Choosing Colors for Geometric Graphs Via Color Space Embeddings

Michael B. Dillencourt
David Eppstein
Michael T. Goodrich

Univ. of California, Irvine
Computer Science Department

Outline

The problem

A brief introduction to color spaces

Our solution

Evaluation

Conclusions

Outline

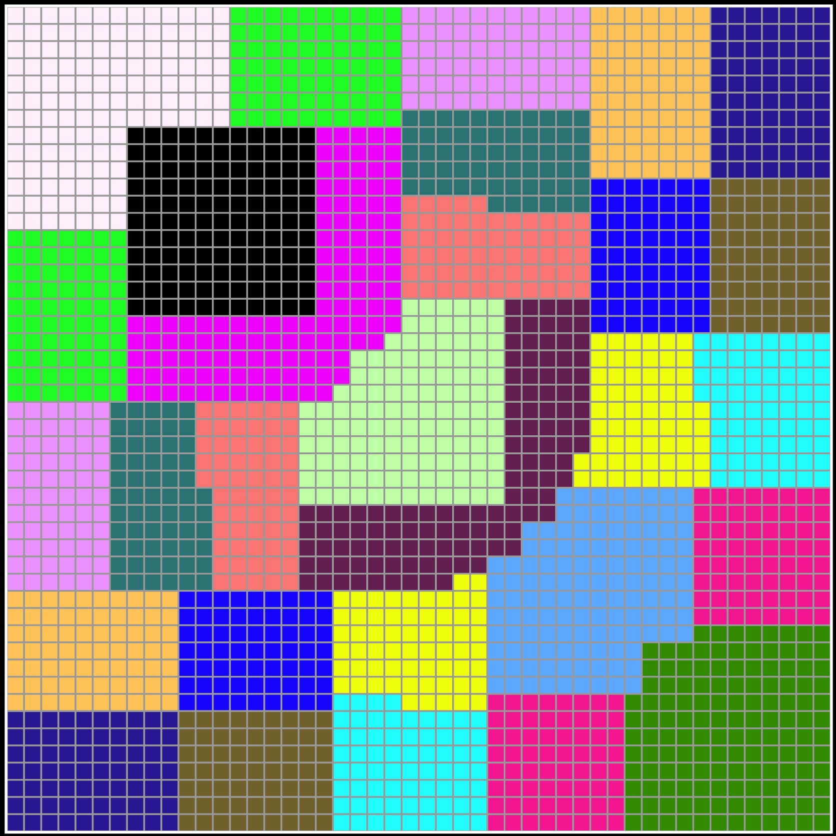
The problem

A brief introduction to color spaces

Our solution

Evaluation

Conclusions

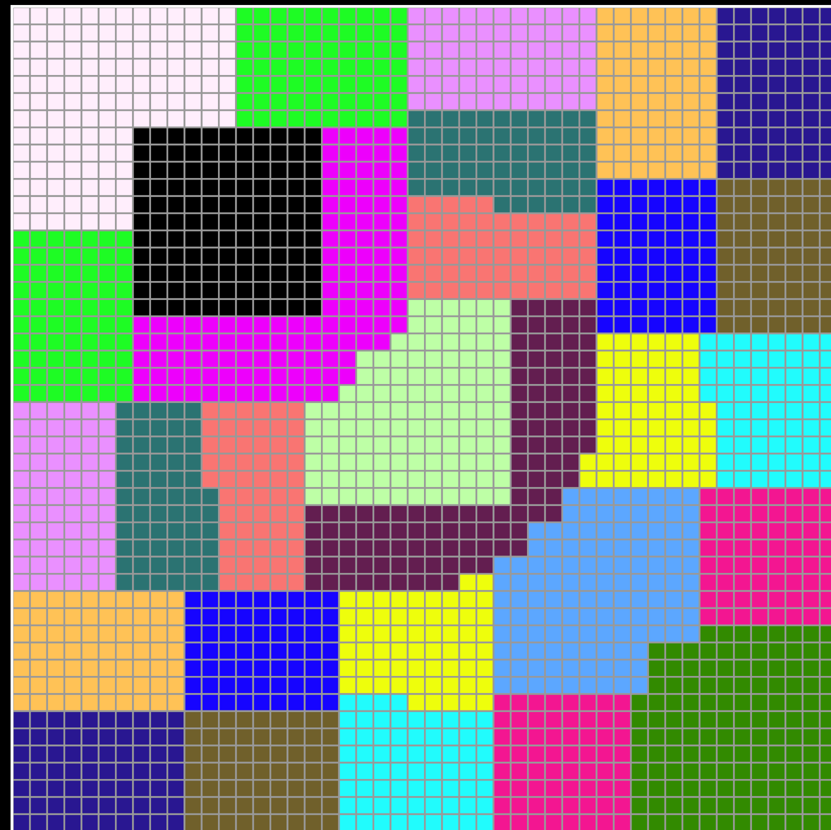


48 x 48 array of data values

Partititioned into 18 subsets for a distributed computing application (subsets shown by cell colors)

Subsets are diagonally symmetric and often disconnected

Matching colors allow viewers to find disconnected pieces of same subset



Goal: Automatically Create Visualizations Like This

Partition sets = vertices in a graph, adjacencies = edges

Choosing colors for sets = embedding graph into color space

All vertices should be well separated from each other
Adjacent vertices should be especially well separated

Outline

The problem

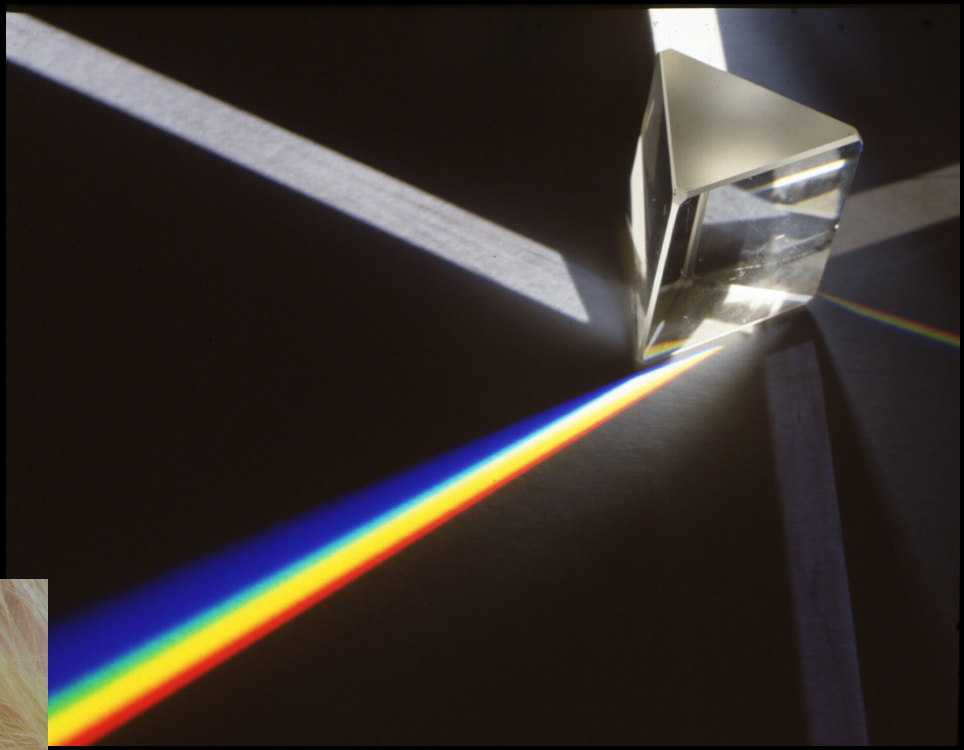
A brief introduction to color spaces

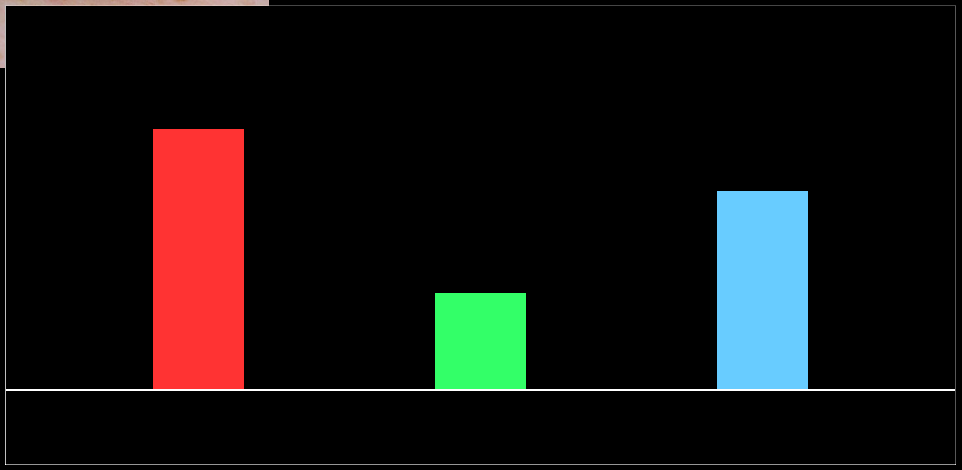
Our solution

Evaluation

Conclusions

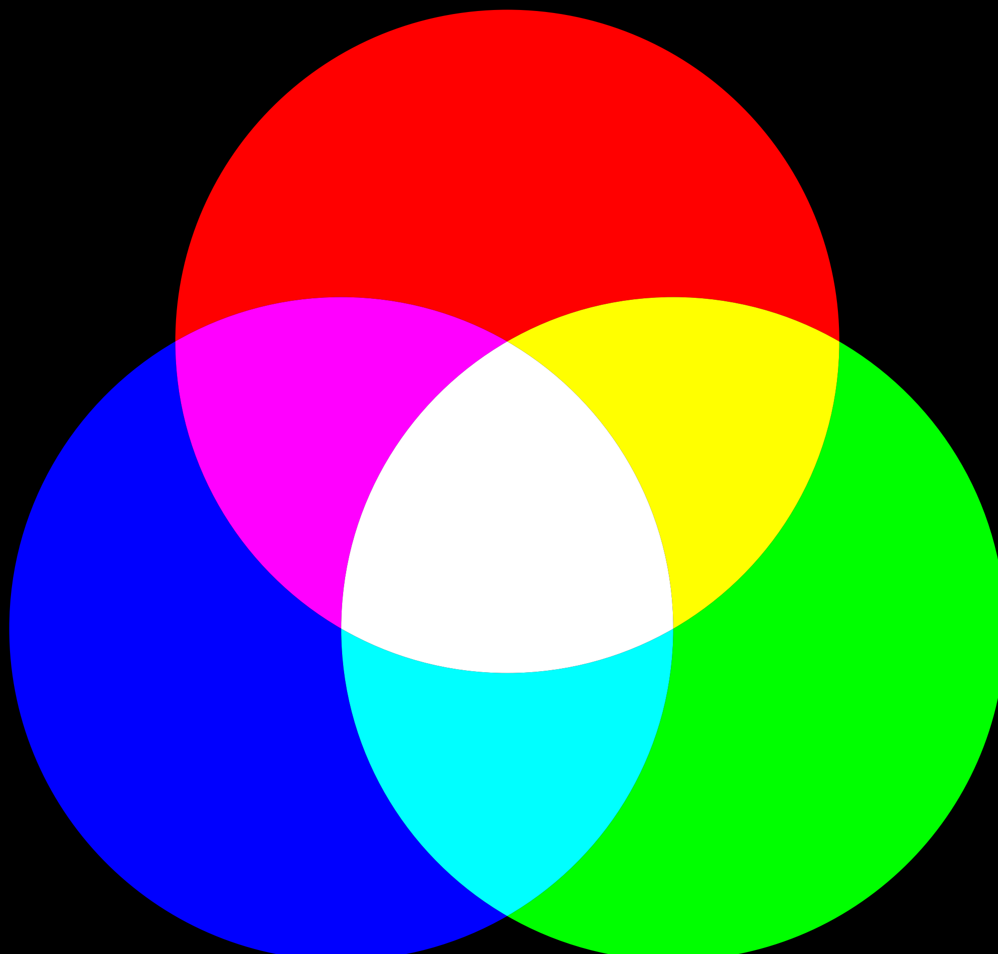
Human color perception is three-dimensional





Additive Color

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



sRGB

Widely used additive color standard for PC displays and web graphics

Color represented as triple (R,G,B) of numbers in range [0,255]

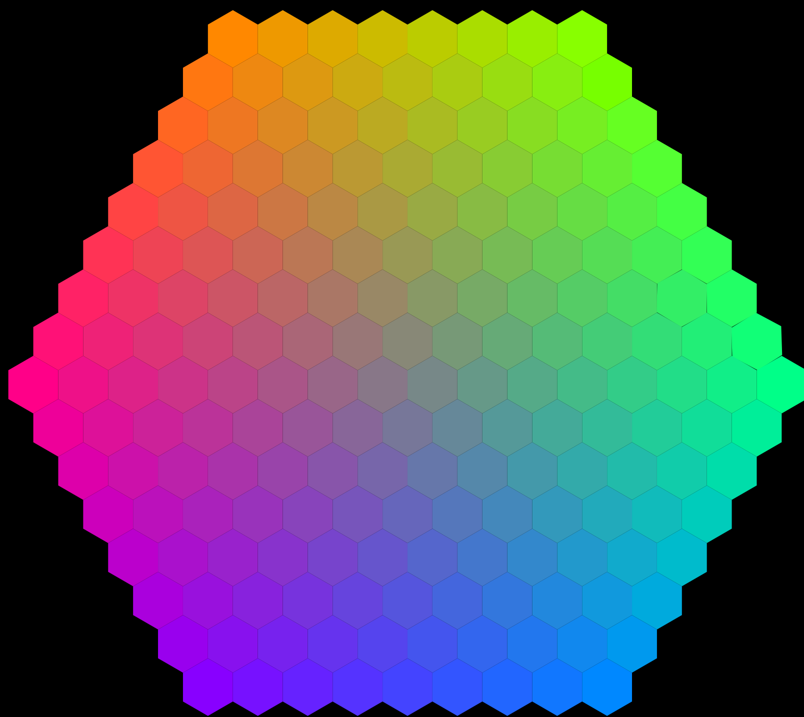
Coordinates transformed nonlinearly then mapped into colors

R: 0 = black, 255 = bright red

G: 0 = black, 255 = bright green

B: 0 = black, 255 = bright blue

Overall color = ∑ (three color channels)



sRGB colors with R+G+B=391

Along the R,G,B axes, coordinates approximate visual similarity but mixed colors with very different coordinates can be very similar

e.g. (0,255,255) ≈ (128,255,255), both cyan

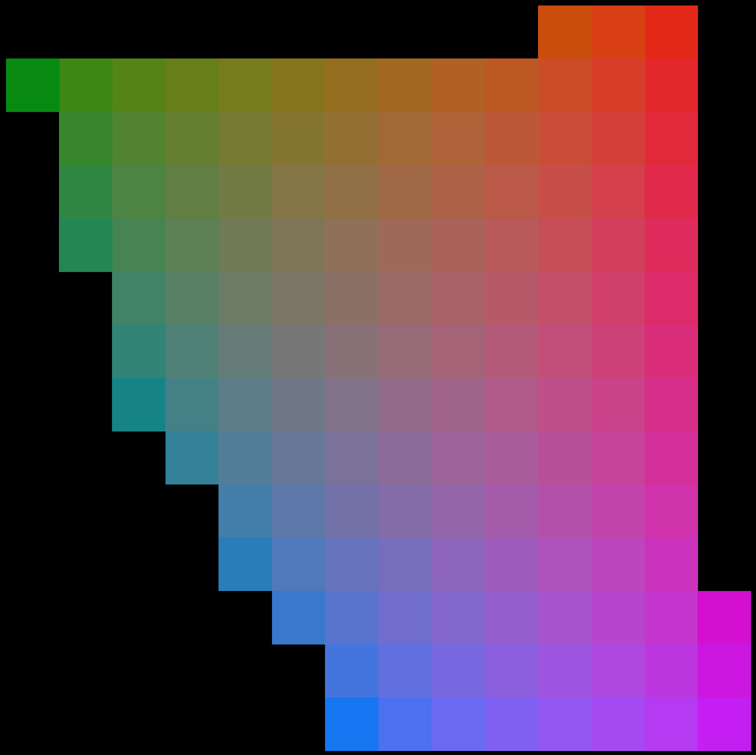
Lab

Non-additive color standard designed to more closely match human vision

Color represented as triple (L,a,b)
L in [0,100], a and b in [-100,100]

L = luminosity (light/dark level)
a,b together specify hue and saturation

Complex formula for transforming
into displayable RGB values



Displayable Lab colors with L=50

Euclidean distances between L,a,b values
give a reasonable approximation to human visual dissimilarity

Outline

The problem

A brief introduction to color spaces

Our solution

Evaluation

Conclusions

Recall our problem:

Embedding a graph into color space
So adjacent vertices are especially far apart

How to find a good embedding?

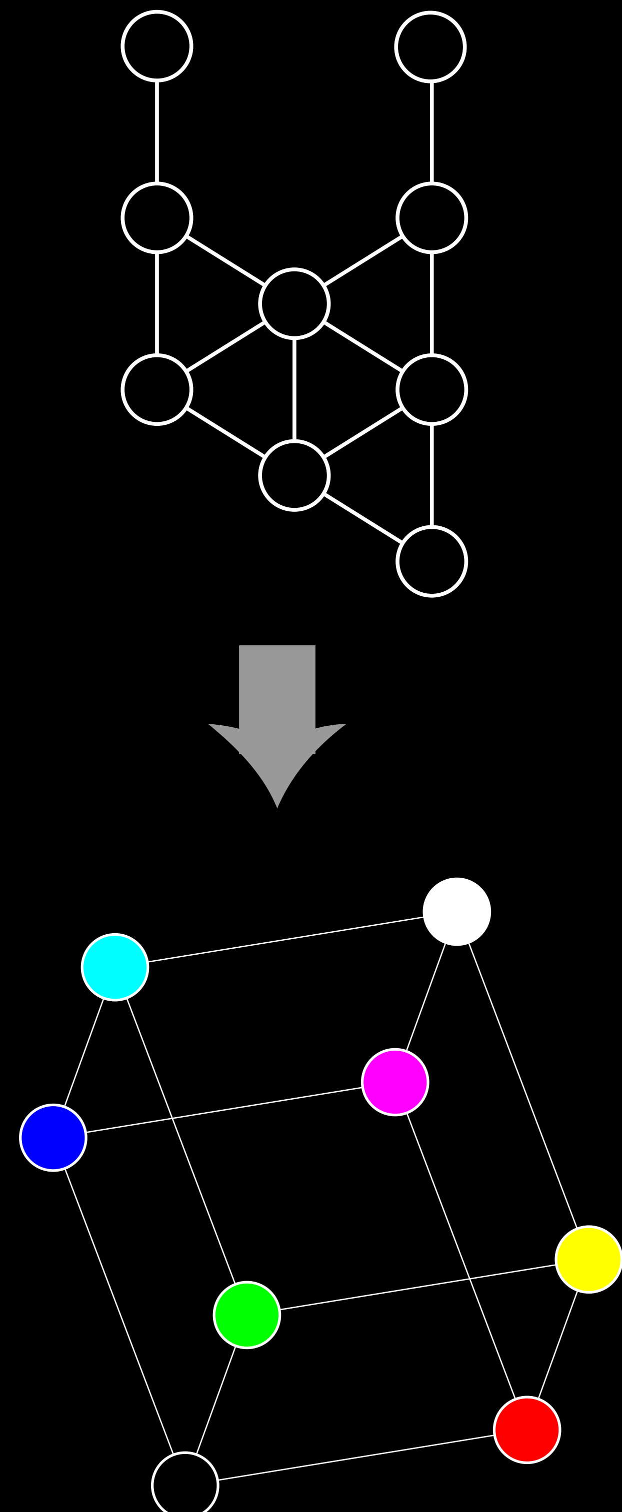
Define numerical measure of embedding quality

$O(1/\text{distance}^4)$ term for each pair of vertices
(high exponent enforces local interactions)

$O(1/\text{distance})$ term for adjacent pairs
(low exponent makes all pairs important)

Normalize so both terms contribute equally

Optimize measure by randomized hill-climbing



Outline

The problem

A brief introduction to color spaces

Our solution

Evaluation

Conclusions

Evaluation: How successful is our method?

Ideal: Human usability studies

Possible future work, beyond scope of this paper

Numerical: compare quality measure scores

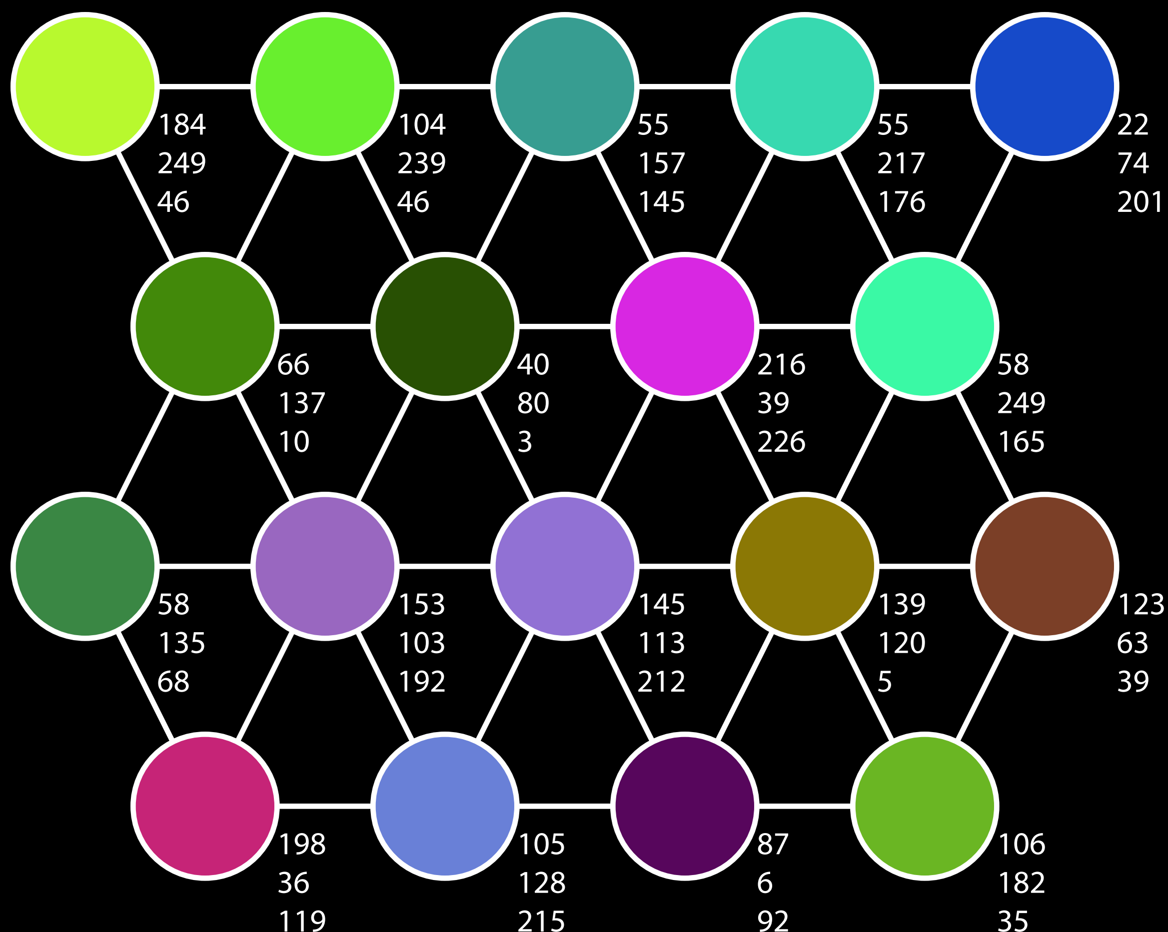
Only makes sense for embeddings in a single color space,
we wish to compare both sRGB and Lab based embeddings

Doesn’t test how well quality measure models the problem

Anecdotal: generate colorings and analyze visually

What we do here

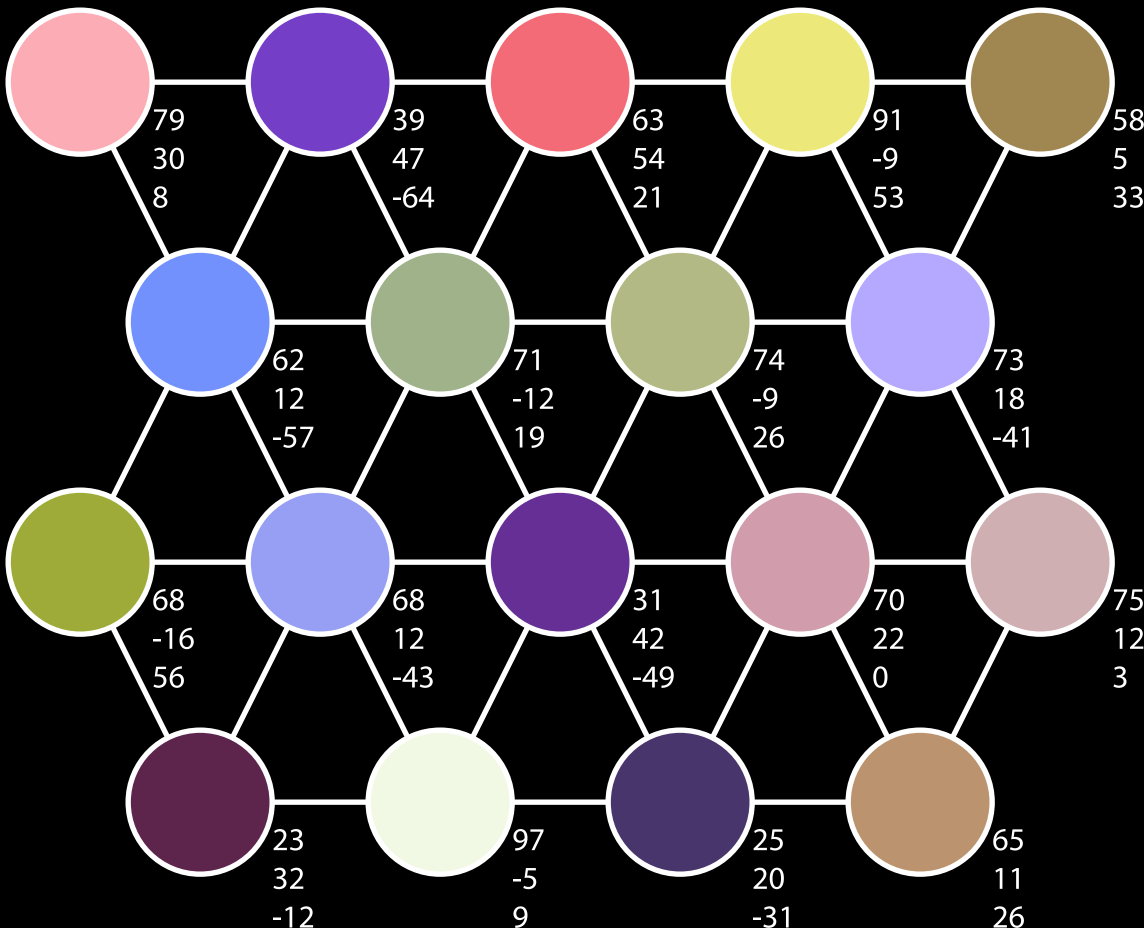
Results: Uniformly random sRGB colors



Many similar pairs of colors

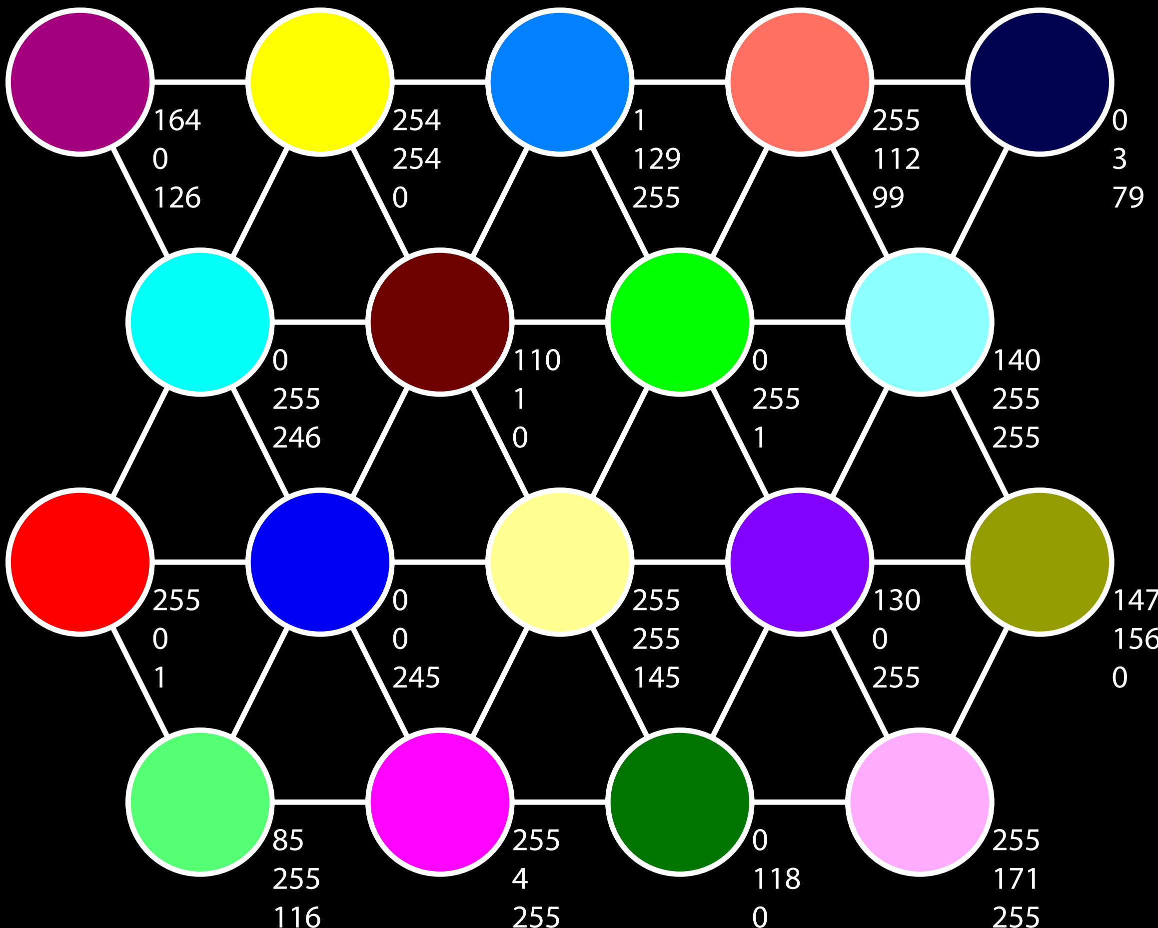
Adjacent pairs of vertices no better than nonadjacent

Results: Uniformly random Lab colors



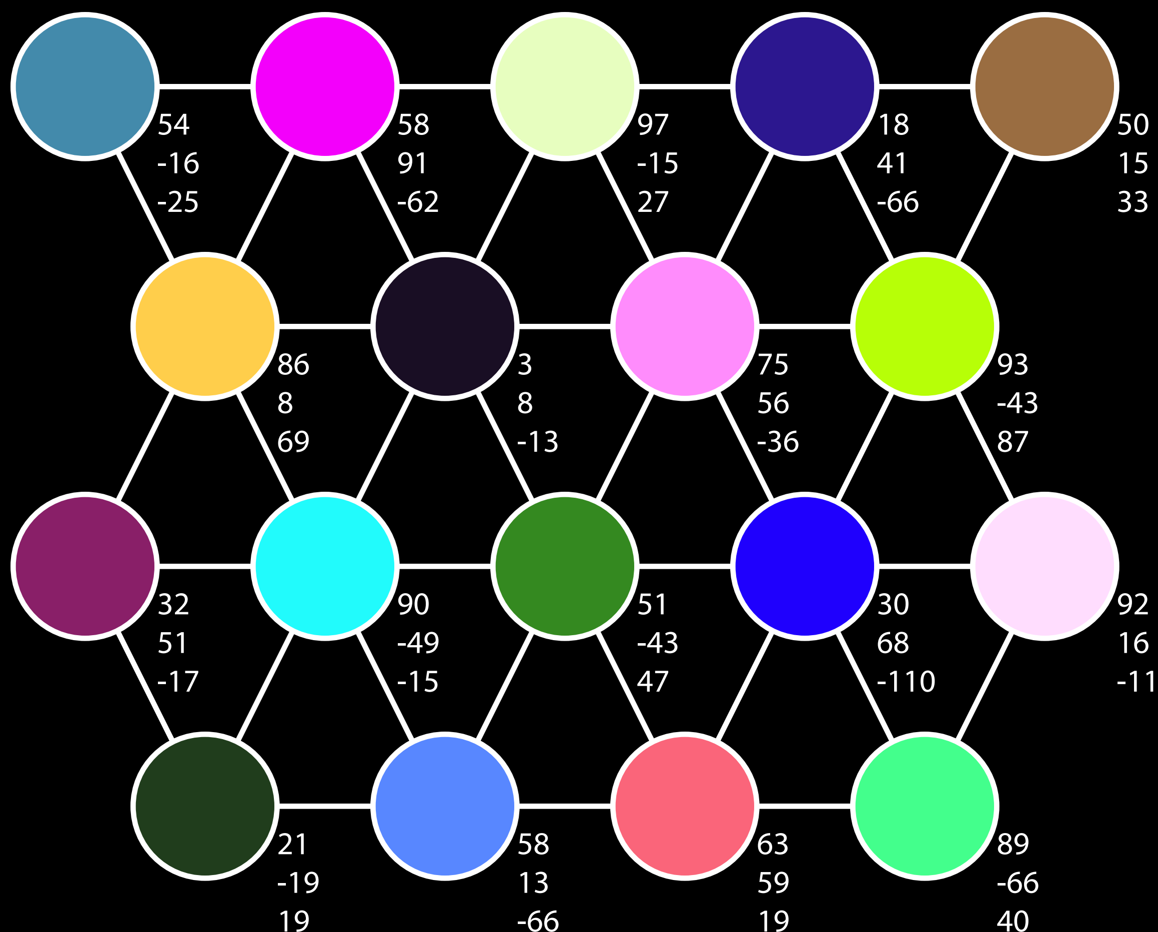
Better distribution of colors than sRGB but still many similar pairs
Adjacent pairs of vertices still no better than nonadjacent

Results: Optimized sRGB colors



Much better separation of adjacent vertices
Still several similar pairs (light green, light blue, yellow, pink)

Results: Optimized Lab colors



Still good separation of adjacent vertices
Even fewer similar pairs (but still some, e.g. pink)

Outline

The problem

A brief introduction to color spaces

Our solution

Evaluation

Conclusions

Conclusions

Have an effective method for automatically coloring drawings

Using Lab seems to be a visible improvement over simpler sRGB version

Still some difficult-to-distinguish color pairs

Future Work

Usability studies?

Embed graph into predetermined color palette?

Exact or approximate embedding maximizing min edge length?