Achieving Seamlessness in Multi-Projector based Tiled Display using Camera Feedback

Pranav Kant Gaur

Graphics and Visualization section,
Computer Division,
Bhabha Atomic Research Centre, Mumbai

Outline

Objective

Create a large image on the projection screen using combination of multiple projectors.

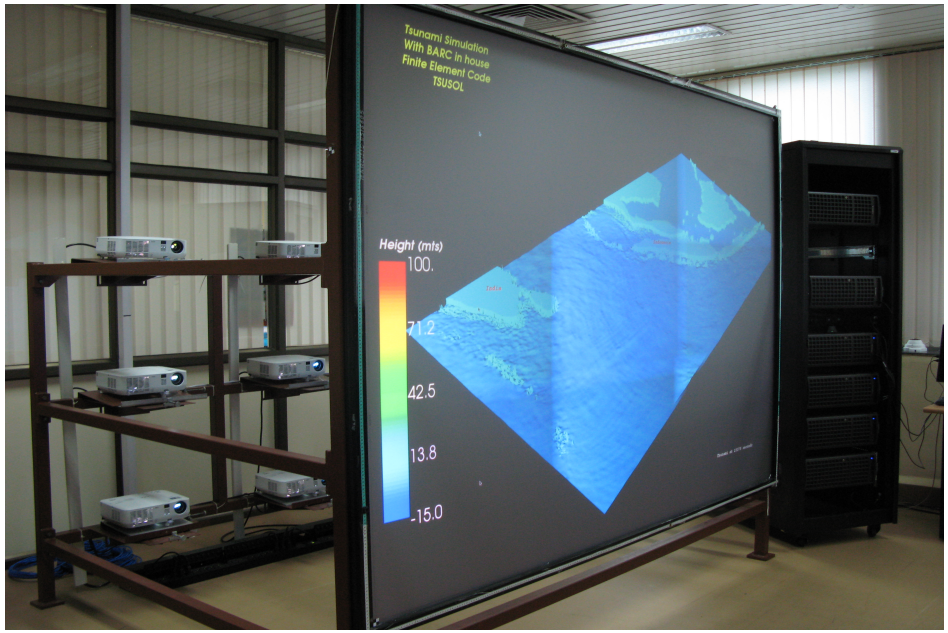


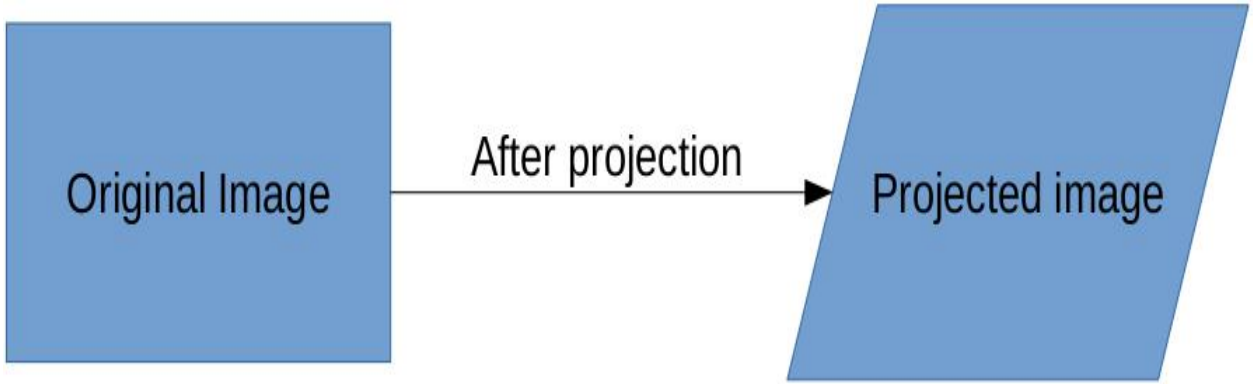
Figure: Multiprojector tiled display

Motivation

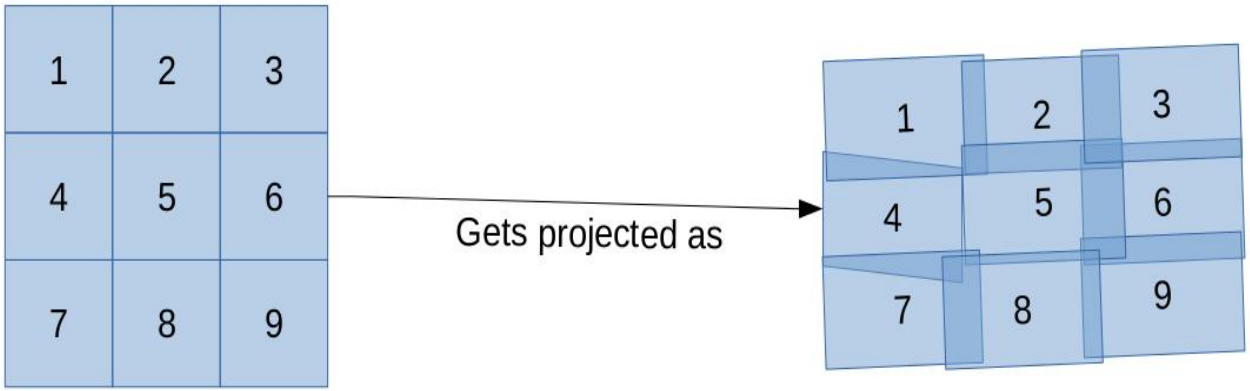
- Why *Tiled*?
  - Covering wider field-of-view
  - Spatially streching high resolution content
- Why *Projectors*?
  - No bezels
  - Variable tile sizes

Challenges

1. Perspective distortion removal:



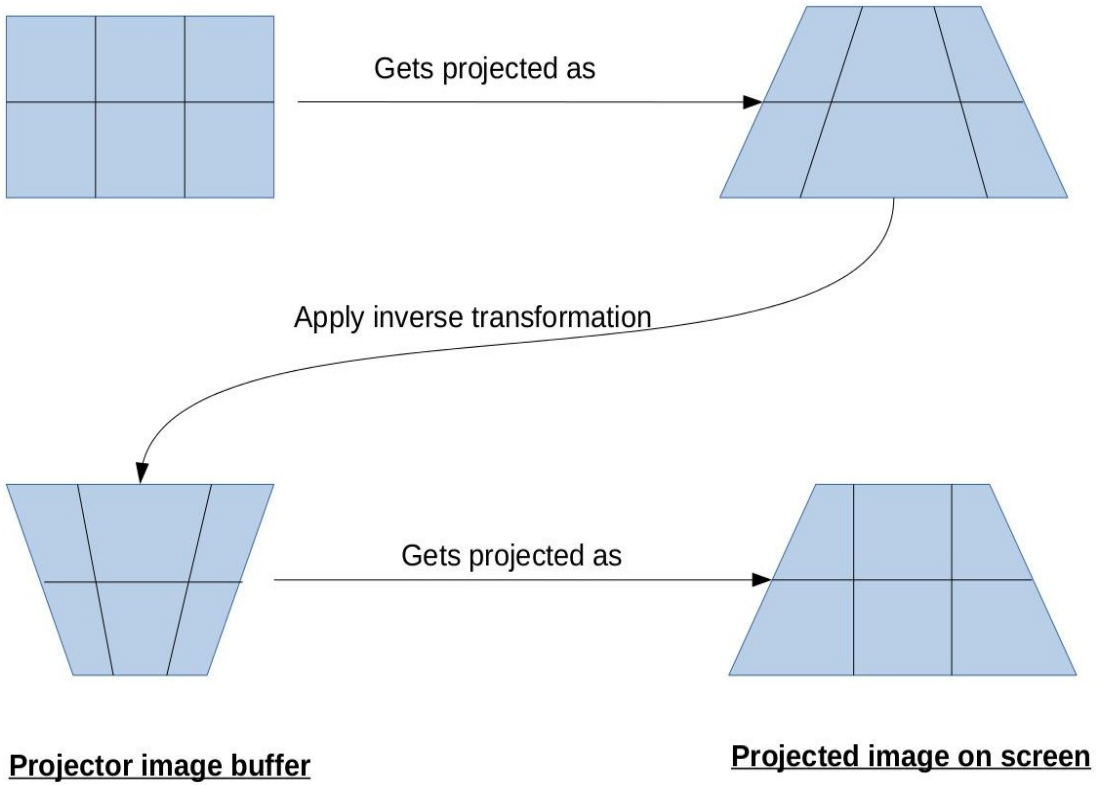
2. Geometric continuity of projected content:



3. Handling intensity in overlap region:

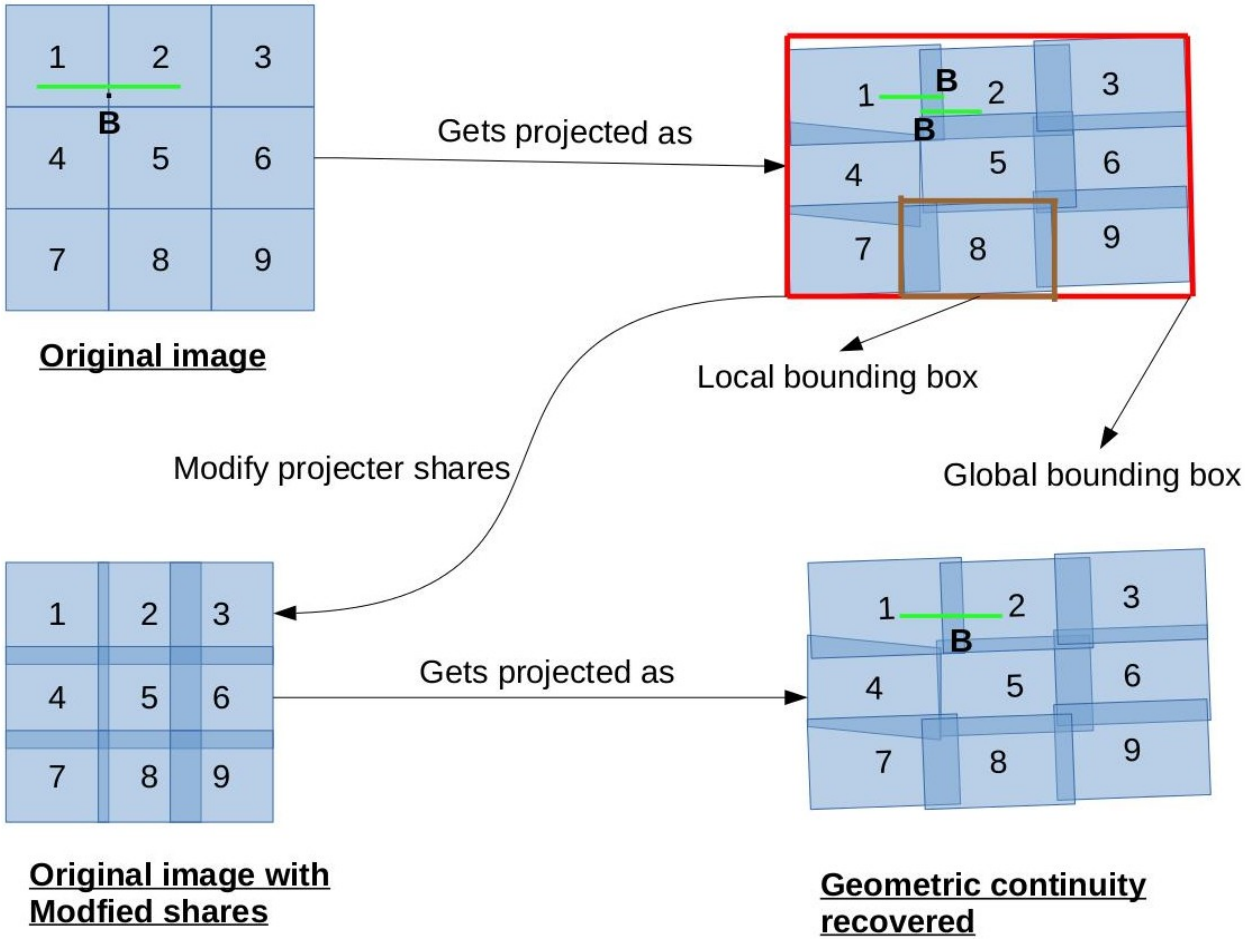
Followed approach

- Removing perspective distortion:



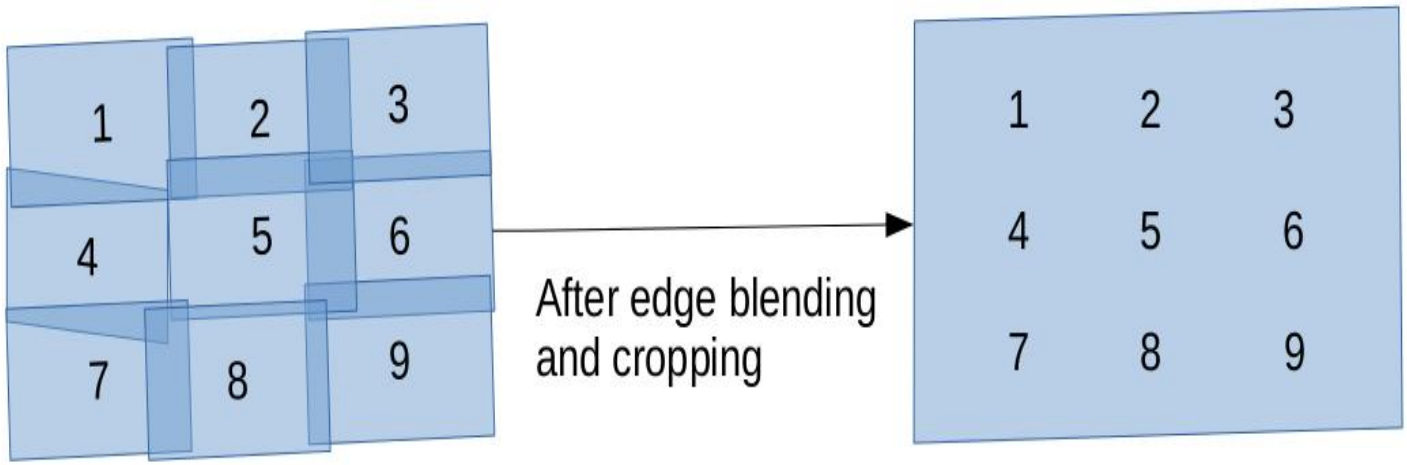
Followed approach(contd.)

- Ensuring geometric continuity:



Followed approach(contd.)

- Handling image intensity at the overlap region:



System setup

Developed system has:

- 3X3 grid of projectors
- Rear projection screen
- 1 digital camera
- Workstations arranged in master-slave configuration

(a) System setup

(b) Projector-array behind the projection screen

Algorithm

- Based on paper 'A practical and flexible tiled display system' by M.S. Brown and W.B. Seales.
- Algorithm overview:
  1. Compute Camera-screen homograhy
  2. Project and detect features(i.e., checkerboard)
  3. Address detected coordinates wrt. local bounding box coordinate system
  4. Address local bounding boxes wrt. global bounding box coordinate system
  5. Compute overlap between adjacent projectors
  6. Attenuate intensities in overlap regions

Algorithm(contd.)

Compute screen to camera relation

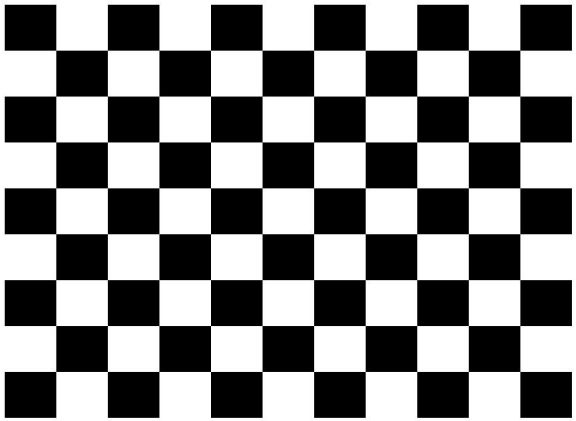
- *Geometrically aligned* and *seamless* content required on projection screen
- Camera is just a *feedback device*
- All later calculations are performed in screen-coordinate system

Figure: View from camera

Algorithm(contd.): Remove perspective distortion

Project and detect features for each projector

- Detected features are mapped to screen coordinate system



(a) Projected features

(b) Low exposure image of detected features for the central projector

Algorithm(contd.): Remove perspective distortion

Compute *local* bounding boxes

- Compute normalized pair of ($coordinate_{original\ image}$, $coordinate_{detected}$) for each checkerboard corner.

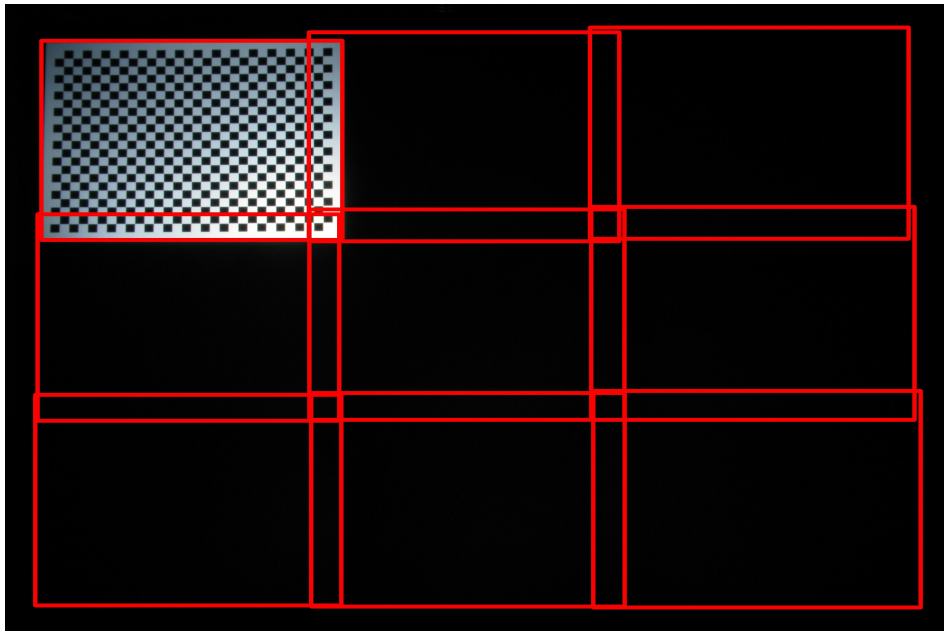
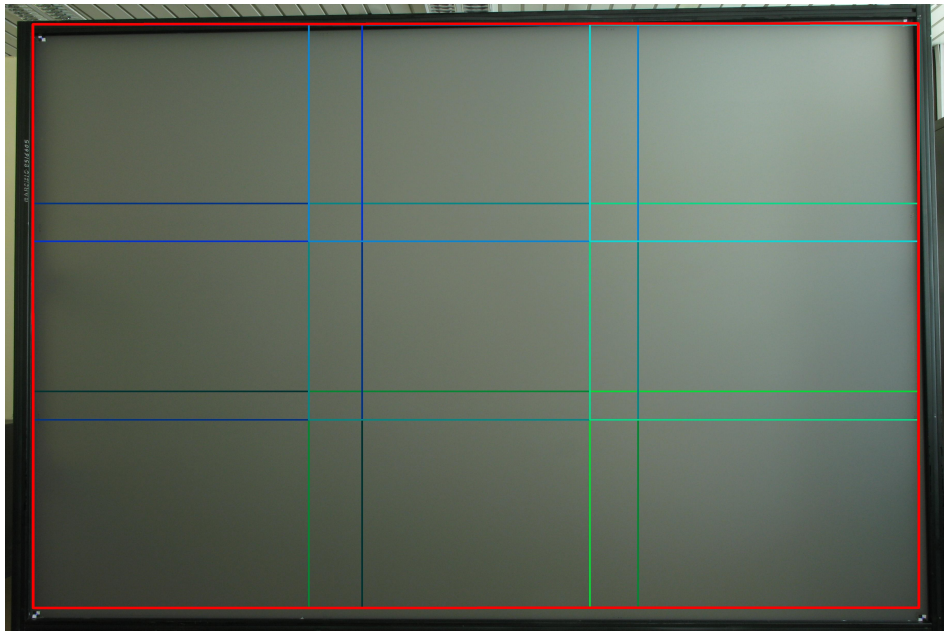


Figure: Boxes bounding the projection region of each projector

Algorithm(contd.): Geometric continuity

Compute *global* bounding box

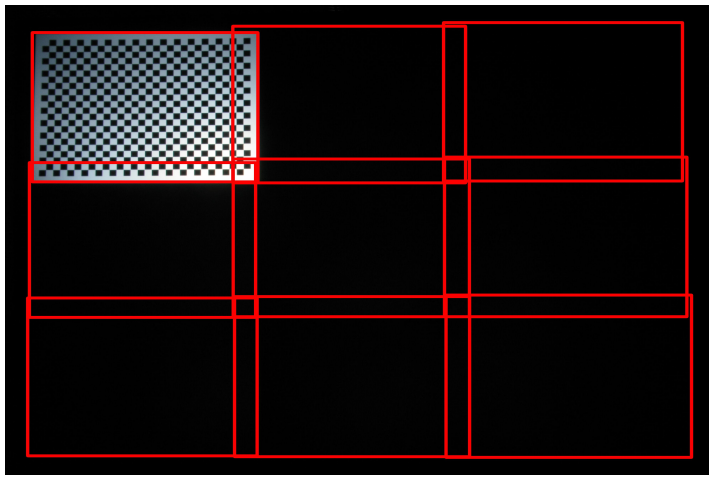
- Addressing all local bounding boxes wrt. a common coordinate system.
- Global bounding box represents the original image to projected on the screen.
- Helps in computing share of each projector in the original image.



Algorithm(contd.): Seamlessness

Compute alpha-map*

1. Compute region of overlap between adjacent projectors.
2. Attenuate image intensity of each overlapping projector.



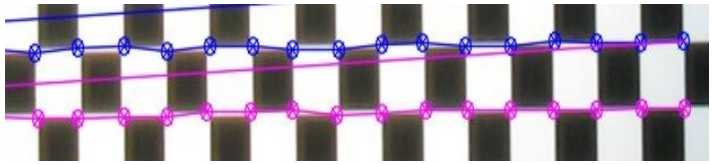
(a) Projection region

(b) Corresponding attenuation map

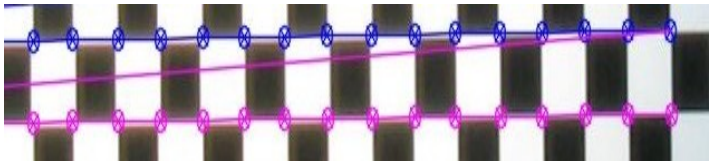
Contributions

Corrected detected corners using line fitting:

- Utilized the fact that collinearity under perspective projection is preserved
- Fitted line on detected corners and corrected the detected coordinates using intersection of fitted lines
- Resulted in more uniform texture mapping



(c) Without line fitting

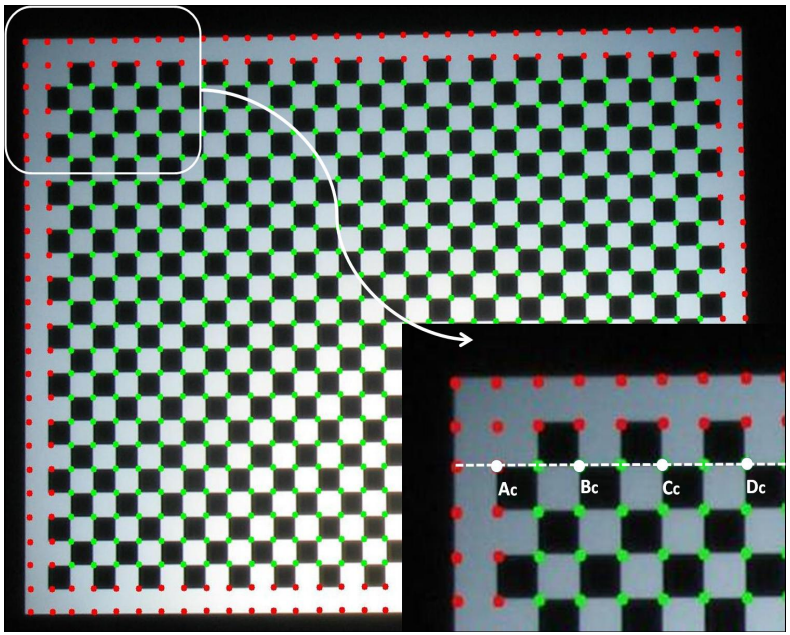


(d) With line fitting

Contributions(contd.)

Used Cross-ratio* invariant to recover full projection region:

- It is a perspective projection invariant
- Relates 4 collinear points: $CR_p = \frac{|AC|_c * |BD|_c}{|BC|_c * |AD|_c}$

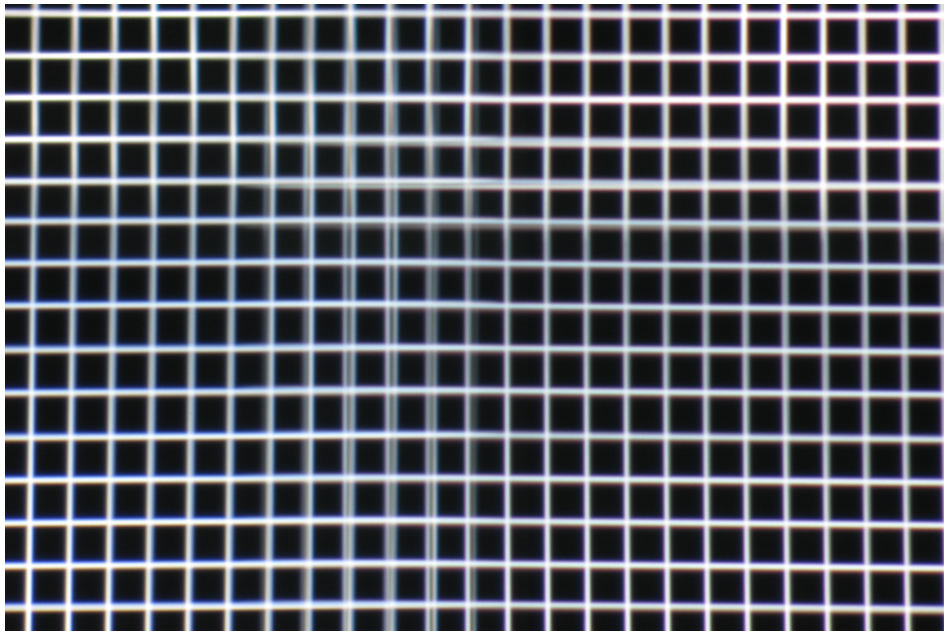


(e) Applying cross ratio invariant

(f) green: without cross ratio, blue: with cross ratio invariant

Results*

- Average misalignment between the grid lines projected at junction between neighbouring projectors was around $\sim 1.0mm$ on a grid size of $\sim$ 14mm.



Results(contd.)

- Recovered $\sim 10\%$ more projection region using cross-ratio invariant.
- Junctions between projectors are more imperceptible using cross ratio invariant.

(g) Without cross ratio: seams more visible

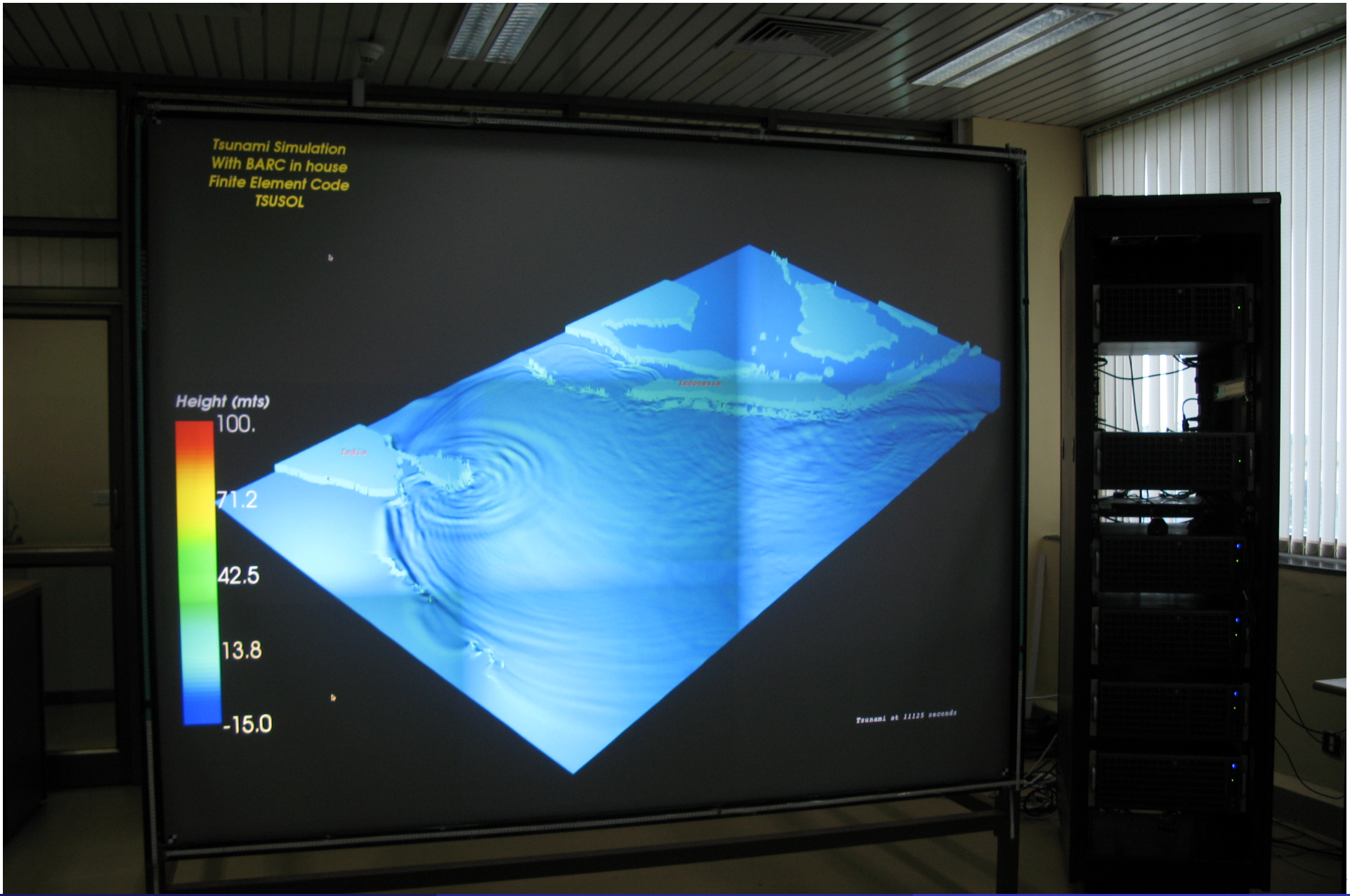
(h) With cross ratio: seams more imperceptible

- Resolution of the display is ~ 6.2 Megapixels.

Conclusion

- Extension of the approach described in Brown's paper:
  - Line-fitting
    - More uniform texture generation
  - Cross ratio:
    - Full projection region
    - Lesser imperceptible seams
    - Completely arbitrary placement of projectors
- Limitations:
  - Cross-ratio works only on planar surface
  - $\text{Resolution}_{\text{Utilized}} < \text{Resolution}_{\text{Native}}$

Thank you!!


Tsunami Simulation
With BARC in house
Finite Element Code
TSUSOL
Height (mts)
100.
71.2
42.5
13.8
-15.0
Indonesia
India
Tsunami at 11125 seconds

Cross ratio equations

$$x_t = B_c^x + t * (D_c^x - B_c^x)$$
$$y_t = B_c^y + t * (D_c^y - B_c^y) \quad (1)$$

$$[t_1, t_2] = \frac{-b \pm \sqrt{b^2 - 4 * a * c}}{2 * a} \quad (2)$$

where,
Let,
$delta = CR_p * (\frac{|BC|_c}{|BD|_c})^2$
Then,

$$a = (1 - delta) * |BD|_c^2$$
$$b = 2 * (D_c^x - B_c^x) * (B_c^x - C_c^x) + (D_c^y - B_c^y) * (B_c^y - C_c^y) + 2 * |BD|_c^2 * delta$$
$$c = |BC|_c^2 - delta * |BD|_c^2$$

Edge blending equations

- Distance transform:

$$\alpha(p, x_p, y_p) = \frac{d_p(x_p, y_p)}{\sum_{i \in shares(i, x_g, y_g)} d_i(x_i, y_i)} \tag{3}$$

- Gamma correction:

$$\alpha(p, x_p, y_p)_\gamma = \alpha(p, x_p, y_p)^{\frac{1}{\gamma(p(x_p, y_p))}} \tag{4}$$

Experiment setup

- Software:
  - Written in C
  - Dependent on OpenCV(v2.4.1) and libgphoto2(v2.5.2)
  - Works on Ubuntu(12.04 LTS) and Scientific Linux(6.1)
- Hardware:
  - 3X3 grid of NEC 200X DLP projectors
  - 2.4mX1.8m acrylic glass based rear projection screen(from ScreenTech,Germany)
  - Canon Powershot G7 digital camera
  - 4 Workstations(1 master+ 3 slave) each with Intel Xeon Six Core Processor X5670, NVIDIA Quadro FX 4600 Graphics card and 48 GB ECC RAM.