Learning Color Constancy

Brian Funt, Vlad Cardei and Kobus Barnard
School of Computing Science, Simon Fraser University, Vancouver, Canada

Abstract

We decided to test a surprisingly simple hypothesis; namely, that the relationship between an image of a scene and the chromaticity of scene illumination could be learned by a neural network. The thought was that if this relationship could be extracted by a neural network, then the trained network would be able to determine a scene's Illuminant from its image, which would then allow correction of the image colors to those relative to a standard illuminance, thereby providing color constancy. Using a database of surface reflectances and illuminants, along with the spectral sensitivity functions of our camera, we generated thousands of images of randomly selected illuminants lighting 'scenes' of 1 to 60 randomly selected reflectances. During the learning phase the network is provided the image data along with the chromaticity of its illuminant. After training, the network outputs (very quickly) the chro-maticity of the illumination given only the image data. We obtained surprisingly good estimates of the ambient illumination lighting from the network even when applied to scenes in our lab that were completely unrelated to the training data.

Descriptive Summary

Existing color constancy algorithms[1,2,3,4,9,6,8] generally employ assumptions about either the surface reflectances that will occur in a scene or about the possible spectral power distributions of scene illuminants. Given the assumptions and 3-band image data (either CIE XYZ specification or camera RGB) these algorithms calculate the chromaticity of the unknown scene illumination. If the assumptions are satisfied—which generally they are not—the estimate of the illumination will be correct and can then be used to adjust the image data so that the image would be the same as if had been taken under some standard, known illuminant. To the extent that the adjusted colors are as they would have been under the standard illuminant, the system can be said to exhibit 'color constancy'.

In contrast, the neural network we have developed has no built-in constraints. It is an adaptive model that makes no explicit assumptions about the input data. All rules are implicitly learned from the training set, which contains a large number of (artificially generated) scenes. The experimental results (see below) show that the neural network outperforms the grey-world and white-patch algorithms, especially in the case of scenes containing a small number (1 to 5) of distinct RGB measurements (Since 'color' is a perceptual quality, in what follows we'll avoid using it and instead simply use RGB to mean the response of the camera at a given pixel). Good performance with only a small number of distinct RGB's means that the network is particularly well suited for processing small, local image regions. This is important because generally a scene contains more than one source of light, so the assumption that the scene illumination is constant will, at best, hold true only locally within an image.

Neural Network's Input and Output

The neural network's input layer consists of a large number (up to 5000 in one test) binary inputs representing the chromaticity of the RGBs present in the scene.

Each image RGB from a scene is transformed into an rg chromaticity space: $r = R/(R + G + B)$ and $g = G/(R + G + B)$. Thus all possible RGB's will map to rg chromaticities inside a triangle having sides of unit length. This space is uniformly sampled with a step S, so that all chromaticities within the same sampling square of size S are taken as equivalent. Each sampling square maps to a distinct network input 'neuron'. The input neuron is set either to 0 indicating that an RGB of chromaticity rg is not present in the scene, or 1 indicating that rg is present. This discretization has the disadvantage that it forgoes some of the resolution in chromaticity due to the sampling, but on the other hand it provides a permutation-independent input to the neural net. This aspect is very important because it reduces the size of the input data set (at the cost of a large input layer).

The output layer of the neural network produces the values r and g (in the chromaticity space) of the illuminant. These values are reals ranging from 0 to 1.

Neural Network Architecture

The neural network we used is a Perceptron[7] with one hidden layer. The first layer is usually large and the input values are binary (0 or 1), as described above. The larger the layer, the better the chromaticity resolution, but a very large layer can lead to a huge increase in the training time. We have made experiments with an input layer of sizes 256, 512, 1250 and 5000 with comparable color constancy results in all cases.

The color-correction experiments described below were done with an input layer of size 1250, which corresponds to a sampling step of 0.020. The hidden layer has a much smaller size, usually about 16-32 neurons and the output layer is composed of only two neurons. The training method was the Back-propagation algorithm, without momentum.[10] During the training, the illuminant chromaticity is also provided to the neural network. The error function was the Euclidean distance in the chromaticity space between the network's estimate of the illuminant chromaticity and its true chromaticity.

Experimental Results and Comparisons

The network was trained with a large number of synthesized scenes, each with a random set of from 1 to 60 surface reflectances. The illuminant database contains the spectral power distributions of 89 different illuminants, that were measured with a Photoresearch PR650 spectrophotometer at different places around the university campus. The reflectance database contains the percent spectral reflectance functions obtained from 368 different surfaces. During the training phase, for each illuminant, the number of scenes used usually ranged from 10 to 1000. There was no noticeable improvement in the behavior of the neural network, when trained on a very large training set.

The number of training epochs was kept relatively small, partly because of the large amount (cpu hours) of training time. It was also a function of the size of the training set. The network used for these experiments had an input size of 1250 neurons, 32 hidden neurons and 2 output neurons. The training set was composed of 8900 scenes (i.e. 100 scenes for each illuminant) and each scene had a random number of colors ranging from 1 to 60. The network was trained for 120 epochs. After the training process was completed, the average error (i.e. Euclidean distance in the chromaticity space between the target output and the output obtained by the neural network) was 0.018608.

After training was completed, the network was tested on a different set of scenes. Scenes were generated by randomly selecting 1, 2, 3, 5 or 10 surface reflectances. For each of these cases 100 scenes were created. The average error obtained by the neural network for 100 scenes for each number of distinct reflectances is compared in Tables I and II to that obtained by three other color constancy algorithms: white-patch algorithm, grey-world algorithm and the 2D convex hull gamut mapping algorithm[3] with and without illumination constraints included.

The grey-world algorithm assumes that the average of all colors in an image is grey, i.e. the red, green and blue components of the average color are equal. The amount the image average departs from grey determines the illuminant RGB. The white-patch algorithm, which is at the heart of many of the various retinex[8] algorithms, presumes that in every image there will be some surface or surfaces such that there will be a point or points of maximal reflectance for each of the R, G, and B bands. The 2D convex hull gamut mapping algorithm considers the set of possible illuminants that could map the observed gamut of image RGB's to a canonical gamut of expected possible RGB's under the standard, known illuminant. We do not include comparisons with the Maloney-Wandell[9] algorithm because it has previously been demonstrated[3] that it works worse than these other algorithms.

The error measures used in Tables I and II are the angular error and the root mean square error. The angular error is computed by converting the rg chromaticities of the illumination's true chromaticity and its estimated chromaticity to 3-vectors and then measuring the angle between the two vectors. For the RMS error the chromaticities of all the surfaces in the scene are corrected on the basis of each algorithm's illumination estimate. This yields an image as the algorithm would expect it to be under the standard illuminant. The difference between the true chromaticities under the standard illuminant and those estimated by the algorithm is measured by the RMS error taken over all surfaces in the scene.

Conclusion

We have shown that color constancy can be learned by a standard neural network. Of course, the disadvantage of the neural network approach is that there is no way to know exactly what it is that the network learned. The neural network performs substantially better than the white-patch and grey-world algorithms on scenes with a limited number of different surfaces. Of course, as the number of surfaces increases the probability of fufilling the grey-world and white-patch assumptions grows and they begin to work better. The neural net performs better than the gamut mapping algorithm when it is allowed to use only the constraints arising from knowledge of the possible gamut of surface reflectances; and almost as well as the gamut mapping algorithm when it is allowed to use the additional constraints provided by knowledge of possible gamut of illuminant spectra.

Table 1. Angular Error Predicting White Under Canonical

Number of Surfaces.	1	1	2	2	3	3	5	5	10	10
Error	**mean**	**st dev**	**mean**	**st dev**	**mean**	**st dev**	**mean**	**st dev**	**mean**	**st dev**
Minimum attainable error using 2D diagonal model.	0.908	0.0	0.908	0.0	0.908	0.000	0.908	0.000	0.908	0.000
Grey World	22.23	12.00	16.17	8.946	12.87	7.759	9.341	5.604	7.112	3.764
White Patch	22.23	12.00	16.93	9.542	14.02	8.910	8.887	5.824	6.871	4.558
Gamut Mapping: surfaces only.	33.64	15.96	23.41	13.65	16.33	12.09	12.61	9.460	7.881	5.346
Gamut Mapping: surfaces plus illumination.	8.051	2.812	7.525	3.350	6.501	2.914	5.918	3.254	4.746	2.281
Neural Net	13.05	3.345	10.40	4.245	8.205	4.291	6.184	3.502	4.902	2.833

Table 2. RMS Chromaticity Mapping Error

Number of Surfaces	1	1	2	2	3	3	5	5	10	10
Error	**mean**	**st dv**	**mean**	**st dv**	**mean**	**st dv**	**mean**	**st dv**	**mean**	**st dv**
No Color Correction.	1.072	0.000	1.072	0.000	1.07	0.00	1.07	0.00	1.07	0.000
Minimum attainable error using 2D diagonal model.	0.126	0.000	0.126	0.000	0.12	0.00	0.12	0.00	0.12	0.000
Greyworld	2.511	5.102	1.096	0.916	0.88	0.90	0.56	0.30	0.45	0.206
WhitePatch	2.511	5.102	1.147	0.976	0.96	1.00	0.54	0.33	0.43	0.241
Gamut Mapping: surfaces only.	20.13	51.11	3.545	5.686	1.80	2.88	1.02	1.15	0.52	0.500
Gamut Mapping: surfaces and illumination.	0.499	0.144	0.466	0.178	0.41	0.15	0.38	0.17	0.31	0.130
Neural Net	0.758	0.157	0.628	0.208	0.51	0.22	0.40	0.18	0.32	0.165

References

1. D. H. Brainard and W. T. Freeman, "Bayesian Method for Recovering Surface and Illuminant Properties from Photosensor Responses", *SPIE Human Vision, Visual Processing and Digital Display V*, 364-376, Feb. 1994.
2. Finlayson, G., "Color Constancy in Diagonal Chromaticity Space," *IEEE Proc. Fifth Intl. Conf. on Comp. Vision*, June 20-23,1995.
3. G. Finlayson, B. Funt and J. Barnard, "Color Constancy Under a Varying Illumination," *Proc. Fifth Intl. Conf. on Comp. Vis.,* Jun 1995.
4. D. Forsyth, "A Novel Algorithm for Color Constancy," *Intl. Journal of Computer Vision* **5**, 5-36, 1990.
5. B. V. Funt and M. S. Drew, "Color Constancy Computation in Near-Mondrian Scenes using a Finite Dimensional Linear Model", *IEEE Proc. Computer Vision and Pattern Recognition Conference*, 544-549, June 1988.
6. R. Gershon, A. Jepson and J. Tsotsos, "From RGB to Surface Reflectance: Computing Color Constant Descriptors in Images," *Perception* 755-758, 1988.
7. J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Company, 1991.
8. E. H. Land and J. J. McCann, "Lightness and Retinex Theory," *J. Opt. Soc. Am,* **61**, 1-11, 1971.
9. L. Maloney and B. Wandell, "Color Constancy: A Method for Recovering Surface Spectral Reflectance," *J. Opt. Soc. Am. A,* **3**, 29-33, 1986.
10. D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning Internal Representations by Error Propagation" in D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol.**1**, Foundations. MIT Press (1986).