

Realizing Super-Resolution with Superimposed Projection

Niranjan Damera-Venkata
Hewlett-Packard Laboratories
Palo Alto, CA

niranjan.damera-venkata@hp.com

Nelson L. Chang
Hewlett-Packard Laboratories
Palo Alto, CA

nelson.chang@hp.com

Abstract

We consider the problem of rendering high-resolution images on a display composed of multiple superimposed lower-resolution projectors. A theoretical analysis of this problem in the literature previously concluded that the multi-projector superimposition of low resolution projectors cannot produce high resolution images. In our recent work, we showed to the contrary that super-resolution via multiple superimposed projectors is indeed theoretically achievable. This paper derives practical algorithms for real multi-projector systems that account for the intra- and inter-projector variations and that render high-quality, high-resolution content at real-time interactive frame rates. A camera is used to estimate the geometric, photometric, and color properties of each component projector in a calibration step. Given this parameter information, we demonstrate novel methods for efficiently generating optimal sub-frames so that the resulting projected image is as close as possible to the given high resolution images.

1. Introduction

Traditional attempts at seamless multi-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 [14, 16, 2, 12, 13, 7]. These tiled approaches offer scalability of resolution and brightness as well as the ability to have arbitrary aspect ratios. However, the performance of the array is often limited by the quality of the worst projector in the array. The brightness and contrast of the system is typically limited by the brightness of the lowest brightness projector and the projector with the worst black state respectively. Additionally, if a single projector in a tiled array drifts from color or luminance calibration, the entire projector array needs to be changed to accommodate the worst projector, or else the drifting projector needs to be replaced. If even one projector goes out due to a lamp

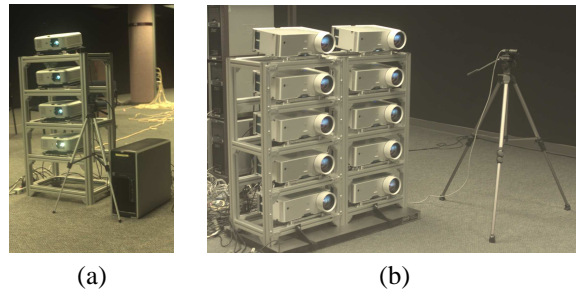


Figure 1: Example multi-projector display systems with (a) four and (b) ten superimposed projectors.

failure, the opportunity cost is incurred on the whole bank.

The key question is how to effectively scale along all image quality attributes and yet ensure a reliable system not limited by the performance or quality of the worst projector. We argue that through superimposed projection (where multiple projectors are substantially superimposed), we obtain a fundamentally more robust system that can better compensate for common intra- and inter-projector variations (e.g. color and luminance drift) that plague conventional tiled displays. Furthermore, superimposed projectors are able to better utilize the full brightness range of all component projectors. The reliability of superimposed projection stems from the fact that a single component projector is only a fractional contribution to the overall image quality. For example, when a single projector fails in a ten-projector superimposed system, the user would barely notice the failure since each projector contributes around 1/10th of the brightness of the bank and humans have a nonlinear response to brightness. Color and luminance drifts are also not noticeable within a wide range of tolerance since the entire image drifts incrementally and opposing drifts often cancel.

A very important issue with superimposed projection is how to avoid the blurring and apparent loss of resolution due to the superimposition. It is natural to examine the problem of multi-projector super-resolution, the dual of multi-

camera super-resolution. The goal of multi-projector super-resolution is to produce a high resolution frame via the superimposition of multiple low resolution images or so-called subframes. While rigorous analysis of the theory and limits of camera super-resolution exists [1, 10], the analysis of multi-projector super-resolution has been limited in scope. Previous work has shown the viability of super-resolution via superimposition only in the special case when the resulting superimposition grid is uniform [11, 15]. This assumption is clearly invalid in the case of multi-projector superimposition. Moreover, Majumder [11] concluded that resolution enhancement is impossible when the superimposed grid is non-uniform.

In recent work [5, 6], we showed that in fact super-resolution via superimposed projection is theoretically possible even in the case of non-uniform sampling. Specifically, we can generate subframes that are themselves severely aliased, but when superimposed combine to form alias-free high resolution images. If the low resolution pixel reconstruction filter (point spread function or PSF) bandwidth is greater than the Nyquist frequency of the low resolution image, each low resolution image can be engineered to contribute frequencies higher than its Nyquist frequency. Of course, this means that each subframe will have low frequency aliasing. By properly generating the complementary subframes, we showed that we can cancel the low frequency aliasing and reconstruct alias-free high frequencies that are *beyond* the Nyquist frequency of a single projector. Thus, the projector subframe generation process can be thought of as similar to the alias cancellation in filter bank theory.

This paper examines subframe generation in the context of practical multi-projector display systems. While previous work [5] focused on theory and validation via computer simulation assuming all the projectors were identical, this work describes practical algorithms to account for color and luminance differences among physical projectors and discusses actual real-time implementation on commodity graphics hardware.

Jaynes and Ramakrishnan [9] first considered the problem of multi-projector superimposed projection and demonstrated some resolution enhancement where the projectors overlapped. They approximate a warped subframe as the sum of regionally shifted subsampled images. Each such image is initially estimated in the frequency domain by phase shifting the frequencies of the target image to be rendered. Then, a greedy heuristic process is used to recursively update pixels with the most error globally. The authors point out that this formulation not only is sub-optimal (by approximating the true homography), but also precludes real-time rendering applications because of its complexity. In contrast, we derive in this paper an optimal, provably convergent subframe generation algorithm that leads to ef-

ficient real-time rendering. Moreover, it also accounts for intra- and inter-projector variations.

Section 2 discusses our model for multi-projector superimposition. This model accounts for relative geometric, photometric and color distortions among the component projectors. Section 3 presents techniques for robustly and accurately estimating these model parameters. Section 4 derives efficient practical algorithms for real-time rendering at interactive frame rates. Section 5 demonstrates experimental results on real multi-projector display systems (see Figure 1) built with commodity projectors and graphics hardware. Finally, Section 6 concludes the paper by summarizing the contributions and pointing to future directions.

2. Modeling Multi-Projector Superimposition

This section models the overall image formation process of multiple superimposed projectors. Assuming a Lambertian screen surface, an estimate of the final projected image is given by the summation of the individual projector images. The key then is to model the imaging process from input image to the framebuffer of each component projector; let the resolution of the frame buffer of a component projector be an $N_1 \times N_2$. We next choose a reference "canvas" with respect to which each simulated projector image is computed. The resolution of the reference canvas is the desired resolution at which the final simulated image is computed. Let the resolution of the display canvas be chosen as $M_1 \times M_2$.

Mathematically, the 2-D model for a K projector system may then be represented as:

$$\begin{aligned} \hat{\mathbf{x}} &= \left(\sum_{k=1}^K \mathbf{A}_k \underbrace{\mathbf{L}_k \mathbf{C}_k}_{\mathbf{z}_k} \mathbf{y}_k \right) + \mathbf{b} & (1) \\ \mathbf{y}_k &= \gamma_k^{-1}(\mathbf{y}'_k) \\ \mathbf{C}_k &= \mathcal{I}_{N_1 N_2 \times N_1 N_2} \otimes \tilde{\mathbf{C}}_k \\ \mathbf{L}_k &= \tilde{\mathbf{L}}_k \otimes \mathcal{I}_{3 \times 3} \\ \mathbf{A}_k &= \tilde{\mathbf{A}}_k \otimes \mathcal{I}_{3 \times 3} \end{aligned}$$

\mathbf{y}'_k is a $3 N_1 N_2 \times 1$ vector that represents the $N_1 \times N_2$ color image (assuming the typical three R,G,B color planes) that is input to the frame buffer of the k^{th} projector. The vector \mathbf{y}'_k is formed by ordering the color image by color plane and by rows. Hence, the first three elements are the R, G, B components of pixel (0,0), while the next three elements are the R, G, B components of pixel (0,1) and so on. The k^{th} projector applies a vector-valued nonlinear inverse gamma function $\gamma_k^{-1}(\cdot)$. Projectors may apply different inverse gamma curves to different color planes. Thus, \mathbf{y}_k represents the linear-space input image to projector k .

The $3 N_1 N_2 \times 3 N_1 N_2$ matrix \mathbf{C}_k represents a color transform that converts from 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. In this paper, we will denote the common reference space as XYZ although it may or may not correspond to the actual CIE XYZ color space. \mathbf{C}_k is completely defined by the 3×3 color transform $\tilde{\mathbf{C}}_k$ via the decomposition $\mathbf{C}_k = \mathcal{I}_{N_1 N_2 \times N_1 N_2} \otimes \tilde{\mathbf{C}}_k$, where \otimes represents the Kronecker product operator and $\mathcal{I}_{N_1 N_2 \times N_1 N_2}$ represents the $N_1 N_2 \times N_1 N_2$ identity matrix.

The $3 N_1 N_2 \times 3 N_1 N_2$ diagonal matrix \mathbf{L}_k represents the 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. \mathbf{L}_k may be decomposed as $\mathbf{L}_k = \tilde{\mathbf{L}}_k \otimes \mathcal{I}_{3 \times 3}$, where $\tilde{\mathbf{L}}_k$ is an $N_1 N_2 \times N_1 N_2$ diagonal matrix. Each element along the diagonal scales the RGB pixel values at a given framebuffer location equally. However, the scaling for different pixel locations is different.

The $3 M_1 M_2 \times 3 N_1 N_2$ matrix \mathbf{A}_k encapsulates the effects of geometric distortion, pixel reconstruction point spread function and resample filtering operations. For clarity, its derivation is given in the Appendix.

Finally, the $3 M_1 M_2 \times 1$ 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, the camera captures some light in this case. This camera image, when warped and resampled to the reference coordinate system, gives the black offset \mathbf{b} of the system.

3. Model Parameter Estimation

This section discusses methods for accurately estimating the parameters of the above model for a practical multi-projector display system. To this end, we introduce a camera into the set up to automatically and correctly characterize the differences (e.g. geometry, luminance, color) among the projectors in a one-time calibration procedure.

The projectors are likely to be geometrically misaligned with respect to one another due to their different optical paths and physical placement. For each projector in sequence, an efficient structured-light coding technique computes the dense projector-camera mapping to cover every subframe location. Specifically, the steps are as follows:

1. Project each coded light pattern onto the scene in succession and capture the result. Horizontally and vertically striped binary light patterns are used to repre-

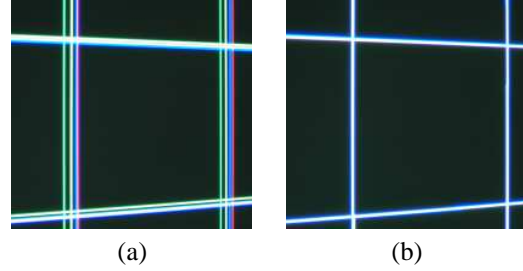


Figure 2: Different colored grid lines projected out of four superimposed projectors (a) before and (b) after geometric calibration.

sent the bitplanes of the Gray-coded binary representation of the row and column indices [3]. Invalid pixels that lie outside the projected space or that do not offer enough discrimination between black and white are automatically masked off.

2. Decode bit sequences at every camera pixel and refine with coarse-to-fine analysis. Through multi-scale corner analysis and interpolation [4], one can accurately establish the subpixel mapping for every camera coordinate and corresponding projector coordinate.
3. Parametrize the dense correspondences if applicable. For simplification with planar display surfaces, each mapping may be further reduced to a 3×3 homography using least squares over the dense set of correspondence pairs. Other higher order models may be used to account for non-planar display surfaces and non-negligible lens distortion.

Unlike traditional approaches, this technique robustly computes the geometric mapping to subpixel accuracy regardless of the projector configuration and underlying display surface (handles even arbitrary and multiple 3-D objects). These projector-camera mappings are subsequently remapped to the canvas coordinate system to finally obtain the warps $\mathbf{m}_k(\cdot)$ from each projector to the reference canvas, and hence operators \mathbf{A}_k (See Appendix). For superimposed projection, the reference canvas typically corresponds to a region that lies entirely in the intersection of all the projectors' outputs. Figure 2 shows an example of different colored grid lines projected from four superimposed projectors. These grid lines become well aligned and crisp after the proposed calibration process.

In addition to geometry, the system also measures the luminance profile and color transform for each projector. The camera response to the maximum red, green and blue of each projector is recorded and remapped to projector coordinates. Note that a pure red from a projector results in red, green and blue camera sensor responses in gen-

eral, and similarly for projecting pure green and pure blue. Thus, the color and luminance characteristics of the k -th projector are jointly defined by the multiplication with a $3N_1 N_2 \times 3N_1 N_2$ block diagonal matrix $\hat{\mathbf{C}}_k$ to form \mathbf{z}_k

$$\begin{aligned} \mathbf{z}_k &= \hat{\mathbf{C}}_k \mathbf{y}_k \\ &= \begin{pmatrix} \hat{\mathbf{C}}_{k,0} & & & \\ & \hat{\mathbf{C}}_{k,1} & & \\ & & \ddots & \\ & & & \hat{\mathbf{C}}_{k,M} \end{pmatrix} \begin{pmatrix} \mathbf{y}_{k,0} \\ \mathbf{y}_{k,1} \\ \vdots \\ \mathbf{y}_{k,M} \end{pmatrix} \end{aligned}$$

where $M = N_1 N_2 - 1$. Thus, the camera response (in projector space) to the the i^{th} projector pixel is $\mathbf{z}_{k,i} = \hat{\mathbf{C}}_{k,i} \mathbf{y}_{k,i}$. The model of equation (1) requires that each $\hat{\mathbf{C}}_{k,i}$ be factorizable as a constant per-projector color matrix $\tilde{\mathbf{C}}_k$ followed by luminance scaling. This means that

$$\hat{\mathbf{C}}_{k,i} = \underbrace{(\tilde{\mathbf{L}}_{k,i})}_{\mathbf{L}_{k,i}} \mathbf{I}_{3 \times 3} \tilde{\mathbf{C}}_k$$

Note that $\tilde{\mathbf{L}}_{k,i}$ is a scalar value that varies with i . A convenient approach to this factorization that works well in practice is to choose

$$\tilde{\mathbf{L}}_{k,i} = \frac{1}{3} \sum_{(u,v)=(1,0)}^{(1,2)} \hat{\mathbf{C}}_{k,i}(u,v) \quad \forall i, \quad \tilde{\mathbf{C}}_k = \frac{1}{N_1 N_2} \sum_i \frac{\hat{\mathbf{C}}_{k,i}}{\tilde{\mathbf{L}}_{k,i}}$$

It is also possible to measure the black offset of each individual projector or instead the overall black offset (i.e. \mathbf{b} in equation (1)). By definition, the projectors always reproduce this offset no matter what image is being projected, so the display system can be no dimmer than this level. Since black offset seams are not typically present with superimposed projection, there may be no need to explicitly account for black offset in the subframe generation process.

4. Subframe Generation Algorithms

In this section, we derive practical subframe generation algorithms for multiple superimposed projectors. First, Section 4.1 discusses the construction of a target rendering space. Each input image frame is mapped into the target rendering space, much like how an image is mapped into printer space via gamut mapping before printing. Section 4.2 presents an iterative algorithm for optimal subframe generation. Section 4.3 derives a near-optimal, efficient, non-iterative algorithm for subframe generation that enables real-time performance. Alias-cancellation is achieved by a bank of filters designed to have compact support.

4.1. Defining the Target Rendering Space

The target rendering space defines the space of allowed colors and intensities. This space is a subset of the feasible space of signals achievable by the system of superimposed projectors. An input image \mathbf{x} , assumed to be in gamma uncorrected linear space, 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 $3M_1 M_2 \times 3M_1 M_2$ matrix $\bar{\mathbf{C}}$ represents a color transform that converts from the color space of the input image (such as sRGB) into the reference XYZ color space. As before, $\bar{\mathbf{C}}$ is completely defined by the 3×3 color transform $\tilde{\mathbf{C}}$ via the Kronecker product with an $M_1 M_2 \times M_1 M_2$ identity matrix. The luminance target may also be decomposed analogously. One way to choose a consistent matrix $\tilde{\mathbf{C}}$ is to define it based on the intersection of individual gamuts of $\tilde{\mathbf{C}}_k$, i.e. $\tilde{\mathbf{C}} = \cap_k \tilde{\mathbf{C}}_k$. The luminance target $\bar{\mathbf{L}}$ may be defined by $\tilde{\mathbf{L}} = \sum_{k=1}^K \tilde{\mathbf{A}}_k \tilde{\mathbf{L}}_k$ to maximize brightness. The target black offset may be chosen as $\bar{\mathbf{b}} = \mathbf{b}$ to maximize contrast.

4.2. Iterative Subframe Generation

In the above equations, one may observe that the $\bar{\mathbf{L}}$ is diagonal and thus commutes with $\bar{\mathbf{C}}$ in equation (2), i.e.

$$\bar{\mathbf{x}} = \bar{\mathbf{C}} \bar{\mathbf{L}} \mathbf{x} + \bar{\mathbf{b}} \quad (3)$$

If we apply the color transform $\mathbf{C}_k^{-1} (\mathcal{I}_{N_1 N_2 \times N_1 N_2} \otimes \tilde{\mathbf{C}})$ to the subframes \mathbf{y}_k , we may rewrite equation (1) as

$$\begin{aligned} \hat{\mathbf{x}} &= \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{C}_k^{-1} (\mathcal{I}_{N_1 N_2 \times N_1 N_2} \otimes \tilde{\mathbf{C}}) \mathbf{y}_k + \mathbf{b} \\ &= \sum_{k=1}^K \mathbf{A}_k (\mathcal{I}_{N_1 N_2 \times N_1 N_2} \otimes \tilde{\mathbf{C}}) \mathbf{L}_k \mathbf{y}_k + \mathbf{b} \\ &= (\mathcal{I}_{M_1 M_2 \times M_1 M_2} \otimes \tilde{\mathbf{C}}) \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{y}_k + \mathbf{b} \quad (4) \end{aligned}$$

$$= \bar{\mathbf{C}} \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{y}_k + \mathbf{b} \quad (5)$$

Equation (4) follows from the fact that the color transformation may be applied after warping to the reference canvas since this is equivalent to warping a color transformed image. Comparing equations (5) and (3), we see that it suffices

to choose subframes \mathbf{y}_k such that

$$\{\mathbf{y}_k\} = \underset{\{\mathbf{y}_k\}}{\operatorname{argmin}} \left\| \bar{\mathbf{L}}\mathbf{x} - \sum_{k=1}^K \mathbf{A}_k \mathbf{L}_k \mathbf{y}_k \right\|^2 \quad (6)$$

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

The optimization problem of equation (6) may be solved using an iterative gradient descent algorithm that may be expressed entirely in terms of localized image processing operations. The optimal subframes are computed using the following iterative gradient descent algorithm:

$$\mathbf{y}_k^{(0)} = \mathbf{L}_k^T \mathbf{A}_k^T \mathbf{x} \quad (\text{initial guess}) \quad (7)$$

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

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

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

$$\psi(\mathbf{y}(i)) = \begin{cases} \mathbf{y}(i) & \mathbf{y}(i) \in [0, 1] \\ 0 & \mathbf{y}(i) < 0 \\ 1 & \mathbf{y}(i) > 1 \end{cases}, \quad \forall i \quad (8)$$

$$\{\mathbf{y}_k^*\} = \lim_{n \rightarrow \infty} \{\mathbf{y}_k^{(n)}\} \quad (9)$$

μ is a momentum parameter indicating the fraction of error to be incorporated at each iteration. The algorithm consists of two passes. In the modeling pass, we compute $\hat{\mathbf{x}}^{(n)}$ from the current guesses of the subframes $\mathbf{y}_k^{(n)}$. Then, a correction pass 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 obtained from equation (14). Since the problem is convex with convex constraints, this process is guaranteed to converge to the optimal solution.

4.3. Fast Filter Bank Subframe Generation

The above iterative algorithm produces optimal subframes for an arbitrary number of component subframes. It is noteworthy that it does however require access to all subframes to compute the model prediction $\hat{\mathbf{x}}^{(n)}$. For systems without shared memory, the algorithm may be unsuitable for scalable real-time implementation, since each computation module must redundantly compute the entire forward pass

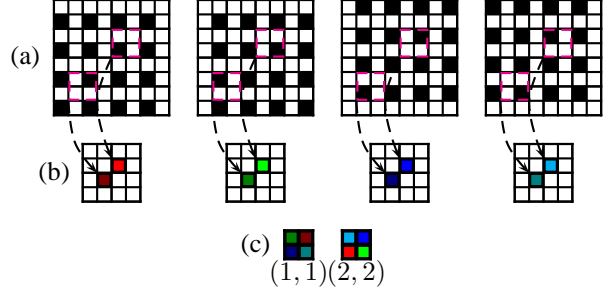


Figure 3: Deriving component subframe generation filters. A 2×2 filter neighborhood is used for convenience. (a) Training impulse subframes. (b) Subframe response at pixel locations (1,1) and (2,2). (c) Resulting per-pixel 2×2 filter coefficients.

before it can derive its subframes. This bottleneck only becomes worse as the number of subframes is increased. Further, even if shared memory is available, performing more than one or two iterations of the algorithm would be computationally infeasible in real-time.

In [5], we showed that scalable subframe generation may be accomplished by bank of linear filters. In this case, the optimal linear filters may be derived by directly finding the impulse response (possibly space varying) of a linear approximation to the nonlinear optimal iterative algorithm of Section 4.2. The linear system approximation to the nonlinear iterative algorithm of (7-9) is made by simply eliminating the nonlinear clipping operation of equation (8). The filter generation method is reviewed here for completeness.

To find the filter coefficients for the k^{th} subframe that filter the neighborhood $\mathcal{N}(\mathbf{m}_k(\mathbf{i}))$ for a given \mathbf{i} , we evaluate the response at location \mathbf{i} in the k^{th} subframe to the linearized iterative algorithm driven with impulse inputs. Successively measuring the responses at \mathbf{i} to single impulses at various locations \mathbf{j} is too time consuming to be practical. Instead, we evaluate subframe responses at multiple subframe pixels in parallel by restricting the neighborhoods to a finite $W \times W$ window while spacing impulses apart by W (See Fig. 3). Thus only W^2 impulse training images are needed to cover all the locations of $\mathcal{N}(\mathbf{m}_k(\mathbf{i}))$. This spacing eliminates spurious aliasing due to multiple impulses contributing to a single observed response at a time. The $W \times W$ filter coefficients $h_{k,\mathbf{i}}[\mathbf{w}]$ are given by

$$h_{k,\mathbf{i}}[\mathbf{w}_{t,k,\mathbf{i}}] = y_k[\mathbf{i}], \quad \forall t$$

where $\mathbf{w}_{t,k,\mathbf{i}}$ represents the location of an impulse in the $W \times W$ neighborhood $\mathcal{N}(\mathbf{m}_k(\mathbf{i}))$ for the t^{th} impulse training image. Since there are W^2 training images, each image determines a unique filter coefficient of $h_{k,\mathbf{i}}[\mathbf{w}]$. Once computed, the filter may be used to compute the response $y_k[\mathbf{i}]$

to an arbitrary input using the following equation

$$y_k[\mathbf{i}] = \sum_{\mathbf{w}} h_{k,\mathbf{i}}[\mathbf{w}] x_{\mathbf{i}}[\mathbf{w}] \quad (10)$$

where \mathbf{w} spans the $W \times W$ neighborhood of x around $\mathbf{m}_k(\mathbf{i})$.

Fig. 3 illustrates the above the process to compute a 2×2 filter for a specific subframe location of a component subframe. The space-varying filter coefficients must be stored. This may seem a daunting task requiring special parameterizations. However, if commodity graphics hardware is used for the rendering, the required texture memory is readily available.

Considering all color planes, this spatially varying filtering operation may be represented as a matrix multiplication of \mathbf{x} with a $3N_1N_2 \times 3M_1M_2$ matrix \mathbf{H} . Considering the color transformation $\mathbf{C}_k^{-1} \left(\mathcal{I}_{N_1N_2 \times N_1N_2} \otimes \tilde{\mathbf{C}} \right)$, we have a complete recipe to compute the optimal subframes from an input frame. Thus, subframe generation is accomplished by the equation

$$\mathbf{y}_k^* = \gamma_k \left(\psi \left(\mathbf{C}_k^{-1} \left(\mathcal{I}_{N_1N_2 \times N_1N_2} \otimes \tilde{\mathbf{C}} \right) \mathbf{H} \mathbf{x} \right) \right) \quad (11)$$

where $\psi(\cdot)$ represents the clipping operation and $\gamma_k(\cdot)$ represents gamma correction. This operation may be understood as spatially varying color separable filtering followed by a per-pixel 3×3 color transformation followed by gamma correction.

5. Experimental Results

We demonstrate in this section experimental results on prototype real-time superimposed display systems. The systems consist of substantially overlapping multiple “low resolution” 1024x768 projectors and displaying higher resolution content (at least 1.25 times higher in resolution) on them. A workstation fitted with multiple commodity graphics cards computes the optimal subframes and drives the projectors using custom code written in C++ and OpenGL. A computer-controlled Firewire camera is used to automatically calibrate the superimposed projectors as described in Section 3. For the discussion, it is assumed that the projectors output onto a gain 1.0 planar screen surface. Also, the projectors’ frame buffers are considered to be the aforementioned subframes.

The fast iterative algorithm from Section 4.3 is used to train and compute appropriate space-varying filters at every projector pixel location. A neighborhood of 4×4 is considered for a total of 16 impulse training images. Because of the filtering nature of the algorithm, it is efficiently implemented

on the pixel shaders of the graphics processing unit (GPU) of commodity graphics cards. It should be emphasized that the construction of these space-varying filters depends only the geometric warps and not on the actual content to be displayed.

Once the above filters have been pre-computed for a given projector configuration, they may be applied in real-time to generate the final displayed output. For a given input image, projector k ’s subframe $y_k[\mathbf{i}]$ is simply constructed by using equation (10). With this construction, the subframes are efficiently computed in parallel on GPU pixel shaders and at real-time rates (60 fps for any number of projectors).

Fig. 4 shows an actual capture of a four-projector superimposed display. The desired image (1600x1200) is higher resolution than any of the component projectors. Since comparable results are observed everywhere in the displayed image, only a portion is shown for detail. A single projector must either blur the high resolution input to prevent aliasing or else alias to display high frequencies. Figure 4(a) shows a single subframe that exhibits severe aliasing. With accurate measurement and proper subframe generation to compensate for non-uniform supersampling, an alias-free high resolution image may be rendered as shown in Figure 4(b). Because of their lower relative brightness, these two images have been contrast enhanced for better comparison. One can observe the dramatic enhancement in resolution and resolvability with the proposed algorithms. One also can mitigate the screen door artifact from typical projectors. Of course, the superimposed display naturally becomes much brighter with the addition of each projector.

Figure 5(a) shows an example superimposed output image after subframe generation, assuming that the projectors are identical. Figure 5(b) shows the effect of incorporating measured luminance. Figure 5(a) shows relatively more residual aliasing than Figure 5(b). In general, when the projectors are fairly similar these differences are subtle.

These examples highlight the feasibility and key advantages of practical multi-projector super-resolution. The prototype systems perform very efficiently using commodity GPU hardware. Geometric mappings are automatically computed in less than eight seconds per projector, and the four projectors’ subframes are generated at real-time 60 fps rates, independent of the content dimensions. In contrast to prior work [9], we believe these systems are the first superimposed displays that demonstrate super-resolution for arbitrary non-uniform sampling and at real-time rates.

6. Conclusion

We have demonstrated the viability of super-resolution via the superimposition of multiple projectors. Theoretical

questions around the possibility of super-resolution have been clearly resolved; more details on the theoretical aspects of superimposed projection may be found in [5, 6]. In addition, we have presented efficient and practical algorithms that for the first time lead to real-time super-resolution display systems. Moreover, these algorithms also factor in differences among projector color and luminance. Through resolution enhancement, we have shown that it is possible to take full advantage of the brightness and reliability of superimposed projection without compromising image quality. We believe that this work highlights some novel benefits of superimposed projection over traditional tiled projection.

Appendix

The matrix \mathbf{A}_k in equation (1) can be derived as follows. Since it applies the same processing to each color plane, it may be decomposed as $\tilde{\mathbf{A}}_k = \tilde{\mathbf{A}}_k \otimes \mathcal{I}_{3 \times 3}$ where $\tilde{\mathbf{A}}_k$ is an $M_1 M_2 \times N_1 N_2$ matrix. $\tilde{\mathbf{A}}_k$ is defined by a geometric mapping $\mathbf{m}_k(\cdot)$, a reconstruction PSF $r(\cdot)$ and a pre-filter $p(\cdot)$. Consider how a color plane l (after color transform and luminance attenuation) is geometrically mapped to the reference coordinate system and resampled. First, the projector pixels are reconstructed using a continuous reconstruction filter $r(\mathbf{u})$ by convolution:

$$z_k^{rec}(\mathbf{u}; l) = \sum_{\mathbf{i}} r(\mathbf{u} - \mathbf{i}) z_k[\mathbf{i}; l]$$

The reconstructed image is then warped to the continuous target coordinate system. The warp is implemented using an inverse warp $\mathbf{m}_k^{-1}(\cdot)$ from target to source:

$$z_k^{warp}(\mathbf{v}; l) = z_k^{rec}(\mathbf{m}_k^{-1}(\mathbf{v}); l)$$

The warped image is pre-filtered with a continuous anti-alias filter $p(\cdot)$

$$z_k^{fil}(\mathbf{v}; l) = \int_{\mathbf{t}} p(\mathbf{v} - \mathbf{t}) z_k^{warp}(\mathbf{t}; l) d\mathbf{t}$$

The continuous pre-filtered images from all component projectors are then sampled to produce a discrete output image at the target resolution:

$$\hat{x}_k[\mathbf{j}; l] = z_k^{fil}(\mathbf{j}; l)$$

The combined model incorporating the above steps is given by

$$\hat{x}_k[\mathbf{j}; l] = \int_{\mathbf{t}} p(\mathbf{j} - \mathbf{t}) \sum_{\mathbf{i}} r(\mathbf{m}_k^{-1}(\mathbf{t}) - \mathbf{i}) z_k[\mathbf{i}; l] d\mathbf{t} \quad (12)$$

Equation (12) may be simplified as:

$$\hat{x}_k[\mathbf{j}; l] = \sum_{\mathbf{i}} \rho_k[\mathbf{i}, \mathbf{j}] z_k[\mathbf{i}; l] \quad (13)$$

where the resampling filters $\rho_k[\mathbf{i}, \mathbf{j}]$ are given by:

$$\begin{aligned} \rho_k[\mathbf{i}, \mathbf{j}] &= \int_{\mathbf{t}} p(\mathbf{j} - \mathbf{t}) r(\mathbf{m}_k^{-1}(\mathbf{t}) - \mathbf{i}) d\mathbf{t} \quad (14) \\ &= \int_{\mathbf{s}} p(\mathbf{j} - \mathbf{m}_k(\mathbf{s})) r(\mathbf{s} - \mathbf{i}) \left| \frac{\partial \mathbf{m}_k(\mathbf{s})}{\partial \mathbf{s}} \right| d\mathbf{s} \end{aligned}$$

where the Jacobian $\frac{\partial \mathbf{m}_k(\mathbf{s})}{\partial \mathbf{s}}$ is introduced by a change of variables. The resampling filters may be computed based on (14) using numerical integration and stored or may be computed efficiently on the fly during mapping if the reconstruction filter and the resampling filters are approximated by Gaussian profiles [8]. Since the model of (13) represents a linear space varying filter bank, it may be represented as an equivalent discrete model using general linear operators \mathbf{A}_k encapsulating the entire resampling operation. Thus

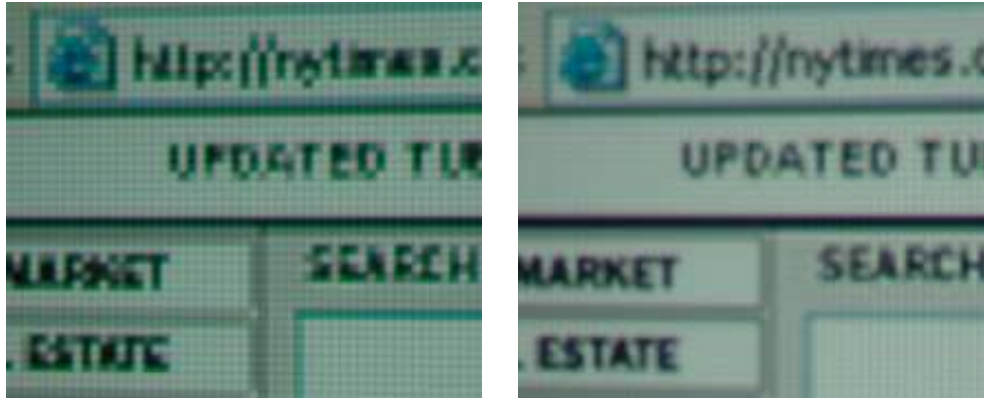
$$\hat{\mathbf{x}}_k^l = \tilde{\mathbf{A}}_k \mathbf{z}_k^l \quad (15)$$

Considering all color planes, we have

$$\hat{\mathbf{x}}_k = \mathbf{A}_k \mathbf{z}_k \quad (16)$$

References

- [1] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Trans. Pattern Anal. Machine Intell.*, 24:1167–1183, Sept. 2002.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] N. Damera-Venkata and N. L. Chang. Display supersampling. *ACM Transactions on Computer Graphics*. In submission.
- [6] N. Damera-Venkata and N. L. Chang. On the resolution limits of superimposed projection. In *Proc. IEEE International Conference on Image Processing (ICIP)*, 2007.
- [7] M. Harville, B. Culbertson, I. Sobel, D. Gelb, A. Fitzhugh, and D. Tanguay. 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.
- [8] P. Heckbert. Fundamentals of texture mapping and image warping. Master's thesis, The University of California at Berkeley, Berkeley, CA, June 1989.



(a)

(b)

Figure 4: Experimental results for four superimposed projectors: (a) Aliased subframe image from a component projector (b) Optimal image for four superimposed projectors. Note how text resolvability increases dramatically due to alias cancellation.



(a)

(b)

Figure 5: Effect of incorporating measured luminances: (a) Assuming all projectors are identical and ignoring measured luminance (b) Incorporating measured luminance. Note that there is more residual aliasing in (a). In general, the more different the projectors are the more aliasing is introduced by ignoring measured luminance. Note that if the projectors are fairly similar, measured luminance may be ignored without any visual artifacts. images ©2007 Animusic. Images used with permission (www.animusic.com).

- [9] 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.
- [10] H. Lin and H. Shum. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE Trans. Pattern Anal. Machine Intell.*, 26:83–97, Jan. 2004.
- [11] 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.
- [12] A. Majumder and R. Stevens. Perceptual photometric seamlessness in tiled projection-based displays. *ACM Trans. on Graphics*, 24:111–134, Jan. 2005.
- [13] R. Raskar, J. V. Baar, and P. Beardsley. iLamps: Geometrically aware and self-configuring projectors. *ACM Trans. on Graphics*, 22:809–818, July 2003.
- [14] R. Raskar et al. Multi-projector displays using camera-based registration. In *Proc. IEEE Visualization Conference*, pages 161–522, San Francisco, USA, Oct. 1999.
- [15] A. Said. Analysis of subframe generation for superimposed images. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 401–404, Atlanta, GA, Oct. 2006.
- [16] R. Surati. *Scalable Self-Calibrating Display Technology for Seamless Large-Scale Displays*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, Jan. 1999.