Resolution in Color Filter Array Images

Cobus Heukelman and Jon Peterson

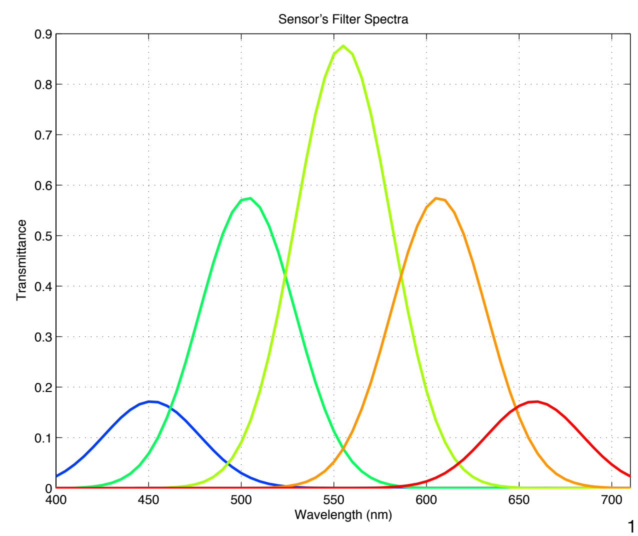


Table of Contents

Problem Statements

- Modern sensors have a surplus of pixels.
- Can we sacrifice spacial resolution to improve perceived color accuracy?
- What do the tradeoffs look like?
- Does adding more color filters always provide a benefit?
- How much color accuracy is gained and how much spacial resolution is lost?

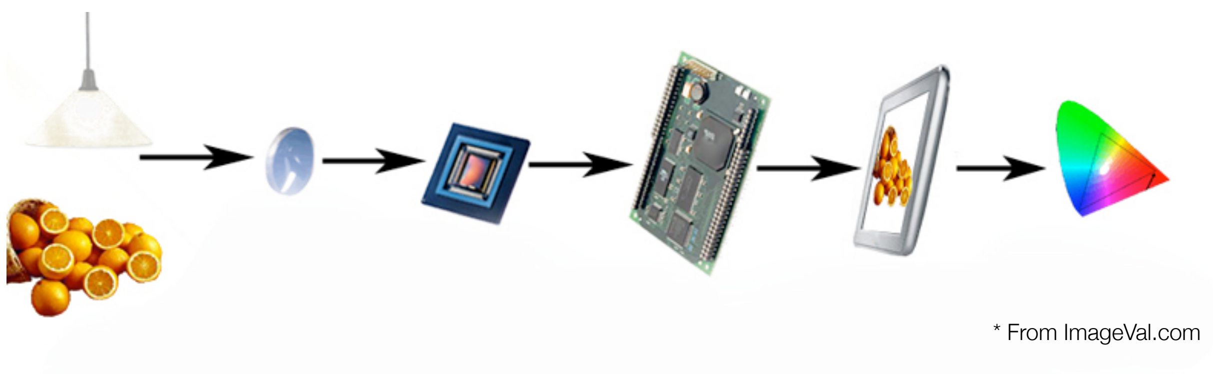
Approaching the Problem

- We used ISET to simulate every aspect of the image pipeline.
- We started by looking at configurations with 3 to 7 different color filters.
- For 3 and 4-color filter arrays, we used a 2 x 2 grid.
- For 5, 6, and 7-color filter arrays, we used a 3 x 3 grid.
- A MacBeth Color Checker (MCC) chart was used to determine color accuracy (average Delta-E).
- An ISO 12233 “Slanted Bar” chart was used to determine spacial frequency response (MTF-50).

The Image Pipeline

The general flow of the test program is as follows:

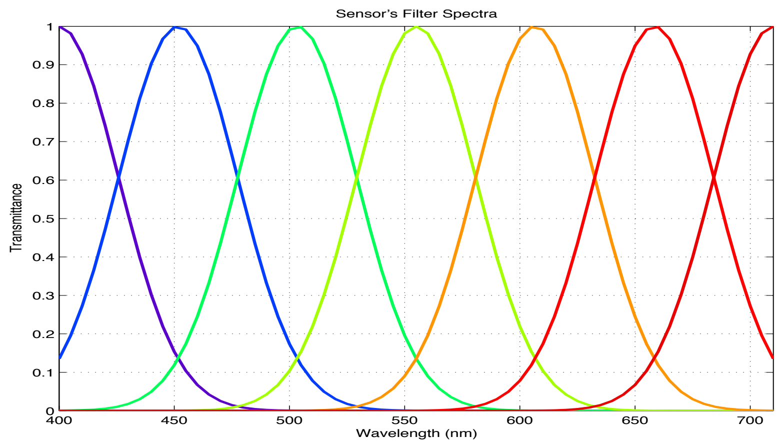
1. The scene is defined (MTF, MCC).
2. The optics are defined.
3. The sensor configurations are defined.
4. Post processing occurs (demosaic, white balance, etc)
5. Metrics are computed and saved.



* From ImageVal.com

Defining Filter Colors - Brute Force Simulation

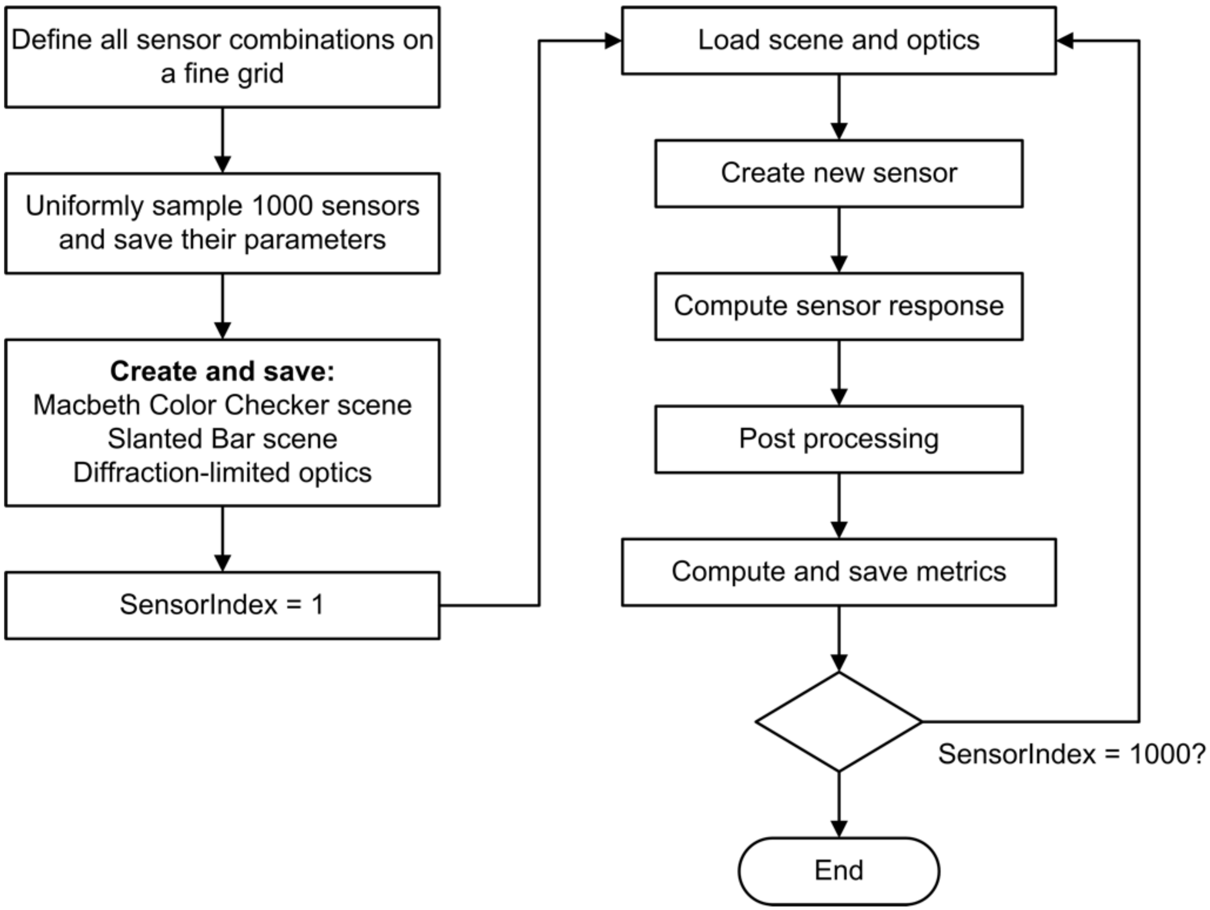
- Number of color filters varied from 3 to 7
- The shape of each filter was defined to be gaussian
- The center wavelength position of the filter varied
- The width of the image filter varied



Defining Filter Colors - Brute Force Simulation

- For each color filter array, the simulation took between 2 and 6 hours.
- For a 5-color sensor with 3 width combinations and 8 filter centers gives roughly 50M sensor configurations.
- Due to the long simulation time, it was necessary to restrict the number of variations we made between each test.
- A large table of possible combinations was first generated and 1000 combinations were uniformly sampled from this list for use in the simulation.

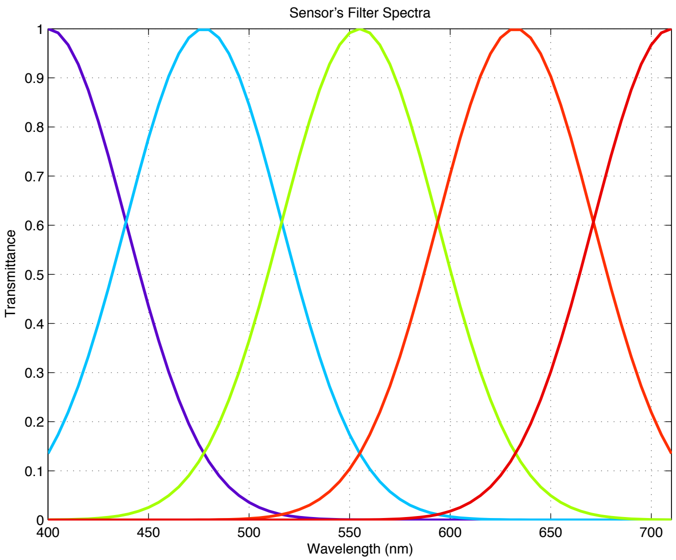
Defining Filter Colors - Brute Force Simulation



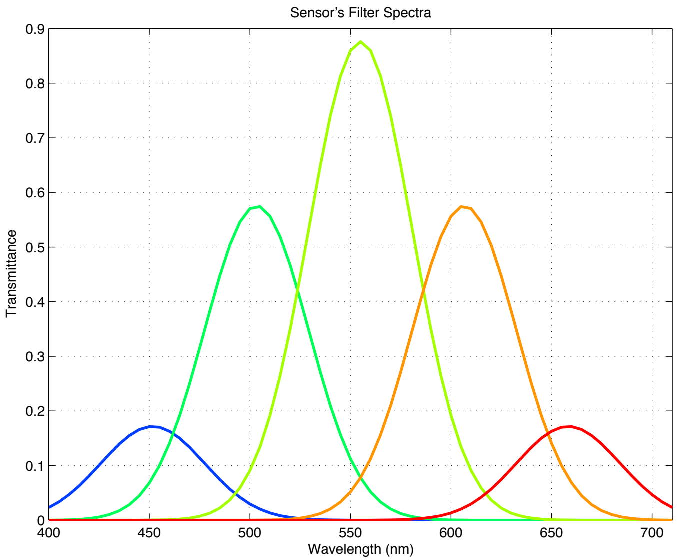
Defining Filter Colors - Theoretical Methods

- In order to match the sensitivity of the eye across the entire visible spectrum, we created a few design rules which limited the overall spectral response of the filter array to be gaussian.
- As a baseline, filter arrays with uniform spacing, widths and magnitudes were used.
- One set of design rules kept the same uniform spacing of the baseline model, but allowed the magnitudes (amount of transmittance) to vary.
- Another set of design rules allowed the center wavelength, filter widths, and amplitudes to vary.

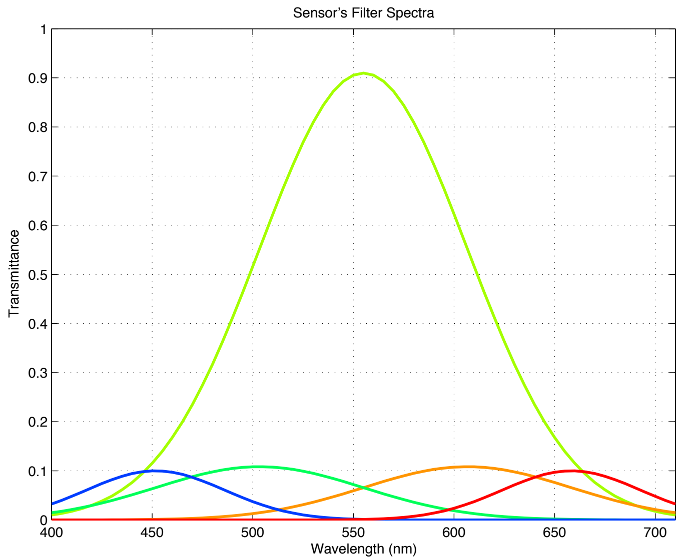
Defining Filter Colors - Theoretical Methods



Uniform



Height-Scaled

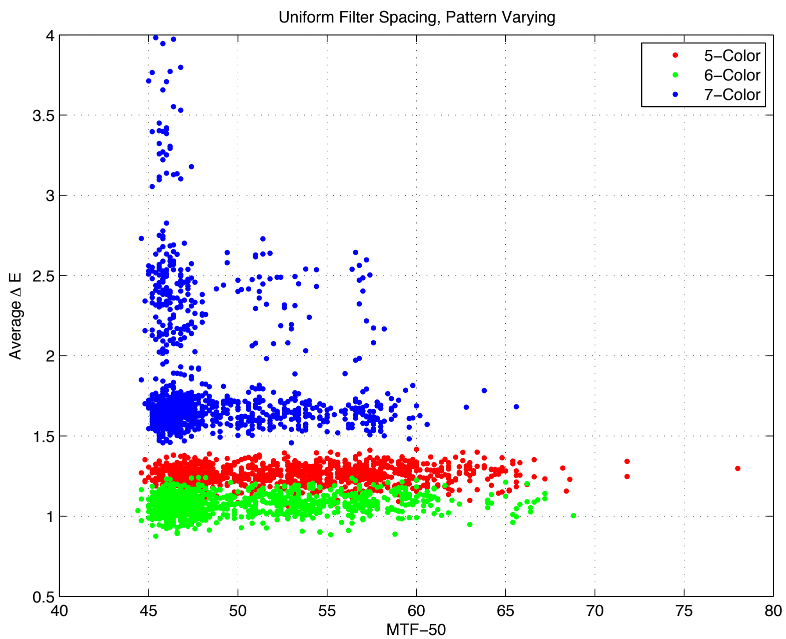


All Scaled

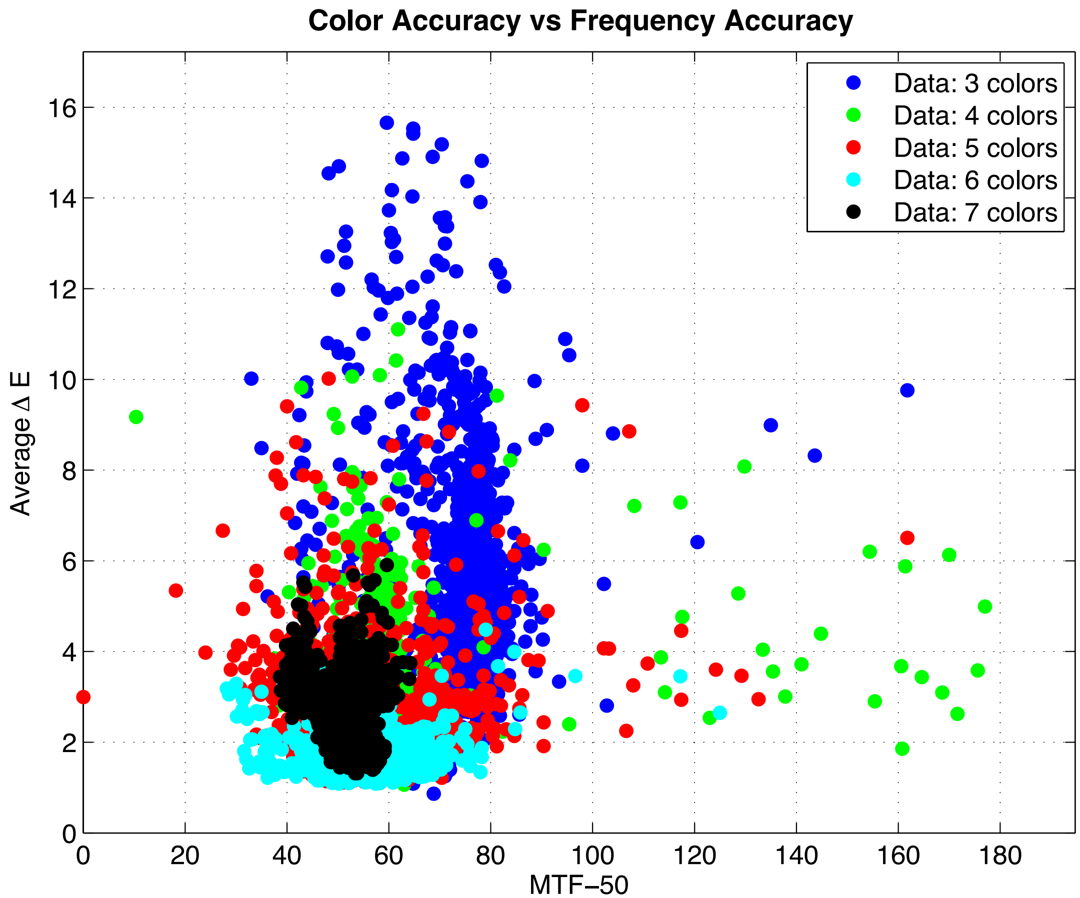
The "all scaled" method fits very well to the gaussian shape, but the design is still sub optimal.

Defining Spatial Patterns

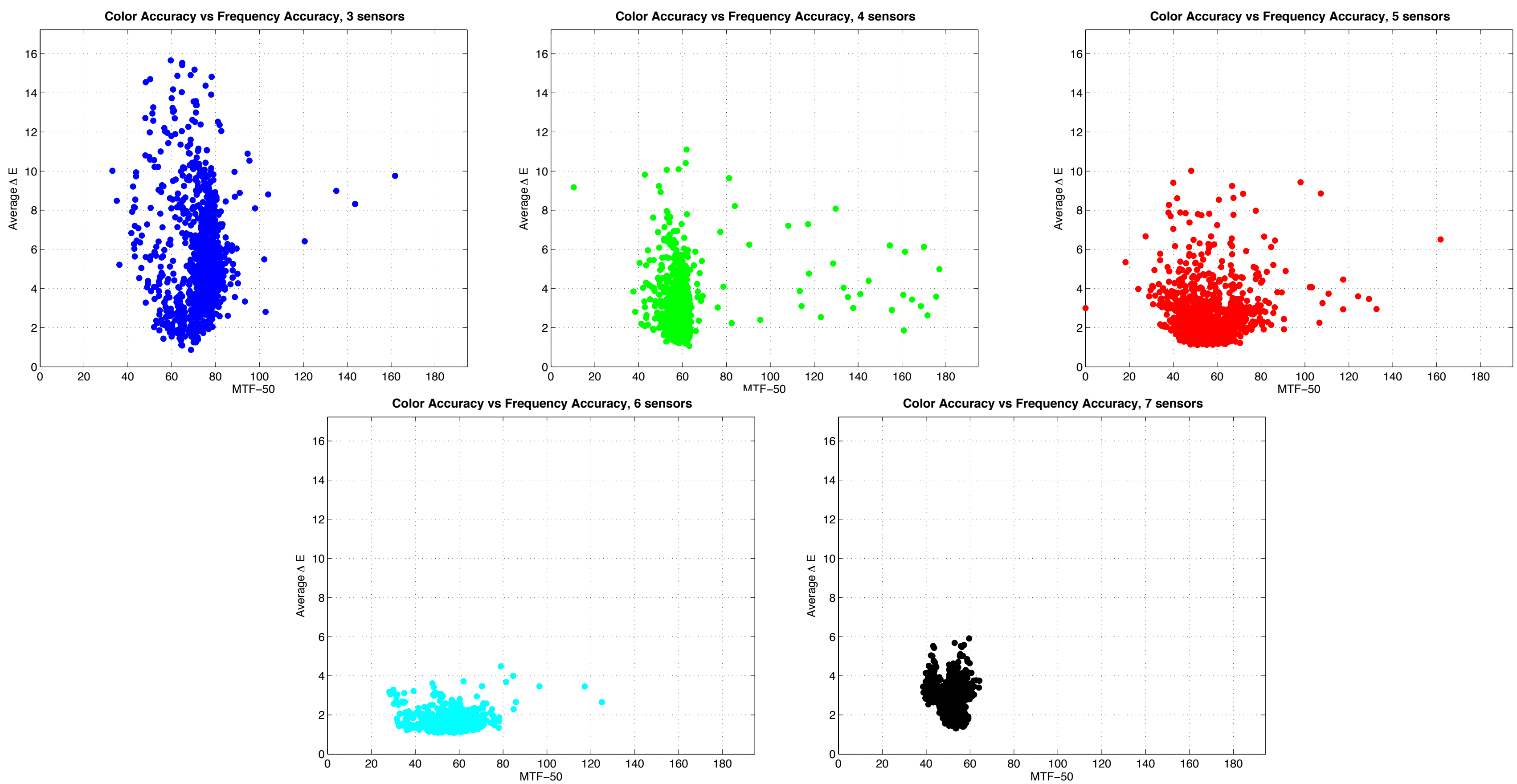
- In order to determine the best spatial pattern, simulations were run with 1000 randomly generated patterns.
- Both MTF-50 and Average Delta-E values were collected for each pattern.



Defining Filter Colors - Brute Force Results



Defining Filter Colors - Brute Force Results

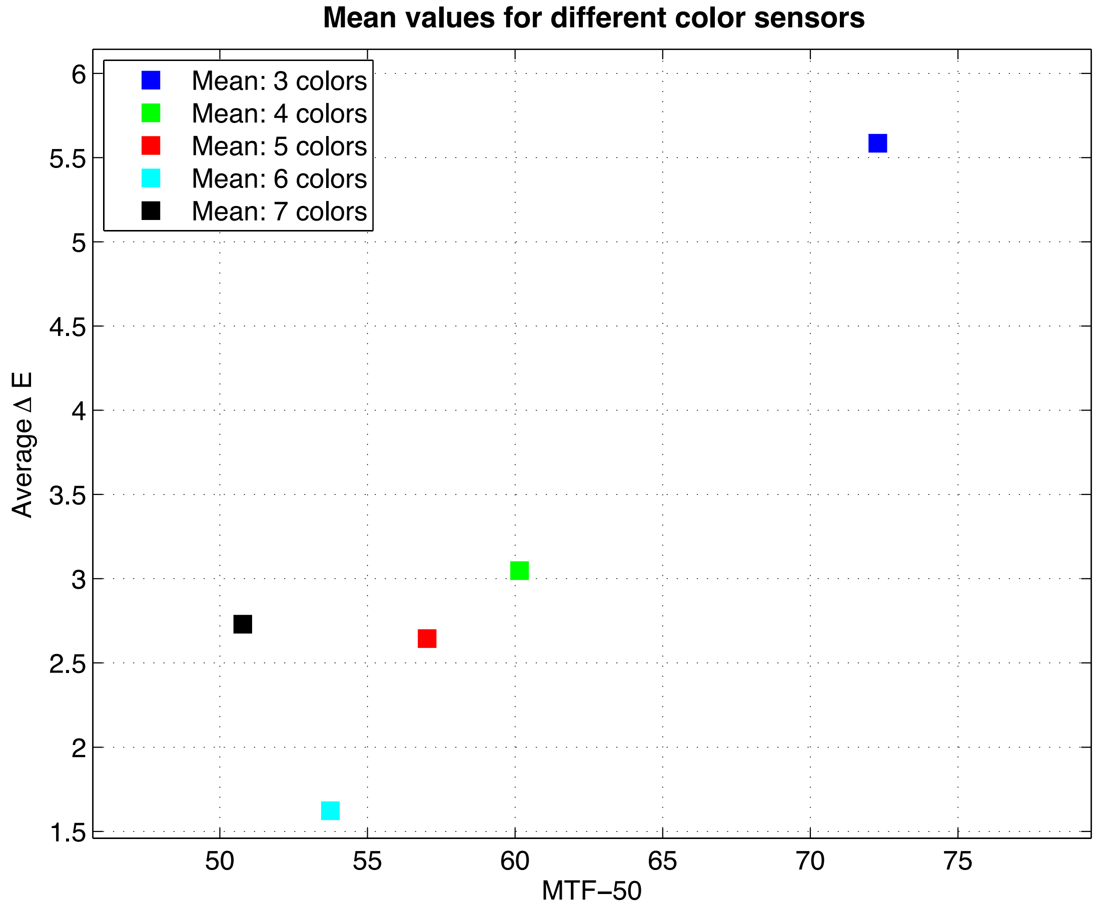


Defining Filter Colors - Brute Force Results

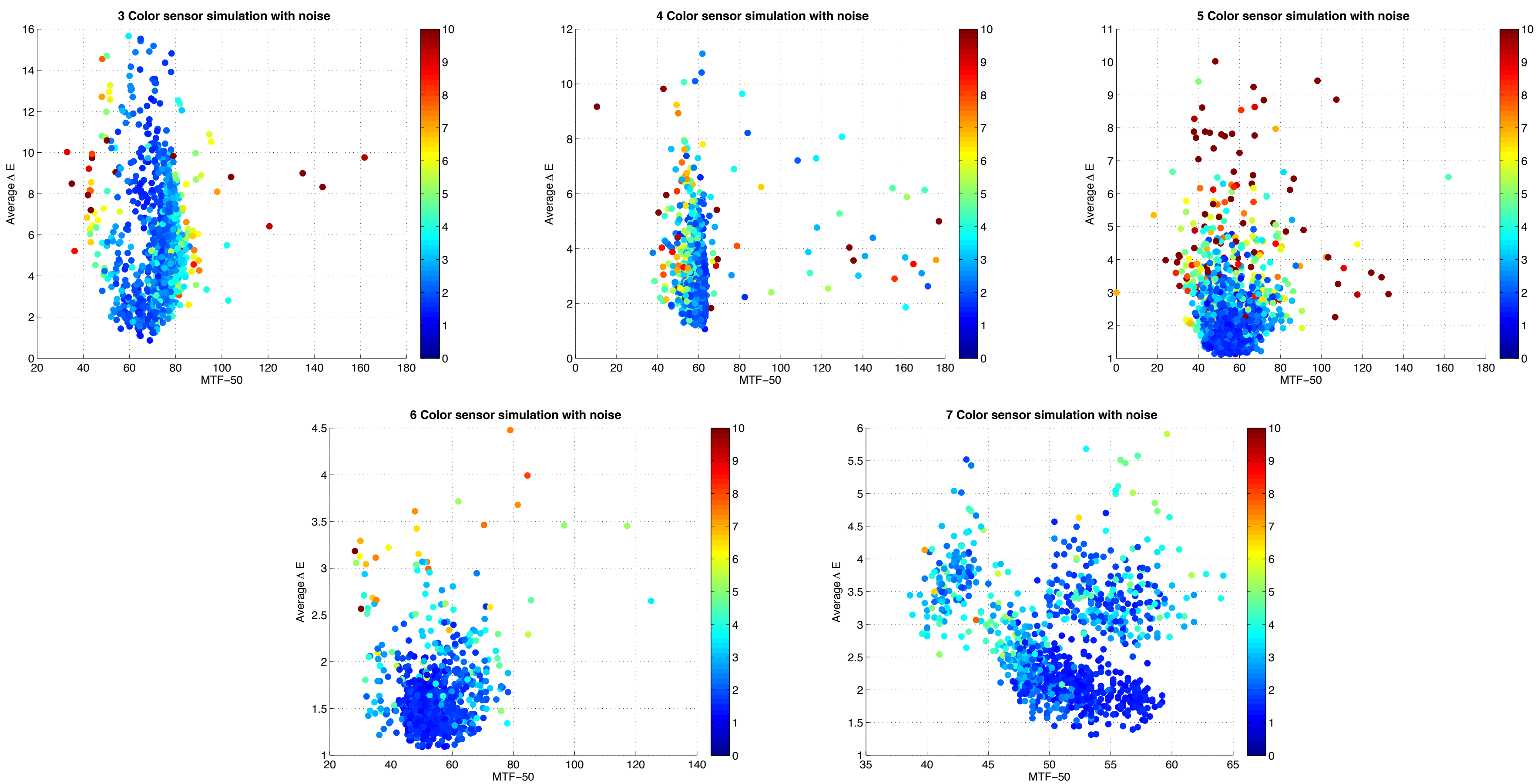
- 3-color sensor: Delta-E between 1 and 16, MTF between 30 and 160.
- 4-color sensor: Lower variance in MTF-50 than 3-color sensor
- 5-color sensor: Loss in spatial resolution compared to 3 and 4-color sensor
- 6-color sensor: Better color performance, lower variation in delta-E
- 7-color sensor: Worse delta-E than 6-color sensor, but smaller variation in MTF-50

Defining Filter Colors - Brute Force Results

- The means clearly show the centers of the distributions.



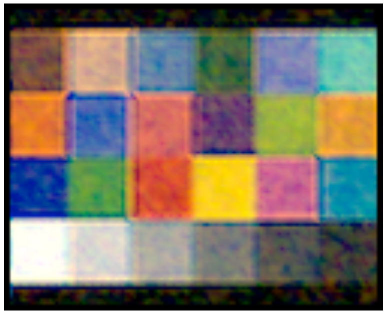
Defining Filter Colors - Brute Force Results



Defining Filter Colors - Brute Force Results

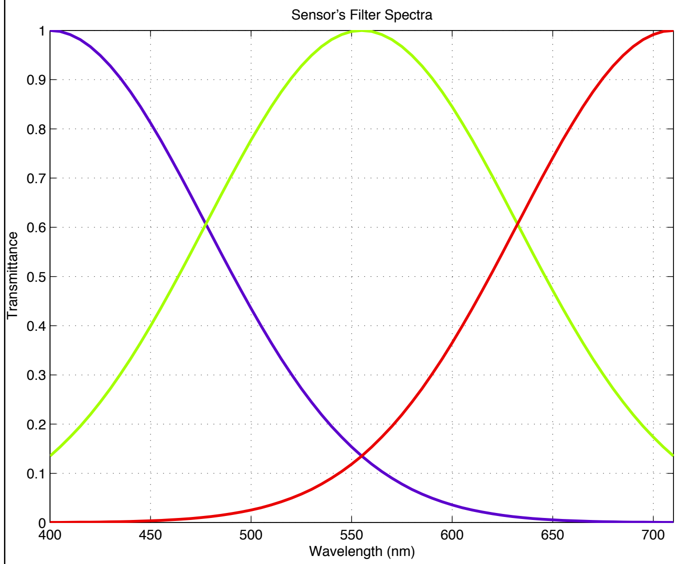
- Luminance noise causes a wide range of brightness values in each color patch.
- $\text{Luminance Noise} = \dfrac{100 \cdot std(Y)}{mean(Y)}$
- Samples further from the mean generally have higher luminance noise.

Lower Noise (2.55)

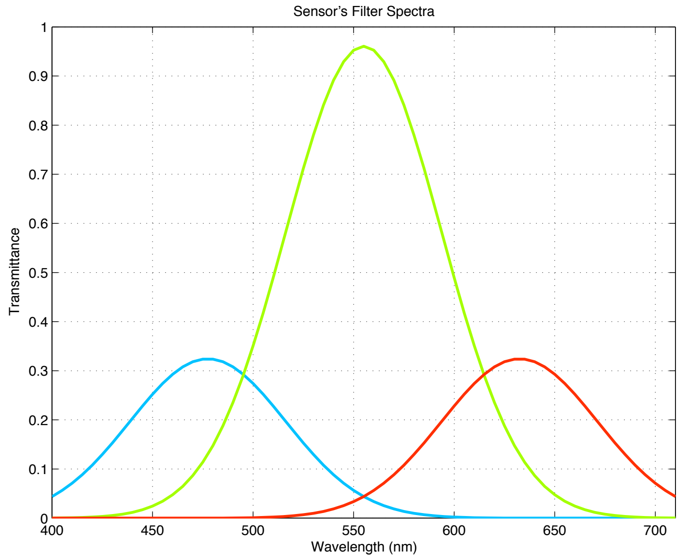


Higher Noise (15.1)

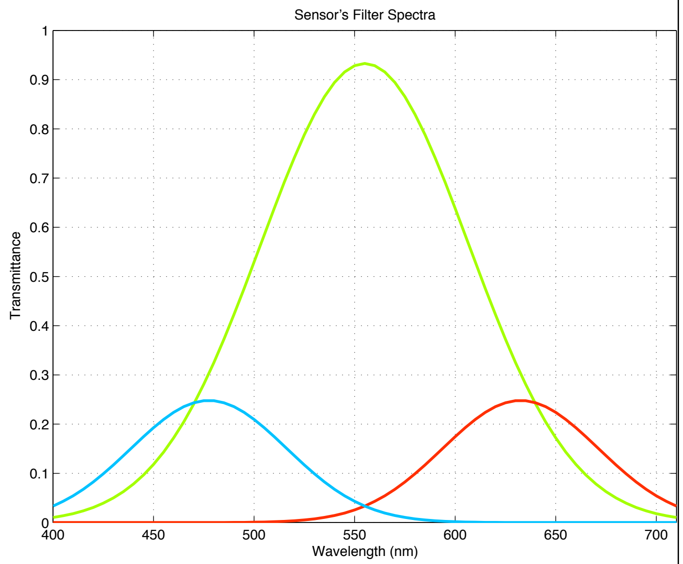
Defining Filter Colors - Theoretical Results



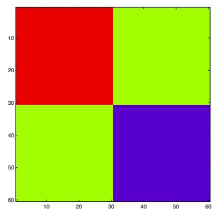
3 Color Uniform



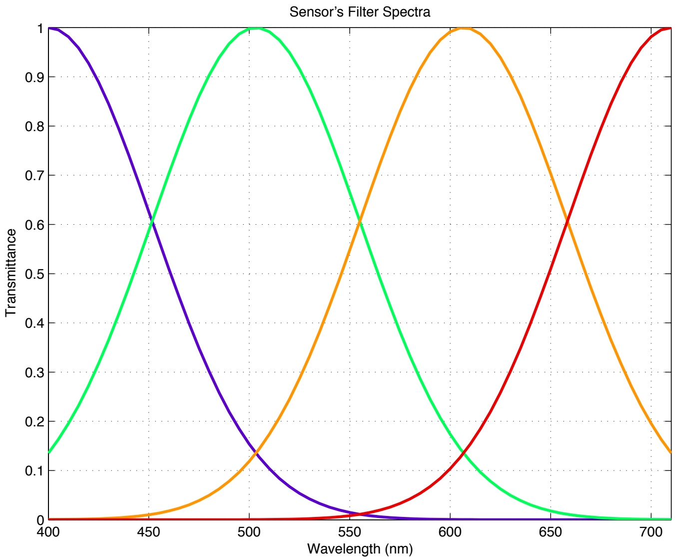
3 Color Height-Scaled



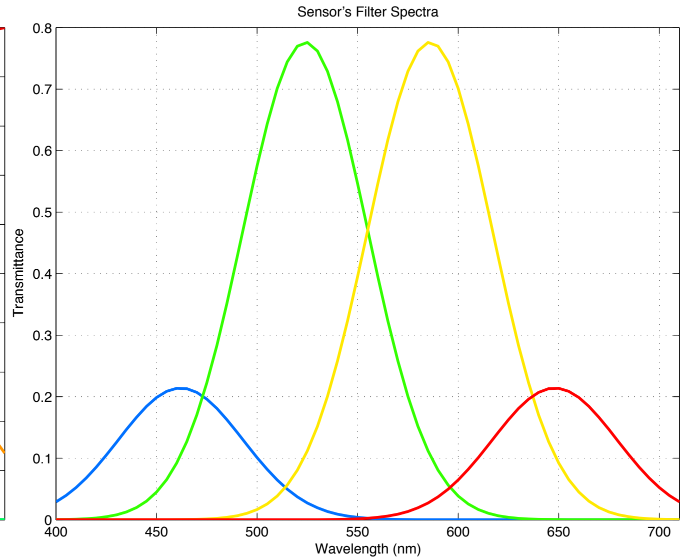
3 Color All-Scaled



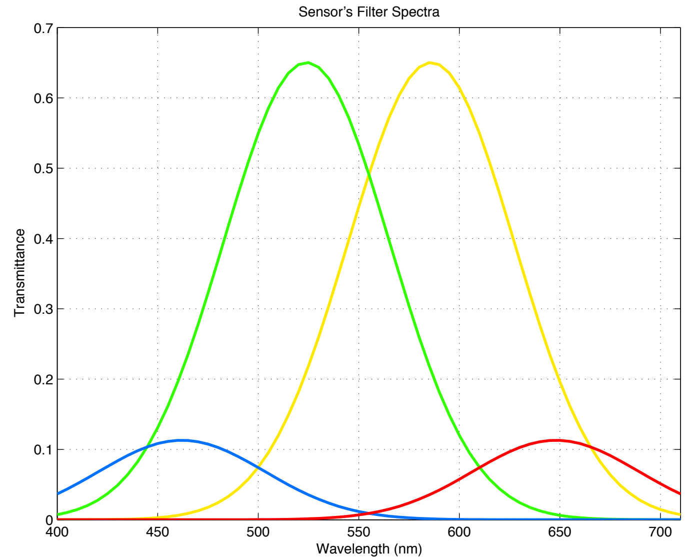
Defining Filter Colors - Theoretical Results



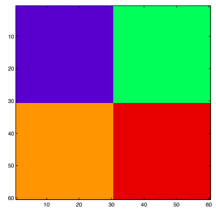
4 Color Uniform



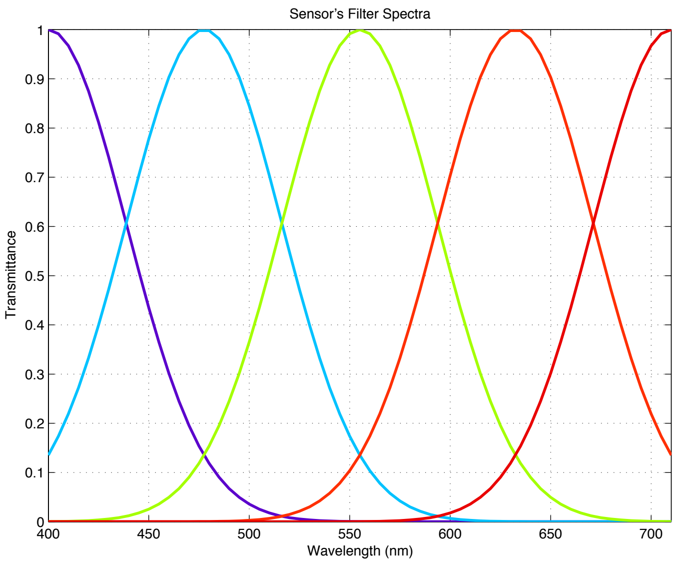
4 Color Height-Scaled



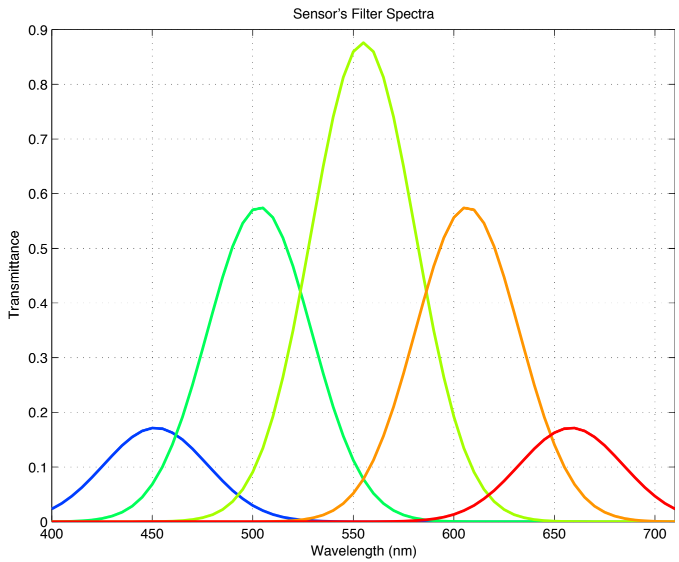
4 Color All-Scaled



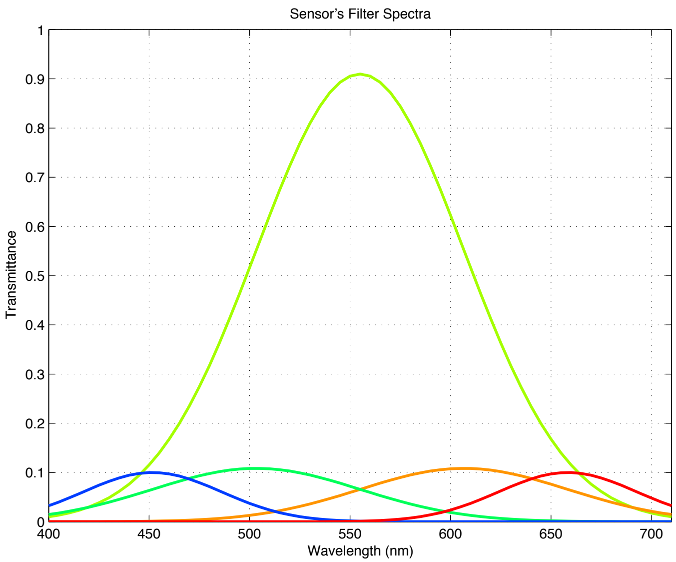
Defining Filter Colors - Theoretical Results



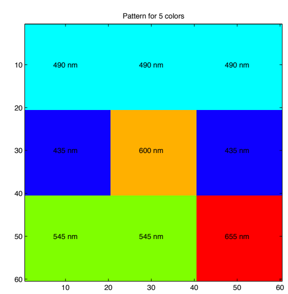
5 Color Uniform



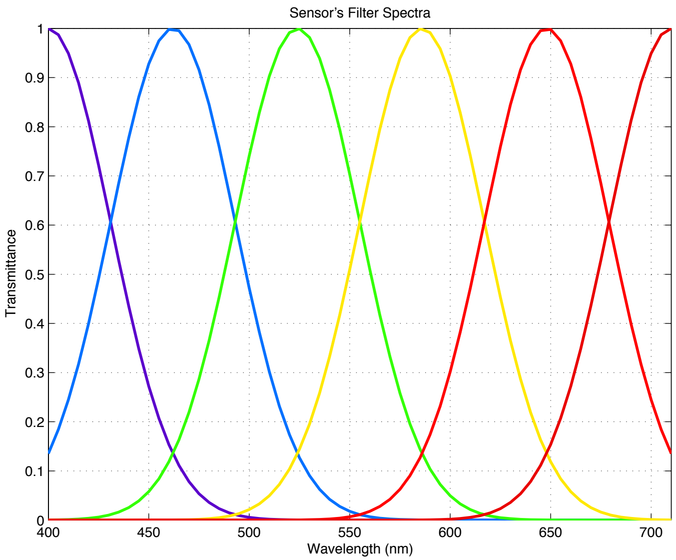
5 Color Height-Scaled



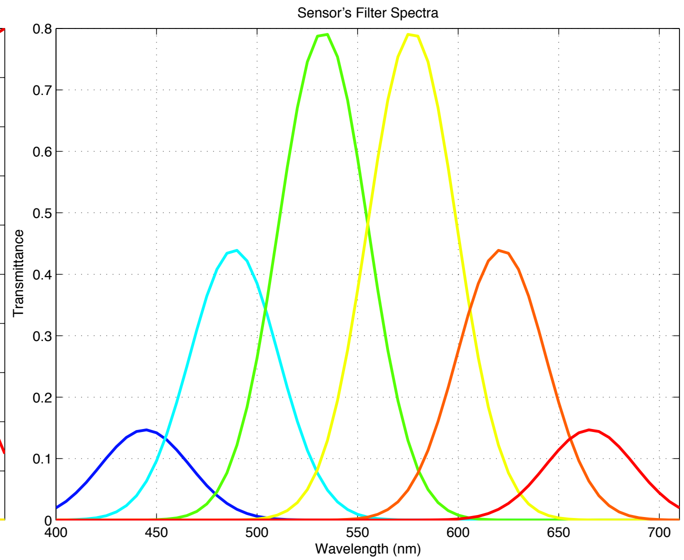
5 Color All-Scaled



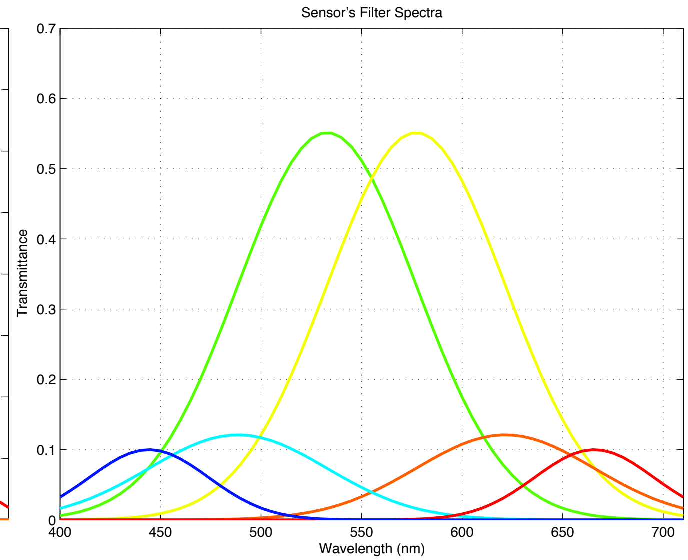
Defining Filter Colors - Theoretical Results



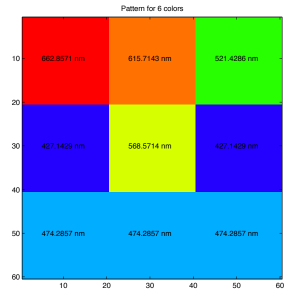
6 Color Uniform



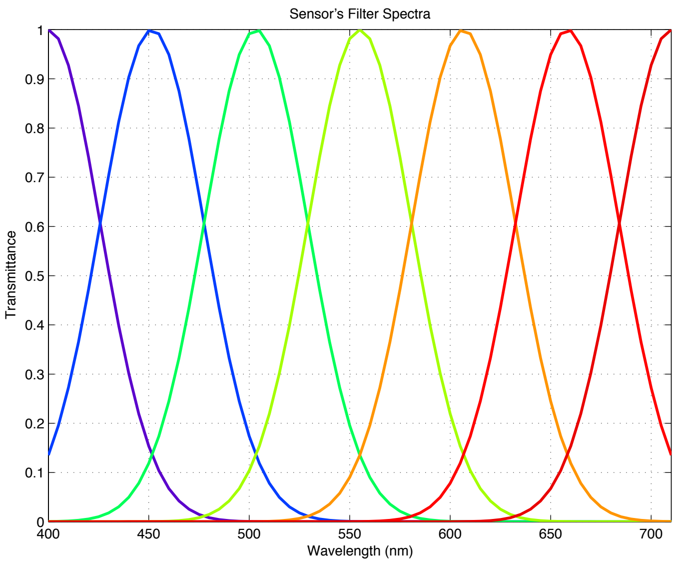
6 Color Height-Scaled



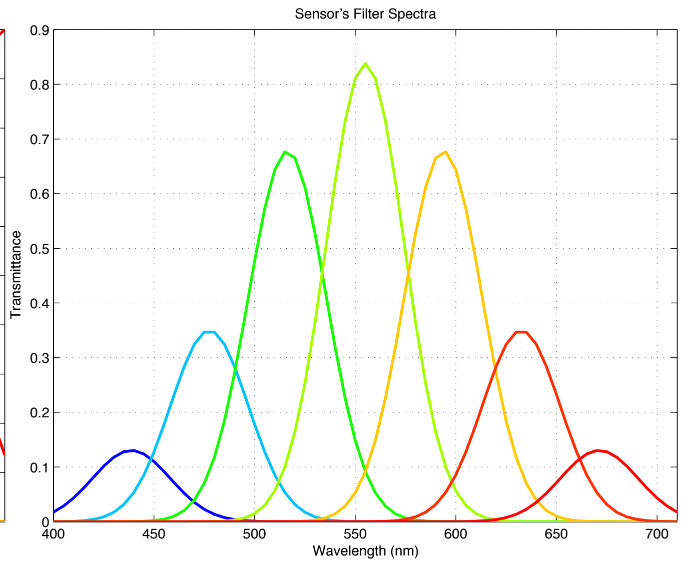
6 Color All-Scaled



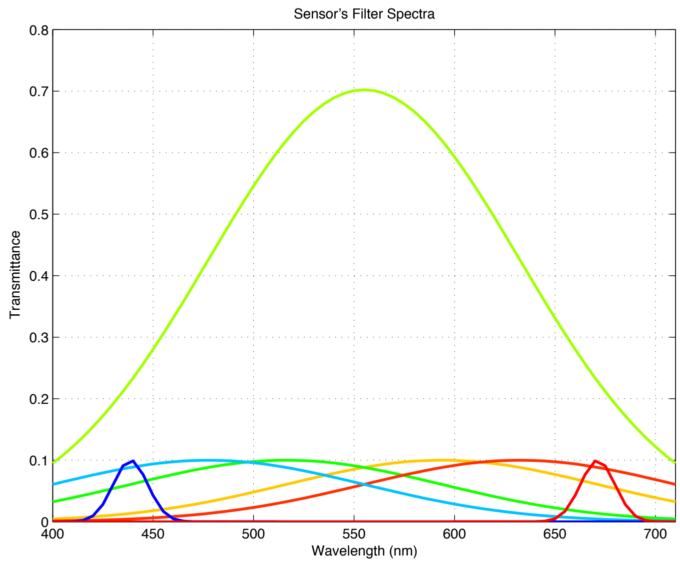
Defining Filter Colors - Theoretical Results



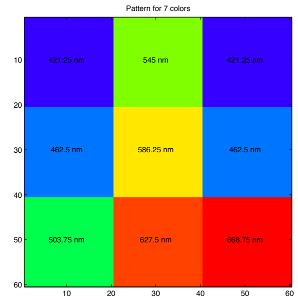
7 Color Uniform



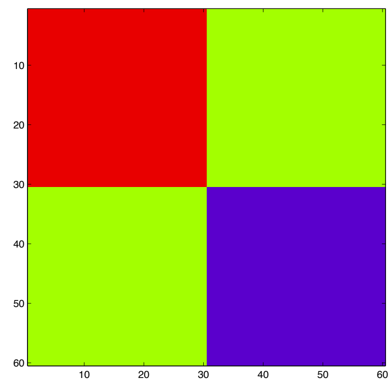
7 Color Height-Scaled

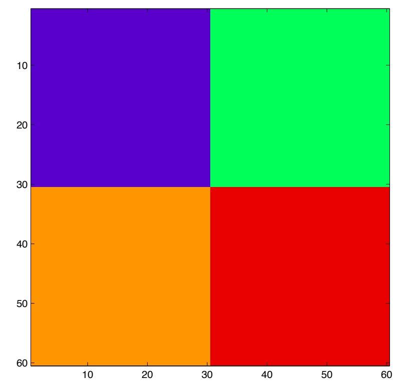


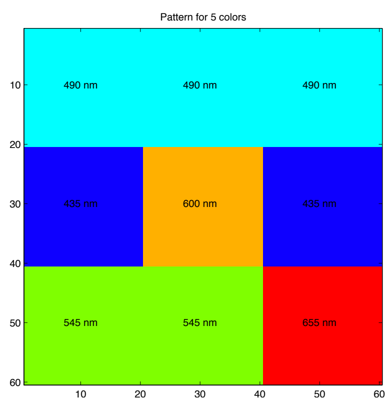
7 Color All-Scaled

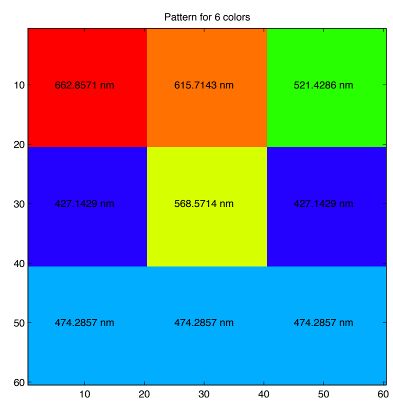


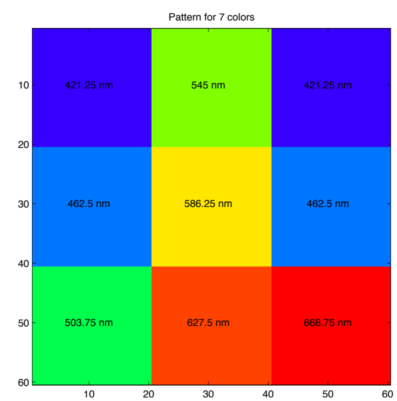
Defining Spatial Patterns - Results



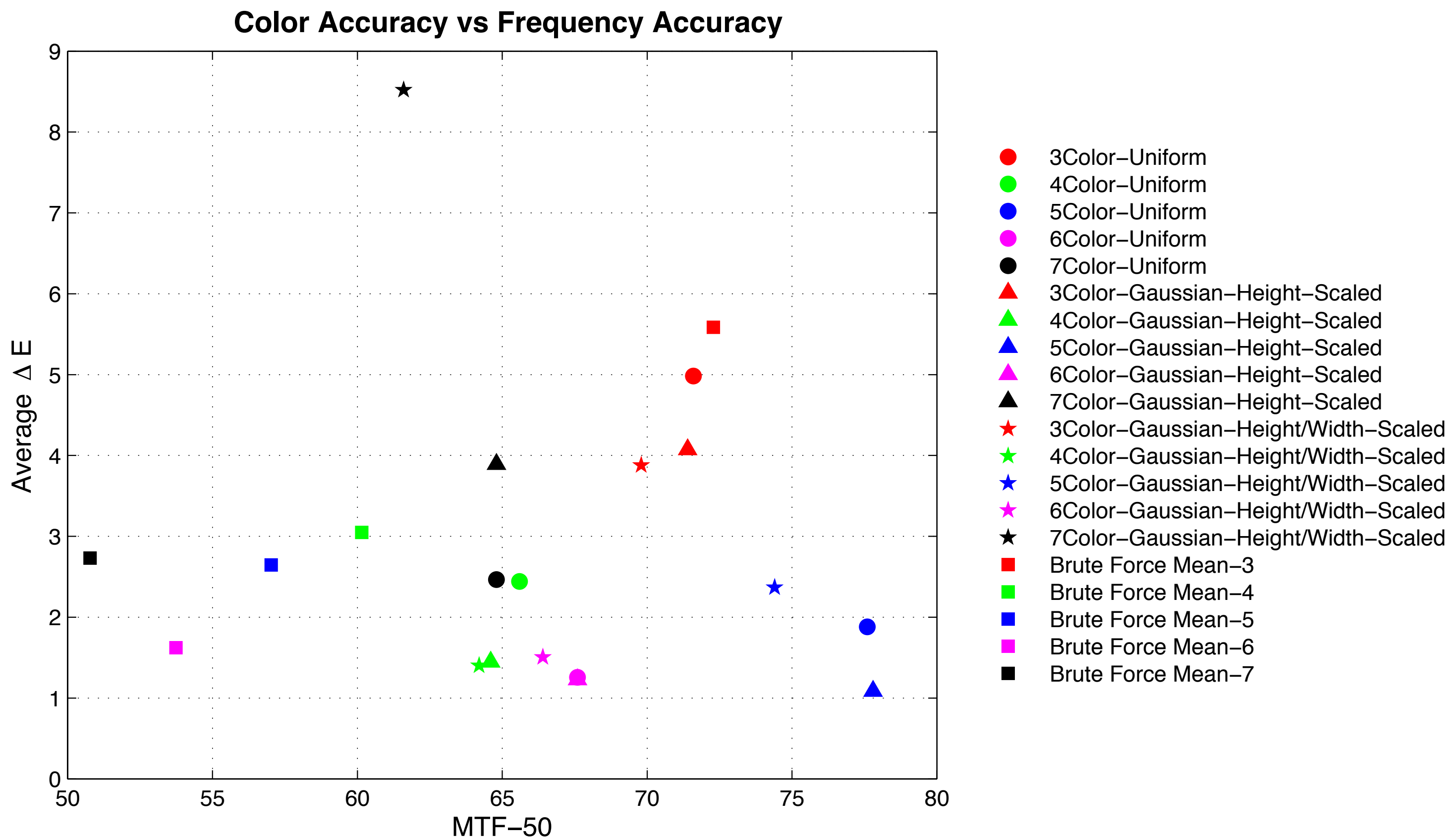




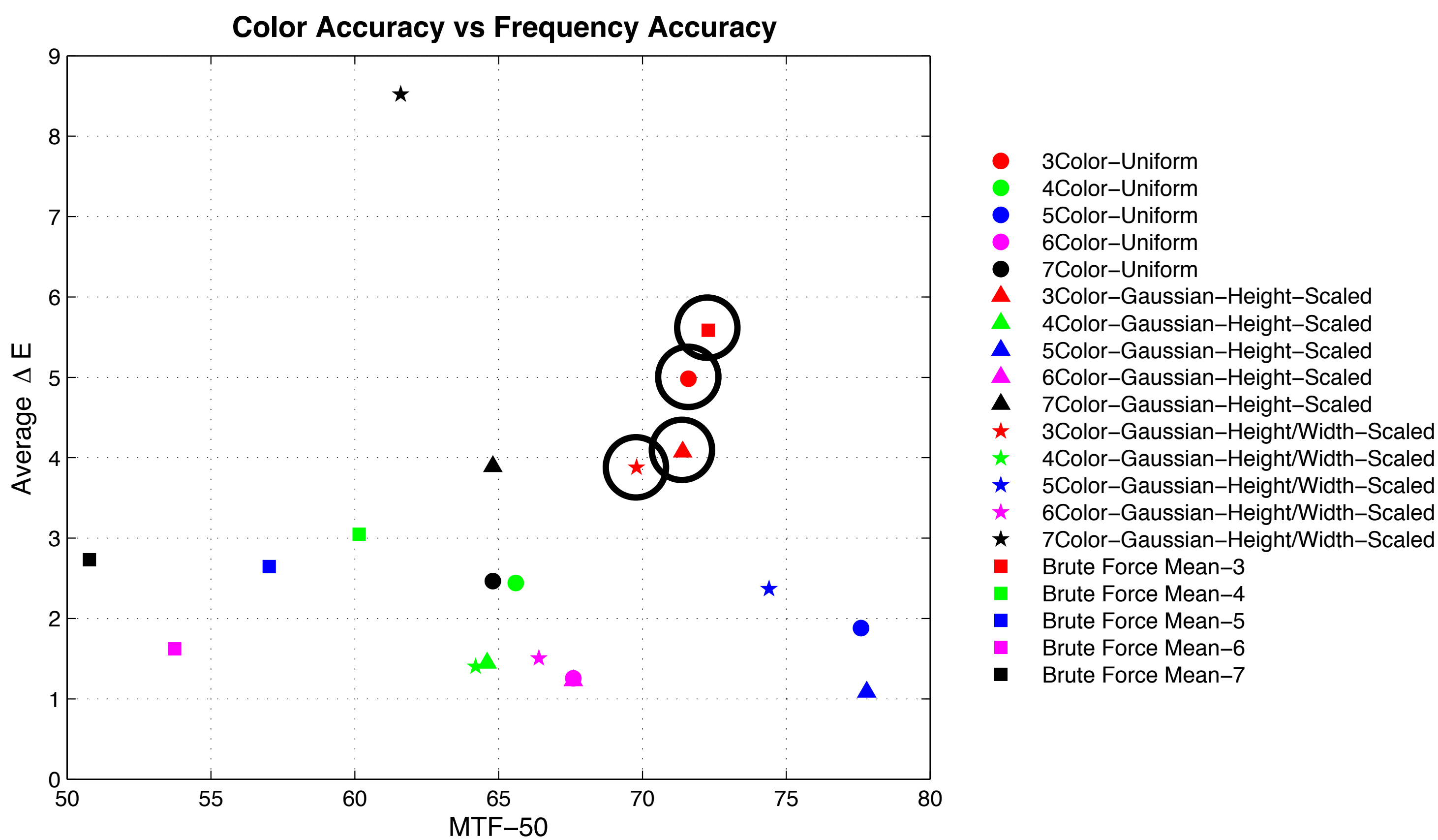




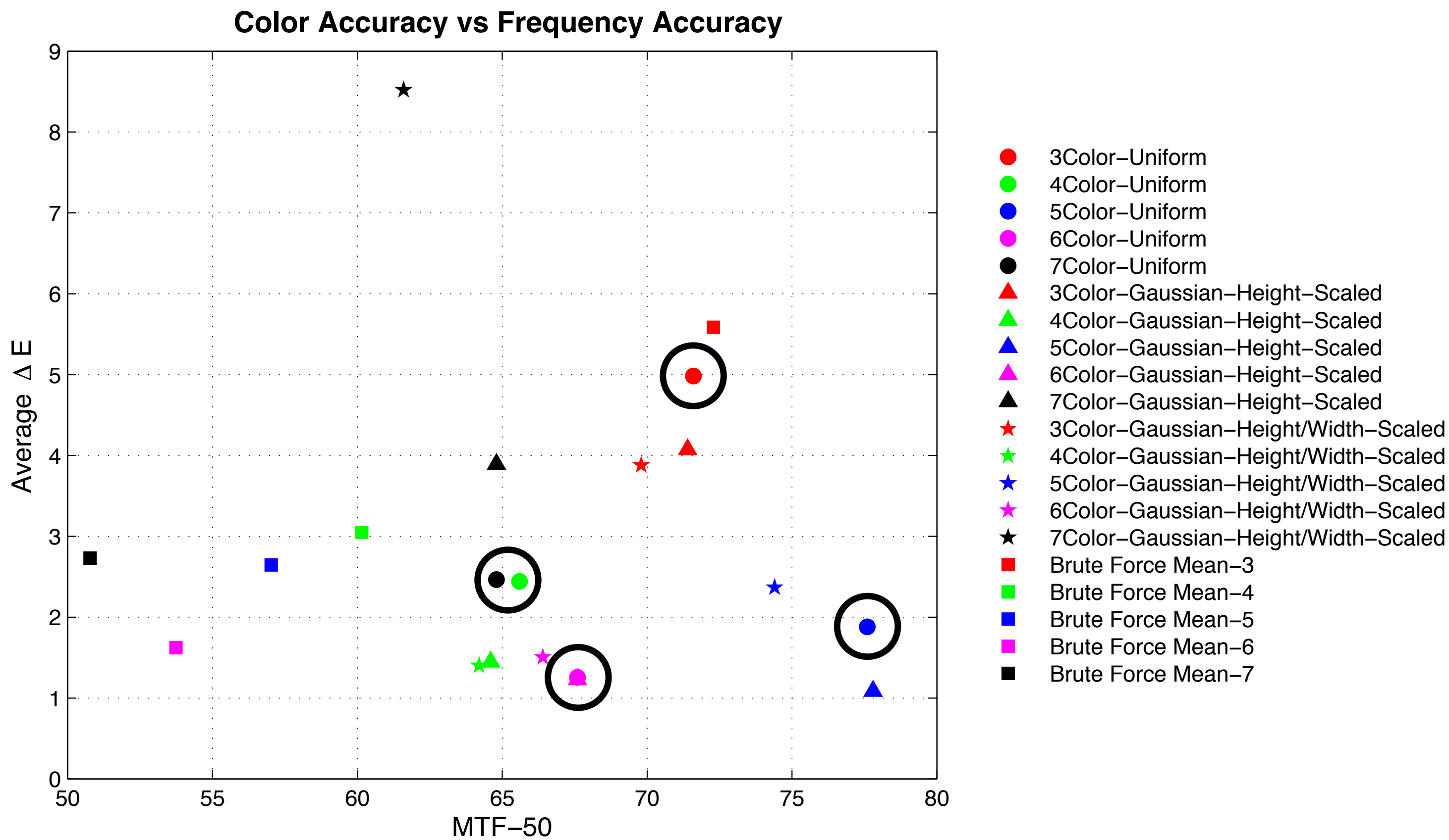
Defining Filter Colors - Theoretical Results



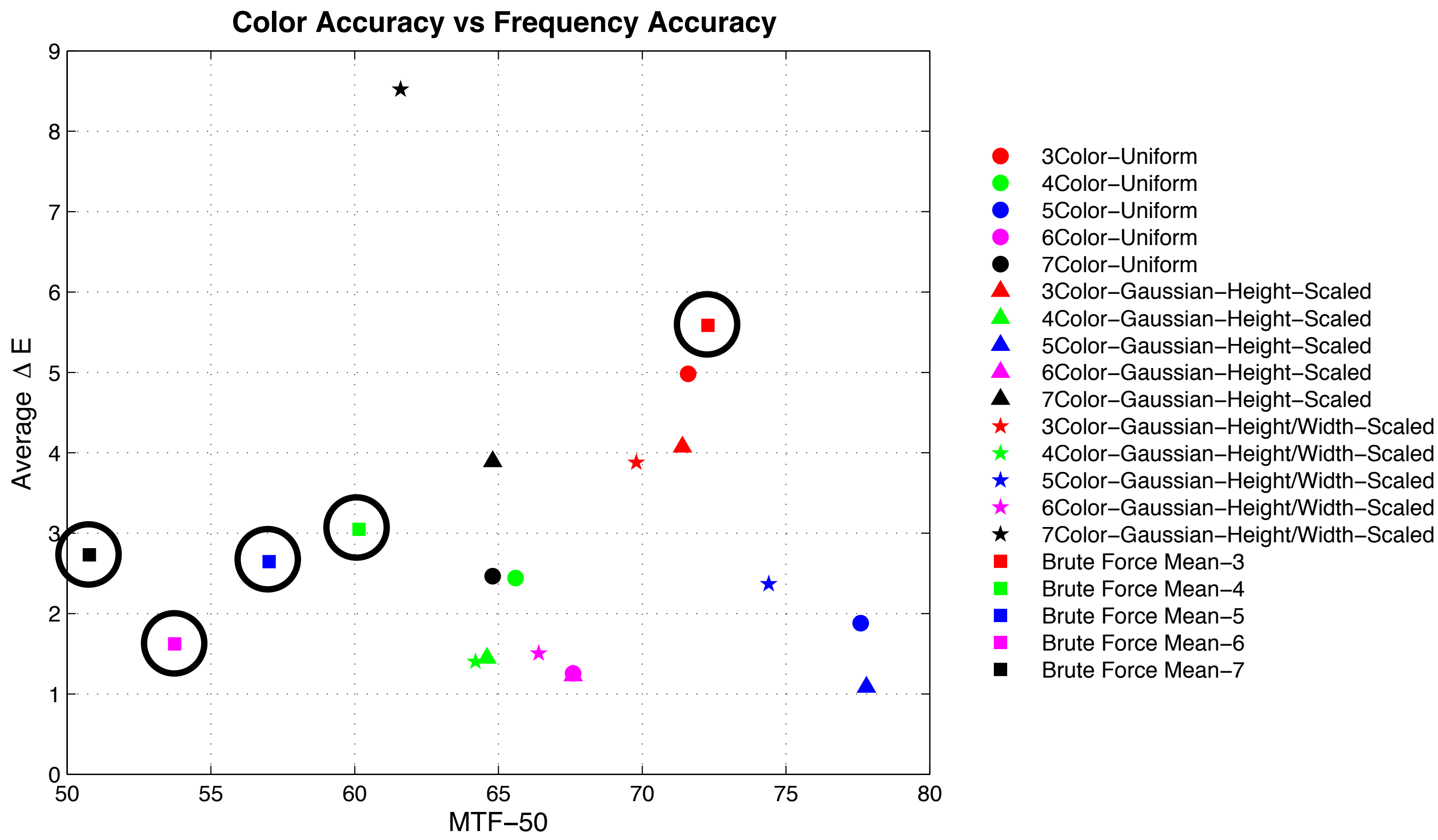
Defining Filter Colors - Theoretical Results



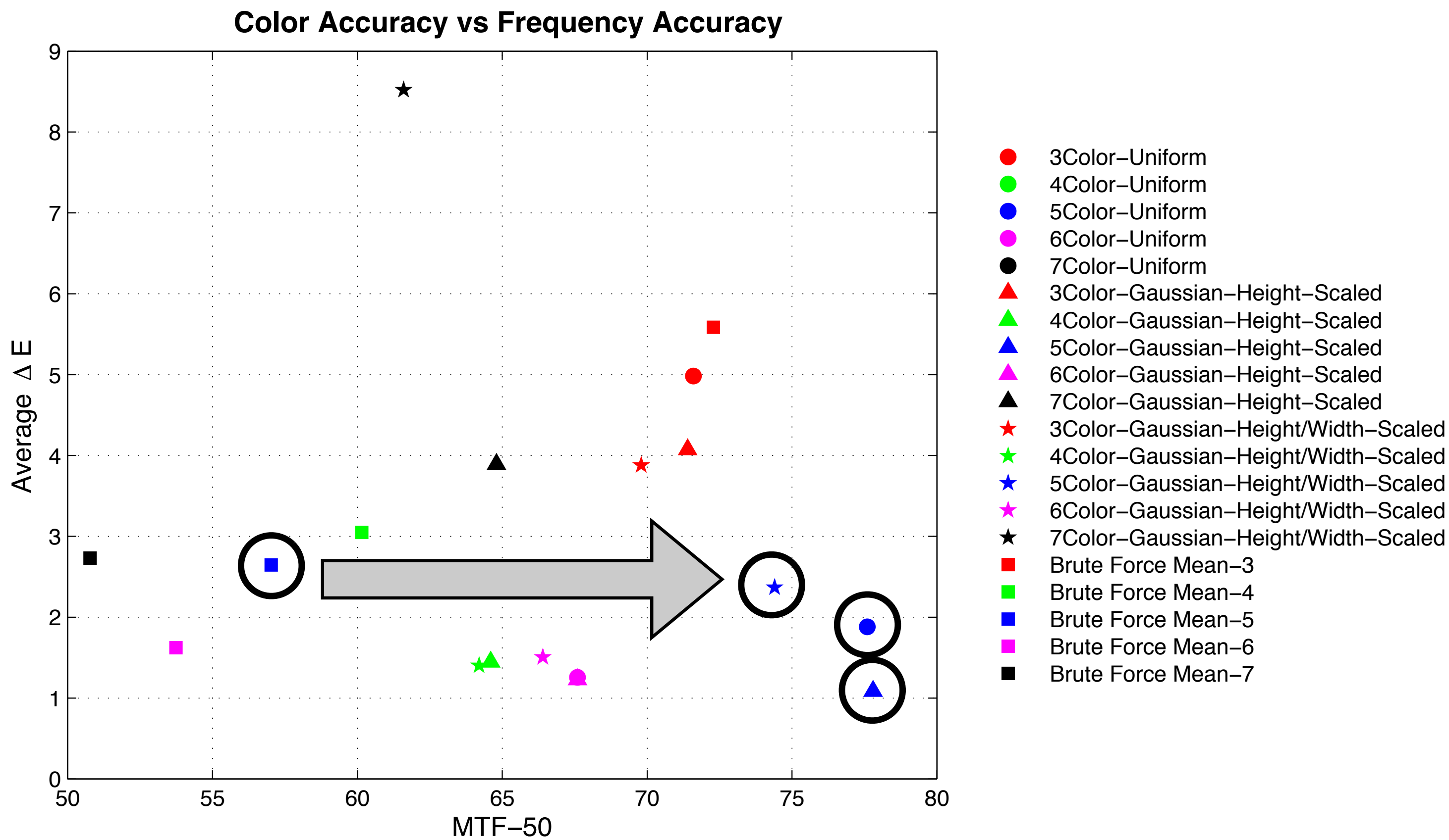
Defining Filter Colors - Theoretical Results



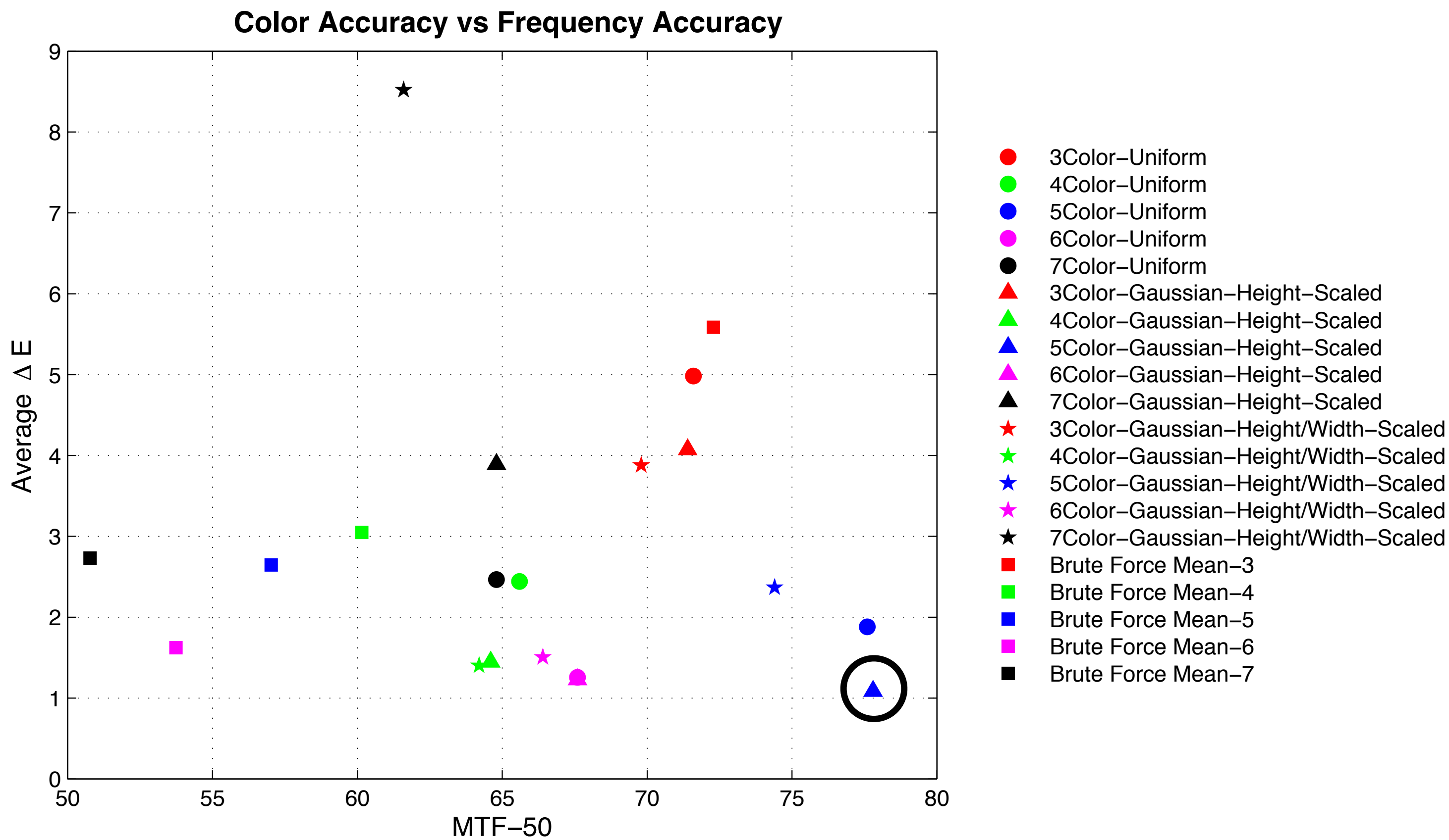
Defining Filter Colors - Theoretical Results



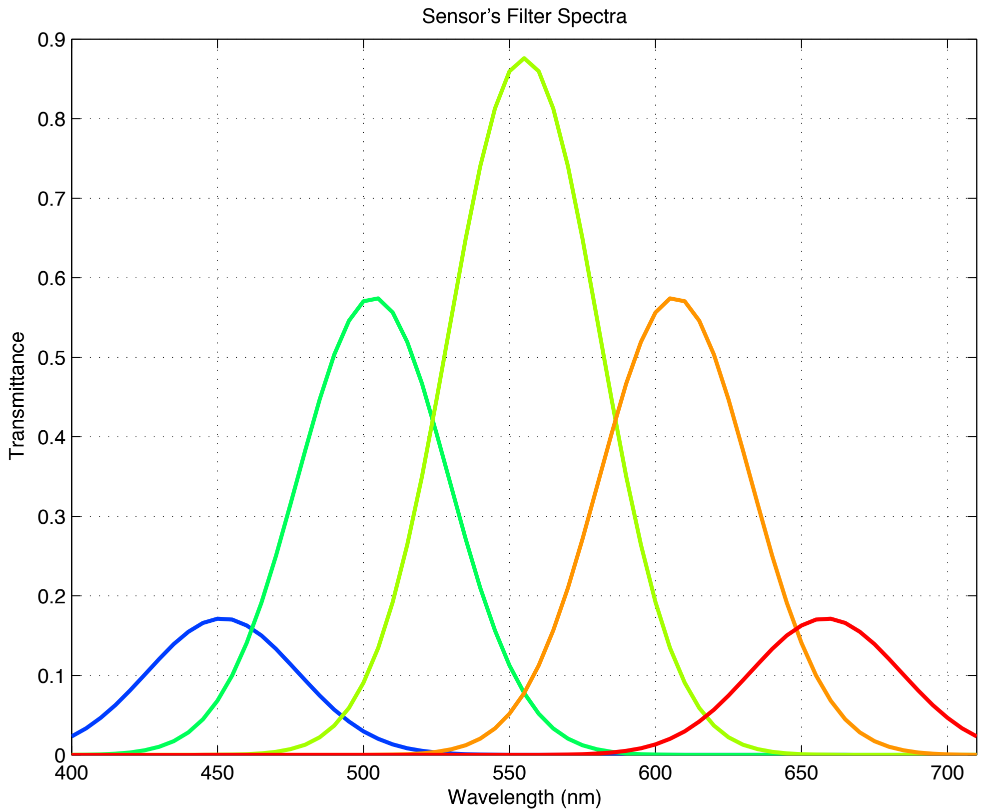
Defining Filter Colors - Theoretical Results



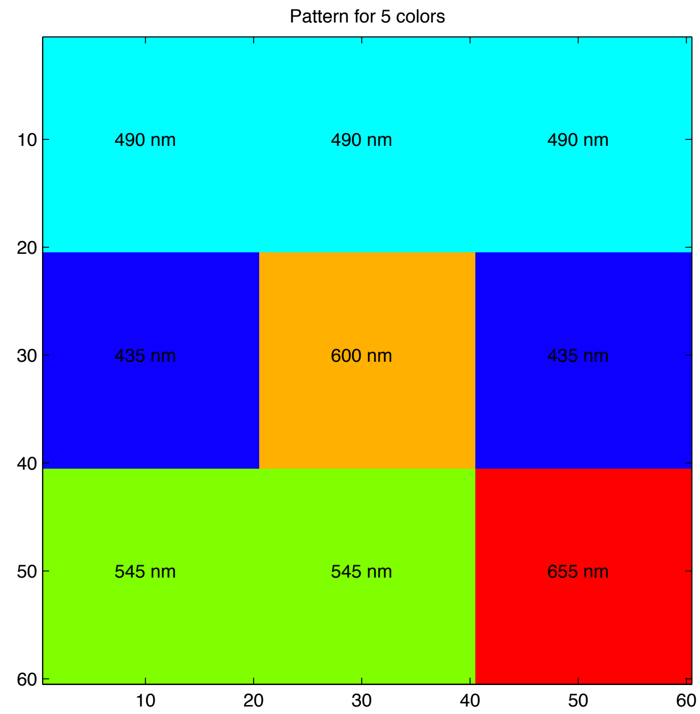
Defining Filter Colors - Theoretical Results



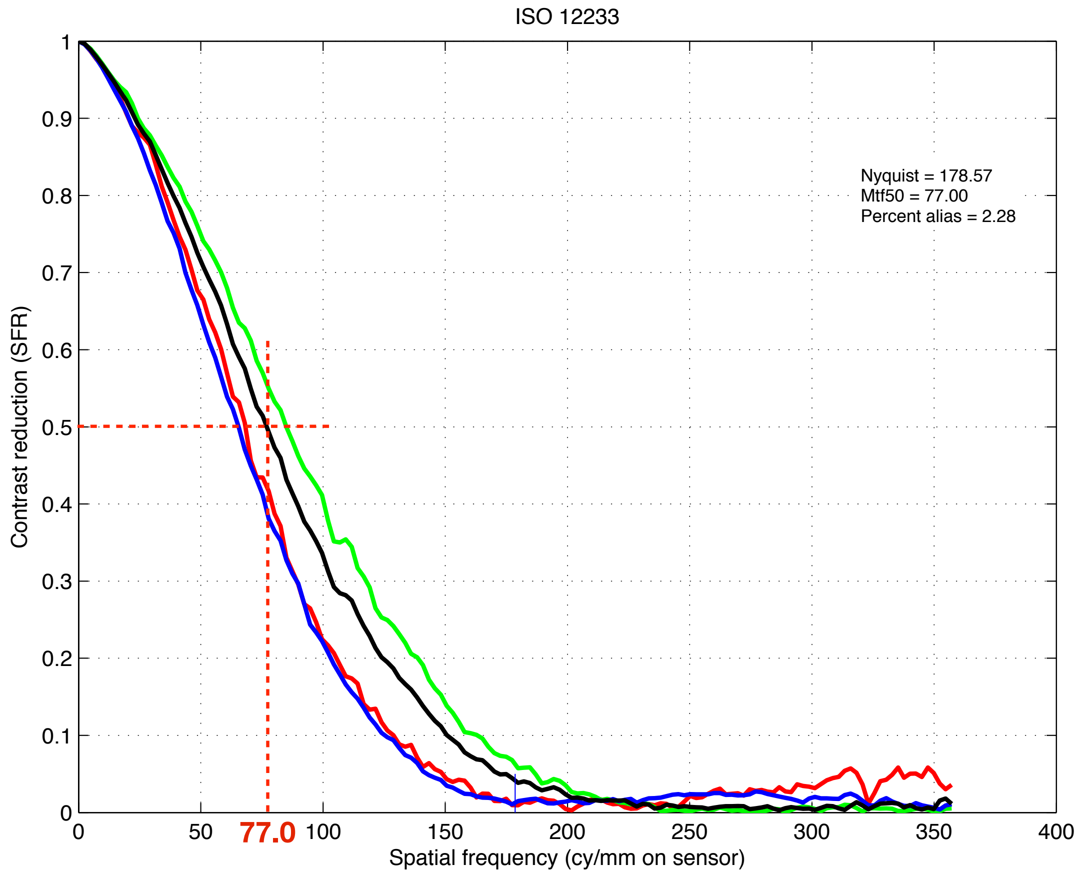
Defining Filter Colors - Theoretical Results



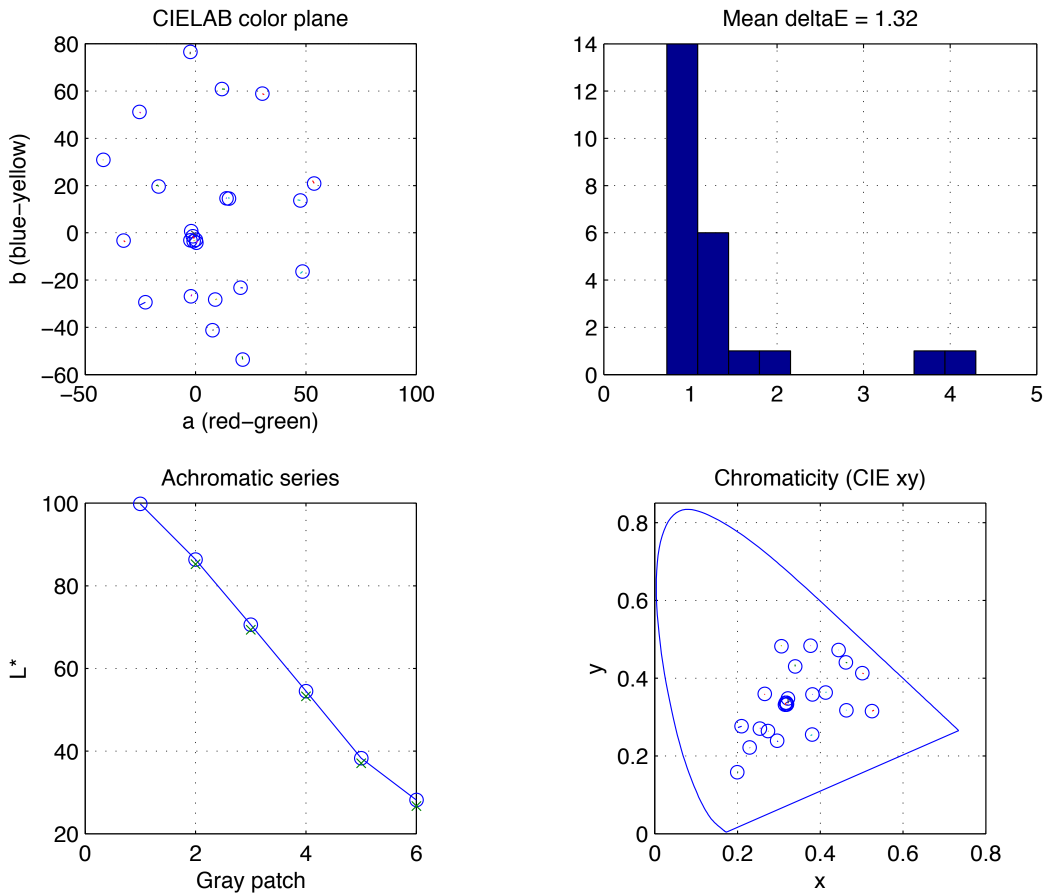
5 Color Height-Scaled



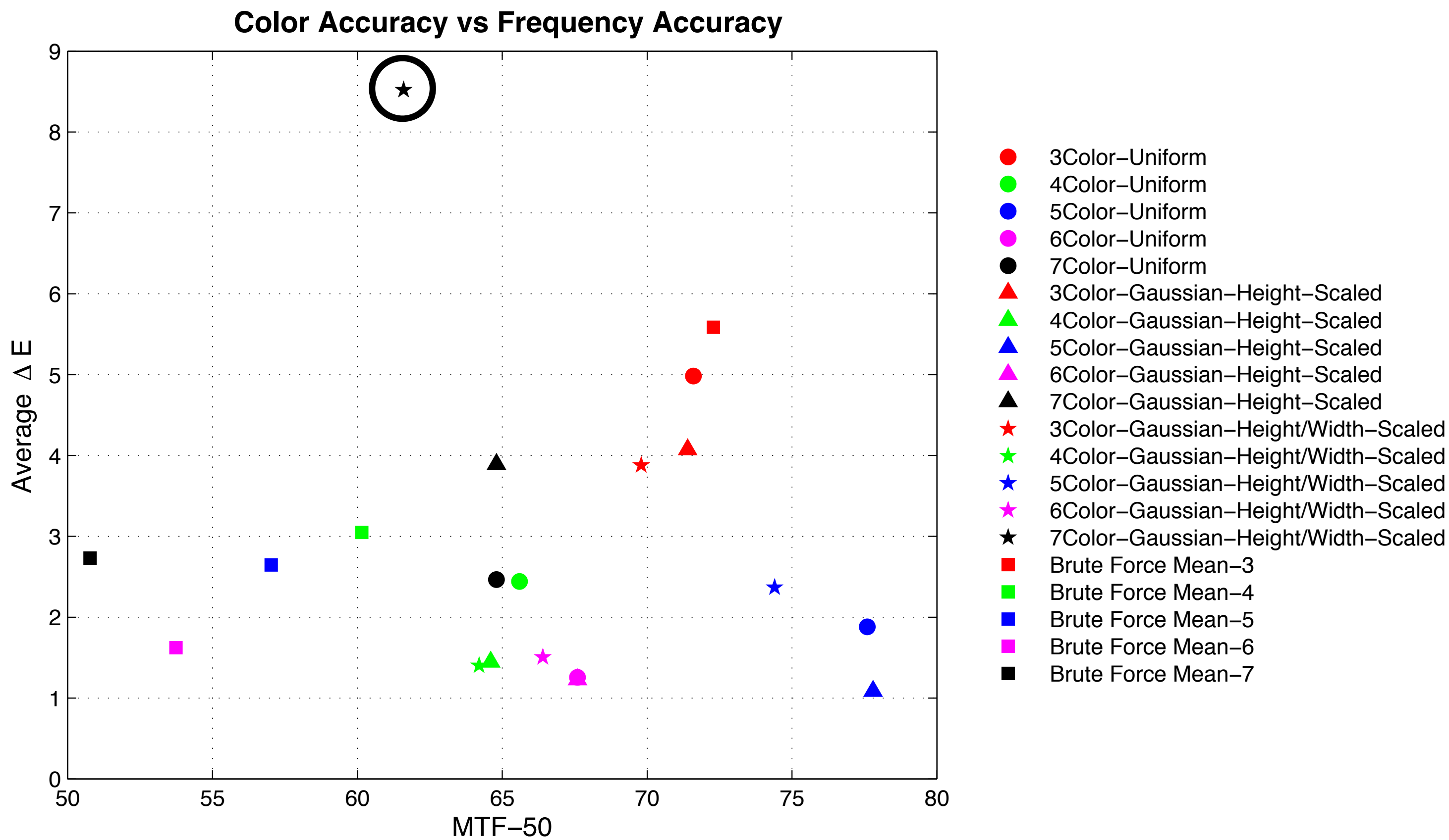
Defining Filter Colors - Theoretical Results



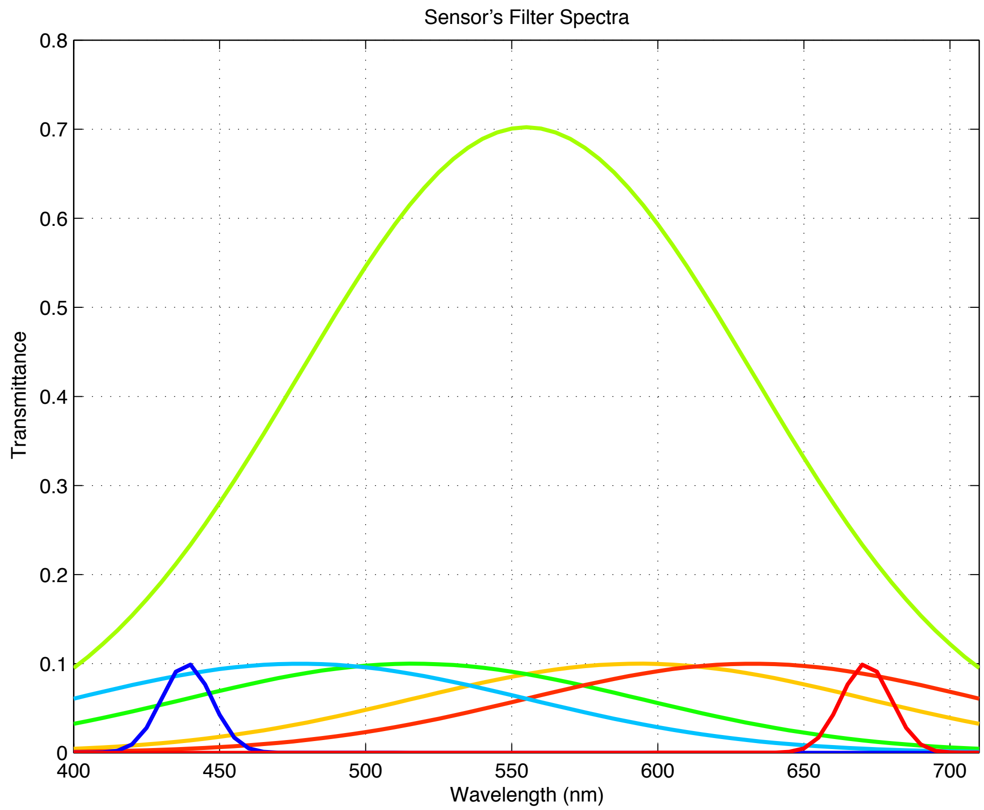
Defining Filter Colors - Theoretical Results

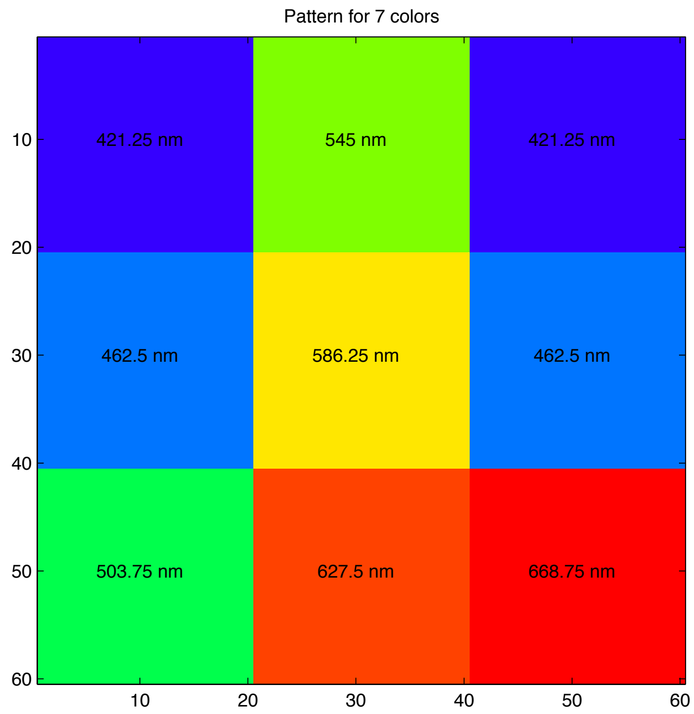


Defining Filter Colors - Theoretical Results



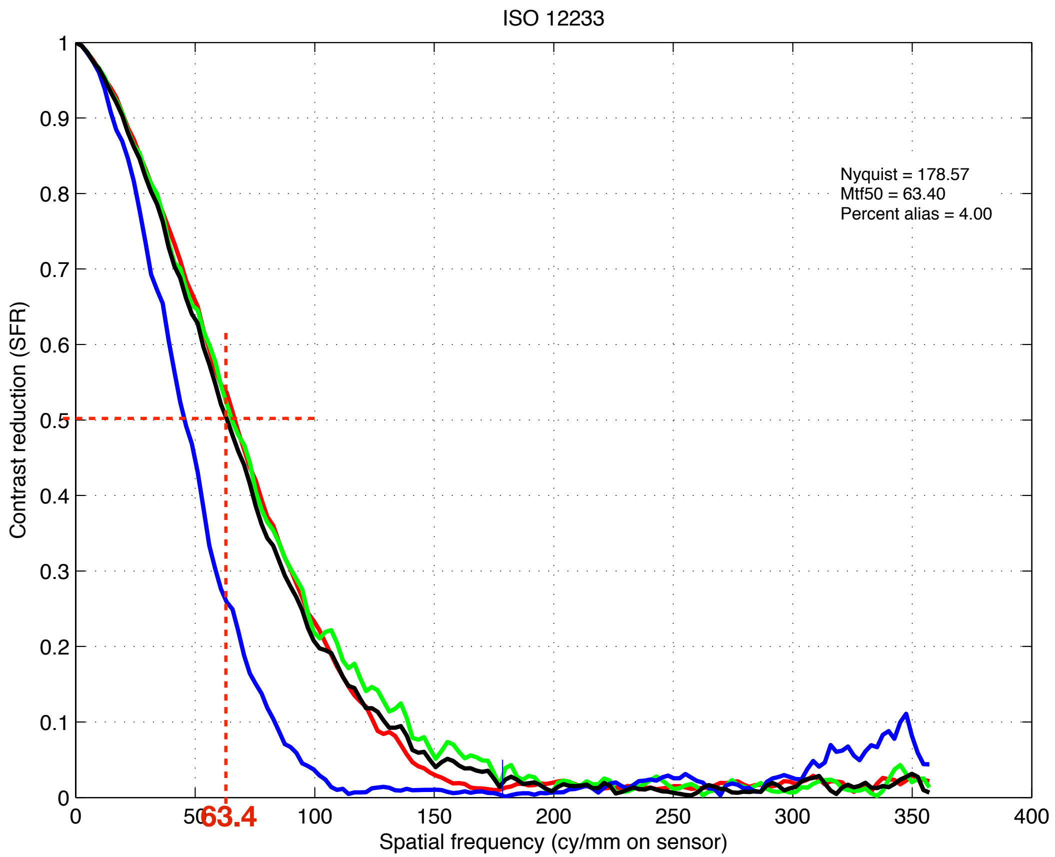
Defining Filter Colors - Theoretical Results



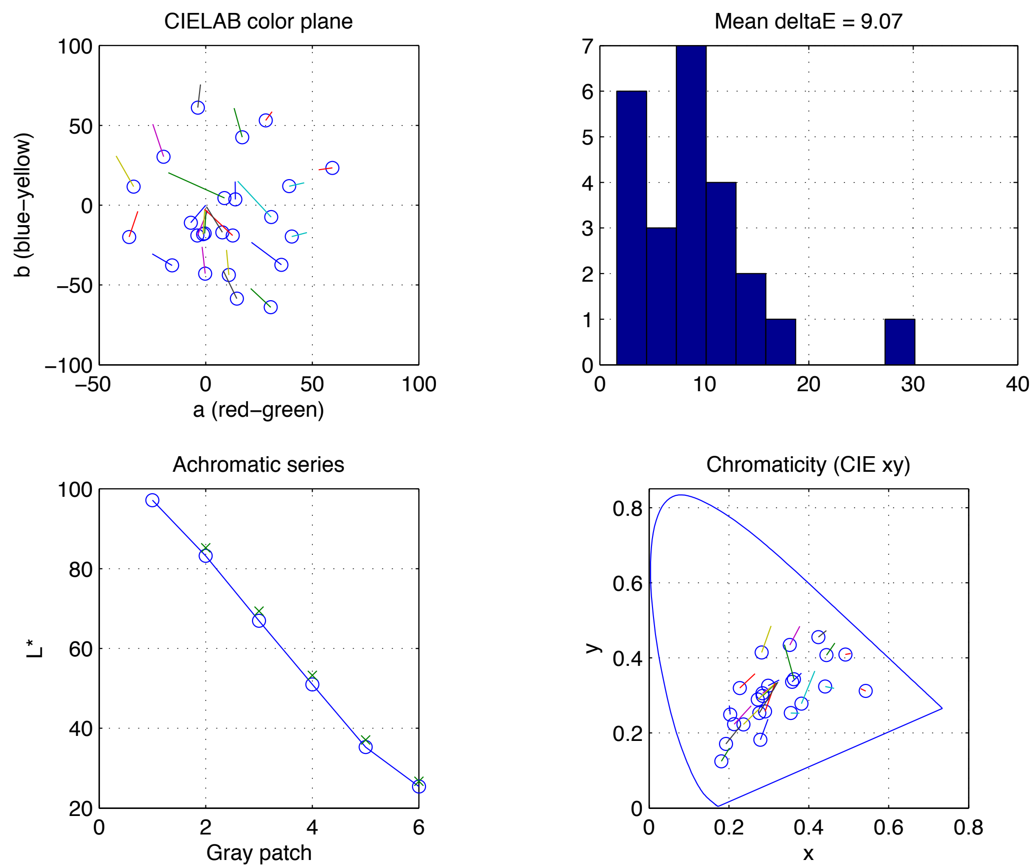


7 Color All-Scaled

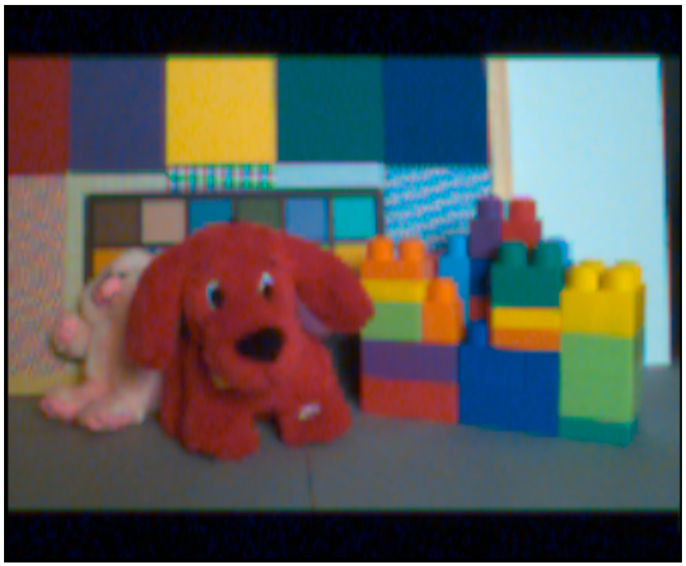
Defining Filter Colors - Theoretical Results



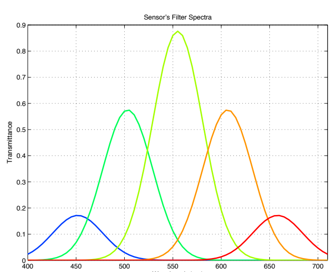
Defining Filter Colors - Theoretical Results

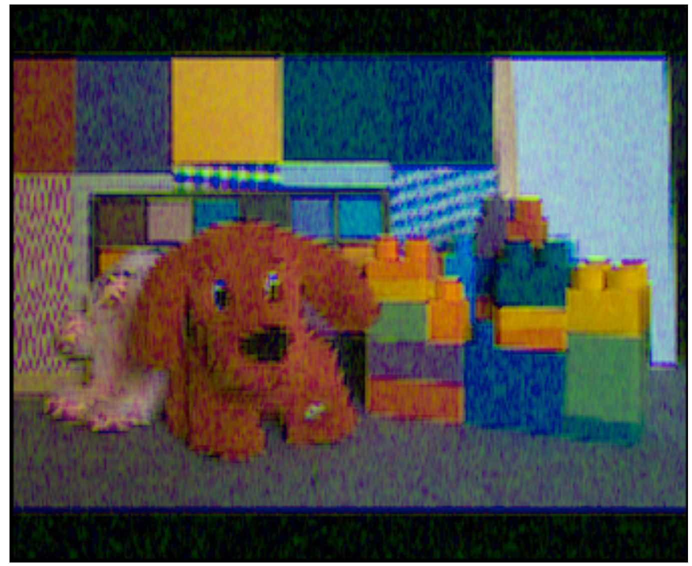


“Real-World” Results

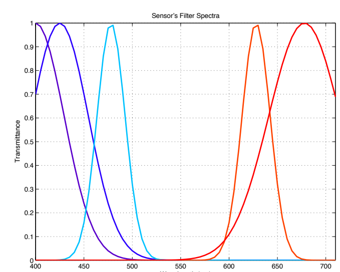


5-Color Height-Scaled





5-Color Uniform (Noisy Sample)



Limitations of the Current Approach

- Brute force simulations take a very long time, which makes it difficult to gather adequate statistics.
- ISET data structures seem to grow with each loop iteration, which means data needs to be cleared after a fixed number of loops.
- Assuming a gaussian distribution for each filter is not representative of manufacturing limitations, and might otherwise be less than ideal.

Future Work

- Simulate different shapes for filter transmittance
- Measure the effect of different illuminations
- Parallelize the simulation to decrease computation time
- Measure performance under varying noise conditions
- Test different demosaic algorithms
- Test different white balancing techniques
- Setup an experiment to determine the perceived quality of “real world” images generated using these new sensor configurations.

Thanks!

A special thanks to Professor Brian Wandell for augmenting ISET with the necessary features for our simulations.

Thank-You to Professor Joyce Farrell for her valuable advice and guidance.

Questions?