# Building Large Area Multi-Projector Displays

Eurographics 2003 Tutorial
Organizers: Michael S. Brown and Aditi Majumder

**Description**

This tutorial provides an introduction to building affordable and flexible large-area multi-projector displays. The goal of the tutorial is to impart a working knowledge of several recent techniques available for making multi-projector large area displays more affordable, flexible, and easier to deploy. We will focus on two fundamental challenges: (1) generating seamless imagery across multiple projectors and (2) low-cost image generation using PC-clusters. In particular, we will provide detail explanations of several proven camera-based geometric and photometric registration techniques and discuss their appropriateness for different display configurations. This is followed by an introduction to PC-cluster rendering approaches for large-scale displays. Finally, integration of geometric and photometric registration into this rendering framework will be presented.

**Topics**

1) Introduction
2) Challenges in Building Large Area Multi-Projector Displays
3) Geometric Registration
4) Photometric Seamlessness
5) Registration with PC Cluster Rendering
6) Conclusion

**Table of Contents** <u>**page**</u>

**Speakers Biographies**

**Michael S. Brown**, organizer and speaker, is an assistant professor of computer science at the Hong Kong University of Science and Technology. He received his BSE and PhD from the University of Kentucky in 1995 and 2001 respectively. From 1998 to 2000, he was a visiting PhD student at the University of North Carolina at Chapel Hill under the direction of Henry Fuchs. His research includes camera-based geometric registration of multi-projector displays and the integration of these approaches into PC-based rendering architectures.

**Aditi Majumder**, organizer and speaker, is an assistant professor in Department of Computer Science at University of California, Irvine. She received her BE in Computer Science and Engineering from Jadavpur University, Calcutta, India in 1996 and PhD from Department of Computer Science, University of North Carolina at Chapel Hill in 2003. She has been a research assistant with Prof. Henry Fuchs for the Office of the Future Project at the Department of Computer Science in UNC Chapel Hill from 1998-2000. She has been collaborating with Rick Stevens of Mathematics and Computer Science Division at Argonne National Laboratory from 2001-2003 in finding solutions towards achieving perceptually seamless multi-projector displays. Her research focuses on color seamlessness of projection based displays and human computer interaction with large tiled displays.

# Building Large Area Multi-Projector Displays

Aditi Majumder and Michael S. Brown

Eurographics 2003

Tutorial



## Speakers


- Aditi Majumder
  Department of Computer Science
  University of California at Irvine

- Michael S. Brown
  Department of Computer Science
  Hong Kong Univ. of Science and Technology

# Why Large Area Displays?

17” Monitor

World through a window

Limited screen real-estate

Large-area display

Large-scale imagery

More pixels

Group interaction

Vivid imagery

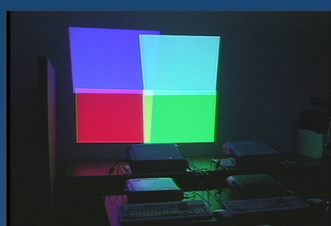


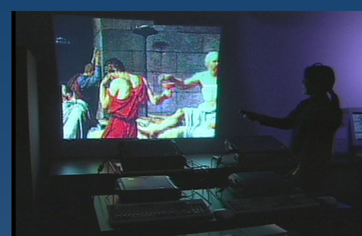

# Course Topics

- Introduction to projector-based large area displays
- Overview of challenges in building these displays
- Details to recent techniques which are making large area display deployment easier and more affordable
  - In particular, we will cover
    - » Geometric registration
    - » Photometric registration
    - » PC-Cluster rendering
- Conclusion and Q/A

# Goals

- To impart a working knowledge of recent techniques that are making the construction of large area displays easier and more affordable







# Challenges for Building Large Area Displays

Michael S. Brown

Hong Kong University of Science and Technology

## Large Area Displays

- Projector-based large area displays
  - Multiple projectors are “tiled” together to create a large viewing area
  - Provides the abstraction of a single logical display
  - Imagery must appear seamless across the projectors



## Commercial Solutions Exist

- Commercial solutions
  - CAVE, Power Wall, Domes, etc . . .
- Such displays are currently available in only a few institutions
  - Research labs, engineering/design labs
- Why are these systems not more accessible?
  - Available at libraries, schools, small businesses

## Commercial Displays Drawbacks

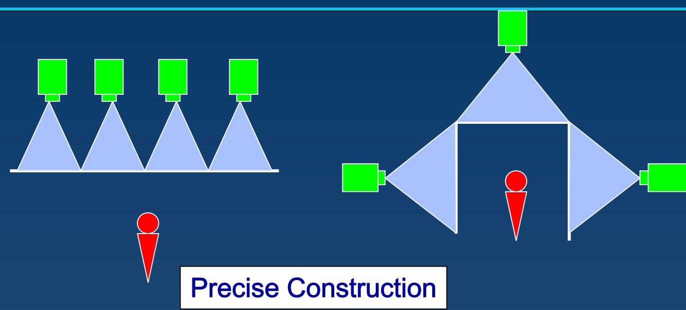
- Drawbacks
  - Price
    - » Large scale display systems are still very expensive
  - Usability
    - » Current designs require expert users
      - » For deployment
      - » For maintenance
      - » For application development
- Why?



## Physical Construction

- Rigid and precise construction of display system
  - Projectors must be in perfect alignment to produce imagery with correct geometry
  - Requires
    - » Expensive special purpose display infra-structure
    - » Expert installation
    - » Continuous maintenance
  - Adds significant cost to the display system
- High-end projectors must be used
  - To ensure color balance between and within projectors
  - Color balance among the projectors must be performed
    - » Tedious manipulation of projector parameters

# Physical Construction




Projector orientation and alignment often “hard coded” in the rendering software

Display becomes a permanent fixture in a dedicated room



# Rendering Hardware

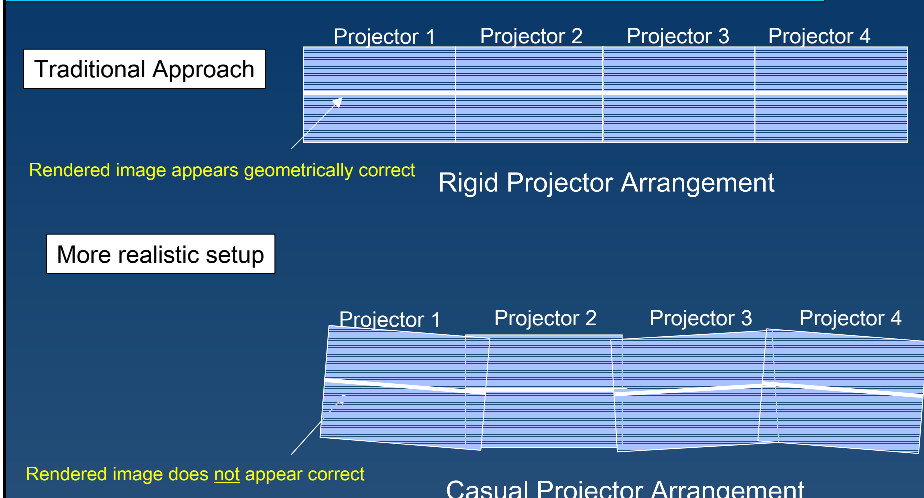
- Rendering
  - – Reliance on expensive rendering engines (Big Iron)
  - – At one time these were needed:
    - » Rendering performance
    - » Multiple, simultaneous display output
- These machines are
  - – Expensive
  - – Not easy to use
  - – Require expert administrators
  - – Often require specialized compilers, software, development environments

## Changing the way we design large scale displays

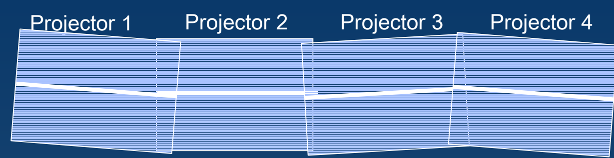
- Casual alignment
  - Reduce restrictions on projectors arrangement
  - Allow quick and flexible deployment
- Automated color balance
  - Correct intra- and inter-projector color variations
  - Correct seams in projector overlap
- PC-Cluster Rendering
  - Remove the reliance on “Big Iron”
  - Exploit the power of commodity graphics cards



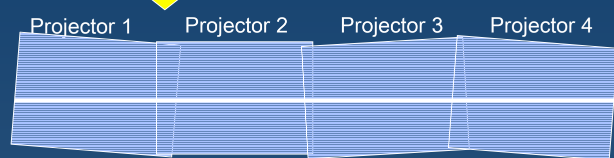
## Challenges:
### Geometric Alignment/Registration



# Challenges:
## Geometric Alignment/Registration




Must correct for projector alignment

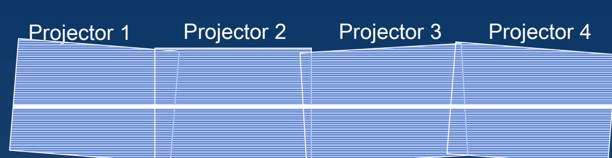



Geometry should appear seamless

Slide 15



# Alignment via image warping




- Projected framebuffers must be warped to compensate for alignment
  - "Software" alignment vs. Manual Alignment
- Not a new idea
  - Keystone correction is built into almost all projectors
    - » Warps image to compensate for a restrictive form of off-axis projection distortion
- Now we need more flexible corrective warping

Slide 16

Building Large Area Multi Projector Displays

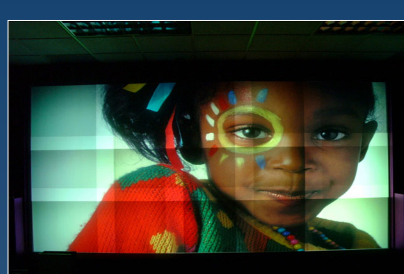
## Alignment Challenges

- How to compute the necessary warps
  - Manually? By hand?
  - Automated?  By camera?
- Applying geometric correction in real-time
  - Is there a performance hit?
  - Resolution?
- Scalable solutions
  - What is the size of the display?
  - 2, 4, 32, 64 projectors!



## Challenges: Photometric Seamlessness

- Color balance between projectors
- Color variation within a single projector
- Overlapping projectors produce visible seams
- Need to remove these artifacts



## Photometric Challenges

- How to model projector color?
- How to compute projector parameters within this model?
  - Manually?
  - Automated? By Camera?
- How to correct color variations?
- How to apply this correction in real-time?

Slide 19

Building Large Area Multi Projector Displays

## Challenges for PC Rendering

- Synchronization
  - Multiple networked PCs
  - Must act like a single logical display
  - Application transparency
- Efficiency/Performance
  - Provides sufficient rendering performance
  - Scalable
- Provide software API support
  - Not just synchronized model viewer app
  - Full API development

Slide 20

Building Large Area Multi Projector Displays

## Solutions

- Camera-based Geometric and Photometric Registration
  - Cameras used to observed projected imagery
    - » Register display geometric
    - » Compute corrective warps
    - » Compute color and intensity matching
- PC-Cluster Rendering
  - Effective PC cluster systems for rendering
    - » Humphreys' Chromium/WireGL
    - » VR Juggler

Slide 21

Building Large Area Multi Projector Displays

## Putting these together

- Geometric and Photometric Registration
  - Flexible display deployment
  - Easier setup and maintenance
- PC cluster rendering solutions
  - *Significantly* cheaper rendering
  - Scalability
- Combination of these techniques/systems
  - Accessible Large Area Displays
  - Affordable, Flexible
    - » Accommodates a variety of users with various budgets, applications, needs, expertise, . . .

Slide 22

Building Large Area Multi Projector Displays

## What do we get by solving these challenges?

- Much easier deployment
  - Even a novice user can setup the display
  - Universities, schools, libraries, businesses, tradeshows
- Less restrictive display infrastructure
  - Can just project on existing wall
  - Suitable for temporary venues
- Ability to use cheaper commodity projectors
- Flexibility
  - Different quality configurations for different budgets
  - Bottleneck becomes price of projectors
    - » Not display infrastructure, installation cost, set up



# Geometric Registration

Michael S. Brown
Hong Kong University of Science and Technology

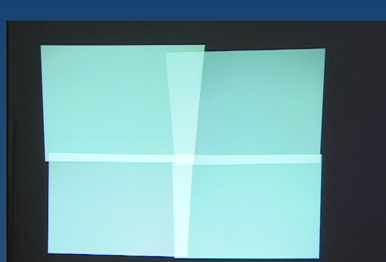
## Problem: Manually tiled projectors

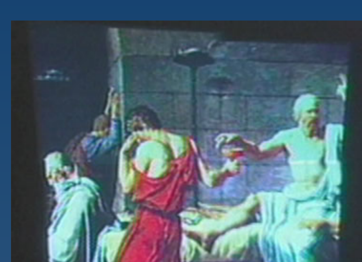
- Projector tiling must be performed very accurately
  - Small discontinuities are noticeable
  - Break in geometry is distracting for viewer
- Manual projector alignment is very difficult
  - Typically requires special mounting hardware
  - Can take hours to perform
  - Limits the type of display surface



## Solution: "Software Alignment" or Geometric Registration

- Instead of accurate alignment
- Allow casual alignment
  - Perform corrective warping
  - Sometimes called "software alignment"
  - We refer to this as "geometric registration"



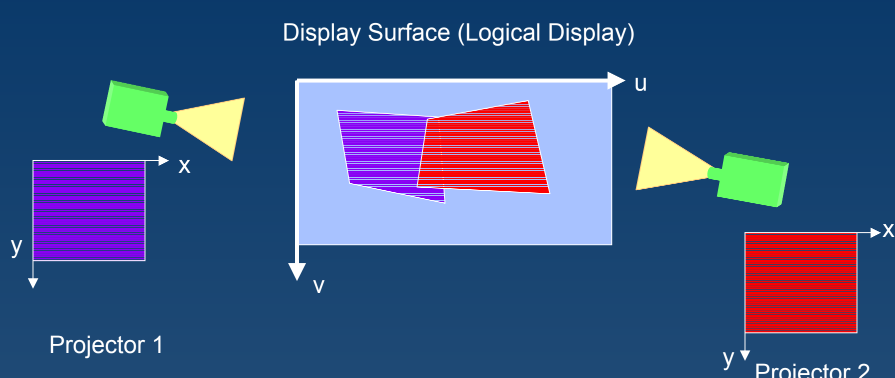


# Geometric Registration

- Two parts
  - Registration
    - » Register individual projectors' geometry into a "common coordinate frame"
    - » Compute the necessary warps to compensate for the alignment
  - Warping
    - » Apply the appropriate warp before projecting the image (post-render warp)
    - » Typically a 2-pass procedure

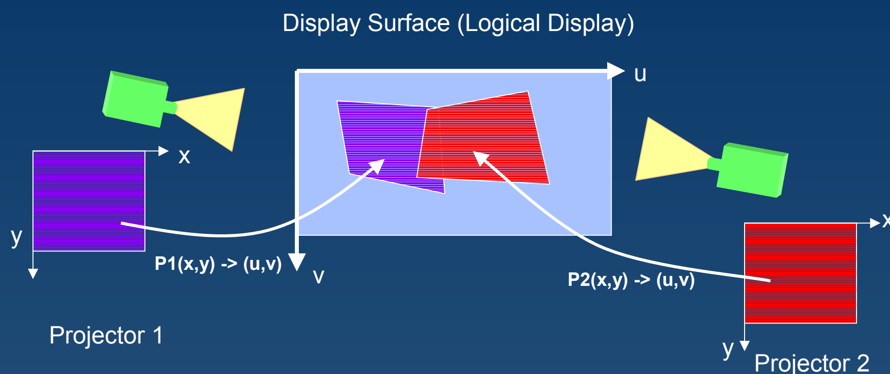


# Registering Projectors in a Common Coordinate Frame




We need to determine the contribution of each individual projector in some common coordinate frame.

# Find Projected Pixels Mapping




Find a mapping between each projector's pixel (x,y) and the common coordinate frame (u,v)



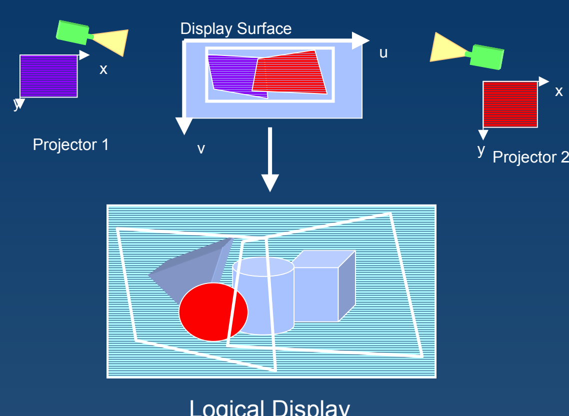
# After Registration

Map common coordinate system to a "logical" display buffer.

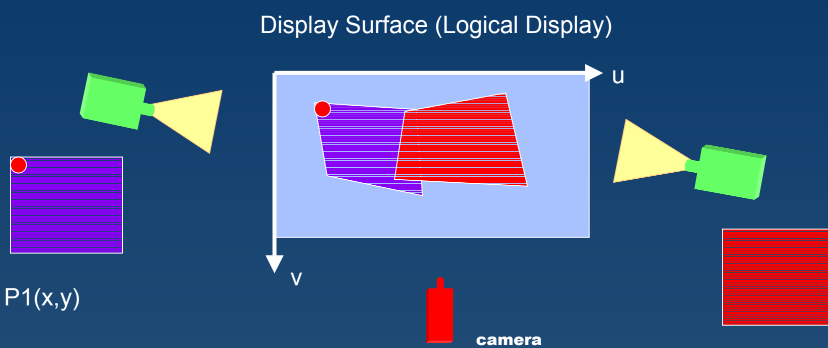
We now know how each projector's pixels maps to the logical display buffer.

If we render the logical display buffer, we can map (warp) it back to the projector's framebuffer. (Discussed in detail in "Corrective Warping" slides)

Projecting this "warped" image will create the desired logical framebuffer on the display surface.



# Automating the Registration: Use a camera to observe features




Illuminate a pixel(x,y) in projector 1.

Observe its location in the coordinate frame using a camera.
Repeat this process for all projector pixels.



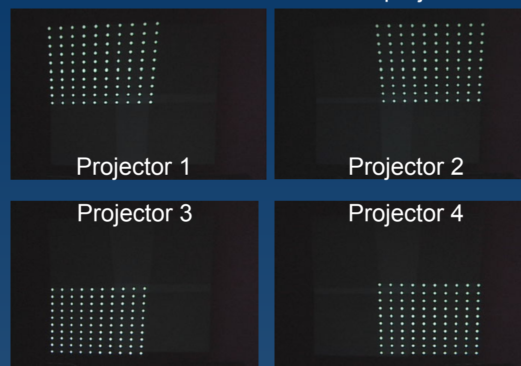


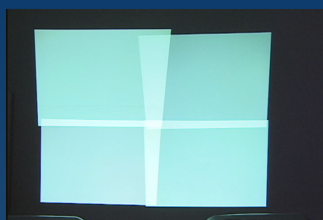
# Registration in practice

- Too hard/slow to observe every projector pixel
- Instead detect projected features
  - Gaussian circles
  - Corners of checker board patterns
  - Line intersections
- Interpolate pixels using the sampled features
- Tools available
  - Camera
  - Image processing routines

# Example

Gaussian blob features from four projectors



Projectors Contribution

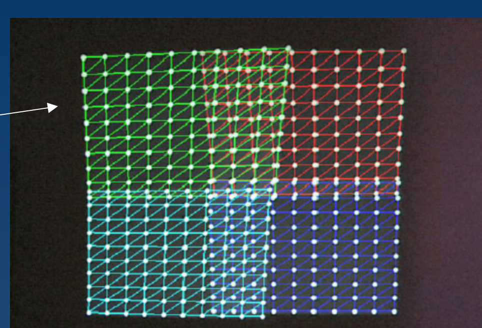
Slide 33

Building Large Area Multi Projector Displays

# Sample Features

Construct a tessellated mesh from the sampled features.

Use this to guide the corrective warping.



Slide 34

Building Large Area Multi Projector Displays

## Registration Application

- App synchronizes projected imagery and camera
  - Projector projects a feature
  - Observe this feature with the camera
    - » Can use techniques to speed up the process
      - » Binary-structured lighting techniques
      - » Colored features
      - » etc
- Output is mapping between projector and logical display

Slide 35

Building Large Area Multi Projector Displays

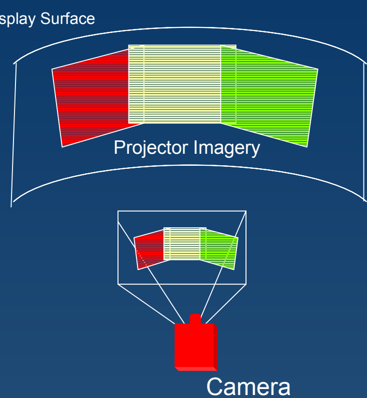
## Coordinate Frames

- Use a coordinate defined on the display surface
  - Especially useful for planar surfaces
  - Approach taken by Princeton's Scalable Display Wall
    - » Requires the camera to be "pre-registered" to the defined coordinate frame
- Use the camera's image plane as a coordinate frame
  - Does not require a planar display surface
  - Camera must be placed at the desired viewing location
  - Camera should be distortion free

Slide 36

Building Large Area Multi Projector Displays

# Registration using the camera image plane.




Camera image plane serves as a Plenoptic sample of the light rays coming from the display to the viewer's eye.

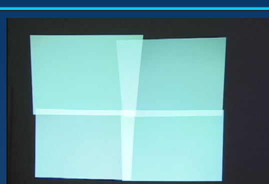
Does not require the display surface to be known.

*Assumes the camera is positioned at the desired viewing location.*

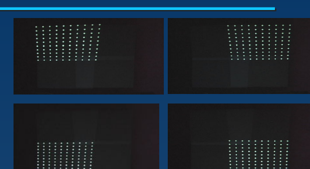
*Assumes distortion free camera.*



# Example



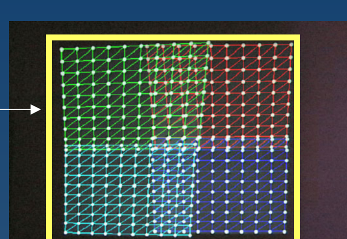
Four Projectors



Features Observed by the Camera

Logical Display Buffer →

(Use the bounding box of the observed features in the camera image plane)



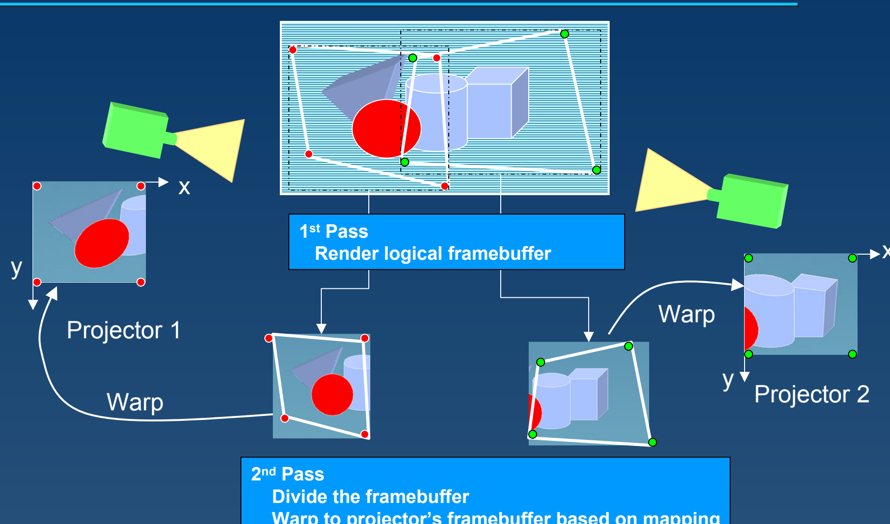
## Corrective Warping

- Two-Pass Rendering Algorithm
  - 1st Pass
    - » Render the “desired” image
      - » i.e. projector’s portion of the logical framebuffer
  - 2nd Pass
    - » Warp the desired image based on the to the projectors framebuffer mapping to the logical framebuffer

Slide 39



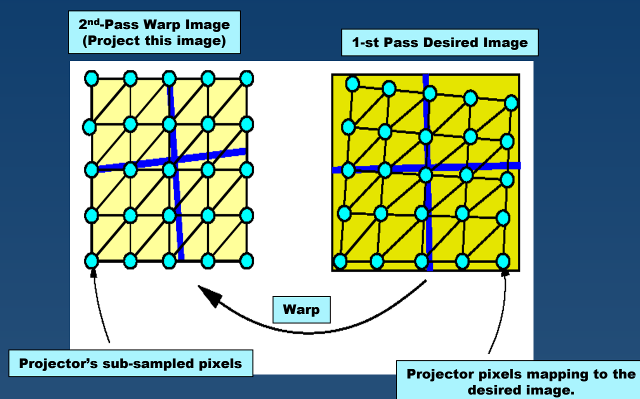
## Corrective Warping




Slide 40

# Warping in Practice

•A triangulated mesh of the sampled features is used.
•Warping is just piece-wise texture mapping of these triangles.




Slide 41

Building Large Area Multi Projector Displays

# Piecewise Warping

- Piecewise Warping is performed by texture mapping
  - Texture map triangles between desired image and projected image
- This can be considered a non-linear warp
  - Compensates for non-planar display surfaces
  - Compensates for projector distortion (lens distortion)

Slide 42

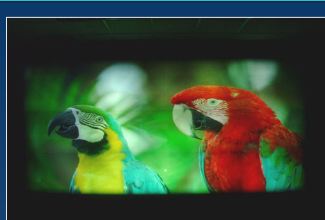
Building Large Area Multi Projector Displays

# Correction in a single pass

- Modify view frustum to reflect mapping into display surface
- Often difficult to do
  - –Assumes no non-linearities in the rendering system
    - » I.E. no lens distortion, completely planar display surface



# Examples



15 projectors on planar surface

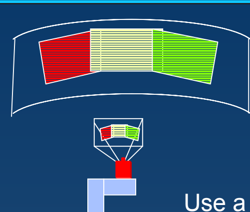

3 projectors on curved surface

## Scalability

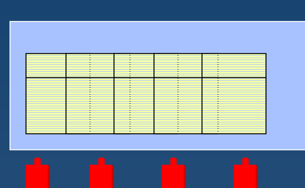
- Single camera needs to observe entire display
  - This limits the number of projectors that can compose the display
- Use Multiple Cameras
  - Works on Planar Display Surface
    - » H. Chen's homography tree
    - » Y. Chen's pan-tilt unit camera
- Still a challenging/open problem



## Scalability Strategies



Use a Pan-Tilt Unit to Move Camera Around
Build a "Mosaiced" Image



Use multiple cameras with overlapping views
Relate camera's together with homography
Build a "Mosaiced" Image

## Registration Accuracy

- Sub-pixel registration of the projector pixels is possible in the virtual display
- Dependent on
  - Camera resolution
  - Accuracy of feature detection algorithms
- Can adjust projected feature size when lower-resolution camera is used
  - Larger size easier to detect
  - May lose some resolution around the borders



## Summary

- Geometric Registration
  - Allow for casual alignment of projectors
  - Correct misalignment in “software”
- Use a camera to automate procedure
  - Registers projected pixels to a common coordinate frame
  - Apply post-render warp to imagery to construct the corrected image

# References

[1] M. S. Brown, et al "A practical and flexible tiled display system", *IEEE Pacific Graphics*, 2002

[2] H. Chen et al. "Scalable Alignment of Large-Format Multi-Projector Displays Using Camera Homography Trees" *IEEE Visualization 2002*

[3] Y. Chen et al. "Automatic Alignment Of High-Resolution Multi-Projector Displays Using An Un-Calibrated Camera." *IEEE Visualization 2000*

[4] R. Raskar et. al "A low-cost projector mosaic with fast registration", *ACCV 2002*

[5] R. Raskar et. al "Efficient image generation for multiprojector and multisurface displays", *Eurographics Workshop, 1998*

[6] R Surati, "Scalable Self-Calibration Display Technology for Seamless Large Scale Displays", PhD thesis, Department of Computer Science, *Massachusetts Institute of Technology*, 1999

[7] R. Yang et al, "Pixeflex: A dynamically configurable display system", *IEEE Visualization 2001*



# Photometric Seamlessness in Multi Projector Displays

Aditi Majumder

University of California at Irvine

## The Problem

Even with perfect geometric alignment, color variation breaks the illusion of a single seamless display

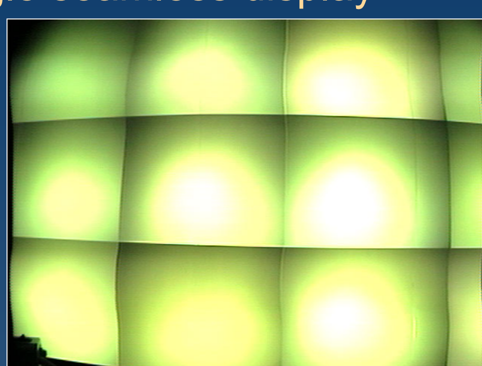
*At ANL*

Slide 51

Building Large Area Multi Projector Displays

## The Problem

Even with perfect geometric alignment, color variation breaks the illusion of a single seamless display



*At ANL*

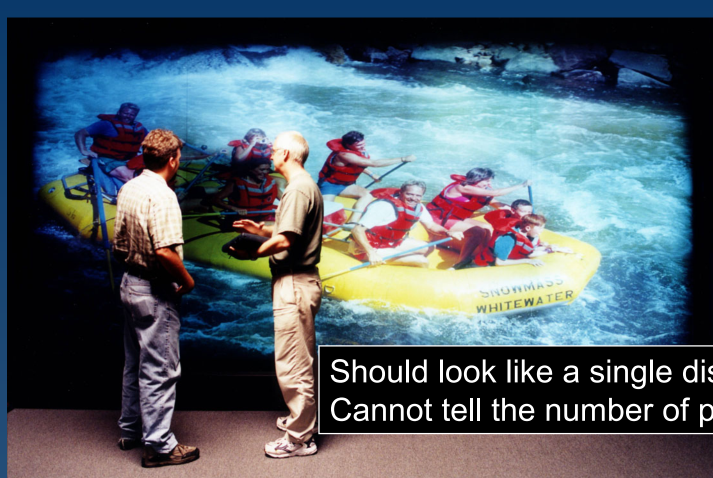
Slide 52

Building Large Area Multi Projector Displays

## The Goal



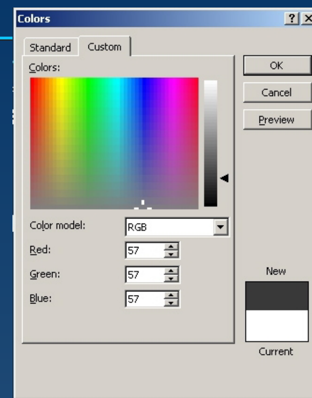

## The Goal




Should look like a single display
Cannot tell the number of projectors

## Background: Color

- Perceptual Representation
  - Luminance or Radiance $(L)$
    - » Brightness
  - Chrominance ( $x$ , $y$ )
    - » Hue and Saturation




- Digital Representation
  - Three channels (Red, Green, Blue)



## Why Is It Difficult?

- No comprehensive model of color variation
- No formal definition of *color seamlessness*
- The problem is inherently five dimensional
  - Color (3D – 1D luminance and 2D chrominance)
  - Display surface (2D)
- Humans are more sensitive spatial variations than to temporal variations in color

## Organization

- Properties of Color Variation
- Previous Work
- Achieving Photometric Seamlessness
- PRISM: Implementation
- Results

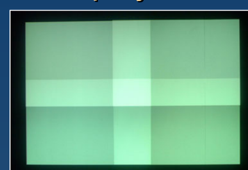


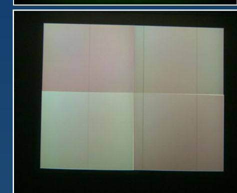
## Organization

- Properties of Color Variation
- Previous Work
- Achieving Photometric Seamlessness
- PRISM: Implementation
- Results

# Properties of Color Variation

- Intra-projector
  – Within a single projector
- Inter-projector
  – Across different projectors
- Overlaps





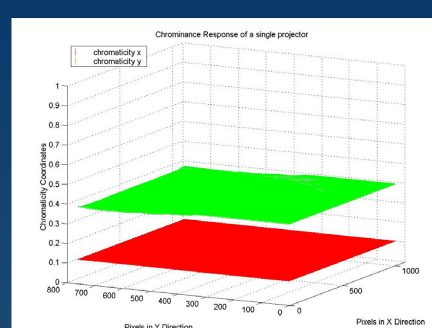
References:
•A. Majumder, Properties of Color Variation in Multi Projector Displays, Proceedings of SID Eurodisplay, 2002.

•A. Majumder and Rick Stevens, Color Nonuniformity in Multi Projector Displays: Analysis and Solutions, IEEE Transactions on Visualization and Computer Graphics, 2003 (To appear).



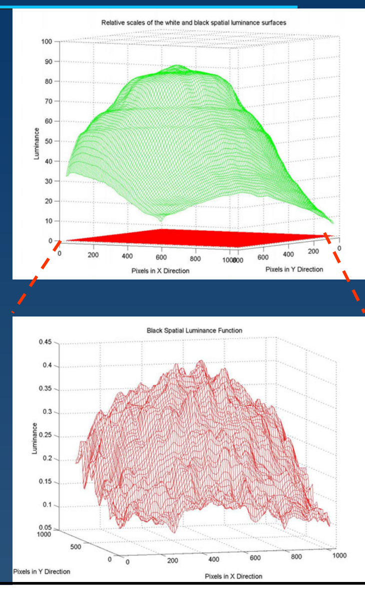
# Intra-Projector Variations

- Chrominance is constant



## Intra-Projector Variations

- Luminance is not
- Black Offset
  - Always present

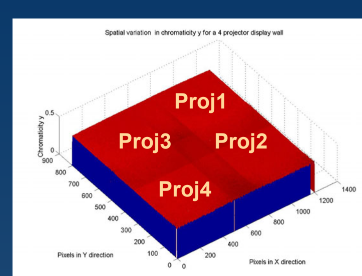



Luminance variation is more significant than chrominance variation

Slide 61

## Inter-Projector Variations

- Projectors of same model
  - Chrominance variation is negligible
  - Luminance variation is significant
- Projectors of different models
  - Chrominance variation is relatively very small
  - Luminance variation is significant



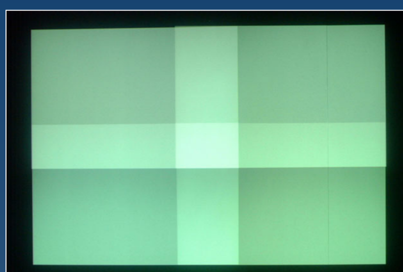

Chrominance $(x)$ of a four projector display

Luminance variation is more significant than chrominance variation

Slide 62

## Overlaps

- For displays made of same model projectors, at overlap regions
  - Chrominance remains almost constant
  - Luminance almost gets multiplied by the number of overlapping projectors



Luminance variation is more significant than chrominance variation

Slide 63

Building Large Area Multi Projector Displays

## Organization

- Properties of Color Variation
- Previous Work
- Achieving Photometric Seamlessness
- PRISM: Implementation
- Results

Slide 64

Building Large Area Multi Projector Displays

## Previous Work

- Controls Manipulation
- Common Bulb
- Blending
- Gamut Matching



## Previous Work

- Controls Manipulation
- Common Bulb
- Blending
- Gamut Matching

## Control Manipulation

- Manipulating Projector Controls
  - Manually  or computer controlled
- Sensor
  - Eye or camera
- Shortcomings
  - Only inter projector variations
  - Time consuming and labor intensive
  - Not scalable to 40-50 projectors



## Previous Work

- Controls Manipulation
- Common Bulb
- Blending
- Gamut Matching

## Common Bulb

- Using common bulb for all projectors
- Shortcomings
  - Bulb is not the only cause of color variation
  - Not scalable
  - Labor intensive ($100,000 for 3x3 display)
  - Addresses only inter projector variation

References:

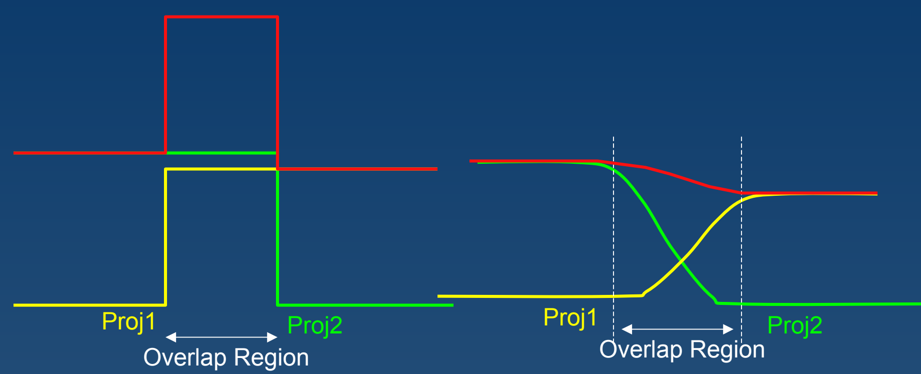
• B. Pailthorpe, N. Bordes, W. Bleha, S. Reinsch, and J. Moreland, *High-resolution display with uniform illumination,* Proceedings Asia Display IDW, 1295-1298, 2001.



## Previous Work

- Controls Manipulation
- Common Bulb
- Blending
- Gamut Matching

## Edge Blending






## Edge Blending

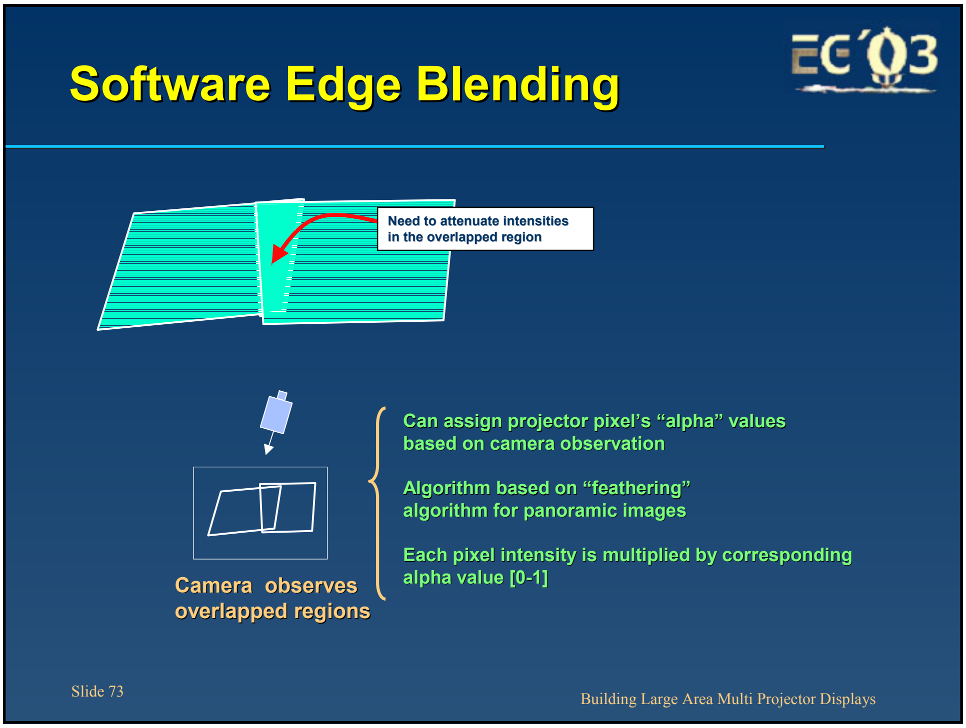
- Software
- Aperture Masking
- Hardware

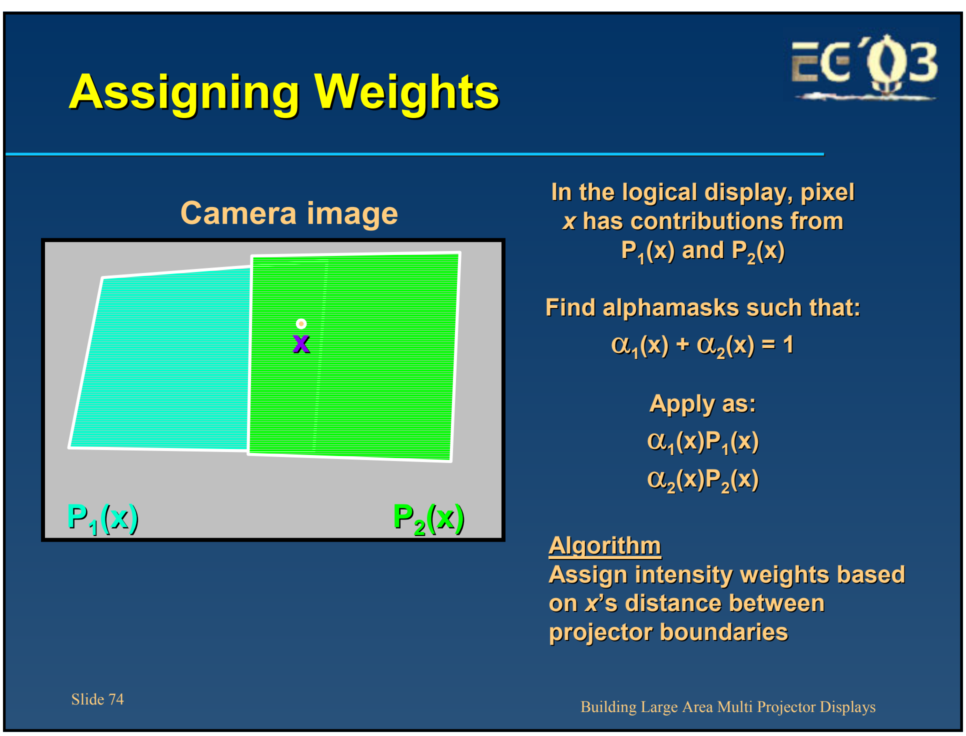
References:

• R. Raskar et al, The Office of the Future: A unied approach to image based modeling and spatially immersive display, Proceedings of ACM Siggraph, 168-176, 1998.

• K. Li et.al, Early experiences and challenges in building and using a scalable display wall system, IEEE Computer Graphics and Applications 20(4), 671-680, 2000.

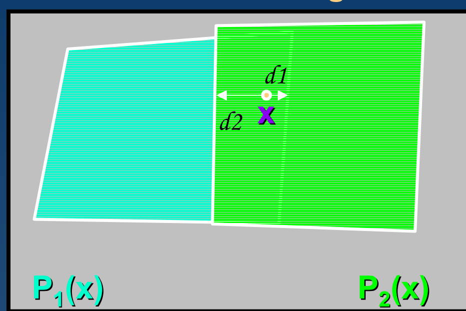
• C.J. Chen, and M. Johnson , Fundamentals of scalable high resolution seamlessly tiled projection system, Proceedings of SPIE Projection Displays VII 4294, 67-74, 2001.


Software Edge Blending
EG'03
Need to attenuate intensities in the overlapped region
Can assign projector pixel's "alpha" values based on camera observation
Algorithm based on "feathering" algorithm for panoramic images
Each pixel intensity is multiplied by corresponding alpha value [0-1]
Camera observes overlapped regions
Slide 73
Building Large Area Multi Projector Displays


Assigning Weights
EG'03
Camera image
x
$P_1(x)$
$P_2(x)$
In the logical display, pixel *x* has contributions from $P_1(x)$ and $P_2(x)$
Find alphamasks such that:
$\alpha_1(x) + \alpha_2(x) = 1$
Apply as:
$\alpha_1(x)P_1(x)$
$\alpha_2(x)P_2(x)$
Algorithm
Assign intensity weights based on *x*'s distance between projector boundaries
Slide 74
Building Large Area Multi Projector Displays

# Assigning Weights

Camera image



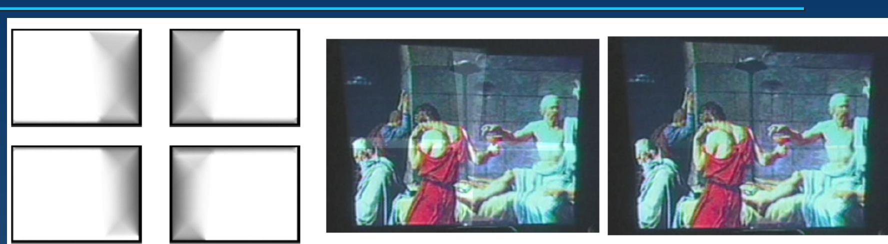

*d1* = x's distance to **P**$_1$'s boundary
*d2* = x's distance to **P**$_2$'s boundary

$$\alpha_1(x) = \frac{d_1}{d_1 + d_2}$$

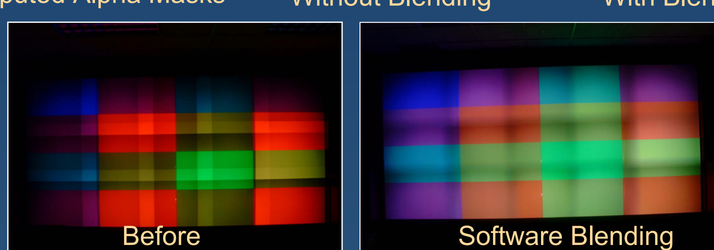
$$\alpha_2(x) = \frac{d_2}{d_1 + d_2}$$



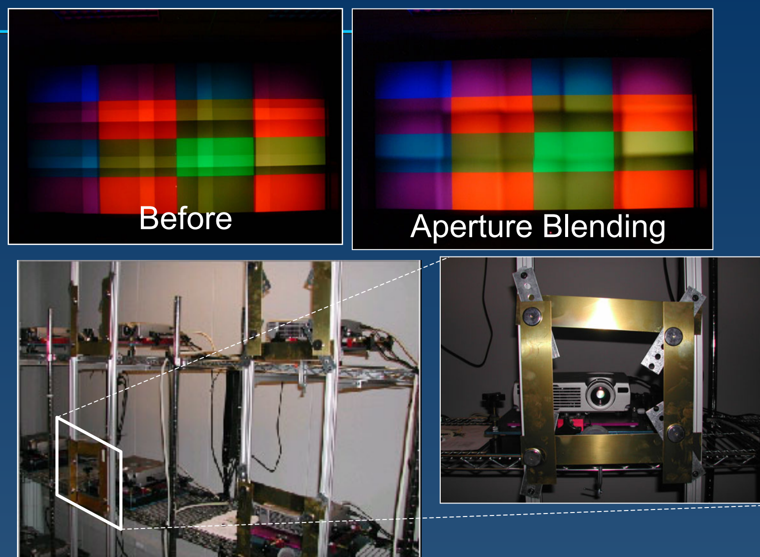
# Software Edge Blending Results



Computed Alpha Masks | Without Blending | With Blending



# Aperture Edge Blending






# Blending

- Only overlap regions
- Aperture Masking
  - Not enough control
  - Rigid
- Assumes linearity of projector response
- Cannot get rid of seams entirely

## Previous Work

- Controls Manipulation
- Common Bulb
- Blending
- Gamut Matching



## Gamut Mapping

- Addresses only inter projector variation
- Not scalable to 40-50 projectors
  - Due to algorithmic complexity

References:

• M.C. Stone, Color balancing experimental projection displays, 9th IS&T/SID Color Imaging Conference, 2001.

• M. C. Stone, Color and brightness appearance issues in tiled displays, IEEE Computer Graphics and Applications, 2001.

• G. Wallace, H. Chen, and K. Li, Color gamut matching for tiled display walls, Immersive Projection Technology Workshop, 2003.

• M. Bern and D. Eppstein, Optimized color gamuts for tiled displays, ACM Computing Research Repository, cs.CG/0212007, 19th ACM Symposium on Computational Geometry, San Diego, 2003, to appear.

## Previous Work

- Time consuming/ Rigid
- No automation
- Not scalable
- Addresses parts of the problem only
  - –Blending : Overlaps
  - –Others: Inter Projector Variations

Uniformity in Response



## Achieving Seamlessness

- To correct, first capture
- Complexity of capture
  - –Input color space : 24 bit color
  - –Need $2^{24}$ images
- Reduce complexity by modeling projector color variations

Emineoptic Function

## Emineoptic Function

- Models color variation
- Provides a framework for correction
- Provides formal definition of color seamlessness
  - –Optimization Problem
- All algorithms can be derived from it

References:

• A. Majumder and Rick Stevens, Identifying and Optimizing the Emineoptic Function for Color Seamlessness in Multi Projector Displays, Argonne National Laboratory Technical Report #260, 2003.



## Photometric Seamlessness

- Most display walls made of same model projectors
  - –Spatial variation in chrominance is negligible
- Humans are more sensitive to luminance variation than to chrominance variation
- Addresses only luminance

## Our Contribution

- Photometric uniformity is not required for photometric seamlessness
- Automation
- Unified way of solving for all kinds of luminance non-uniformities
  - » Intra-projector spatial variation
  - » Inter-projector spatial variation
  - » Overlaps
- Real-time Correction



## Organization

- Properties of Color Variation
- Previous Work
- Achieving Photometric Seamlessness
- PRISM: Implementation
- Results

# Let us assume…

- No black offset
- Projectors have identical gamma
- Projectors are linear devices

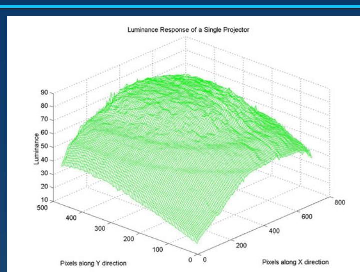
Gamma function is a straight line between zero and one for all projectors

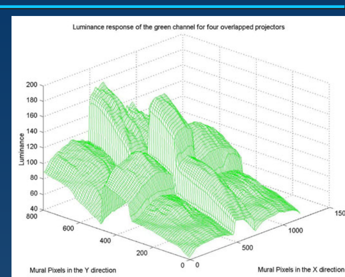
Slide 87

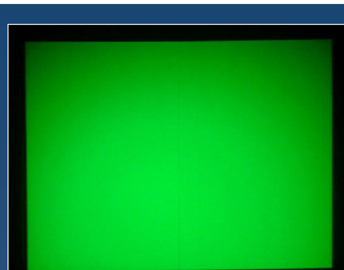


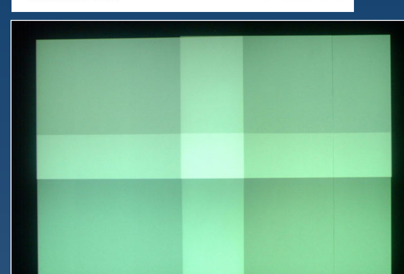
# Why do we see seams?

Single Projector Display





Four Projector Display

Slide 88

## Why do we see seams?

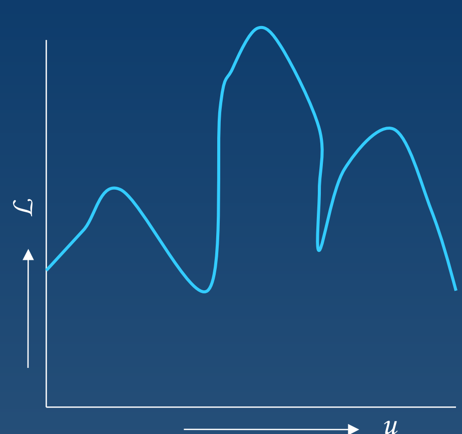
- Sharp discontinuities are the cause of photometric seams
- Remove the sharp discontinuities

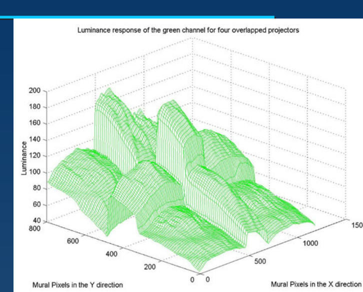
Slide 89



## The Problem

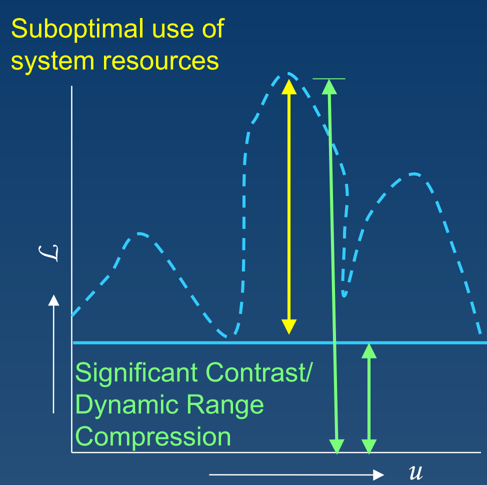
Maximum Channel Luminance

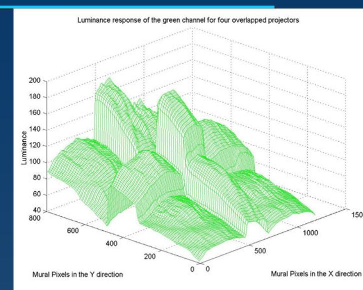


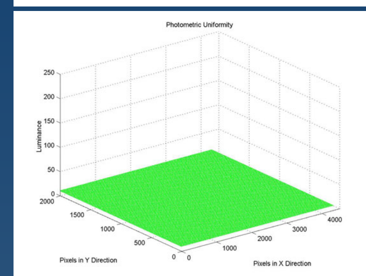




Slide 90

# Photometric Uniformity



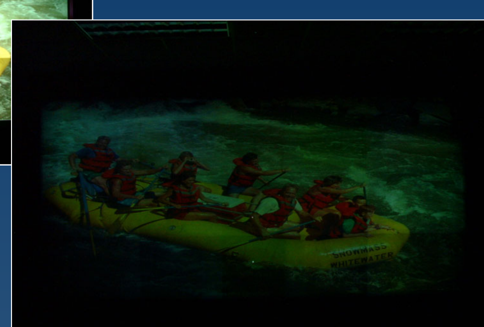






# Photometric Uniformity : Display Quality

Which one is better?

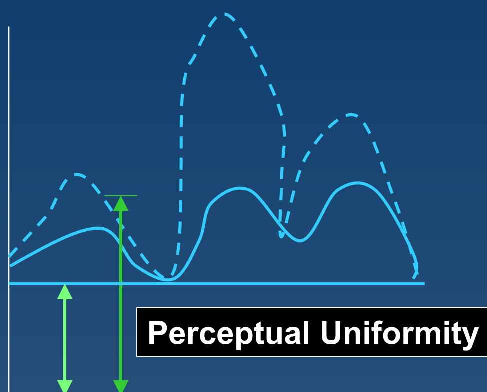


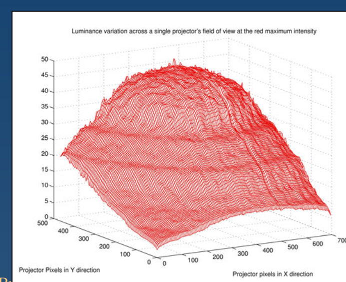
Reference:

- A. Majumder and Rick Stevens, LAM: Luminance Attenuation for Photometric Seamlessness in Projection Based Displays, Proceedings of Virtual Reality Software and Technology, 2002.

## Is Photometric Uniformity Required?

Humans cannot detect smooth luminance variations



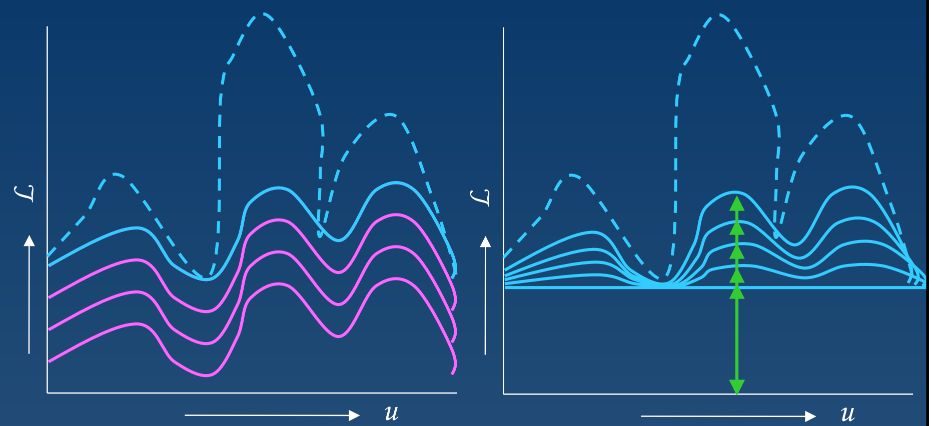



## Achieving Photometric Seamlessness

- Optimization Problem
  - –Perceptual Uniformity
    - » Creates the *perception* of uniformity
  - –Display Quality
    - » Maintains *high* display quality


References:
•A. Majumder and Rick Stevens, Identifying and Optimizing the Emineoptic Function for Color Seamlessness in Multi Projector Displays, Argonne National Laboratory Technical Report #260, 2003.

## Optimization Problem




Photometric uniformity is a special case of perceptual uniformity

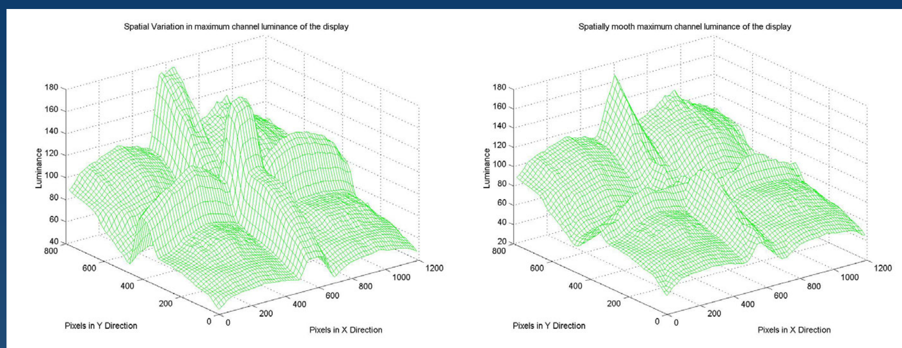


## Constrained Gradient Based Smoothing Algorithm

- Perceptual Uniformity Constraint
  - Gradient Based Linear Constraint
- Display Quality
  - Maximize a Linear Objective Function
- Linear Programming
- Alternate Iterative Algorithm

## Constrained Gradient Based Smoothing

*2 x 2 array of four projectors*



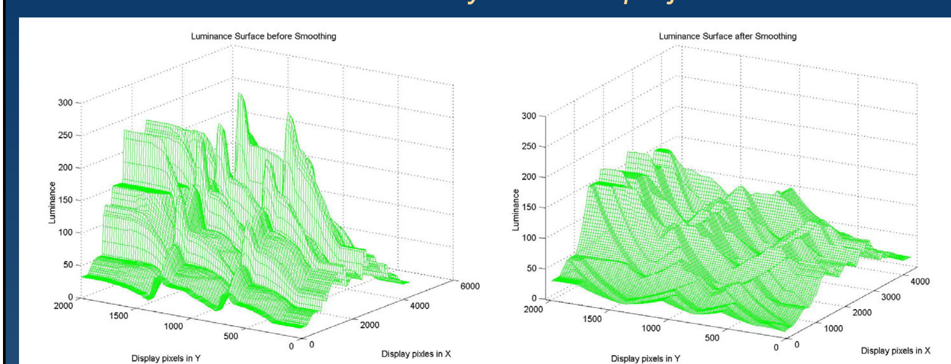

*Before smoothing* *After smoothing*

Slide 97



## Constrained Gradient Based Smoothing

*5 x 3 array of fifteen projectors*




*Before smoothing* *After smoothing*

Slide 98

## Photometric Uniformity

Slide 99

Building Large Area Multi Projector Displays

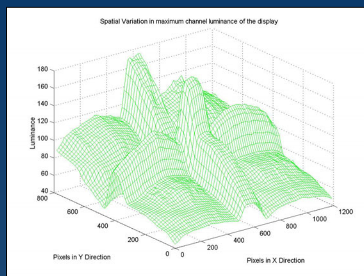
## Photometric Seamlessness

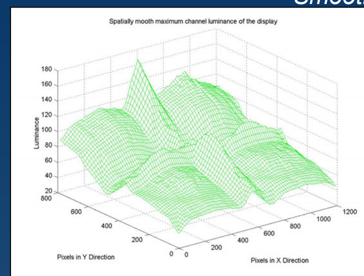
Slide 100

Building Large Area Multi Projector Displays

# Different Smoothing Parameter (2x2 array of four projectors)

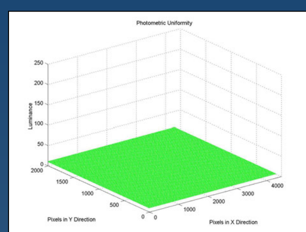
Smooth

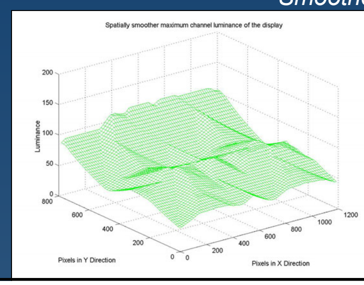



Original

Smoother



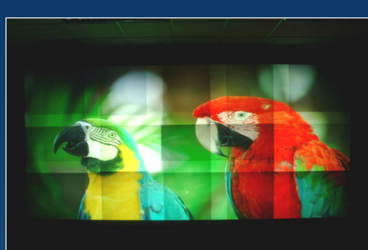



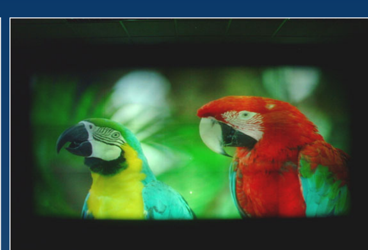

Flat

Slide 101

# Different Smoothing Parameter (3x5 array of 15 projectors)

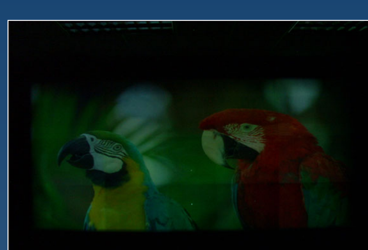
Smooth

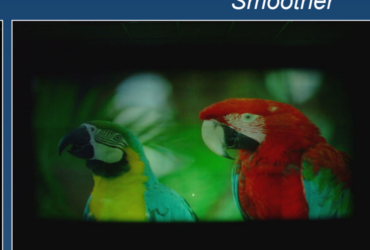




Original

Smoother





Flat

Slide 102

## Organization

- Properties of Color Variation
- Previous Work
- Achieving Photometric Seamlessness
- PRISM: Implementation
  - –Perceptual Radiance Seamlessness in Multi-projector Displays
- Results



## PRISM: Overview

- Off-line Calibration
  - –Repeated periodically
- Online Image Correction
  - –Applied in real-time to all images displayed on the wall

References:

•A. Majumder and Rick Stevens, Identifying and Optimizing the Emineoptic Function for Color Seamlessness in Multi Projector Displays, Argonne National Laboratory Technical Report #260, 2003.

## PRISM: Overview

- Off-line Calibration
- Online Image Correction



## PRISM: Calibration

- Reconstructing the display luminance surface *automatically*
  - –Using a inexpensive digital camera
- Generating the smooth luminance surface
- Encoding the correction as a luminance attenuation map

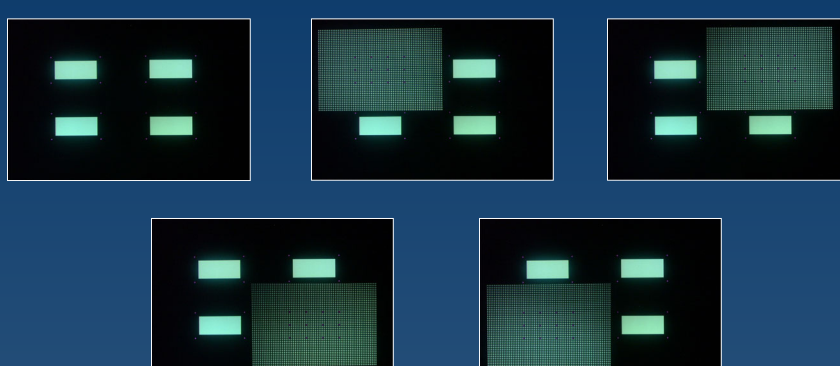
## PRISM: Calibration

- Reconstructing the display luminance surface *automatically*
- Generating the smooth luminance surface
- Encoding the correction as a luminance attenuation map



## Reconstruction of Display Luminance Surface

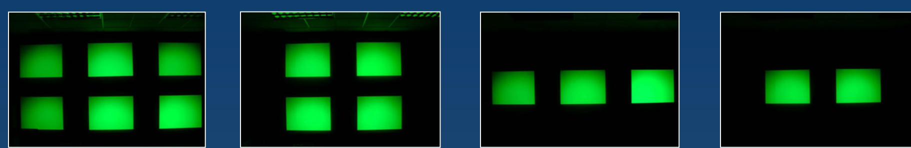
- Find geometric correspondences



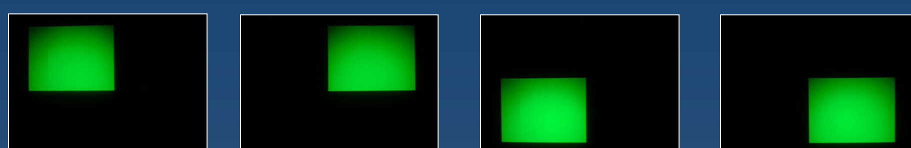
*Four projector wall*

# Reconstruction of Display Luminance Surface

- Need to click four pictures per channel



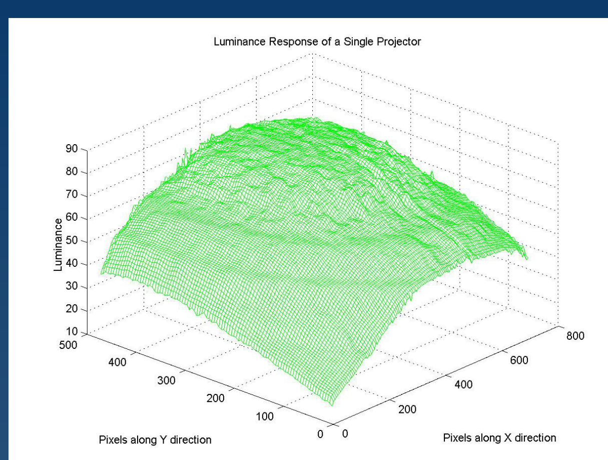
*15 projector wall*



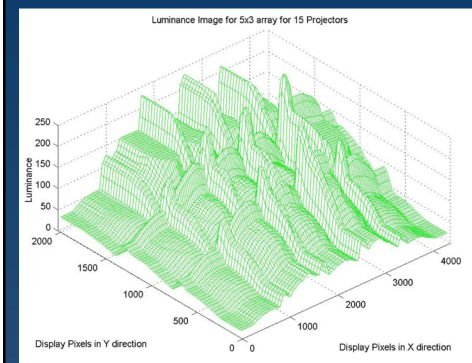
*Four projector wall*



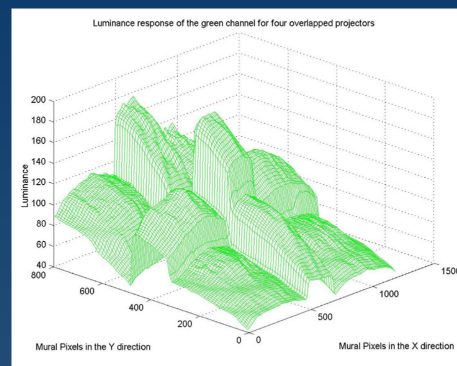
# Projector Luminance Surface



## Display Luminance Surface




*15 projector wall*




*Four projector wall*

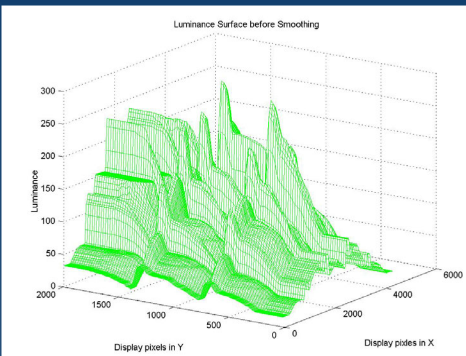


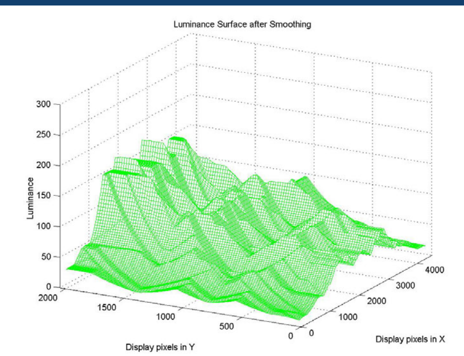
## Calibration

- Reconstructing the display luminance surface *automatically*
- Generating the smooth luminance surface
- Encoding the correction as a luminance attenuation map

## Generate Smooth Luminance Surface

- Using linear programming
- Or, alternate iterative algorithm




Slide 113

Building Large Area Multi Projector Displays

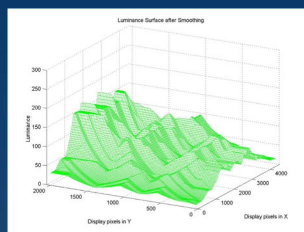
## Calibration

- Reconstructing the display luminance surface *automatically*
- Generating the smooth luminance surface
- Encoding the correction as a luminance attenuation map

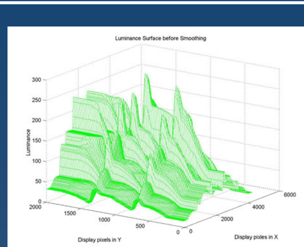
Slide 114

Building Large Area Multi Projector Displays

# Generation of Attenuation Map




Smooth Luminance Surface



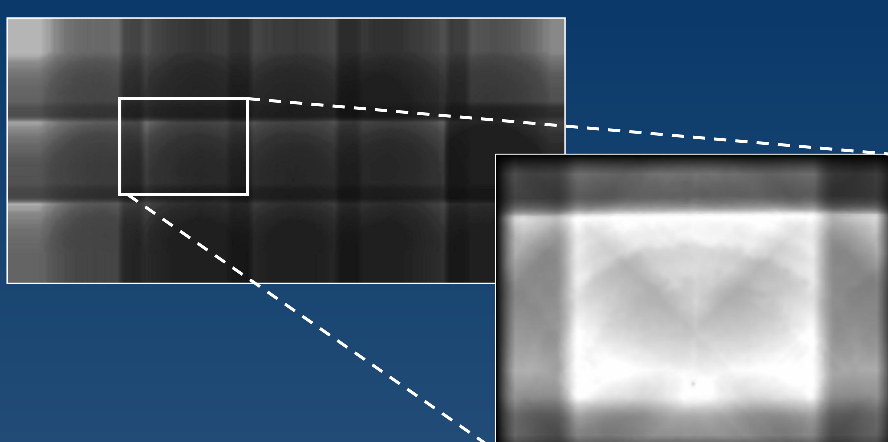

Reconstructed Luminance Surface

=

Display Attenuation Map



# Projector Attenuation Map

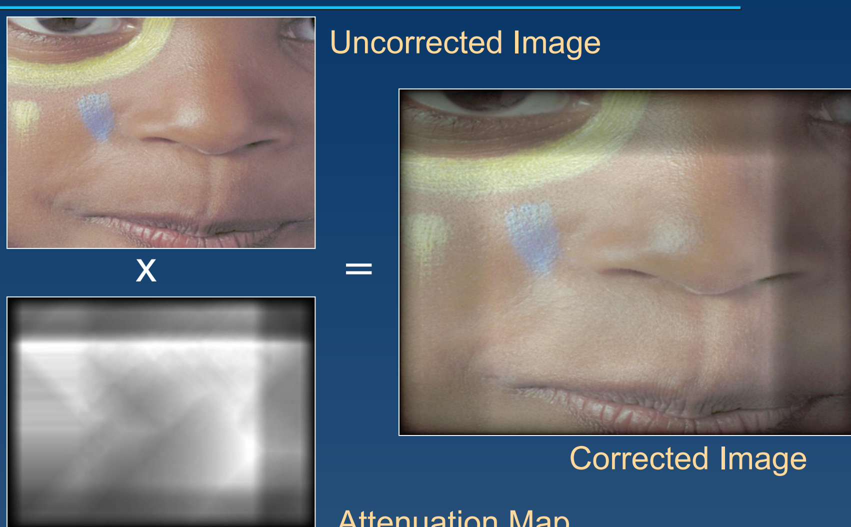


# PRISM: Overview

- Off-line Calibration
- Online Image Correction



# PRISM: Image correction



## Let us assume…

- No black offset
- Projectors have identical gamma
- Projectors are linear devices

Gamma function is a straight line between zero and one for all projectors

Slide 119



## Let us assume…

- No black offset
- Projectors have identical gamma
- Projectors are linear devices
  - Convert the image to be put up to linear space
    - » Common gamma function
  - Convert the image back to non linear space of each projector
    - » Projector dependent inverse gamma function

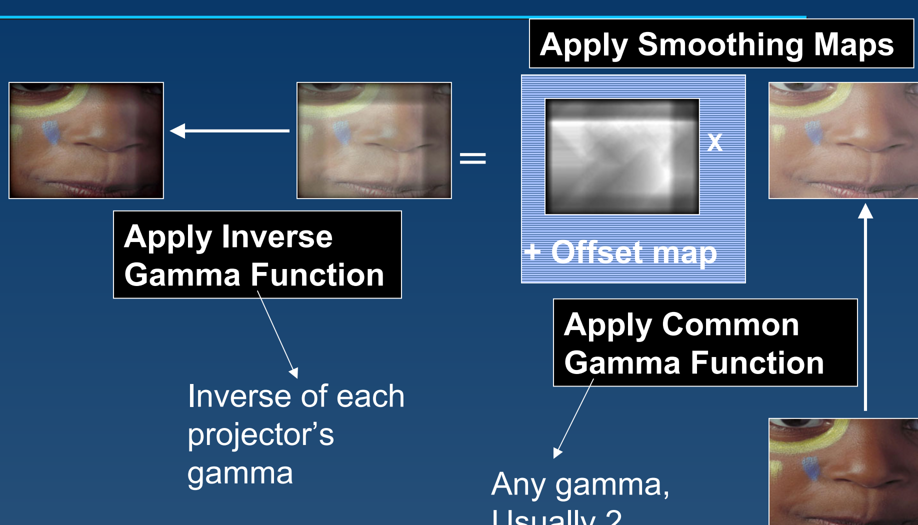
Slide 120

# Let us assume…

- No black offset
  - Generate and apply offset maps
- Projectors have identical gamma
- Projectors are linear devices



# PRISM: Image Correction



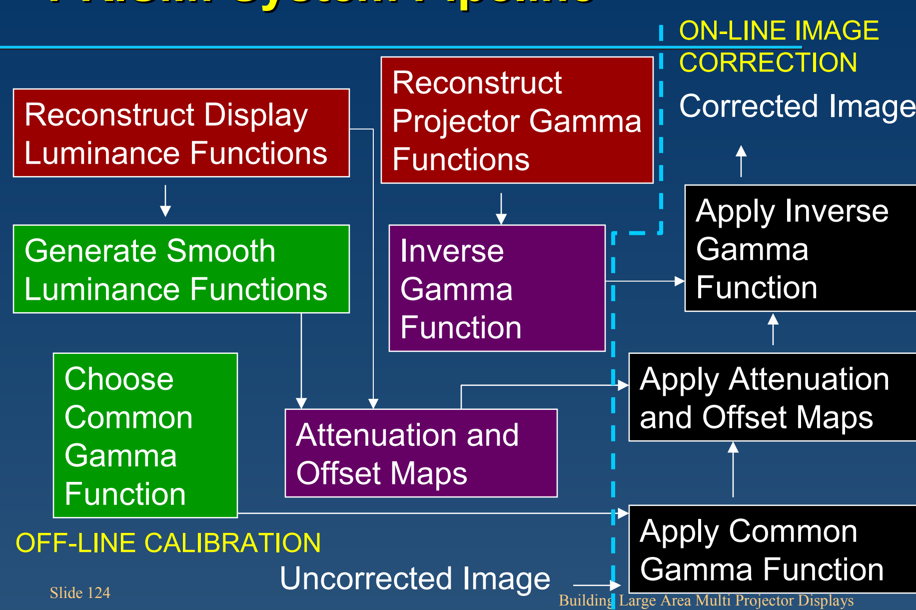
## Real-time Image Correction

- Pixel Shaders
  - Multiply image by itself
  - Multiply with Attenuation Map
  - Add Offset Map
  - Apply Inverse Gamma Function
    - Dependent texture look up

Slide 123

Building Large Area Multi Projector Displays

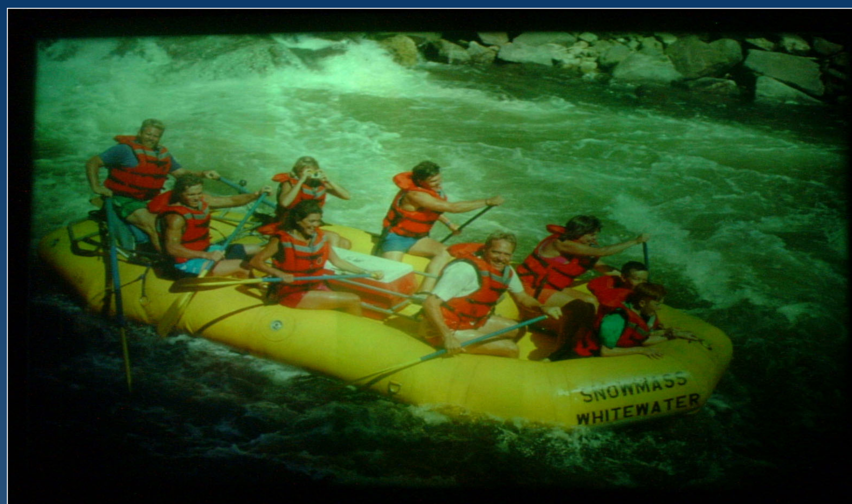
## PRISM: System Pipeline




Slide 124

Building Large Area Multi Projector Displays

## Results (After PRISM)




Slide 125

Building Large Area Multi Projector Displays

## Results (Before)


Slide 126

*6 Projector Display*

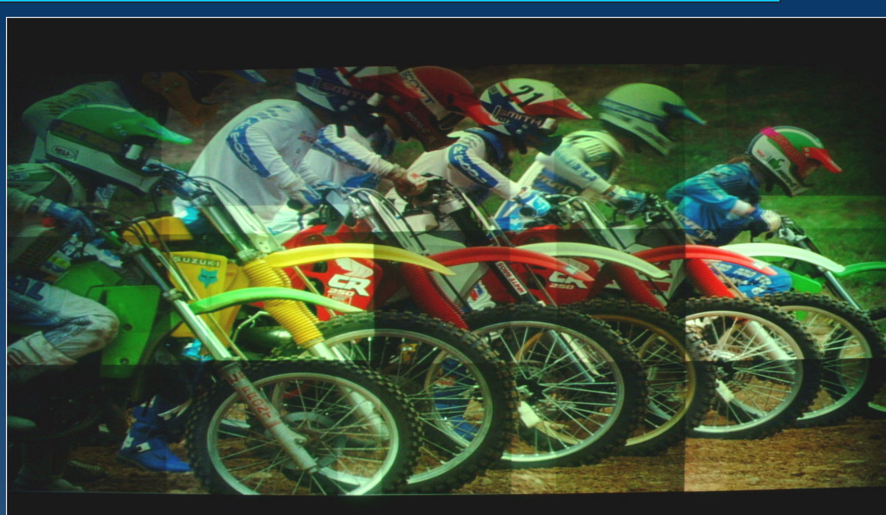
Building Large Area Multi Projector Displays

## Results (After PRISM)

Slide 127

Building Large Area Multi Projector Displays

## Results (Before)



Slide 128

*15 Projector Display*

Building Large Area Multi Projector Displays

## Calibrating the Camera

- Non-linearity
  - Reconstructed using HDR images
  - Each camera image is linearized
- Generate Luminance
  - Standard RGB to YUV conversion
- Operate below F8 aperture
- Exposure adjustments



## Summary

- Manual Manipulation
  - Inter projector luminance and chrominance
  - Not scalable
  - May work reasonable for small systems
  - High maintainence
- Common Bulb
  - Inter projector luminance and chrominance
  - Labor intensive
  - Not scalable

## Summary

- Gamut Mapping
  - Inter projector luminance and chrominance
  - Not scalable due to algorithmic issues
- Blending
  - Overlap region luminance
  - Automated or non automated
  - Scalable



## Summary

- LAM
  - Intra, inter and overlap region luminance
  - Automatic
  - Scalable
  - Low display quality
- PRISM
  - Intra, inter and overlap region luminance
  - Perceptually uniform high quality display
  - Automatic
  - Scalable

## Registration with PC-Cluster Rendering

Michael S. Brown

Hong Kong University of Science and Technology



## Techniques Making Large-Scale Displays Accessible

- PC-Cluster Rendering
  - Makes rendering affordable
  - Exploits "cheap" graphics cards/network
- Camera-Based Display Registration
  - Enables "easy" set up and maintenance displays
  - Reduces need for expensive mounting and display infrastructure
  - Allows flexible projector configurations

## Techniques Making Large-Scale Displays Accessible

PC-Cluster Rendering

**Affordable**

Camera-Based Registration

**Flexible**

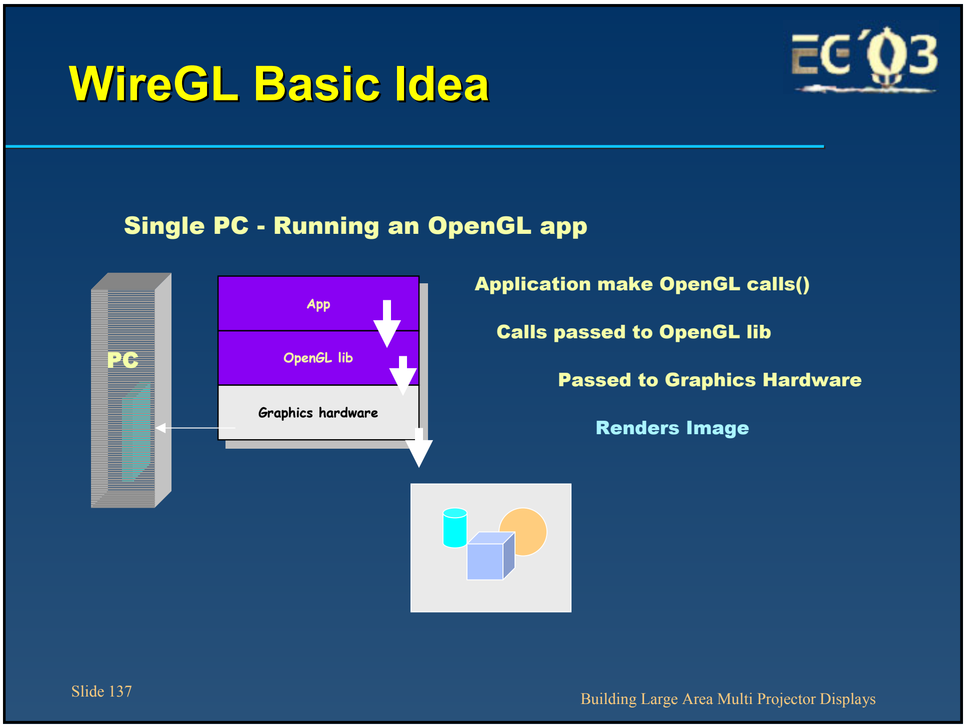
Obvious progression . . combine the two

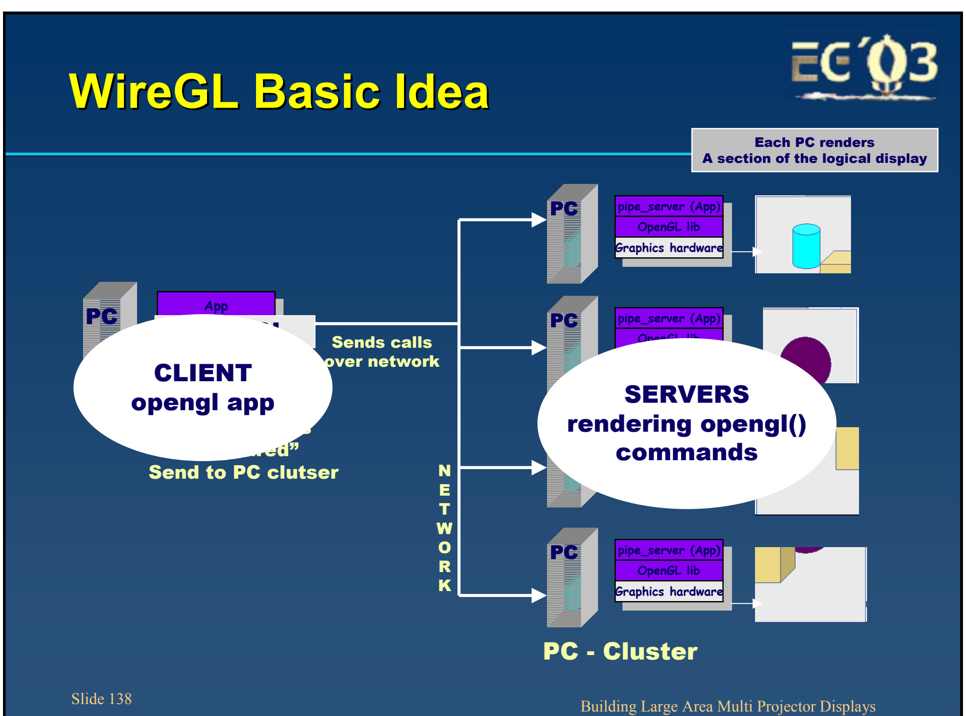
**Affordable and Flexible Large-Scale Display Systems**



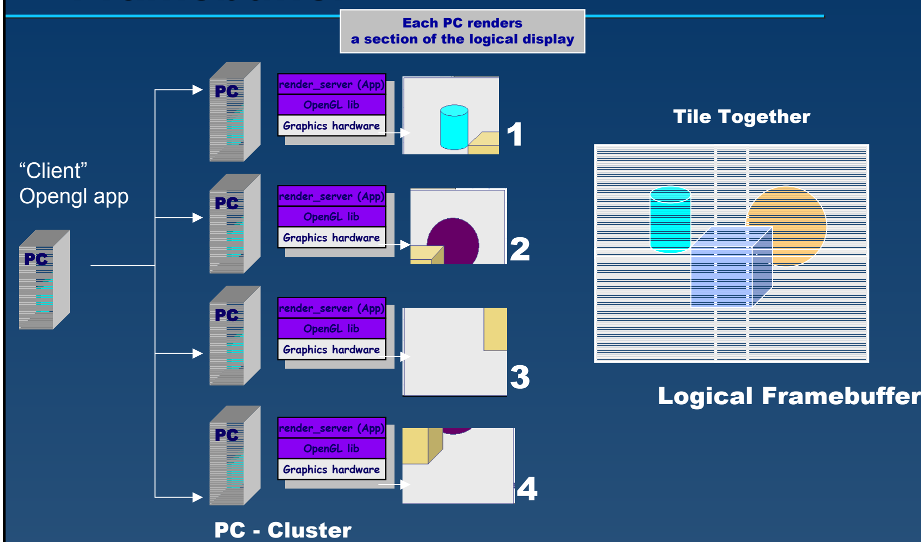
## WireGL/Chromium

- Modify WireGL/Chromium to perform real-time geometric correction
- Why WireGL/Chromium?
  - Distributed PC-cluster rendering architecture that supports the *OpenGL API*
  - Can support existing OpenGL apps with *no* modifications
  - *Open Source*
    - » Source code is available, easy to understand, and can be modified


WireGL Basic Idea
Single PC - Running an OpenGL app
PC
App
OpenGL lib
Graphics hardware
Application make OpenGL calls()
Calls passed to OpenGL lib
Passed to Graphics Hardware
Renders Image
Slide 137
Building Large Area Multi Projector Displays


WireGL Basic Idea
Each PC renders
A section of the logical display
PC
App
CLIENT
opengl app
Sends calls
over network
Send to PC clutser
NETWORK
PC
pipe_server (App)
OpenGL lib
Graphics hardware
SERVERS
rendering opengl()
commands
PC - Cluster
Slide 138
Building Large Area Multi Projector Displays

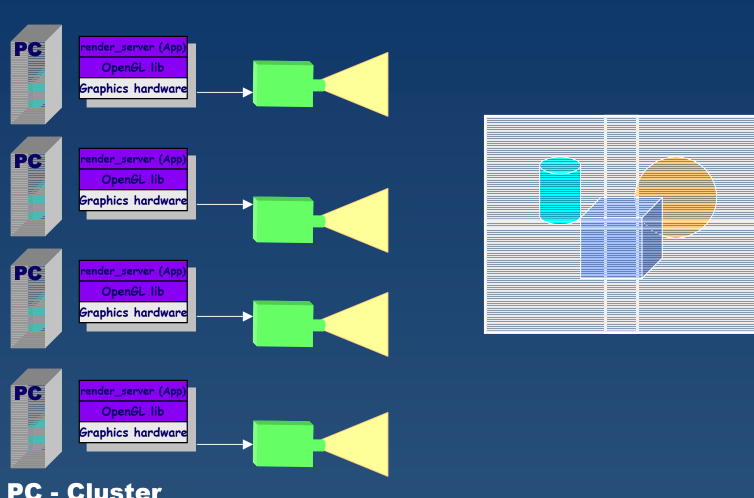
# WireGL "Tiled" Logical Framebuffer




Slide 139

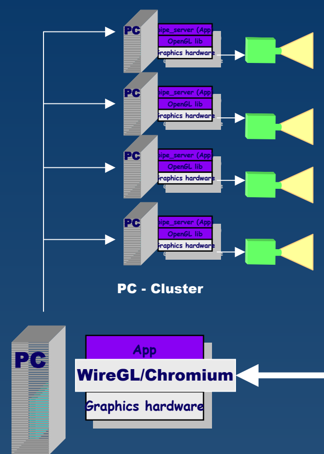


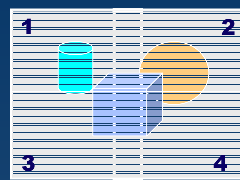
# WireGL:
# Perfect for Tiled Displays




Slide 140

# Configuration File: Describes each rendering server's contribution to the display



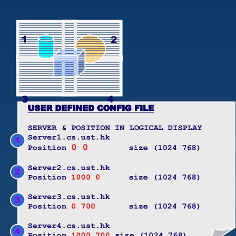

**USER DEFINED CONFIG FILE**

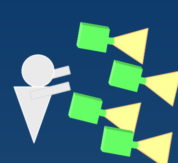
```
SERVER & POSITION IN LOGICAL DISPLAY
Server1.cs.ust.hk
Position 0 0        size (1024 768)

Server2.cs.ust.hk
Position 1000 0     size (1024 768)

Server3.cs.ust.hk
Position 0 700      size (1024 768)

Server4.cs.ust.hk
Position 1000 700 size (1024 768)
```

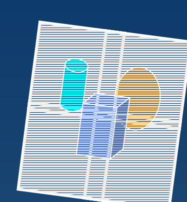




# Configuration: Requires rectilinear alignment







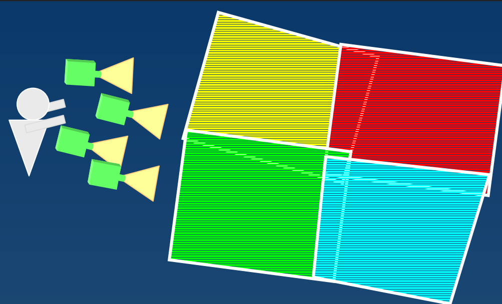
**AS WE KNOW REALIZING THIS CONFIGURATION IS DIFFICULT!!!**

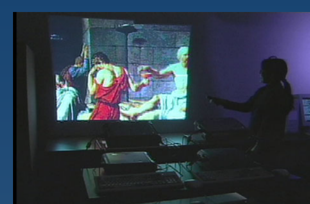
- **Manually align projectors to pixel accuracy**
  - **Difficult and time consuming**
- **Display surface must be planar**
- **Requires special mounting hardware for projectors**
- **Not FLEXIBLE**



Building Large Area Multi Projector Displays

## Desired Functionality





**We would prefer "casual" alignment**
**Remove the need for precise alignment**

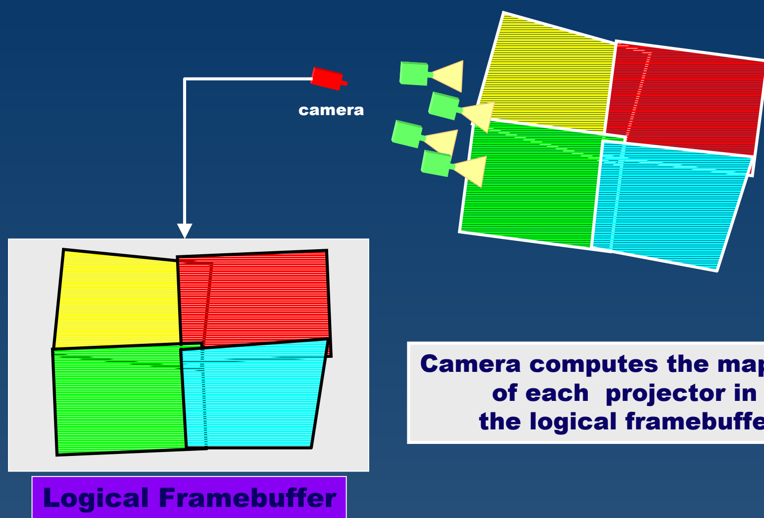


## How to provide this flexibility?

- First
  - Need a "geometric registration" application to . . .
    - » Determine display's geometry
    - » Generate the configuration file needed by WireGL/Chromium
    - » Compute per-projector corrective warping info
- Second
  - Modify PC-rendering architecture
    - » Read in the corrective warping information
    - » Perform real-time corrective warping

## Camera-Based Registration

EG'03




**Camera computes the mapping of each projector in the logical framebuffer**

Slide 145

Building Large Area Multi Projector Displays

## Registration Application

EG'03

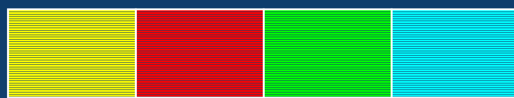
- Applications Details
  - Standard OpenGL application + camera capture
  - App displays a set of equally spaced Gaussian features for each projector
  - Camera observes these projected features
    - » Synchronized – i.e., we knows which projector the camera is observing
    - » Can compute mapping between projector pixels and camera image
- Application runs on top of WireGL/Chromium
  - Uses a dummy configuration file which specifies each projector in a disjoint configuration (no overlap)

Slide 146

Building Large Area Multi Projector Displays

# Registration Application

- Dummy Configuration file



Dummy configuration file specifying non-overlapping tiled arrangement

**USER DEFINED CONFIG FILE**

```
SERVER & POSITION IN LOGICAL DISPLAY
Server1.cs.ust.hk
Position 0 0    size (1024 768)

Server2.cs.ust.hk
Position 1024 0               size (1024 768)

Server3.cs.ust.hk
Position 2048 700             size (1024 768)

Server4.cs.ust.hk
Position 3036 700 size (1024 768)
```

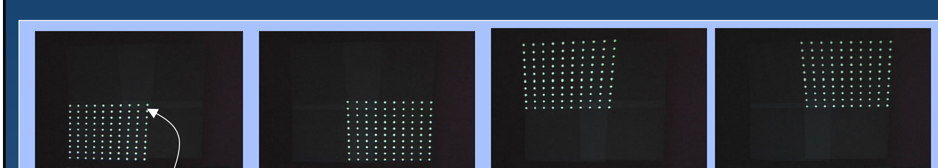
- Registration app also reads in dummy configuration
  - Determines number of projectors, host names, etc . .



# Registration Application

**For each projector**

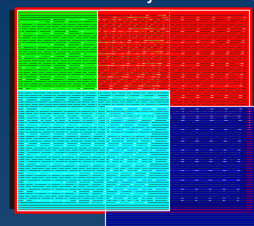
- Display a pattern of features
  (use configuration file to determine projector's position in logical display)
  - Observe features with the camera
    - » Compute mapping for each feature
      - » Feature's centroid in projector and camera: P(x,y) → C(u,v)

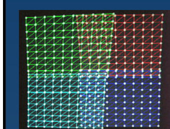


Each observed feature encodes:
P(x,y) → C(u,v)
Feature's location in projector image, P(x,y)
Feature's location in the camera image C(u,v)

# Computing Display Geometry

Determine Projector "Tiling"





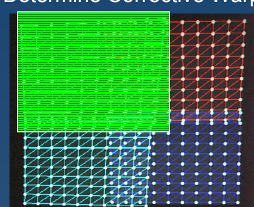
Observed Camera Info

**Configuration File**

**(1) Compute Bounding Box**
**(2) Compute rectilinear "tile" to surround each projector**

**Use this information to compute a configuration file for WireGL/Chromium**

Determine Corrective Warp



**Warping Information**

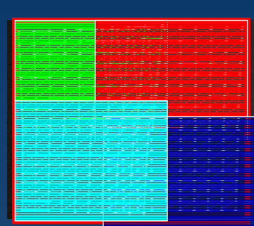
For each projector compute a "warping" mesh.

Features are nodes in the mesh:

Nodes encode the following:
Feature's (x,y) in projector's image
Feature's (x,y) location in assigned Tile
(Projector.x,Projector.y) → (Tile.x, Tile.y)

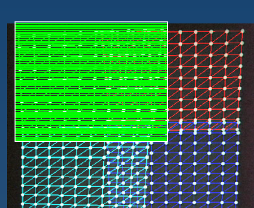


# Registration Application's Output



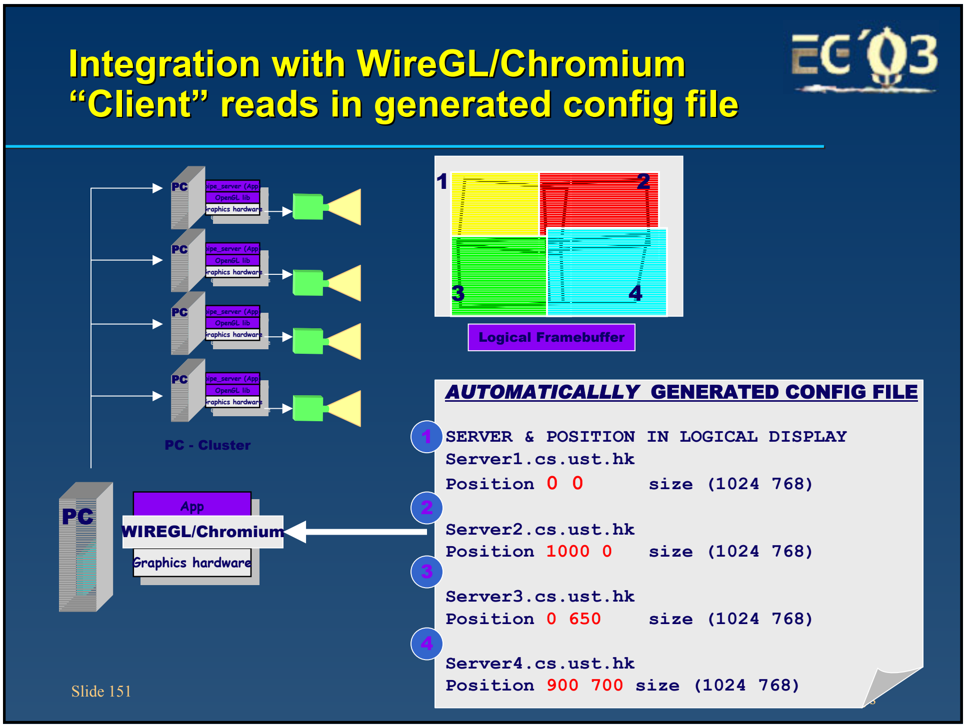
***Automatically Generated Configuration File***

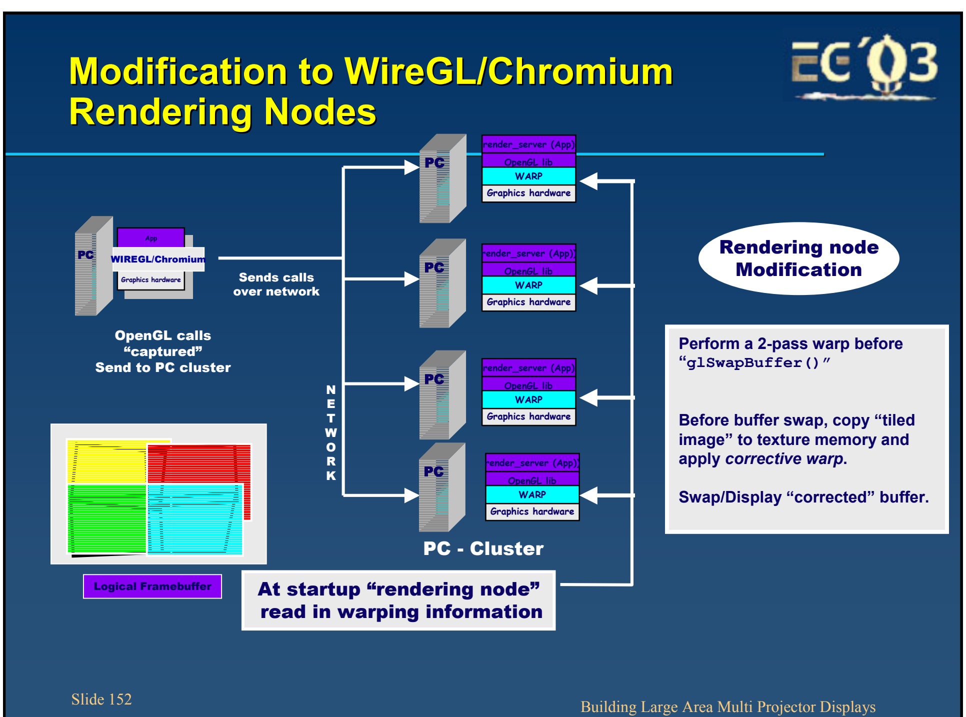
```
SERVER & POSITION IN LOGICAL DISPLAY
Server1.cs.ust.hk
Position 0 0    size (1024 768)

Server2.cs.ust.hk
Position 1000 0  size (1024 768)

Server3.cs.ust.hk
Position 0 650  size (1024 768)

Server4.cs.ust.hk
Position 900 700 size (1024 768)
```



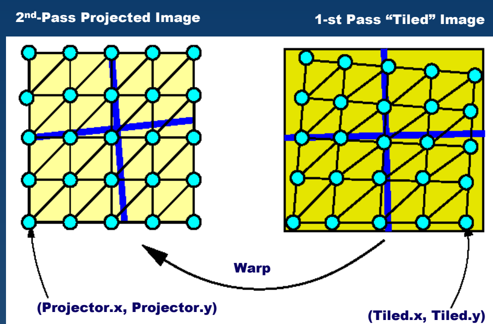
***Per-Projector* Corrective Warp Information**

***Projector 1***
**Vertex List**
**v1 = Projector_x, Projector_y, Tile_x, Tile_y**
**v2 = Projector_x, Projector_y, Tile_x, Tile_y**
**. .**
**Triangle List**
**t1 = v1 v2 v3**
**t2 = v2 v4 v1**
**. . .**
***Projector 2***
**Vertex List**
**. . .**


Integration with WireGL/Chromium
"Client" reads in generated config file
EG'03
PC - Cluster
PC
App
WIREGL/Chromium
Graphics hardware
1
2
3
4
Logical Framebuffer
AUTOMATICALLLY GENERATED CONFIG FILE
SERVER & POSITION IN LOGICAL DISPLAY
Server1.cs.ust.hk
Position 0 0 size (1024 768)
Server2.cs.ust.hk
Position 1000 0 size (1024 768)
Server3.cs.ust.hk
Position 0 650 size (1024 768)
Server4.cs.ust.hk
Position 900 700 size (1024 768)
Slide 151


Modification to WireGL/Chromium
Rendering Nodes
EG'03
PC
WIREGL/Chromium
Sends calls
over network
OpenGL calls
"captured"
Send to PC cluster
NETWORK
render_server (App)
OpenGL lib
WARP
Graphics hardware
PC - Cluster
Rendering node
Modification
Perform a 2-pass warp before "glSwapBuffer()"
Before buffer swap, copy "tiled image" to texture memory and apply corrective warp.
Swap/Display "corrected" buffer.
Logical Framebuffer
At startup "rendering node" read in warping information
Slide 152
Building Large Area Multi Projector Displays

# "Software Alignment" Use 2-pass Rendering Algorithm




**Corrective warping is performed by piecewise texture mapping**

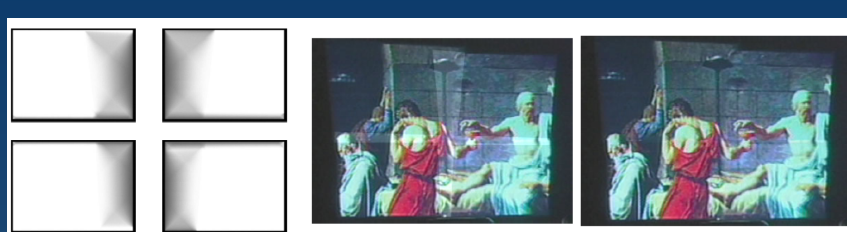
**(use a 2D control mesh)**

Note: Piecewise warping is suitable for non-planar display surfaces



# Can also include intensity blending/photometric correction

- **Automatically compute intensity blending masks (alphamasks)**



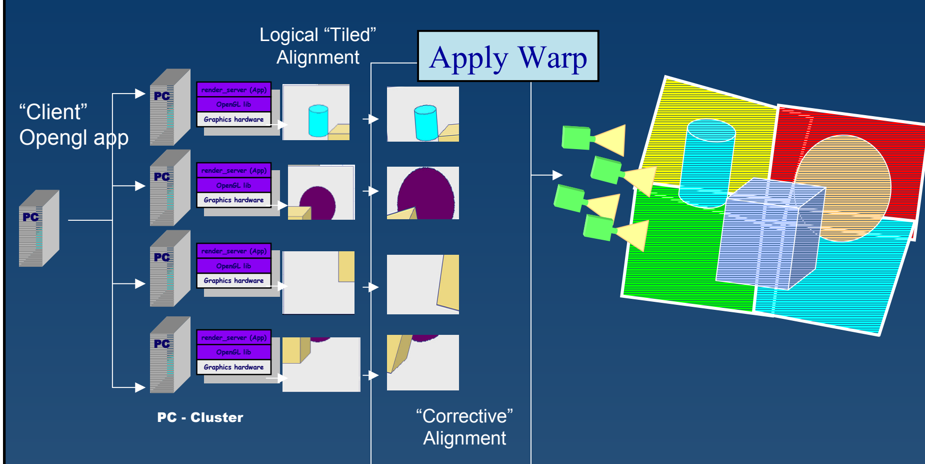
- **Piecewise texture mapping can be considered a "non-linear" warp**
  - **-Corrects for non-planar display surfaces**
  - **-Corrects for projector distortion**

# Corrective Warping Pseudo Code
# 2-pass Rendering

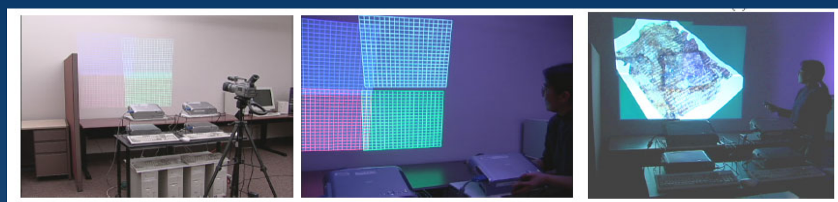
```
// GEOMETRIC WARP PROCEDURE
//  Copy framebuffer Tile to texture memory

glBindTexture(_WarpTexture_);   // use warping texture
glCopyTexSubImage2D(…)          // copy current framebuffer to texture memory


// PERFORM NONLINEAR WARP
// T are the triangles in the tessellated mesh

glBegin(GL_TRIANGLES);
   for each triangle T

       for each vertex V in triangle T

            glTexCoord2D(V.tileX, V.tileY );             // texture coords tile(x,y)

            glVertex2D(V.projectorX, V.projectorY );     // warp to projector(x,y)

        end

    end

glEnd();


// Intensity Blending
glBindTextureAlphaMask(_AlphaMask_);                      // bind alpha mask
render_textured_quad(. . . );                             // draw a screen sized quad
                                                          //   to perform alpha-blending

// display corrected image
glSwapbuffer();
```



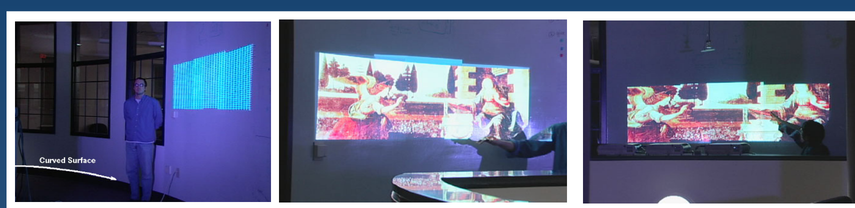
# Putting it together . . .



# Examples



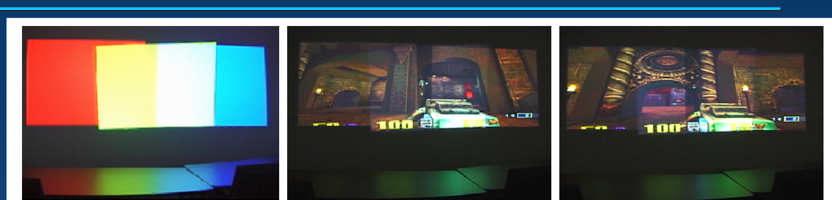
Example Deployment




Display on Curved Surface



# Examples



Display Running Quake III

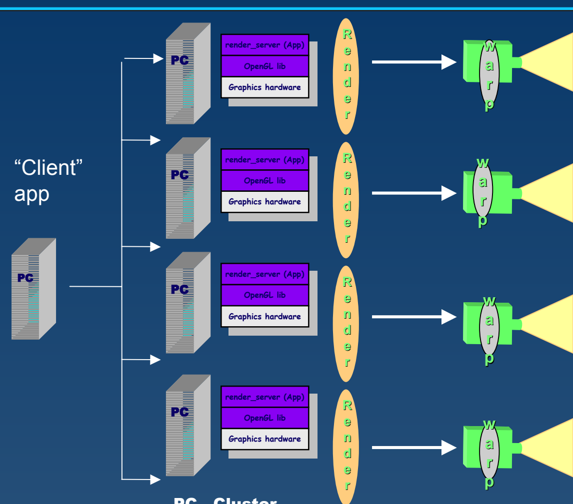
Reconfigured in less than one minute

# Remarks

- Perform warp on the graphics hardware
  - Use a display-list
- Resource Overhead
  - Texture memory on the graphics card
  - Warping time
  - Constant overhead
- Performance Hit
  - Depending on graphics card
  - 2nd pass can reduce speed (60 -> 30fps)
  - Bottleneck is _framebuffer_ copy
    - » Rendering directly to texture memory will eliminate this problem



# Near Future Systems




**Perform warping on the projector**

## Summary

- Affordable and Flexible Tiled Display System
  - Camera-based Registration + PC-cluster rendering
  - Supports **OpenGL** applications
  - Very Flexible, Fully Automated
- Low-cost, easy to deploy
- Flexible, mobile, and suitable for temporary venues

Slide 161



## Course Wrap-up

Aditi Majumder

University of California at Irvine

Slide 162

## Course Summary

- Challenges
  - Casual Alignment
  - Color Seamlessness
  - PC Rendering

Slide 163



## Course Summary

- Challenges
  - Casual Alignment
    - » Automated camera based geometric registration
  - Color Seamlessness
    - » Gamut matching, blending, LAM, PRISM
  - PC Rendering
    - » WireGL, Chromium

Slide 164

## Future Work

- Calibration from any content
  - Feedback loop
- Different Sensors
- Advanced Projectors
  - Smart Projectors

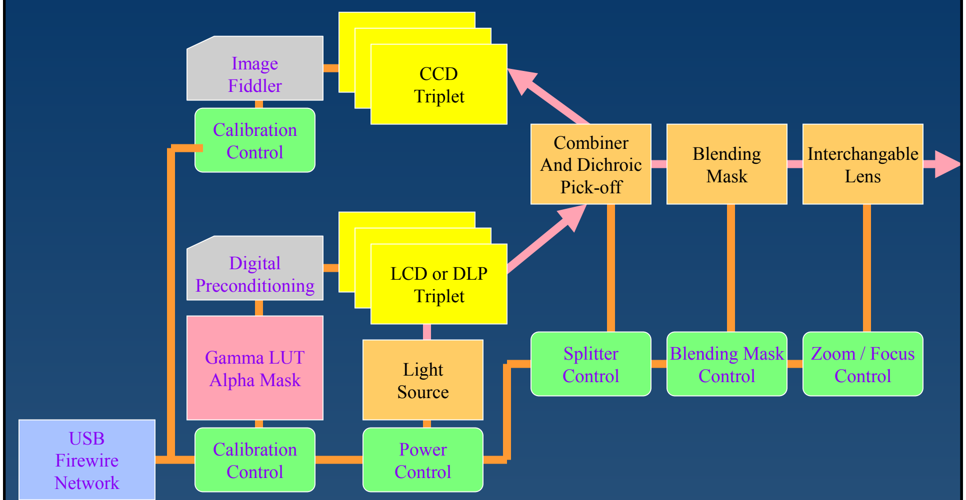


## Projector Wish List

- Geometric Control
- Photometric Control
- Stereo Operation
- Imperceptible Structured Light
- Re-configurable / Front-projection Issues
- Miscellaneous


Credit: Rick Stevens, Argonne National Laboratory

## The Modular Professional Projector




 Credit: Rick Stevens, Argonne National Laboratory 

## Open Projector Specification Might Address

- Physical Image Resolution and Compression/Expansion
- High Contrast and High Absolute Dynamic Range
- High Luminance Output
- Projection Geometry and Configuration Options
- Color Convergence Properties
- Colorimetry and Color Balance Properties
- Illumination Field Properties
- Optical Path Models and Properties
- Support for High Quality Illumination Sources
- Packaging and Mounting, Fine Physical Adjustments
- Serviceability/Modularity (MTTR)
- Stability (Thermal, Temporal, Colorimetry)
- Known Reliability (MTBF)
- Standard Digital Computer Interfaces and Network Interfaces
- Implementation of Standard Control and Calibration Parameters
- External and Internal Control Software APIs

 Credit: Rick Stevens, Argonne National Laboratory 

## Possible Features for Advanced Projectors

- Internal Graphics Interfaces (non-Pixel protocols)
- Stereo Support and High-Frequency Inputs
- Internal or Externally Integrated Calibration Cameras
- Integrated Wireless/Fiber Networking Interfaces
- Integrated Codex for Compressed Streams/Stills
- Non-Uniform Resolution/Non-Raster Imaging
- Support for Multiresolution Display Systems
- Non-Uniform or Irregular Imaging Geometry
- Off Axis Projection Capability
- Projector Array/Cluster Awareness
- Distributed Frame Buffer Support
- Non-Uniform Tiling Support
- Built in Calibration Agents and APIs
- Internal Edge Blending Logic/Hardware

Slide 169 Credit: Rick Stevens, Argonne National Laboratory



## More Radical Proposals (I)

- Projectors are the largest cost component in a TD system so lets improve them
- Smart projectors
  - A rendering engine in each projector
  - Virtual frame buffer elements
    - With neighborhood/self organization logic
  - Edge blending logic/hardware mechanisms
  - Networking interface chips
  - Cameras in the optical path to provide self-calibration
- Could these get cheap enough (say via Games) to make interesting ?
- Make the projectors simpler and cheaper
  - Think of a projector as a high-tech light bulb

Slide 170 Credit: Rick Stevens, Argonne National Laboratory

Building Large Area Multi Projector Displays

# More Radical Proposals (II)

- Build displays out of a very large number of low resolution and lower performance (and therefore inexpensive) devices with integrated computing + networking
- Something like active "post-it" notes
  - Develop low "power" visualization techniques
  - Leveraging ideas from imaging, illustration, and graphics design, etc
- Building tiled displays with 10,000 color palm pilots velcro'd to the wall with wireless interconnects?



Credit: Rick Stevens, Argonne National Laboratory