linear optimal iterative algorithm of Section 4.1.2.

Subframe generation is then accomplished by a bank of K space-varying filters, each operating on the high resolution image to produce a component subframe [5, 7]. Considering all color planes, this spatially varying filtering operation for subframe k may be represented as a matrix multiplication of \mathbf{x} with a matrix \mathbf{H}_k . Thus, fast real-time subframe generation can be accomplished by the equation

$$\mathbf{y}_k = \psi \left(\mathbf{C}_k^{-1} \bar{\mathbf{C}} \mathbf{H}_k \mathbf{x} + \mathbf{y}_k^0 \right) \quad (17)$$

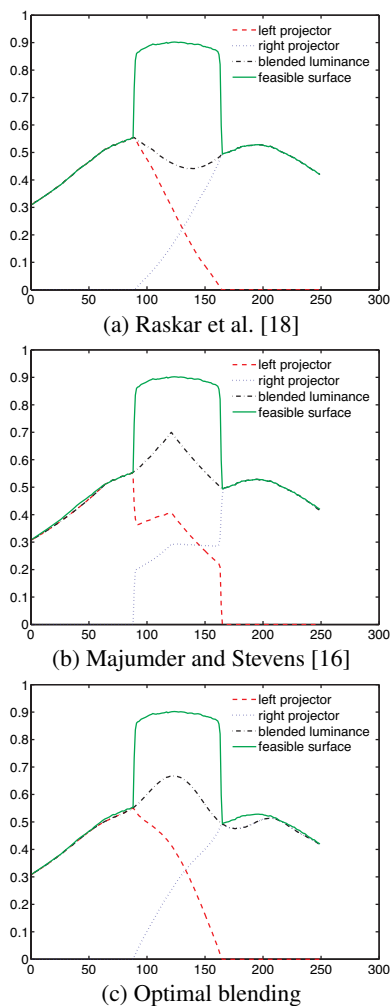


Fig. 5. Performance of various blending algorithms for two similar projectors with substantial overlap.

These operators may be efficiently implemented in the pixel shaders of commodity GPUs for real-time rendering [7].

5 RESULTS

In this section, we consider examples to compare our method with prior work and to illustrate the benefits of the unified framework. We include results from prototype multi-projector display systems leveraging the unified framework. As shown in Figure 1, they consist of four-to-six XGA (1024×768) DLP projectors running on a single HP xw9300 workstation. The system uses a Point Grey Research 1600×1200 Scorpion Firewire camera to perform the automatic calibration in a one-time process. The entire calibration and modeling process discussed in Sections 2 through 4 takes less than ten minutes total for all six projectors. It runs a custom C++/OpenGL application on a single nVidia GeForce 8800 GTX graphics card to perform real-time filtering and rendering (equation (17)) for all six projectors at 30 fps or higher. The same system may be reconfigured to address different display configurations and only a few representative examples are included due to space limitations. Additional results are available at our web site (<http://hpl.hp.com/research/pluribus>).

5.1 Tiled projection

Consider an example where two fairly similar projectors are tiled. First, we consider the traditional approach to tiled projection via edge blending. Representative state of the art geometric blending methods like [18] are based on empirical blend functions. These blend func-

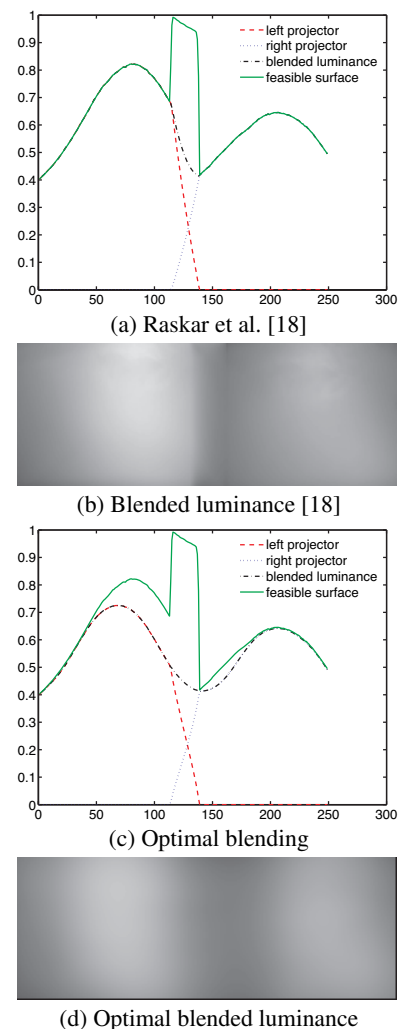


Fig. 6. Conventional vs. optimal blending for two different projectors with small overlap. The seam in (b) is visible when projected, while the blended luminance in (d) has no visible seam.

tions taper off projector outputs in overlap regions, causing the sum of the luminances in the overlap region to vary smoothly. Figure 5(a) shows a one-dimensional horizontal cross-sectional view of the luminance surfaces used in the blending process. In this case, a significant amount of light in the overlap region is wasted due to the natural tapering of the blend functions. It is easy to see that as the overlap increases, even more light becomes wasted in the overlap region.

To overcome the brightness, contrast (if we make the similar arguments for black levels), and visual inefficiencies of classical geometric blending, Majumder and Stevens [16] proposed a fast dynamic programming approach to fitting a target luminance surface to maximize brightness while minimizing visible brightness fluctuations. However, as shown in Figure 5(b), Majumder and Stevens' blended luminance target surface is not smooth and exhibits substantial gradient jumps. Even if this issue is ignored and a perfectly smooth target surface were specified, the generated blend functions still exhibit steep jumps in the blend maps for each projector. This means that this method of generating blend maps is very sensitive to calibration/model error or drift.

Harville et al. [9] found that applying geometric blending (similar to Raskar et al. [18]) prior to using Majumder and Stevens' algorithm produced smooth blend maps. However, this increased robustness is achieved at the cost of brightness/contrast maximization. Since the target surfaces are constrained to be below the geometrically blended surfaces, it is then impossible to increase brightness as projector over-

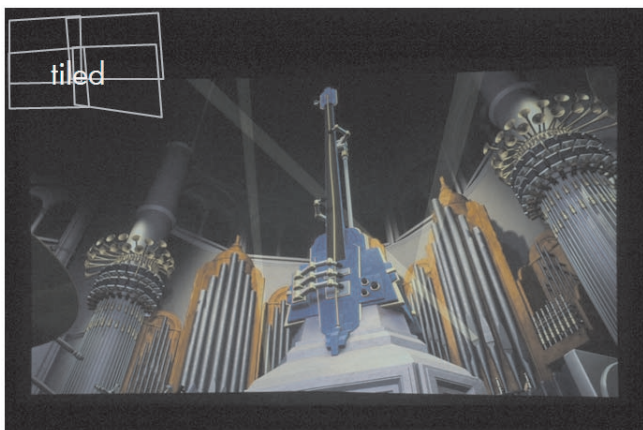


Fig. 7. Experimental results with four projector in a tiled configuration. Note the seamless nature of the final result. ©2007 Animusic. Image used with permission (www.animusic.com).

lap increases.

No prior method exists that generates smooth blend maps, while at the same time maximizing brightness and contrast subject to luminance fluctuation visibility constraints. Our proposed optimal blend map generation process described in Section 4.1.1 achieves these objectives for the first time. Figure 5(c) shows that when the projector overlap is significant, our surface fitting procedure produces brightness levels in the overlap region that are significantly higher than those of any individual projector. Further, the generated blend maps that achieve the brightness maximization are smooth. In comparison with the surface fitting procedure of [16], our method based on the cubic B-spline bases is smooth with continuous second derivatives.

Consider now an example when the left projector is significantly brighter than the right projector and the overlap area is reduced. In this situation, conventional blending transitions rapidly from a high brightness projector to a low brightness projector over a narrow overlap region (Figure 6(a)). This causes the transition to appear as a visible seam in the display (Figure 6(b)). With the proposed technique, the bound δ (defined in Section 3) on the magnitude of the second derivative bounds the variation of the luminance surface, preventing steep jumps at the boundary of the overlap region. As shown in Figures 6(c) and (d), brightness is optimally reduced to preserve seamlessness in this case. If more brightness were desired, we can simply increase δ at any point to make the tradeoff with seam visibility.

Figure 7 is a screen capture of an actual four-projector tiled system in a two-wide, two-tall configuration (as diagrammed in the upper left corner). The system automatically solves for the optimal blend maps and produces a seamless final result for 1920×1080 video as shown in the figure. Note that the black offset is virtually invisible.

5.2 Superimposed projection

We next consider when all the projectors are superimposed. Brightness is typically increased with superimposed projection. Conventional algorithms, however, compromise the resolution of the display, especially when the resolution of the input to be displayed is higher than any individual projector resolution, resulting in either blurred or aliased images [5].

Using the rendering algorithms designed for tiled displays on a superimposed configuration also leads to sub-optimal results. As we saw earlier, these algorithms function poorly with respect to brightness and contrast when projector overlap is increased. In fact, if a conventional blending algorithm is used for superimposed projection, the brightness of the system is effectively that of a single projector. Note that the converse is also true: rendering algorithms designed for superimposed displays [11, 5, 7] are focused on resolution enhancement and perform poorly for tiled configurations. They ignore target surface fitting and

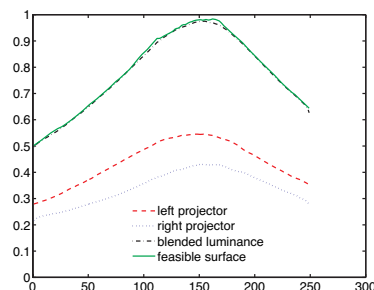


Fig. 8. Unified framework applied to superimposed projection: luminance profiles.



Fig. 9. Experimental results with four superimposed projectors: (a) single aliased subframe (contrast enhanced for visibility) versus (b) super-resolved output. Note the resolution enhancement as well as brightness gain of the superimposed result.

smoothing that is essential for handling luminance discontinuities in tiled displays.

In contrast, our proposed paradigm automatically computes the maximal brightness and solves for the optimal resampling filters, even enhancing resolution when possible. Figure 8 shows the performance of our system for superimposed projection with two projectors. Our target surface fitting algorithm automatically utilizes the full brightness capability of the combined system. Furthermore, the blend maps are not tapered in this case, allowing the target surface to be achieved without compromise.

Through the unified framework, one can observe the super-Nyquist resolution gain achieved over a single projector as shown in Figure 9. In this case, we compare the single aliased subframe versus four superimposed projectors for displaying higher resolution content (1600×1200). Because of its relative dimness, the single projector image has actually been contrast enhanced to improve visibility. There is clear improvement in text resolvability in the displayed Excel spreadsheet in the superimposed case.

5.3 Hybrid projection

In the previous sections, we showed how our framework can handle both tiled and superimposed projection seamlessly, with no configuration-specific modifications. We demonstrate in this section that it can also handle arbitrary configurations.

Figure 10 shows an actual screenshot of an example novel hybrid six-projector configuration (three projectors wide, two deep) displaying a high resolution 2700×800 video. In this case, the unified framework ensures color consistency, as well as maximizes brightness, contrast, and resolution for any configuration without any per configuration tweaking. Moreover, there is no visible seams or color shifts with this automatically produced result. This configuration has the added benefit of being fault tolerant to the loss of one projector.

Figure 11 shows a second hybrid example with the same six projectors arranged in another non-standard configuration (two tall, three deep), again resulting in a seamless and high quality final result. By design, this configuration is fault tolerant to the loss of two projectors. These configurations are representative examples of the freedom the paradigm offers in trading off brightness, resolution, aspect ratio, and



Fig. 10. The proposed paradigm automatically corrects the inherent geometry, color, luminance, and black offset variations among six projectors in a three-wide, two-deep hybrid configuration (top) to create a seamless and high quality result on a 16' wide screen (bottom).

fault tolerance while producing high quality and seamless images.

6 CONCLUSION

We have presented a unified paradigm that automatically optimizes image quality (i.e. color, brightness, contrast, resolution) of multi-projector systems regardless of how they are configured. We showed that our configuration-agnostic framework produces results that outperform the state-of-the-art even when we consider purely tiled or purely superimposed system configurations. Moreover, the prior algorithms for tiled projection and those for superimposed projection are shown to be significantly sub-optimal when applied to the other paradigm. In addition, we demonstrated robust modeling and measurement techniques that lead to seamless and high quality results in real-time using commodity GPUs. With current graphics technology, the same paradigm can handle up to twelve projectors on a single workstation, limited only by the bus bandwidth, size of texture memory, and number of physical display outputs. Further scalability can be achieved by linking multiple such workstations together.

The proposed paradigm allows for the first time explicit tradeoffs to be made and novel configurations to be supported. We believe this work opens new opportunities in the area of large display visualization through practical scalable multi-projector displays.

ACKNOWLEDGEMENTS

The authors wish to thank Arun Paruchuri, Tom Carrico, Si-Jie Yu, Suk-Hwan Lim, Qian Lin, Jerry Liu, and Bob Ulichney for their help with and support of this work.

REFERENCES

- [1] M. Brown, A. Majumder, and R. Yang. Camera-based calibration techniques for seamless multiprojector displays. *IEEE Trans. Visual. Comput. Graphics*, 11:193–206, Mar. 2005.
- [2] N. L. Chang. Efficient dense correspondences using temporally encoded light patterns. In *Proc. IEEE International Workshop on Projector-Camera Systems (ProCams)*, Nice, France, Oct. 2003.
- [3] N. L. Chang. Creating interactive 3-D media with projector-camera system. In *Proc. SPIE Visual Communications and Image Processing Conference (VCIP)*, volume 5308, pages 850–861, San Jose, CA, Jan. 2004.
- [4] N. L. Chang, N. Damera-Venkata, J. M. DiCarlo, and S. H. Lim. Camera-based measurement for projector-camera systems. Technical report, Hewlett-Packard Laboratories, Palo Alto, CA, In preparation.
- [5] N. Damera-Venkata and N. L. Chang. Display supersampling. *ACM Transactions on Computer Graphics*, 2007. Accepted with revisions.

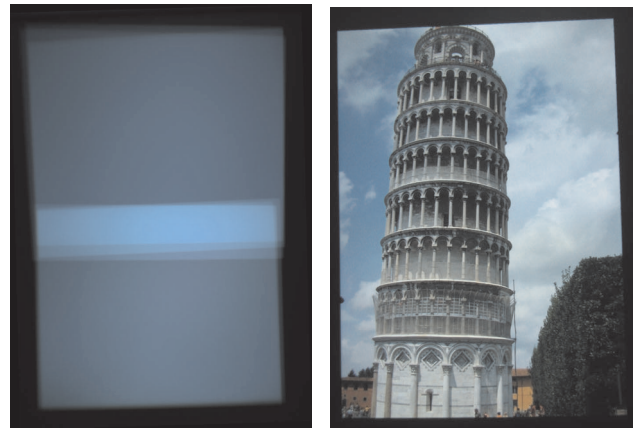


Fig. 11. Hybrid configuration where six projectors arranged in a two-tall, three-deep configuration (left) to create a seamless and high quality result for a 9' tall screen (right).

- [6] N. Damera-Venkata and N. L. Chang. On the resolution limits of superimposed projection. In *Proc. IEEE International Conference on Image Processing (ICIP)*, San Antonio, TX, Sept. 2007.
- [7] N. Damera-Venkata and N. L. Chang. Realizing super-resolution with superimposed projection. In *Proc. IEEE International Workshop on Projector-Camera Systems (ProCams)*, Minneapolis, MN, June 2007.
- [8] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range image compression. In *Proc. ACM SIGGRAPH*, pages 249–256, San Antonio, TX, July 2002.
- [9] M. Harville, B. Culbertson, I. Sobel, D. Gelb, A. Fitzhugh, and D. Tang. Practical methods for geometric and photometric correction of tiled projector displays on curved surfaces. In *Proc. IEEE International Workshop on Projector-Camera Systems (ProCams)*, pages 52–59, New York, USA, June 2006.
- [10] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ, 1988.
- [11] C. Jaynes and D. Ramakrishnan. Super-resolution composition in multi-projector displays. In *Proc. IEEE International Workshop on Projector-Camera Systems (ProCams)*, Nice, France, Oct. 2003.
- [12] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *Proc. European Conference on Computer Vision (ECCV)*, pages 377–389, Prague, CZ, May 2004.
- [13] A. Majumder. Contrast enhancement of multi-displays using human contrast sensitivity. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 377–382, San Diego, USA, June 2005.
- [14] A. Majumder. Is spatial super-resolution feasible using overlapping projectors? In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 209–212, Philadelphia, USA, Mar. 2005.
- [15] A. Majumder and R. Stevens. Color nonuniformity in projection-based displays: Analysis and solutions. *IEEE Trans. Visual. Comput. Graphics*, 10:177–188, Mar. 2004.
- [16] A. Majumder and R. Stevens. Perceptual photometric seamlessness in tiled projection-based displays. *ACM Trans. on Graphics*, 24:111–134, Jan. 2005.
- [17] P. Perez, M. Gagnat, and A. Blake. Poisson image editing. In *Proc. ACM SIGGRAPH*, pages 313–318, San Diego, CA, July 2003.
- [18] R. Raskar, M. Brown, R. Yang, W. Chen, G. Welch, H. Towles, B. Seales, and H. Fuchs. Multi-projector displays using camera-based registration. In *Proc. Visualization*, pages 161–522, San Francisco, USA, Oct. 1999.
- [19] M. Stone. Color and brightness appearance issues in tiled displays. *IEEE Comp. Graphics and Applications*, 21:58–66, Sept. 2001.
- [20] R. Surati. *Scalable Self-Calibrating Display Technology for Seamless Large-Scale Displays*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, Jan. 1999.
- [21] P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley and Sons, Chichester, UK, 1992.
- [22] A. Zomet, A. Levin, S. Peleg, and Y. Weiss. Seamless image stitching by minimizing false edges. *IEEE Trans. Image Processing*, 15:969–977, Apr. 2006.