Journal of Electronic Imaging 13(1), 58–64 (January 2004).

Tuning Retinex parameters

Florian Ciurea
Brian Funt
Simon Fraser University
School of Computing Science
Burnaby, British Columbia, V5A 1S6
Canada

Abstract. *Our goal is to understand how the Retinex parameters affect the predictions of the model. A simplified Retinex computation is specified in the recent MATLAB™ implementation; however, there remain several free parameters that introduce significant variability into the model's predictions. We extend previous work on specifying these parameters. In particular, instead of looking for fixed values for the parameters, we establish methods that automatically determine values for them based on the input image. These methods are tested on the McCann-McKee-Taylor asymmetric matching data, along with some previously unpublished data that include simultaneous contrast targets.*
[DOI: 10.1117/1.1635365]

1 Introduction

The MATLAB™ implementation[1] of the Retinex model has three important input parameters: the number of iterations the algorithm performs at each level of its multilevel computation, the output lookup table function (postLUT), and the input image size. The model's final output depends strongly on the values chosen for the parameters.[1]

The Retinex model aims to predict the sensory response of lightness. In previous work[2] we suggested values for the parameters based on fitting the model's predictions to the data originally described over 35 years ago by McCann, McKee, and Taylor.[3] This fit led to the conclusion that 33 iterations had the lowest global average of the differences between observer data and computed values, assuming that the number of iterations was constant for all levels of the multiresolution computation. However, McCann felt that 33 was too high a number, and would not lead to a good model of simultaneous contrast. Hence, together we began the current series of experiments by including previously unpublished data from lightness matching experiments with simultaneous contrast targets. We also added other unpublished data for targets containing a fixed set of patches of various shades of gray appearing on a background that varied from black to gray to white.

For the simultaneous contrast data, we indeed did find that a much smaller value is required for the iteration parameter to make a good fit. However, we could no longer find a universal value for the number of iterations that simultaneously would minimize the error for the combined data from the McCann-McKee-Taylor (MMT), fixed scale of grays on different backgrounds (SB), simultaneous contrast (SC), and gray on white (GW). This led us to consider a method of automatically calculating how many iterations to use based on how the computation was proceeding. As described earlier,[1] the postLUT processing needs to change as a function of the number of iterations, so this led us to a method of automatically calculating the appropriate postLUT.

2 Number of Iterations

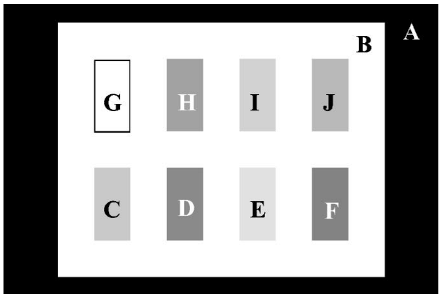
The two MATLAB™ implementations of Retinex in Funt, Ciurea, and McCann[1] are referred to as McCann99 Retinex and Frankle-McCann Retinex. For brevity, we concentrate here only on McCann99 Retinex, but the results are similar for both versions. McCann99 Retinex creates a multiresolution pyramid from the input by averaging image data. It begins the pixel comparisons at the most highly averaged, or top level, of the pyramid. After computing so called new products (precursors to the final lightness estimates) on the image at a reduced resolution, the resulting new product values are propagated down, by pixel replication, to the pyramid's next level as initial estimates at that level. Further pixel comparisons refine the estimates at the higher resolution level, and then those new estimates are again propagated down a level in the pyramid. This process continues until values have been computed for the pyramid's bottom level.

At each level, the basic step is the comparison of each pixel to each of its immediate neighbors. The number of iterations refers to the number of times all the immediate neighbors are cycled through before moving down to the next level in the pyramid. Since pixels are only directly compared to immediate neighbors, comparisons to more distant pixels at the current pyramid level are only made implicitly by propagation of information from pixel to pixel during these iterations. Hence, increasing the number of iterations increases the spatial distance across which pixels are related during the computation. McCann99 Retinex uses the same number of iterations at all levels, and so there is only a single iteration parameter to specify, and we have limited this work to consider a single value for all levels.

3 PostLUT Processing

PostLUT processing refers to applying a function f uniformly to every image pixel, $I(x,y)=f[I(x,y)]$, for all im-

Paper RTX-05 received Mar. 4, 2003; revised manuscript received Oct. 15, 2003; accepted for publication Oct. 15, 2003.
1017-9909/2004/$15.00 © 2004 SPIE and IS&T.



Patch	Luminance	Pixel value	Match	Std dev	Calc.	Calc-Match
G	1001	255	8.75	0.15	9.01	-0.26
E	595	236	7.55	0.20	7.66	-0.11
I	439	225	6.25	0.25	6.97	-0.72
C	336	215	5.94	0.31	6.40	-0.46
J	228	200	5.19	0.19	5.63	-0.44
H	125	178	4.36	0.26	4.66	-0.30
D	63	153	3.37	0.50	3.76	-0.39
F	50	145	2.80	0.30	3.51	-0.71
B	1001	255	8.80	0.20		
A	1	0	1.0	0		

Fig. 1 "Scale on white" target along with patch identification, the luminance values measured in the original display, the digit representing log luminance, and the mean and standard deviation of observer matches in Munsell value units. The sixth column lists the calculated lightness for all iterations above three. The seventh column lists the errors between observed and calculated lightness.

age locations (x,y) immediately after the main Retinex computation. The term postLUT derives from historical use of image processing hardware using a lookup table (LUT) as a final postprocessing step. PostLUT processing is important in bringing the final result into the appropriate dynamic range, compensating for differences in overall illumination intensity between test targets, and in converting to the coordinates of the Munsell value scale used in recording the experimental data. Although all these factors can be thought of separately, they are all eventually combined into a single postLUT function.

The first postLUT step adjusts the dynamic range. Retinex output from the pyramidal spatial comparison stage falls in the $[0,1]$ range. Because the value 1 represents white and Retinex assumes there is at least one white pixel in every image, the value 1 necessarily arises in the output. However, the lowest output value depends on the image content and varies with the number of iterations used. The fewer the iterations, the more local the spatial comparisons will be, and therefore, the less the likelihood of big intensity differences being found. As a result, the fewer the iterations, the higher the minimum Retinex output value (Fig. 1 in Ref. 1 illustrates this effect). The first purpose of the postLUT is to stretch the Retinex output to a reasonable range. Since the amount of stretching needed depends on the number of iterations, and we vary the number of iterations in our experiments, we decided to always linearly scale the Retinex output to the full $[0,1]$ range. This stretch does not correct for the fact that the number of iterations performs a nonlinear compression of the image. The postLUT is not fixed, but rather depends on the input image and number of iterations used. This decision effectively means that we are assuming that there is at least on black location in the test target. While this assumption need not be true for images in general and could lead to errors in Retinex predictions, it is true for all the test target subjects viewed.

After scaling to the $[0,1]$ range, the postLUT then converts the Retinex output values r to the lightness scale used for recording the subject's matches. For the MMT dataset, the conversion is to Munsell value scale V:[4]

$$V = 2.539 r^{1/3} - 1.838 \quad \text{for} \quad r > 0.384.$$

For the SB, SC, and GW datasets, the conversion is to a lightness scale described by Stiehl, McCann, and Savoy.[5] Based on a fit to the raw data, we use the following function to convert the log luminance to the lightness scale values L:

$$L = 129.6 r^{1/100} - 132.45.$$

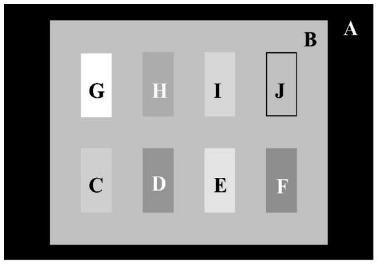
The final postLUT component compensates for differences in overall illumination intensity between the test and match conditions. Only the MMT experiments involved such intensity differences. The compensation is based on data from Fig. 8 of McCann, Land, and Tatnall.[6] Generally, the effect of this correction is slight. Details are provided by Funt and Ciurea.[2]

4 Lightness Matching Data

The experimental technique for the MMT matching experiments was reported a long time ago.[6] The "new" data we report here is based on experiments by McCann, which were also conducted earlier, but not previously reported in the literature. These experiments involve transparent gray-scale targets lit from behind with uniform illumination. Subjects were asked to report the lightness of each patch in the target display using a standard lightness transparency display as a reference. The standard lightness display consists of 25 squares of different lightness values against a white surround. The squares are arranged in a serpentine path, such that the change in lightness from any of the 25 squares to the next is constant.[5] In the resulting lightness scale, 1.0 corresponds to an opaque area and 9.0 to the brightest area. The experiments were based on four to seven subjects, with each subject repeating the matches on three different occasions.

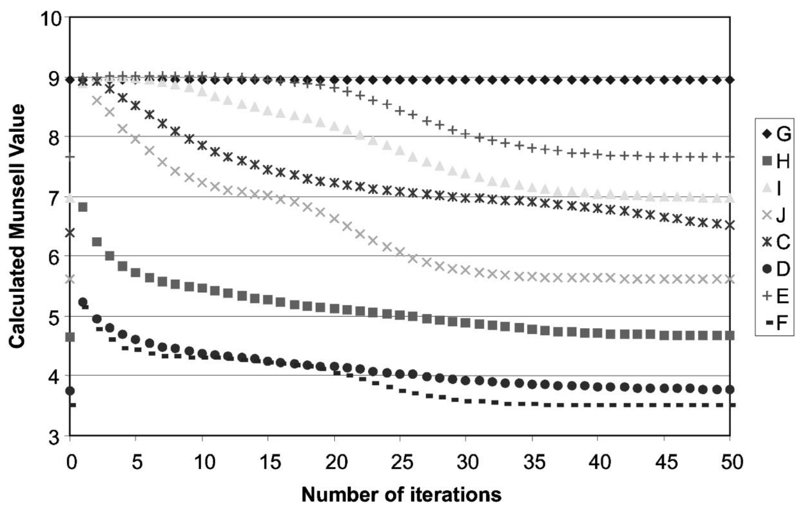
The matching procedure was set up such that in the normal viewing position, the subject saw the test display as the only thing in the field of view. By turning 90 deg to the right, the subject would see instead the standard lightness display as the only thing in the field of view. Subjects were allowed to look back and forth between viewing the test display and the standard display as many times as desired without a time constraint.[6] The test display and the standard lightness display had the same level of luminance.

Figures 1 through 6 illustrate the targets along with the corresponding luminance, pixel value for each patch as input to the Retinex algorithm, and average observed lightness reported for each patch. All the patches have uniform luminance. It should be noted that the figures are intended



Patch	Luminance	Pixel value	Match	Std dev	Calc.	Iterations
G	1001	255	9.00	0.20	8.95	1
E	595	236	8.50	0.50	8.53	24
I	439	225	7.31	0.31	7.32	31
C	336	215	7.06	0.31	7.06	26
J	228	200	5.88	0.38	5.88	28
H	125	178	4.98	0.28	4.98	26
D	63	153	4.08	0.48	4.08	23
F	50	145	3.05	0.55	3.51	50
B	228	200	5.75	0.25		
A	1	0	1.0	0		

(a)



(b)

Fig. 2 (a) ''Scale on gray'' target along with patch identification, the luminance values measured in the original display, the digit representing log luminance, the mean and standard deviation of observer matches in Munsell value units, and the calculated values for the best fit to observer match. The iterations column lists the number of iterations for best fit for calculated to observed lightness. The average number of iterations for best fit from areas E, I, C, J, H, and D is 26.33 ± 2.88, while the average that included areas G and F is 26.13 ± 13.32. The best fit for ''Scale on gray'' is 26 iterations. (b) ''Scale on gray'' calculated lightness as a function of ''Number of iterations.'' In a gray surround, all gray patches except white decrease with the increase of number of iterations. The number of iterations has significant effect on the calculated values of grays. Area E, the lightest gray, has a calculated lightness equal to white up until 20 iterations. Areas E, I, C, H, D, and F show different degrees of nonmonitonic decrease in calculated Munsell lightness. The darkest gray, area F, and midgray, area J, both show second phase starting at 20 iterations. A slightly lighter gray, area C, shows a similar change in slope at 35 iterations.

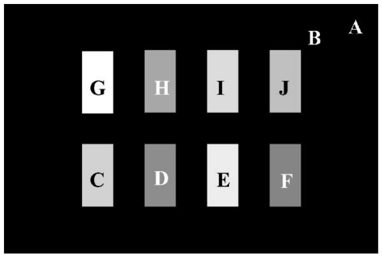
only to illustrate the corresponding targets. They are not accurate reproductions of the targets. Their printed appearance is not the same as under the controlled experimental conditions.

The calculated lightness for the ''Scale on white'' display are nearly constant with changes in ''Number of iterations.'' In a solid white surround, all gray patches have a constant value after the third iteration. As shown in the table in Fig. 1, the calculated lightness values (sixth column) are close to the observer matches (fourth column). There are residual errors (seventh column) with an average value of 0.42 ± 0.2. Since the white surround is the control case that establishes the shape of the LUT, the lack of perfect correlation is due to experimental and LUT error. These errors have no effect on the analysis of number of cycles, but contribute to any global average.

5 Discussion of Results

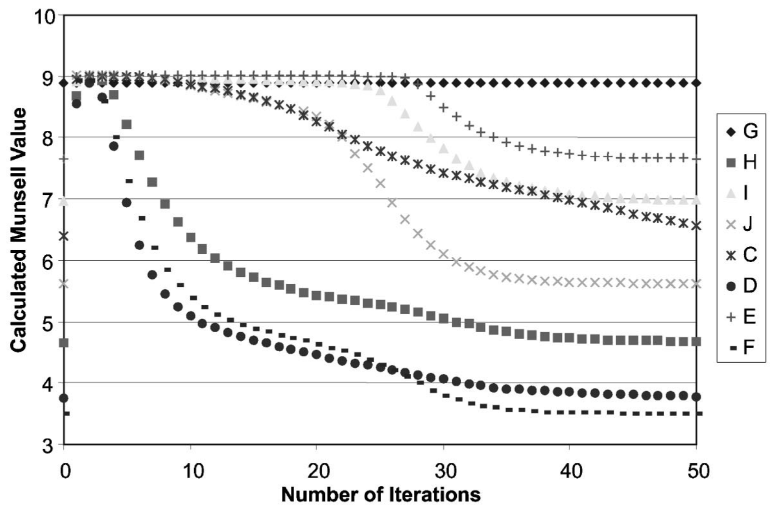
The principle effect of selecting the number of iterations is to establish the degree of local versus global influence from spatial comparisons. As seen in the previous data, it has no effect on grays in a white surround and significant effect on grays in a black surround. Using a very large number of iterations, so as to have the lightness asymptote to the limit of the calculation, makes the output approach the input.[7] That special case serves no purpose. Human observers make matches consistent with mechanisms that are between local and global. McCann, McKee, and Taylor[3] reported good fit from their experimental data using a path algorithm of length 200 hops, a moderately global process. We have previously[2] reported 33 iterations for an experiment that applied the same number of iterations for all spatial channels. In these experiments, it is clear that an intermediate number gives the best results for the ''Scale on gray'' target (Fig. 2) and ''Scale on black'' target (Fig. 3). In addition, the best fit to observer data is with very few iterations with larger gray patches in the ''Simultaneous contrast'' series. Seven iterations gave the best fit.

The displays that required the fewest iterations had large uniform surrounds. The scale displays had slightly smaller



Patch	Luminance	Pixel value	Match	Std dev	Calc.	Iterations
G	1001	255	9.00	0.05	8.88	1
E	595	236	8.55	0.45	8.60	3
I	439	225	7.53	0.78	7.54	26
C	336	215	7.29	0.54	7.28	33
J	228	200	6.80	0.50	6.67	27
H	125	178	5.65	0.35	5.65	16
D	63	153	5.20	0.50	5.25	9
F	50	145	4.68	0.38	4.69	19
B	1	0	0.90	0.10		
A	1	0	0.83	0.08		

(a)

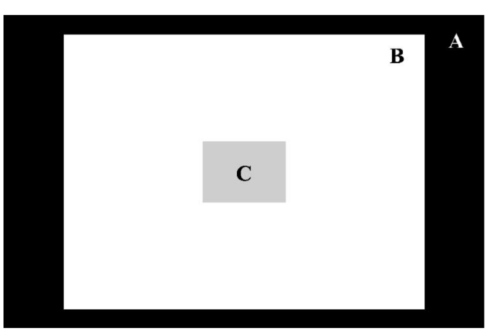


(b)

Fig. 3 (a) "Scale on black" target along with patch identification, the luminance values measured in the original display, the digit representing log luminance, the mean and standard deviation of observer matches in Munsell value units, and number of iterations and calculated lightness. In a black surround, the calculated lightness for all gray patches, except white, decreases with the increase of number of iterations. The average for best fit from areas G, E, I, C, J, H, and D is 16.8 ± 11.7 (b) "Scale on black" calculated lightness as a function of "Number of iterations." In a black surround, all gray patches except white decrease with an increase of number of iterations. Area E, the lightest gray, has a calculated lightness equal to white, up until 30 iterations. Areas I calculated lightness begins to fall at 25 iterations. C and J calculated lightness begin to fall at 12 iterations. The darkest grays begin to fall at five iterations.

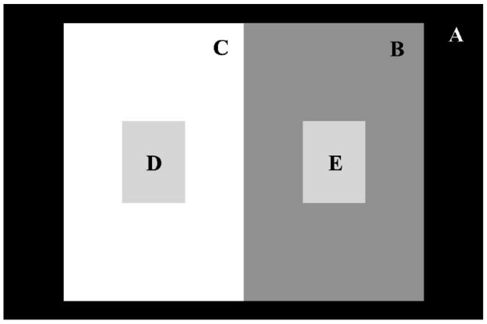
test patches and there were many more of them. The Mondrians had many more patches with smaller angular subtends. This, combined with results of other recent experiments,[8,9] suggests that the different number of iterations in each spatial channel will give the best overall fit to experimental data. Frankle and McCann used a table to control the number and direction of comparisons for each spatial channel.

Larger simple displays generate large signals in the low spatial frequencies or highest levels of the image pyramid.



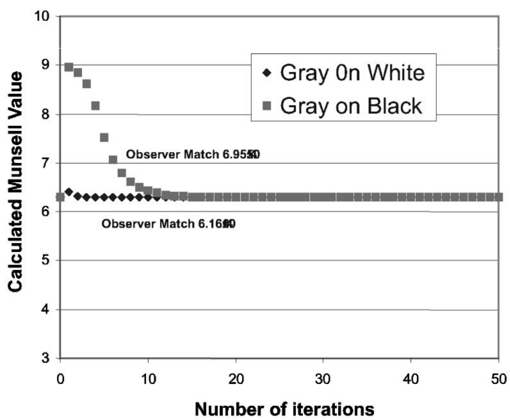
Patch	Luminance	Pixel value	Match	Std dev
B	1001	255	9.03	0.23
C	321	213	6.15	0.52
A	1	1	1.13	0.13

Fig. 4 "Gray on white" target along with patch identification, the luminance values measured in the original display, the digit representing log luminance, and the mean and standard deviation of observer matches in Munsell value units. There is no significant change in calculated values for white and gray. Black values vary for iterations of 1 to 7. The best fit is three iterations with a calculated value of 1.16, while the observed value is 1.13. The calculated asymptotes are 1.00, 6.29, and 9.01.



Patch	Luminance	Pixel value	Match	Std dev
C	1001	255	8.85	0.18
D	321	213	6.16	0.40
E	321	213	6.95	0.45
B	50	145	3.95	0.45
A	1	1	1.08	0.27

(a)



(b)

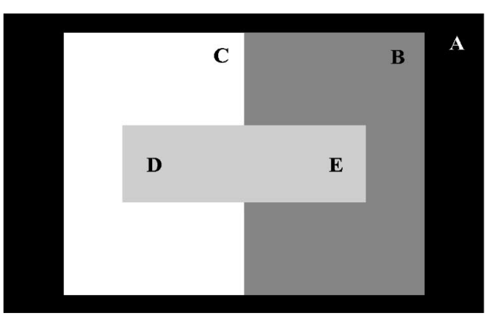
Fig. 5 (a) "Simultaneous contrast" target along with patch identification, the luminance values measured in the original display, the digit representing log luminance, and the mean and standard deviation of observer matches in Munsell value units. (b) The best fits are at six iterations for gray on white and eight for gray on black.

These channels need few spatial comparisons. Scales displays generate signals with higher spatial-frequency information, and these channels best fit the observer data with more iterations. The color Mondrians have the most high spatial-frequency information, and these channels need the highest degree of spatial comparisons.

6 Automatic Selection of the Number of Iterations

To investigate the advisability of automatic processes to measure the optimal number of iterations (i.e., cycles of comparing a pixel to its neighbors at each pyramid level), we plotted the rms error between the mean lightness values reported by human subjects and those predicted by Retinex as a function of the number of iterations. The variation in error is shown in Fig. 7 for the case of the SC and GW data from Figs. 4 through 7.

Since subjects reported a single lightness value for each patch, we calculate the Retinex lightness of a patch as the mean of the Retinex lightness values for all pixels from the patch. The Retinex prediction error for a patch, therefore, reflects the difference between the Retinex lightness estimate and the mean across all subjects of the lightness of the matches made for that patch. The overall prediction error for a target is simply the rms of the errors for the individual patches it contains.



Patch	Luminance	Pixel value	Match	Std dev
C	1001	255	9.09	0.29
D	321	213	6.2	0.50
E	321	213	6.7	0.58
B	50	145	4.04	0.54
A	1	1	1	0.25

Fig. 6 "Simultaneous contrast strip." Best fit is six iterations for gray on white, and eight iterations for gray on black.

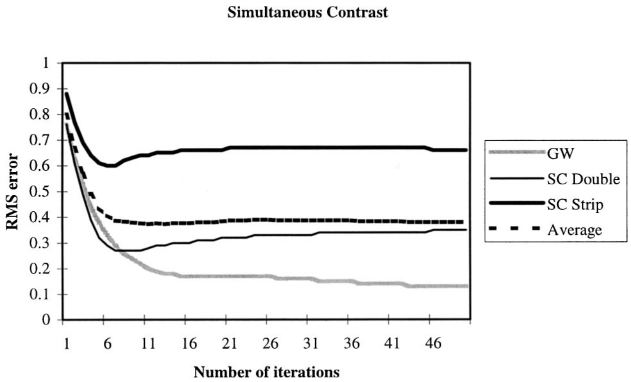


Fig. 7 SC and GW targets: rms error measuring the difference between Retinex lightness predictions and subjects' reported matching lightness as a function of the number of iterations.

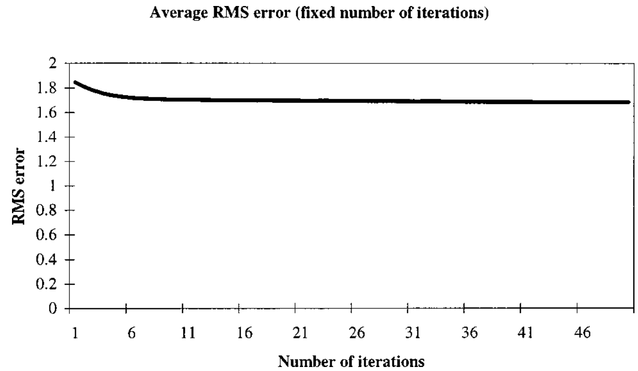


Fig. 8 Rms error in Retinex lightness prediction averaged across MMT, SB, SC, and GW experiments as a function of the number of iterations. For each choice of the number of iterations parameter, the same choice is then used for Retinex for all targets.

For the SC targets, the minimum target prediction error occurs when the number of iterations is small, as can be seen from Fig. 7. The line labeled GW shows the average rms error of Retinex predictions in lightness units for the case of a target (Fig. 4) in which there are three areas: the gray center, the white surround, and the black background. At one iteration, with a linear postLUT that expands the dynamic range of the raw Retinex output to [0...1], the rms value is 0.9. That is much larger than the standard deviation of observer results of 0.52, 0.23, and 0.13. Increasing the number of iterations to ten causes a drop in rms values to 0.2 units. From 10 to 50 iterations, the values drop from 0.2 to 0.1. For this target, any number of iterations over five does reasonably well at matching the observer data.

The thin line labeled ''double'' represents the data from Fig. 5. In this simultaneous contrast target, the prediction error (average error over all patches) is at a minimum around six or seven iterations. This is because the dark gray surround and the gray area within the dark gray surround are very sensitive to the number of iterations. This target is of particular interest, because the two central grays have different perceived lightness values, although the patches have the same luminance. With too few iterations, the calculated value for the gray in black is too high. At the point of minimum error, the calculation renders the gray-in-black one lightness unit higher than the gray-in-white. This actually conforms to the observer's predictions for this target. When the number of iterations is increased beyond seven, Retinex reports that the two grays are almost identical in lightness. This means that with too many iterations, the simultaneous contrast effect is no longer predicted correctly.

Figure 8 shows the average error for the targets from the combined MMT, SB, SC, and GW datasets versus the number of iterations. The minimum error now occurs when the number of iterations is quite large, although the curve is quite flat so the minimum is also not very distinct.

From Fig. 8 it is clear that there is no single optimal choice for the number of iterations based on minimizing the rms error measurement alone. The number of iterations required to minimize the error for one target does not necessarily minimize the error for other targets. Therefore, a stopping condition providing a method of adjusting the number of iterations automatically on a case-by-case basis is required. Note that the stopping condition cannot be based on minimizing the rms error directly, since the subjects' matches are not available to Retinex. The lightness matches are, after all, what Retinex is supposed to be predicting.

We introduce and test two possible stopping conditions: one based on the relative change in Retinex output,[10] the second based on the average brightness of the Retinex output. We refer to them as the change-based and brightness-based stopping conditions. The change-based condition measures the change in Retinex output as the number of iterations is increased from n to $n+1$, and stops when the change becomes small. Although this is analogous to the situation of numerical solution of a typical optimization problem, where the minimization process is iterated until the change becomes small enough, it is not precisely the same. The difference is in the meaning of the term ''iteration.'' In the optimization case, the entire process is repeated until convergence; whereas, in the Retinex case, the process is not being repeated in its entirety. Here the number of iterations denotes the number of times the process of cycling through the neighbors is repeated at each level.

Let R_x^n be the Retinex output at location x, when Retinex's iterations parameter has been set to n. The change-based Retinex stopping condition for an image of N pixels and threshold ϵ can be expressed as:

$$\left[\frac{\Sigma_x (R_x^{n+1} - R_x^n)^2}{N}\right]^{1/2} \leqslant \epsilon.$$

Using this stopping condition, the number of Retinex iterations will vary with the input target. What is the optimal value of ϵ? We determined an optimal value for it by a brute force search. In other words, we chose an initial high value for ϵ, ran Retinex on all the test targets and calculated the rms prediction error, decreased ϵ by a small amount, and repeated the process. A minimum occurs at $\epsilon = 0.015$. The average prediction error drops to 0.62. In comparison, the minimum average error for any fixed choice of the number of iterations (as shown in Fig. 8) was 1.71.

The second brightness-based stopping condition is based on the observation that Retinex reaches an optimal solution for bright targets (ones for which the average of all image pixel values is high) at fewer iterations than for dark ones. This effect can be seen in the ''Scale on white,'' ''Scale on gray,'' and ''Scale on black,'' targets (Figs. 1 through 3). The ''Scale on white'' target, a quite bright one, requires just three iterations. On the other hand, the darker ''Scale on gray'' and ''Scale on black'' targets require 28 iterations and 30 iterations, respectively. These are the individual number of iterations for each target that would give the best correlation with the observer matches. Intuitively, the correlation between average brightness and the optimal number of iterations is to be expected, because Retinex proceeds by subtracting from white, which has the highest average brightness. At 0 iterations, the Retinex output consists of a white image (all pixels set to 1). After each successive iteration, the average brightness of the image goes down. At an infinite number of iterations, the Retinex output image would equal the input image scaled by the maximum value in each channel.

As with the change-based stopping condition, we run the Retinex algorithm at 1,2,...*n* iterations until the stopping condition is reached. The brightness-based stopping condition is reached when the current average brightness of the Retinex output image exceeds 110% of the average brightness of the input scaled by its maximum value. The 110% value was determined empirically. The resulting slight increase in the overall image brightness can be compensated for in the Retinex postLUT. Since scaling the input by the its maximum value is equivalent to the Retinex output in the limit as the number of iterations approaches infinity, the stopping condition in essence is comparing the average lightness estimate at *n* iterations to what it would converge to at an infinite number of iterations.

This new brightness-based stopping condition yields better results than the previous incremental-change-based stopping condition,[10] in that the Retinex lightness estimates correlate better with the observer predictions. The average prediction error drops to 0.51 (brightness-based) from 0.62 (change-based). Either stopping condition error is substantially less than the 1.71 obtained in the optimal fixed-iteration case. If we look at each target individually and manually choose a number of iterations yielding the best prediction, we get an average error of 0.39. This gives a lower bound on the error that we could obtain with a perfect stopping condition.

7 Conclusion

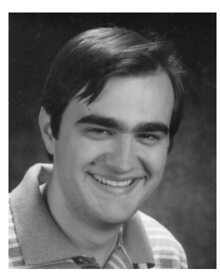
Our goal is to study the effects of the number of iterations in the special case where all spatial channels use the same number of iterations. Further, this study uses the same pattern of spatial comparisons. However, Retinex requires the parameters' postLUT and number of iterations be set. We introduce methods for setting these parameters automatically. Using these methods, Retinex yields an average rms prediction error of only 0.51 units on a 1-to-9 lightness scale in predicting the available psychophysical data. By comparison, optimization for a fixed setting for the number of iterations resulted in an overall average rms error of 1.71, so the new automatic-stopping-condition technique constitutes a significant improvement over a single choice for the number of iterations. Since the method changes only Retinex's input parameters, the Retinex model itself has not changed. However, the advantage of using the Retinex model in conjunction with automatic parameter selection is that it can be applied in a hands-off manner without requiring further intervention. Future work will include modifying Retinex to employ different numbers of iterations automatically at each pyramid level.

Acknowledgments

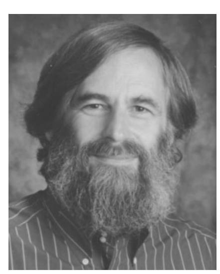
The authors gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada. We thank John McCann for supplying data and comments.

References

1. B. Funt, F. Ciurea, and J. J. McCann, ''Retinex in Matlab,'' *J. Electron. Imaging* **13**, 68–72 (2004) (in this issue).
2. B. Funt and F. Ciurea, ''Control parameters for Retinex,'' *Proc. 9th Cong. Intl. Color Assoc.*, pp. 287–290 (2001).
3. J. J. McCann, S. P. McKee, and T. H. Taylor, ''Quantitative studies in Retinex theory,'' *Vision Res.* **16**, 445–458 (1976).
4. L. G. Glasser *et al.*, ''Cube-root color coordinate system,'' *J. Opt. Soc. Am. A* **48**, 736–740 (1958).
5. W. A. Stiehl, J. J. McCann, and R. L. Savoy, ''Influence of intraocular scattered light on lightness-scaling experiments,'' *J. Opt. Soc. Am.* **73**, 1143–1148 (1983).
6. J. J. McCann, E. H. Land, and S. M. V. Tatnall, ''A technique for comparing human visual responses with a mathematical model for lightness,'' *Am. J. Optometry Arch. Am. Academy Optometry*, **47**, 845–855 (1970).
7. J. J. McCann, ''The role of nonlinear operations in modeling human color sensations,'' *Proc. SPIE* **1077**, 355–363 (1989).
8. J. J. McCann and A. Rizzi, ''The spatial properties of contrast,'' *Proc. IS&T/SID 11th Color Imag. Conf.* (in press).
9. R. Sobol, ''Improving the Retinex algorithm for rendering wide dynamic range photographs,'' *Proc. SPIE* **4662**, 341–348 (2002).
10. B. Funt, F. Ciurea, and J. J. McCann, ''Tuning Retinex parameters,'' *Proc. SPIE* **4662**, 358–366 (2002).



Florian Ciurea received his BSc in computer science from Politehnica University of Bucharest, Romania. He is currently a PhD student in computing science at Simon Fraser University. He is interested in color imaging, with a focus on color correction and computational models for color constancy.



Brian Funt is a professor of computing science at Simon Fraser University. After receiving his PhD from the University of British Columbia in 1976, he spent two years at Stanford as a postdoctoral researcher, followed by two years as a professor at the State University New York, Buffalo. He has been at Simon Fraser University since 1980. His work on computational models of color perception began in the early 1980s and has covered color constancy, color object recognition, high dynamic range color imaging, and color in computer vision. He is a Marr Prize recipient. This year he is the general co-chair of the Eleventh Color Imaging Conference.