Exploring Histogram-Based Color Constancy

David R. Treadwell IV   Yunxuan Rao   Daniel Bi   Bruce A. Maxwell

Northeastern University

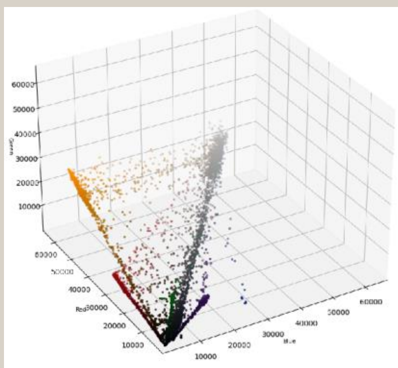
Abstract

We present a lightweight fully convolutional network for color constancy (LHCC). The network uses multiple 2-D projections of the 3-D log RGB histogram of an image in order to predict the color correction coefficients.

In developing the network, we explored whether to use linear RGB or log RGB data, the network structure (width and depth), how to handle dark pixels, how to generate the 2-D mappings of the 3-D histogram, and how to normalize or transform the bin counts in order to preserve the fine histogram structure. Our results show that attention to each of these details makes a difference in overall performance.

3-D Histograms Represent the Colors in an Image

Original Image



Linear RGB histogram

Gamut-based methods (including DNNs) use the color distribution to estimate the most likely illumination color

Image Model:

I = Image

A = Ambient Illuminant

R = Body Reflectance

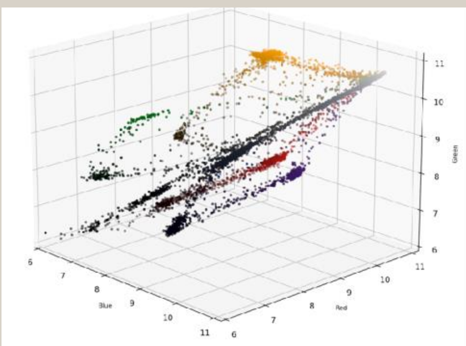
L = Direct Illuminant

1 = Strength of L 2 [0,1]

$$I = AR + \quad LR = R(A + \quad L)$$

Log RGB separates reflectance and illumination

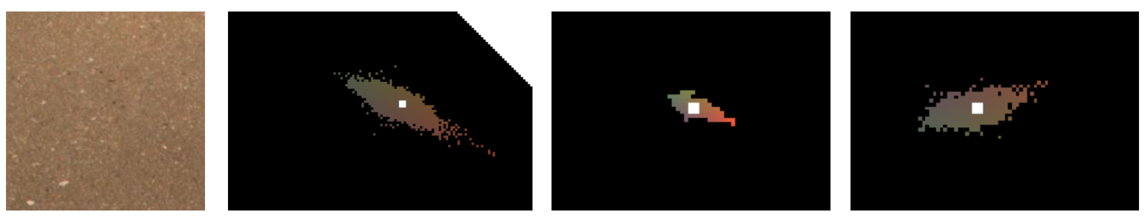
$$\log I = \log[R(A + ,L)] = \log R + \log(A + ,L)$$



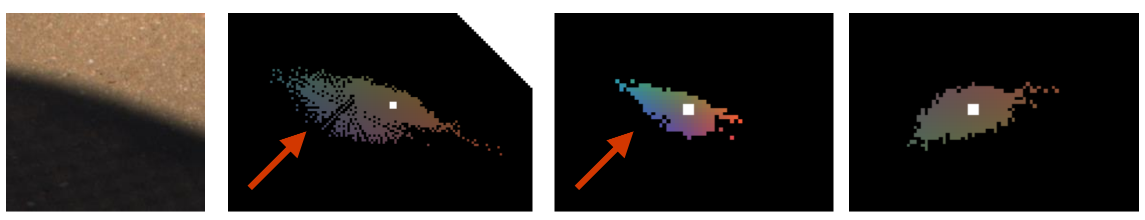
Log RGB histogram

Note consistent lines with clear separation for each material

Differing ambient and direct illuminants create separate, overlapping distributions of colors



Original Image (no shadow) | rg chromaticity histogram | log rg chromaticity histogram | ISD chromaticity histogram



Original Image (shadow) | rg chromaticity histogram | log rg chromaticity histogram | ISD chromaticity histogram

secondary shadow distribution

Grey world correction (RGB) | Grey world correction (RGB) | Grey world correction (ISD)

Efficiently Analyzing a 3-D Distribution

- 3-D convolutions are too computationally complex for color constancy
- Project the 3-D data onto multiple planes
- Independently extract features and then combine

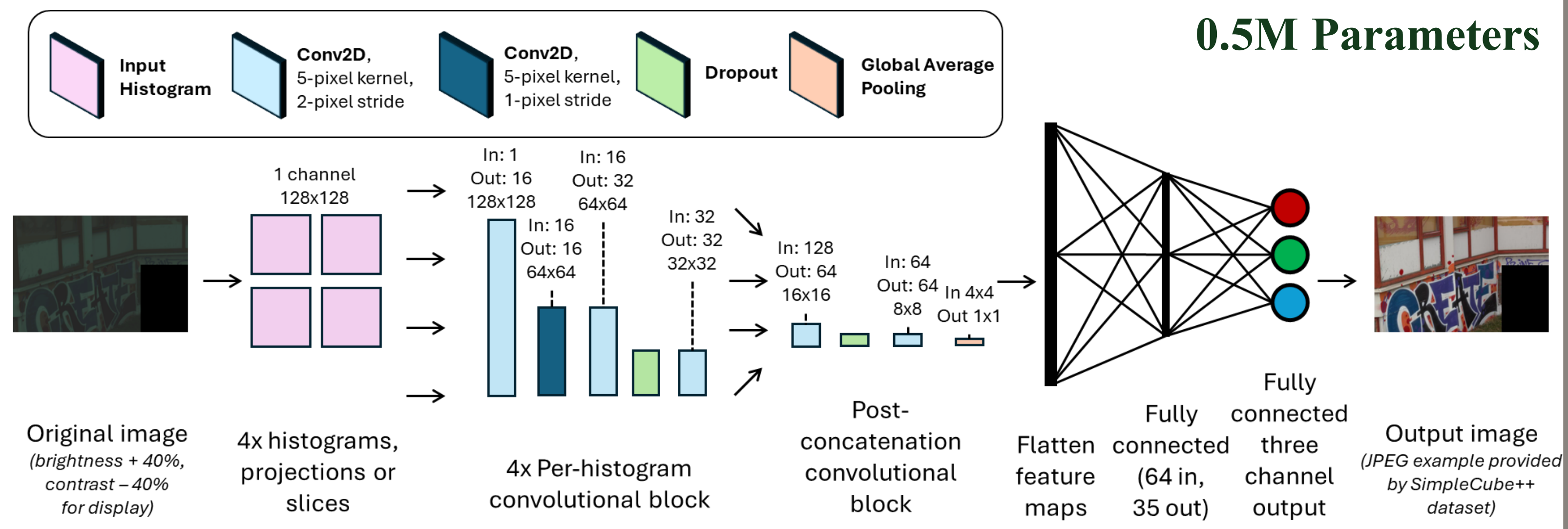
Implementation Details

- What color space to use: RGB or log RGB
- Pre-processing log space: clip to [1, 255] or [257, 65535] before taking the log
- What planes to use: {RG, RB, GB, (1, 1, 1)} or 4 slices along the 2nd eigenvector
- How to present the histogram values: raw, thresholding, hyperbolic tangent, or log
- How to design the network to integrate the separate plane information

Exploration Results

- Color space: log RGB outperforms linear RGB for this task and network
- Planes: planar projections slightly outperform eigenvector slices
- Histogram counts: All saturation methods outperformed raw, log was the best
- Network design: slow fusion with 4 convolutions before and 2 after concatenation

0.5M Parameters



SimpleCube++ Method	Mean	Med	Tri	25% Best	25% Worst	Parameters
Gray World [31]	3.18	2.00	2.37	-	-	-
Grey World 2OE [31]	3.06	1.75	2.08	-	-	-
FFCC (train/test)	2.64	1.75	1.85	0.52	6.35	12,288
FFCC (3-fold)	1.26	0.59	0.69	**0.18**	3.55	12,288
FC4 (log) [14] [24]	2.83	1.17	1.51	0.35	8.31	1,705,284
FC4 (default) [14] [24]	1.65	1.03	1.15	0.32	4.02	1,705,284
Afifi & Brown (SIIE, **Cube**) [1]	1.98	1.36	-	0.40	4.64	1,008,044
Afifi & Brown (SIIE, **Cube+**) [1]	2.14	1.44	-	0.44	5.06	1,008,044
CCMNet (**Cube+**) [16]	1.68	1.16	1.26	0.38	3.89	-
CauNet (**Cube+**) [18]	1.61	0.97	1.08	-	4.08	-
GoogLeNet (linear, ours)	1.53	0.95	1.05	0.25	3.90	6,797,700
GoogLeNet (log, ours)	1.35	0.75	0.88	0.22	3.52	6,797,700
LHCC (log, slices, ours)	1.12	0.63	0.69	0.19	2.93	490,895
LHCC (linear, proj, ours)	1.11	0.60	0.66	**0.18**	2.99	490,895
LHCC (log, proj, ours)	**1.03**	**0.57**	**0.64**	**0.18**	**2.72**	490,895

NUS-8 Method	Mean	Med	Tri	25% Best	25% Worst	Parameters
Grey World [2]	4.59	3.46	3.81	1.16	9.85	-
Grey World 2OE [2]	3.36	2.70	2.80	0.89	7.14	-
CCMNet [16]	2.32	1.71	1.83	0.53	5.18	
FC4 (SqueezeNet-FC) [14]	2.23	1.57	1.72	0.47	5.15	1,705,284
Afifi & Brown (SqNet) [1]	2.05	1.50	-	0.52	4.48	1,008,044
FFCC [2]	1.99	1.31	1.43	**0.35**	4.75	61,480
C4 (SqNet) [32]	1.96	1.42	1.53	0.48	4.40	1.2M+
TLCC [28]	**1.61**	**1.27**	**1.33**	0.44	**3.35**	6,406,549
LHCC (pretrained, linear, ours)	2.09	1.51	1.63	0.49	4.68	490,895
LHCC (pretrained, log, ours)	2.05	1.46	1.58	0.46	4.64	490,895

Conclusions

- Training on log RGB data consistently improves performance on all data sets
- Normalizing histogram bin counts using a log transform improves performance
- LHCC is a light-weight 0.5M parameter network robust to overtraining
- LHCC gives a new state-of-the-art on SimpleCube++ with a 2.72° error on the worst 25%