Optimized Color Gamuts for Tiled Displays

Marshall Bern

Palo Alto Research Center

David Eppstein

Univ. of California, Irvine
School of Information and Computer Science

Tiled projector systems

Problem: display system for collaborative workspaces
high resolution (≥ 6Mp), large scale (conference room wall)

Solution: combine output from multiple LCD projectors

Commercially available 20-projector system from Visbox, Inc.

Issues in creating seamless appearance to display walls

Data streaming and control

How to get large quantities of data to many projectors, update and interact with data in real time?

Physical alignment

How to make screen images line up without seams?
How to combine coordinate systems of screens?
How to measure and adjust screen alignment?

Color alignment

How to handle radial brightness fade (vignetting)?
How to reduce brightness where projections overlap?
How to get matching colors from different projectors?
How to measure and adjust color alignment?

Issues in creating seamless appearance to display walls

Data streaming and control

How to get large quantities of data to many projectors,
update and interact with data in real time?

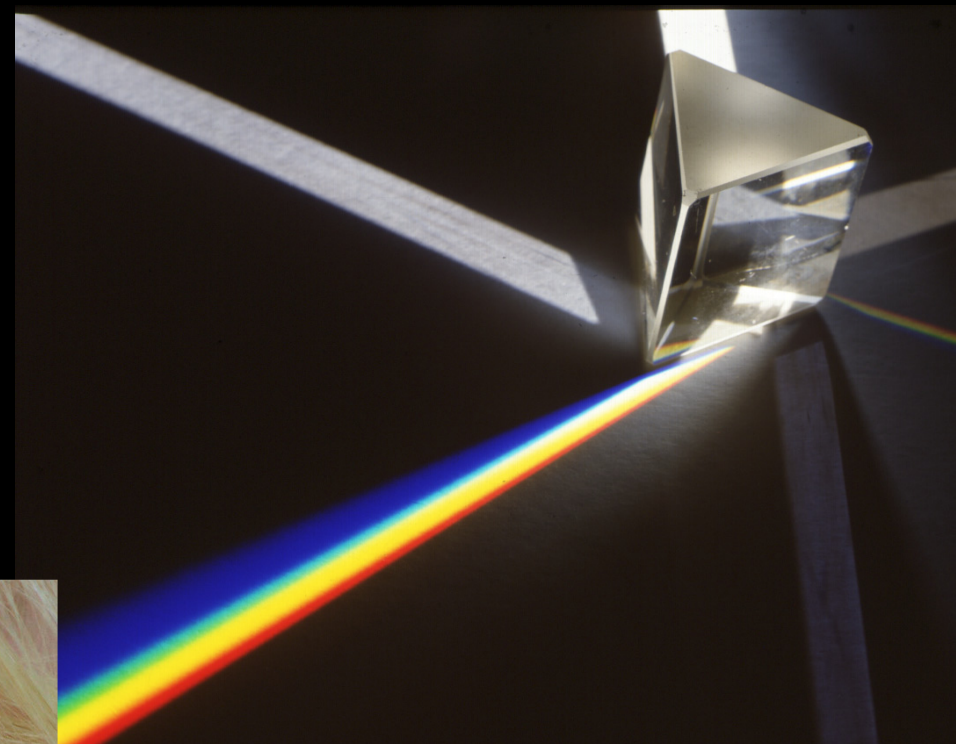
Physical alignment

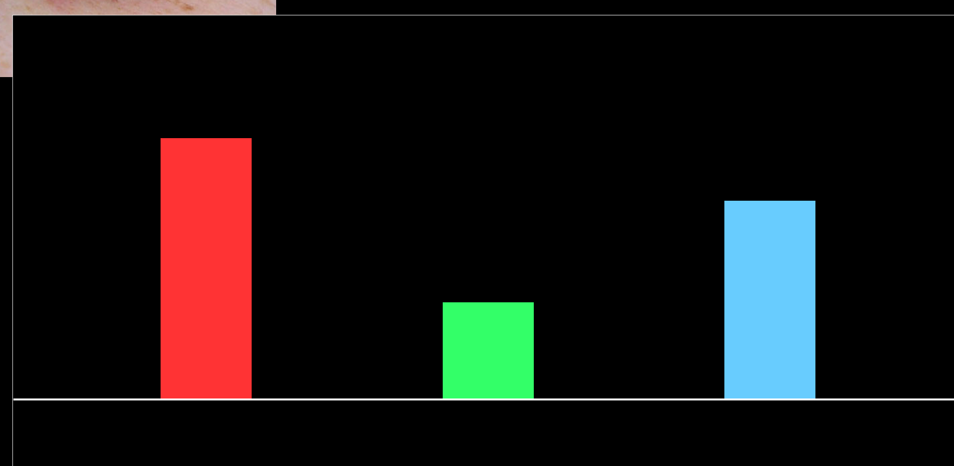
How to make screen images line up without seams?
How to combine coordinate systems of screens?
How to measure and adjust screen alignment?

Color alignment

How to handle radial brightness fade (vignetting)?
How to reduce brightness where projections overlap?
How to get matching colors from different projectors?
How to measure and adjust color alignment?

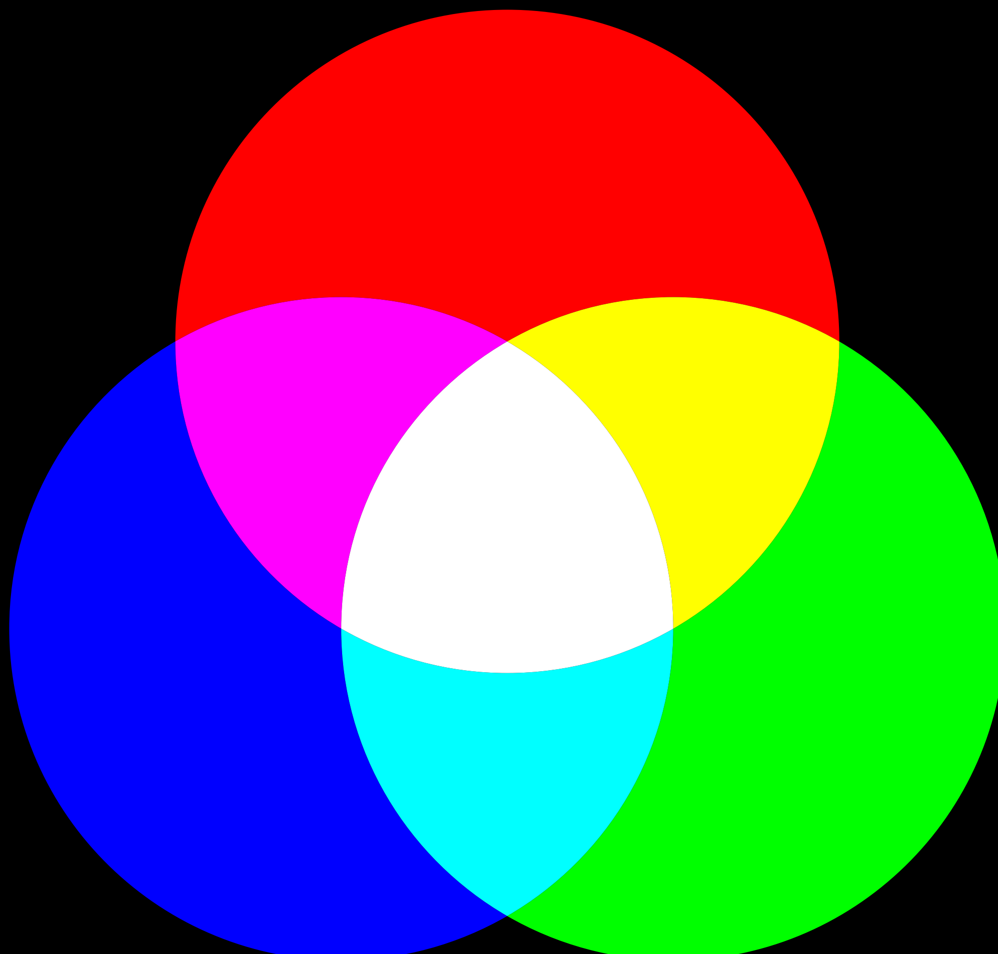
Human vision is 3-dimensional





Additive Color

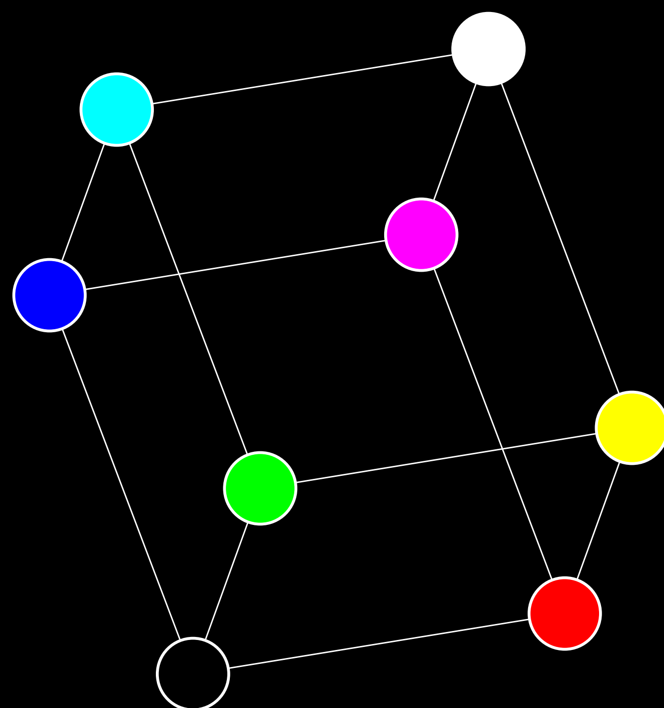
Build up colors from black by adding primary color light
Used in most computer displays (CRT, LCD, etc)



Additive color gamuts

Gamut = set of colors that can be produced by a display

Gamuts of additive color devices (such as projectors)
form parallelepipeds in 3d device-independent color space



Twelve degrees of freedom (black point + three primary colors)

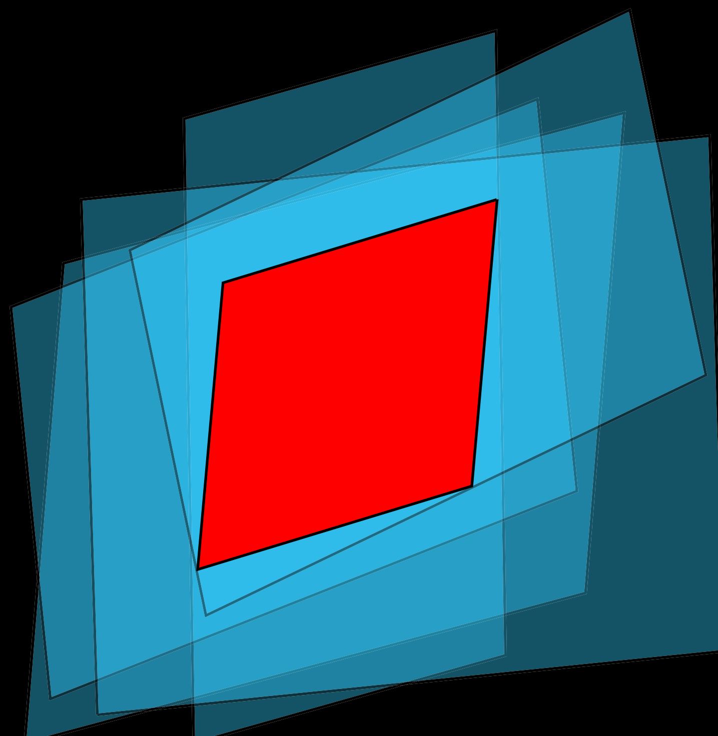
Different projectors in tiled display have different gamuts

Even within same model (different color filter batch, light bulb age...)

If uncorrected, photographic images will show seams

Problem: find additive gamut displayable by all projectors

Geometrically: find large parallelepiped inside intersection of parallelepipeds



Prior Work

Many papers on tiled displays...

Majumder et al., IEEE Visualization 2000

Attacks combined problems of vignetting, projector mismatch, and feathering overlaps

Matches luminosity (dark-light values) only, not color

Stone, *IEEE Computer Graphics & Appl*. 2001

Formulates gamut intersection problem

Provides incompletely-specified non-automated heuristic for finding large gamuts

Unclear what objective is being optimized

Naïve volume-based approach

Objective function: maximum volume gamut

Space of parallelepipeds is 12-dimensional (coordinates of four vertices)
$48n$ halfspace constraints (vertex of output within facet of input)
Objective function is nonconvex

Test each face of halfspace intersection: $O(n^6)$

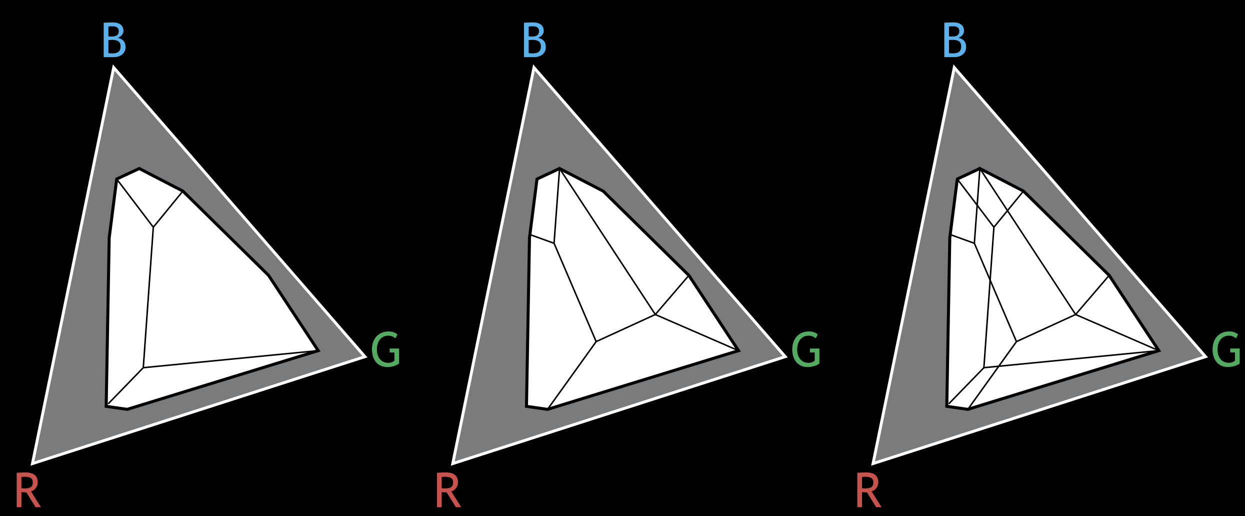
Too slow, too little connection to visual appearance

Combining volume with colorimetrics

"Chromaticity diagram" = 2d central projection of gamut onto black point

Overlay diagram of darkest values in gamut intersection
(lower part of convex polytope) with diagram of lightest values (upper polytope)

Search overlay for black and white points with same color, max luminance ratio
(avoid color shifts, maximize contrast)



Choose remaining six gamut parameters to maximize volume: $O(n^3)$

Alternative objective functions

Cubic time should work for few projectors

But what if we want an input gamut for each pixel of each projector?

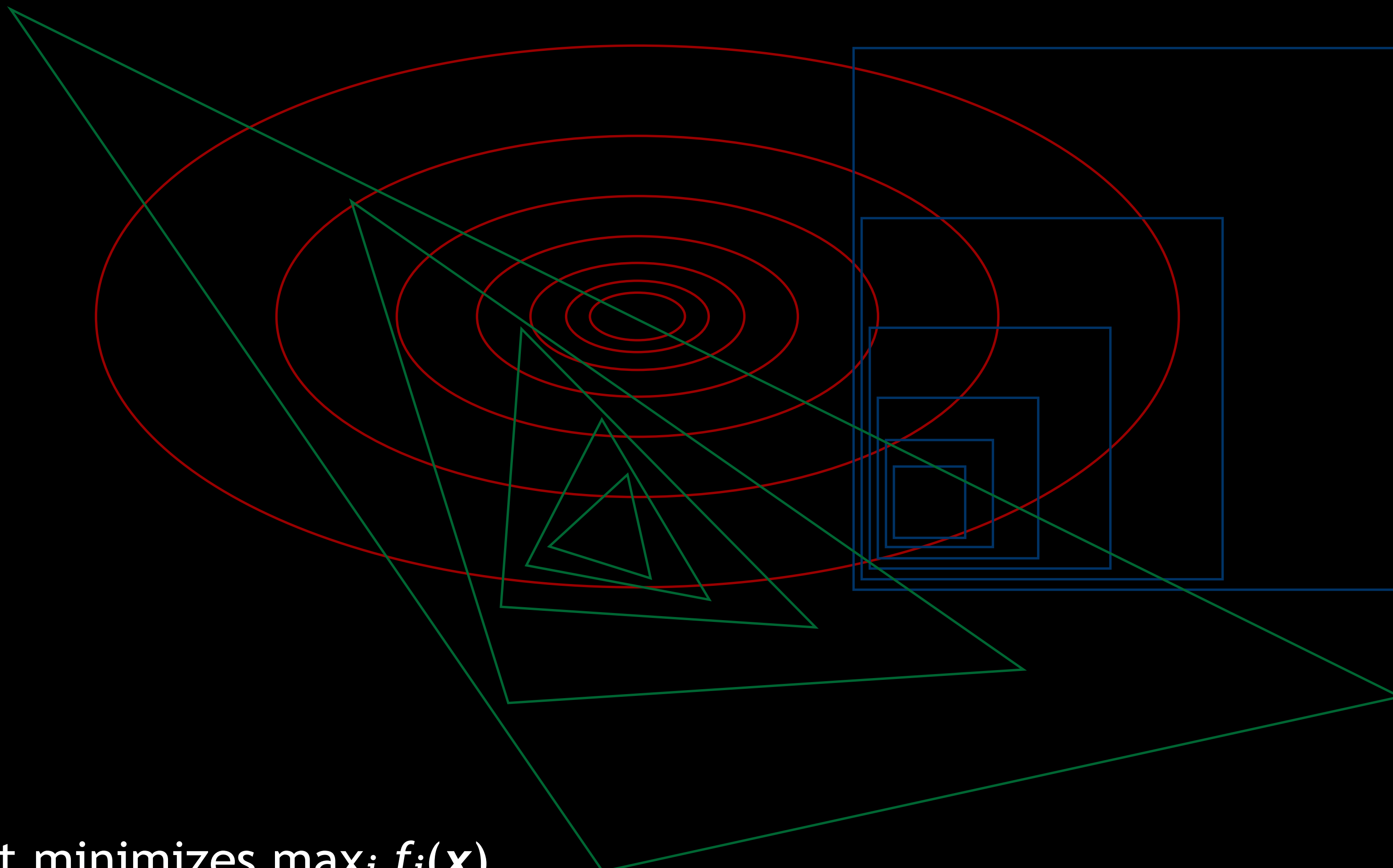
Need an algorithm that scales better

Idea: use quasiconvex programming

Leads to linear time LP-type algorithms

Quasiconvex program:

Input: family of quasiconvex functions $f_i(\boldsymbol{x})$, $\boldsymbol{x}$ in $\mathbf{R}^d$



Output: $\boldsymbol{x}$ that minimizes $\max_i f_i(\boldsymbol{x})$

Quasiconvex gamut optimization

Find eight 3d quasiconvex functions f_K, f_R, f_G, f_B, f_C, f_M, f_Y, f_W
measuring quality of each gamut corner location
Lift each function to 12*d* function of gamut location (still quasiconvex)

Add 48*n* halfspace constraints (quasiconvex step functions)

Quasiconvex program value = gamut optimizing worst color corner

Scales linearly with number of input gamuts
Can treat some colors (black, white) as more important than others

Conclusions

Cubic-time optimal gamut algorithm
using colorimetric volume-based objective function

Linear-time quasiconvex optimization algorithms
using separate quality measure for each color corner

Still needed

Experimental tests of feasibility of cubic algorithm?
(need gamut data from Stone)

Colorimetric expertise: which objective function is best?

Fast approximation algorithms?