2015/06/16 v0.5.1 IPOL article class

Automatic Color Enhancement (ACE) and its Fast Implementation

Pascal Getreuer

CMLA, ENS Cachan, France (getreuer@cmla.ens-cachan.fr)

Communicated by Vicent Caselles   *Demo edited by* Pascal Gertreuer

Abstract

Automatic Color Enhancement "ACE" is an effective method for color image enhancement introduced by Gatta, Rizzi, and Marini based on modeling several low level mechanisms of the human visual system. The direct computation of ACE on an $N \times N$ image costs $\mathcal{O}(N^4)$ operations. This article describes two fast approximations of ACE. First, the algorithm of Bertalmío, Caselles, Provenzi, and Rizzi uses a polynomial approximation of the slope function to decomposes the main computation into convolutions, reducing the cost to $\mathcal{O}(N^2 \log N)$. Second, an algorithm based on interpolating intensity levels also reduces the main computation to convolutions. The use of ACE for image enhancement and color correction is demonstrated.

Source Code

ANSI C source code to produce the same results as the demo is accessible on the article web page https://doi.org/10.5201/ipol.2012.g-ace.

Keywords: image enhancement, image color correction

1 Introduction

The Automatic Color Enhancement "ACE" method of Gatta, Rizzi, and Marini [9] and further developed by Rizzi, Gatta, and Marini [10, 11] and Bertalmío, Caselles, Provenzi, and Rizzi [13] is an effective color correction and enhancement method based on a simple model of the human visual system. The method is inspired by the following low level mechanisms:

- "gray world," the average perceived color is gray [5]
- "white patch," normalization toward a white reference [3]
- lateral inhibition [2]
- local-global adaptation [6]

By modeling these mechanisms, an image enhancement method can simulate the process of perception. The enhanced image appears natural because the input image is adjusted in a manner consistent with perception. This motivation is similar to the Retinex color perception model of Land and McCann [4]. ACE and Retinex are compared in detail in other works [11, 15, 16].

2   Histogram Equalization

Bertalmío et al. [13] proved that ACE has a connection with histogram equalization, which we briefly review here. Uniform histogram equalization is a method to modify an image so that it has a uniform histogram.

Let $I : \Omega \to [0,1]$ denote the input grayscale image with domain Ω and intensities scaled in $[0,1]$. For a color image, the following is performed independently on the red, green, and blue (RGB) channels. Define the normalized cumulative histogram F,

$$F(\lambda) = \tfrac{1}{|\Omega|} \big| \{ x \in \Omega : I(x) \le \lambda \} \big|, \quad \lambda \in [0,1], \tag{1}$$

here $|\Omega|$ is the total number of pixels in the image. Considering image intensity I as a random variable, the histogram computes its cumulative distribution $F(\lambda) = \mathbf{P}(I \le \lambda)$. The histogram equalized image is obtained as $F\big(I(x)\big)$. Provided F is invertible, the equalized image $F\big(I(x)\big)$ has uniformly distributed intensity because

$$\mathbf{P}\big(F(I) \le \lambda\big) = \mathbf{P}\big(I \le F^{-1}(\lambda)\big) = F\big(F^{-1}(\lambda)\big) = \lambda. \tag{2}$$

In practice, F is often not invertible due to quantization, however, the resulting image still has an approximately uniform histogram.

3   ACE

Let I denote the input grayscale image or a given chromatic channel in a color image with domain Ω and intensity values scaled in $[0,1]$. For a color image, the following operation is performed independently on the RGB channels:

$$R(x) = \sum_{y \in \Omega \setminus x} \frac{s_\alpha\big(I(x) - I(y)\big)}{\|x - y\|}, \quad x \in \Omega, \tag{3}$$

where $\Omega \setminus x$ denotes $\{ y \in \Omega : y \ne x \}$, $\|x - y\|$ denotes Euclidean distance, and $s_\alpha : [-1,1] \to \mathbb{R}$ is the slope function $s_\alpha(t) = \min\{ \max\{ \alpha t, -1 \}, 1 \}$ for some $\alpha \ge 1$.
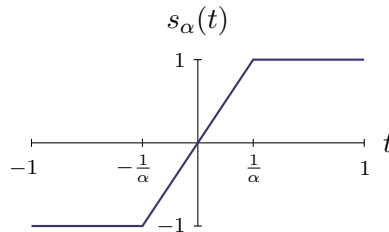
In the limit $\alpha \to \infty$, it is the signum function $s_\infty(t) = \mathrm{sign}(t)$. In the second stage, the enhanced channel is computed by stretching R to $[0,1]$ as

$$L(x) = \frac{R(x) - \min R}{\max R - \min R}. \tag{4}$$

The first stage of the method (3) adapts local image contrast. Lateral inhibition is simulated by neighbor differences $I(x) - I(y)$ and weighting according to distance $\|x - y\|$. The function s_α amplifies small differences and saturates large differences, which has the effect of expanding or compressing the dynamic range according to the local image content. The second stage (4) adapts

the image to obtain a global white balance. By implementing these mechanisms, ACE is a simplified model of the human visual system: the enhancement process is consistent with perception [10, 11].

Bertalmío et al. [13] developed a variational interpretation of ACE. It is shown that ACE is equivalent to solving the minimization problem

$$\arg\min_I \frac{1}{2}\sum_x \left(I(x) - \tfrac{1}{2}\right)^2 - \frac{1}{4M}\sum_x\sum_{y\neq x} \omega(x,y)S_\alpha\big(I(x) - I(y)\big), \tag{5}$$

where $S'_\alpha = s_\alpha$, $\omega(x,y) = 1/\|x - y\|$, and $M = \max_x R(x)$. Furthermore, it is shown that uniform histogram equalization is a minimizer when $\alpha = \infty$ and $\omega(x,y) = 1$. Thus ACE may be seen as a smoothed and localized modification of uniform histogram equalization
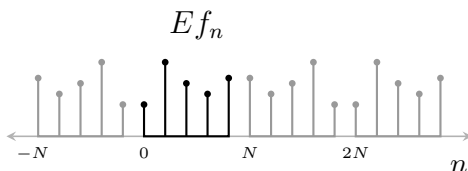
Variations of the ACE algorithm have also been considered [9, 10, 11]:

- other slope functions s_α
- weight functions other than $1/\|x - y\|$
- y can be restricted to a window around x in the summation
- other normalizations for L (4)

While ACE produces high-quality enhancement, a significant obstacle is that direct computation of (3) is impractically expensive, costing $\mathcal{O}(N^4)$ operations for an $N \times N$ image. The following sections describe two fast approximations of ACE that reduce the cost to $\mathcal{O}(N^2 \log N)$.

3.1 Boundary Handling and Convolutions

Both algorithms presented here rely ultimately on performing fast convolutions, and for this reason, we change ACE's boundary handling to the more convolution friendly half-sample symmetric extension. Define the half-sample symmetric extension Ef of an N-sample sequence f,

$$Ef_n = \begin{cases} f_n & \text{if } n = 0, \ldots, N-1, \\ Ef_{-1-n} & \text{if } n < 0, \\ Ef_{2N-1-n} & \text{if } n \geq N. \end{cases}$$

The definition is recursive since multiple reflections may be needed to obtain an index between 0 and $N - 1$. We also consider the tensor product of this extension applied to an $N \times N$ image $u_{i,j}$, $i = 0, \ldots, N - 1$, $j = 0, \ldots, N - 1$. Noting that Ef is $2N$-periodic, it can also be defined as the periodization of the reflected sequence $f_0, \ldots, f_{N-1}, f_{N-1}, \ldots, f_0$. The domain can be interpreted to be a circle of $2N$ samples. In two dimensions, the domain is the $2N \times 2N$-periodic torus $\mathbb{T}^2$.

For any $x, y \in \mathbb{T}^2$, distance is defined on the torus as

$$d(x,y) := \min_{\bar{x},\bar{y}}\big\{|\bar{x} - \bar{y}| : \bar{x} \equiv x, \bar{y} \equiv y\big\}, \tag{6}$$

where $|v| := \sqrt{v_1^2 + v_2^2}$ and $\equiv$ denotes equivalence on the torus. It can be shown that $d(x,y)$ is an even $2N \times 2N$-periodic function of $(x - y)$, which allows us to write $d(x,y) = d(x - y) = d(y - x)$.

The summation $R(x)$ (3) is redefined as a summation over the torus $\mathbb{T}^2\backslash x$, and Euclidean distance $\|x - y\|$ is replaced by torus distance $d(x - y)$:

$$R(x) = \sum_{y\in\mathbb{T}^2\backslash x} \frac{s_\alpha\big(I(x) - I(y)\big)}{d(x - y)}, \quad x \in \Omega. \tag{7}$$
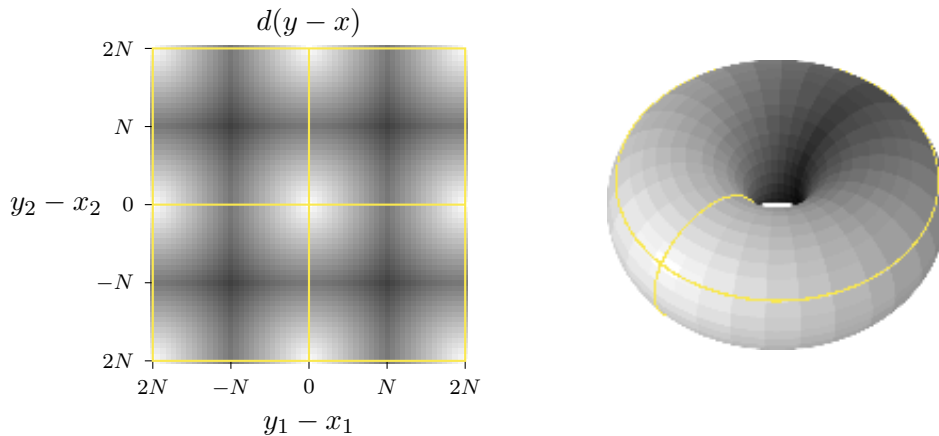
268

Figure 1: Distance on $\mathbb{T}^2$ (6).

Notice that by defining

$$\omega(x - y) = \begin{cases} 0 & \text{if } x = y, \\ 1/d(x - y) & \text{if } x \neq y, \end{cases} \tag{8}$$

the domain of summation can be extended to the whole torus $\mathbb{T}^2$. Henceforth we compute R as

$$R(x) = \sum_{y \in \mathbb{T}^2} \omega(x - y) s_\alpha\big(I(x) - I(y)\big). \tag{9}$$

Both algorithms will approximate R in terms of convolutions with ω on $\mathbb{T}^2$. Fast Fourier transforms (FFT) may be used to evaluate these convolutions in $\mathcal{O}(N^2 \log N)$ operations. However, since ω is even in both coordinates, the discrete cosine transform (DCT) may be used instead for even greater computational efficiency. The data does not need to be padded in this case because symmetric boundaries are implied by the transforms, which reduces the cost compared to FFTs. Martucci [7] showed that convolution with half-sample symmetric boundaries can be performed through DCT transforms as

$$\omega * I = \mathcal{C}_{2e}^{-1}\big(\mathcal{C}_{1e}(\omega) \cdot \mathcal{C}_{2e}(I)\big),$$

where $\mathcal{C}_{1e}$ and $\mathcal{C}_{2e}$ are the unnormalized DCT-I and DCT-II transforms of the same period lengths. In one dimension, these transforms are

$$\mathcal{C}_{1e}(h) = h_0 + (-1)^k h_N + 2 \sum_{n=1}^{N-1} h_n \cos(\pi n k/N) \qquad k = 0, \ldots, N, \tag{10}$$

$$\mathcal{C}_{2e}(x) = 2 \sum_{n=0}^{N-1} x_n \cos\big(\pi(n + \tfrac{1}{2})k/N\big) \qquad k = 0, \ldots, N-1. \tag{11}$$

The transforms in higher dimensions are obtained by tensor product. Note that the DCT-I transform is one sample longer ($k = 0, \ldots, N$) than the other two transforms ($k = 0, \ldots, N-1$). In the pointwise multiplication ($\cdot$), the first N coefficients are multiplied. See Martucci [7] and Getreuer [17] for further details.

4   Polynomial Slope Function

Bertalmío, Caselles, Provenzi, and Rizzi [13] showed that by replacing s_α with a polynomial, the summation in R can be decomposed into convolutions, reducing the complexity to $\mathcal{O}(N^2 \log N)$.

4.1 Fast Computation with Convolutions

The key change to the ACE method is to approximate $\min\{\max\{\alpha t, -1\}, 1\}$ with an odd polynomial approximation,

$$s_\alpha(t) \approx \sum_{m=1}^{M} c_m t^m. \tag{12}$$

The exact choice for the approximating polynomial will be discussed in the next section. It is then possible to decompose R into a sum of convolutions:

$$
\begin{aligned}
R(x) &= \sum_{y \in \mathbb{T}^2} \omega(x-y) \sum_{m=1}^{M} c_m \big(I(x) - I(y)\big)^m \\
&= -\sum_{y \in \mathbb{T}^2} \omega(x-y) \sum_{m=1}^{M} c_m \big(I(y) - I(x)\big)^m \\
&= -\sum_{y \in \mathbb{T}^2} \omega(x-y) \sum_{m=1}^{M} c_m \sum_{n=0}^{m} \binom{m}{n} I(y)^n \big(-I(x)\big)^{m-n} \\
&= \sum_{n=0}^{M} \underbrace{\left(\sum_{m=n}^{M} c_n \binom{m}{n} (-1)^{m-n+1} I(x)^{m-n} \right)}_{a_n(x)} \sum_{y \in \mathbb{T}^2} \omega(y-x) I(y)^n \\
&= \sum_{n=0}^{M} a_n(x)(\omega * I^n)(x), \tag{13}
\end{aligned}
$$

where $*$ is cyclic convolution over $\mathbb{T}^2$. For each x, the evaluation of $a_n(x)$ costs $\mathcal{O}(1)$ operations. The convolutions can be efficient computed with DCT transforms in $\mathcal{O}(N^2 \log N)$ operations. Note that for the term $n = 0$, explicit computation of the convolution is not needed since $I^0 \equiv 1$. For an RGB color image, $3M$ convolutions need to be computed.

The summation over n can be parallelized since the evaluation of $a_n(x)(\omega * I^n)(x)$ is independent for each n. OpenMP is used in the implementation included with this article to evaluate the summands concurrently.

4.2 Polynomial Approximation

The slope function $s_\alpha(t) = \min\{\max\{\alpha t, -1\}, 1\}$ is approximated with an odd polynomial,

$$s_\alpha(t) \approx \sum_{m=1}^{M} c_m t^m.$$

The input image is assumed to have intensities scaled in $[0, 1]$, so the argument t is guaranteed to be between -1 and 1. By the Stone–Weierstrass theorem, the continuous function $s_\alpha(t)$ can be uniformly approximated on $[-1, 1]$ by a polynomial with any desired precision. However, the degree of the polynomial affects the computational cost ($3M$ convolutions must be evaluated for an Mth degree polynomial), so a compromise is necessary between accuracy and speed.

For a fixed polynomial degree M, we select the coefficients c_m to minimize the maximum absolute error over $[-1, 1]$,

$$\min_{c} \max_{t \in [-1,1]} \left| s_\alpha(t) - \sum_{m=1}^{M} c_m t^m \right|. \tag{14}$$

α	Polynomial	Max error
1	t	0.000
2	$1.85623249t + 3.82397125t^3 - 19.70879455t^5 + 26.15510902t^7 - 11.15375327t^9$	0.028
3	$3.51036396t - 6.31644952t^3 + 0.92439798t^5 + 9.32834829t^7 - 6.50264005t^9$	0.057
4	$4.76270090t - 18.23743983t^3 + 36.10529118t^5 - 31.35677926t^7 + 9.66532431t^9$	0.061
5	$5.64305564t - 28.94026159t^3 + 74.52401661t^5 - 83.54012582t^7 + 33.39343065t^9$	0.081
6	$6.19837979t - 35.18789052t^3 + 95.28157108t^5 - 109.95601312t^7 + 44.78177264t^9$	0.118
7	$6.69888108t - 41.02503190t^3 + 115.02784036t^5 - 135.35603880t^7 + 55.81014424t^9$	0.156
8	$7.15179080t - 46.43557440t^3 + 133.54648929t^5 - 159.34156394t^7 + 66.27157886t^9$	0.193

Table 1: Optimal 9th degree approximation of s_α for different α. Computed using CVX [12].
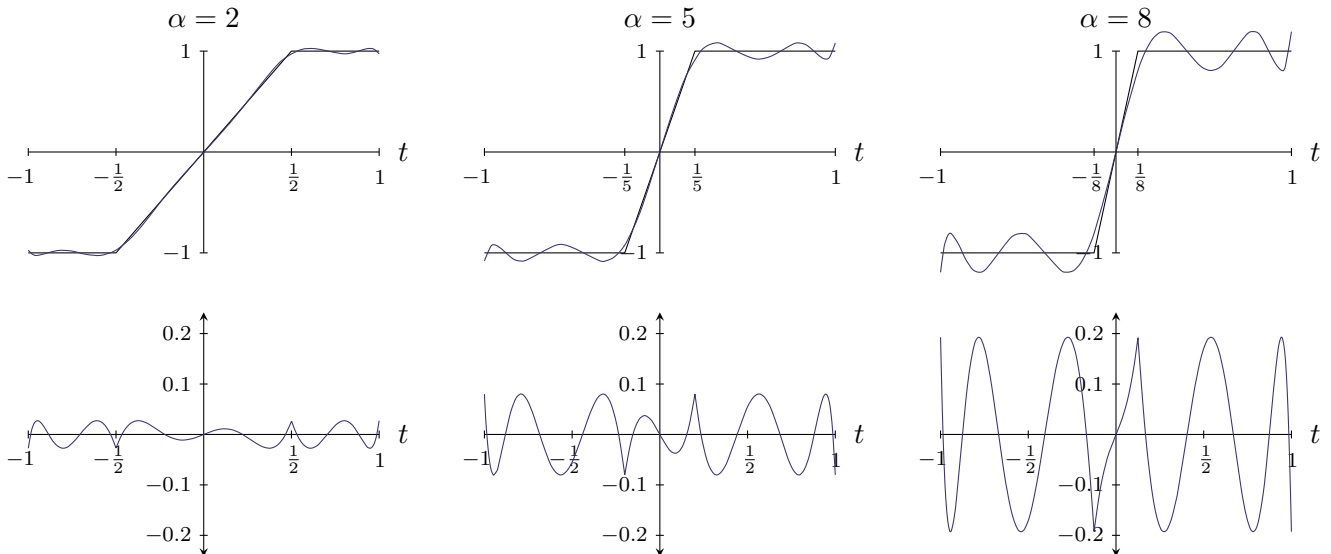

Figure 2: *Top row:* s_α and its 9th degree approximation. *Bottom row:* approximation error.

The optimal c can be found using the Remez algorithm [1]. Table 1 lists the optimal 9th degree coefficients for integer values of α from 1 to 8. These approximations are used by the implementation included with this article. For fixed polynomial degree, the approximation error increases with α (Figure 2).

5 Interpolation

Here we describe another algorithm, which was generously suggested by an anonymous reviewer, that decomposes the computation of R (3) into convolutions by using interpolation. We again use the boundary handling developed in §3.1 and DCT-based convolutions. Define the sum

$$R(x; L) = \sum_{y \in \mathbb{T}^2} \omega(x - y) s_\alpha\big(L - I(y)\big), \tag{15}$$

where $I(x)$ in (9) has been replaced by a constant L. Since the argument of s_α now depends only on y, the sum is a convolution and can be computed in $\mathcal{O}(N^2 \log N)$ operations. This allows for a fast algorithm to approximate ACE.

Let (L_j) be a sequence such that $\min I = L_1 < L_2 < \cdots < L_J = \max I$, and compute $R(x; L_j)$,

$j = 1, \ldots, J$. Then approximate $R(x) = R(x; I(x))$ by piecewise linear interpolation,

$$R(x) \approx R(x; L_j) + \frac{R(x; L_{j+1}) - R(x; L_j)}{L_{j+1} - L_j}(I(x) - L_j), \quad j \text{ such that } L_j \le I(x) \le L_{j+1}. \qquad (16)$$

For an RGB color image, this algorithm costs $3J$ convolutions, which can be computed in parallel. The interpolations (16) can also be computed in parallel by dividing the task as thread j interpolating the half-closed interval $[L_j, L_{j+1})$ (or $[L_{J-1}, L_J]$ for the rightmost interval).

A direct implementation strategy would begin by computing and storing all $3J$ convolutions, but this is a demanding amount of memory. To reduce the memory cost, the included implementation computes (concurrently) and stores convolutions for just a few consecutive levels $L_i, \ldots, L_{i+K}$ at a time, interpolates over $[L_j, L_{j+1})$, $j = i, \ldots, i+K-1$, and then the process is repeated with $i \leftarrow i+K$ until all intervals have been covered.

In the examples, (L_j) is uniformly spaced, $L_j = \min I + (\max I - \min I)\frac{j-1}{J-1}$. We find that using $J = 8$ levels provides an accurate approximation for typical images.

6 Examples

This section shows the effects of the parameters and compares the two approximate algorithms. Computation times are reported using the implementation included with this article running on a laptop with two cores.[1] For reference, some examples include the result with uniform histogram equalization (HE). Unless otherwise specified, ACE is performed with $\alpha = 5$, $\omega(x, y) = 1/\sqrt{x^2 + y^2}$ using the level interpolation algorithm with $J = 8$ levels.

6.1 Effect of Varying α

The α parameter specifies the slope at $t = 0$ of the function $s_\alpha(t) = \min\{\max\{\alpha t, -1\}, 1\}$. The larger this parameter, the stronger the enhancement. Ignoring the spatial weighting, uniform histogram equalization corresponds to $\alpha = \infty$.

| Input (352×480) | ACE, $\alpha = 2$ | ACE, $\alpha = 4$ | ACE, $\alpha = 8$ | HE |
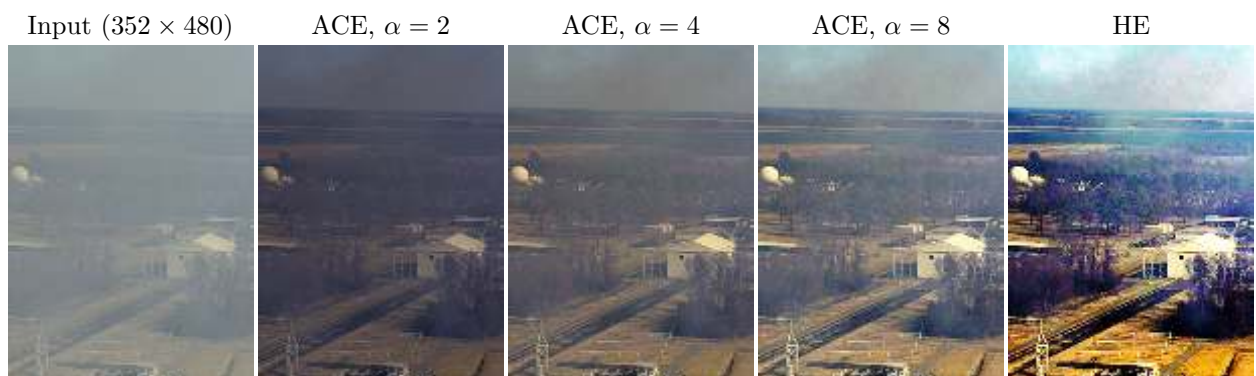
6.2 Effect of Varying ω

The following experiments show the enhancement results for $\alpha = 8$ when ω is a Gaussian with standard deviation 5, 25, or 100, the default $\omega(x, y) = 1/\sqrt{x^2 + y^2}$, or uniform weighting $\omega \equiv 1$. The leftmost image shows that ACE acts like an edge detector when ω is finely localized.

---

[1] 2.40GHz Intel® Core™ 2 Duo T7700 with 2GB RAM

6.3 Approximation

The following experiment investigates the cost/quality tradeoff of the two fast ACE approximations, polynomial slope approximation (§4) and level interpolation (§5). Polynomial slope approximation is tested with degrees $3, 5, 7, 9, 11$ and level interpolation is tested with number of levels $4, 5, \ldots, 12$. For the test images, we use the Kodak Image Suite,[2] a set of 24 natural color images. The method parameters are $\alpha = 5$ and $\omega(x, y) = 1/\sqrt{x^2 + y^2}$.

The scatter plot below shows runtime vs. root-mean-square error (RMSE)[3] for every combination of image and method order. This allows to assess cost/quality tradeoff as well as to visualize how RMSE varies across different images. Polynomial slope approximations are labeled p3, p5, etc., and for each degree, the ◆ markers indicate the median RMSE over the 24 images. Level interpolations are labeled i4, i6, etc., and ● markers indicate the median RMSE for each number of levels.
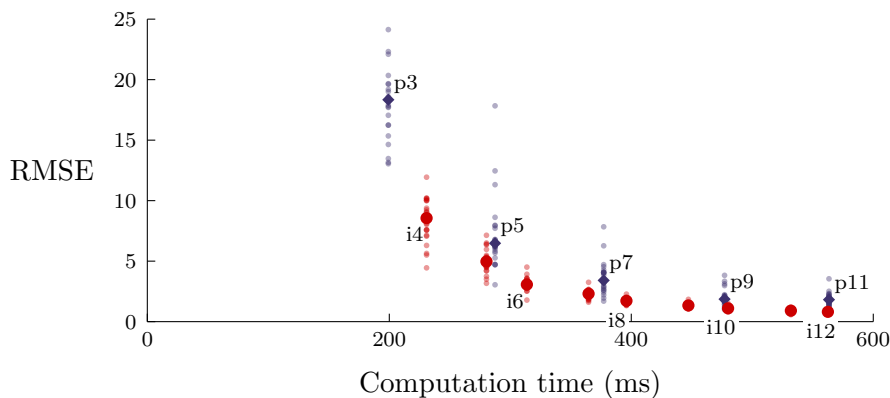
The scatter plot suggests that the two algorithms have similar cost/quality tradeoff with level interpolation being slightly more efficient. We observe that level interpolation is faster for the same number of convolutions (i.e., polynomial degree = number of levels) since the linear interval interpolations are simpler than evaluating the $a_n(x)$ polynomials. Several results for image #21 are shown below. All results appear very similar, even for low degree or number of levels.

---

[2]Available online at http://www.cipr.rpi.edu/resource/stills/kodak.html

[3]RMSE values are relative to the intensity range $[0, 255]$. The exact solution was computed using the interpolation algorithm with 256 levels.

Input (768×512)	Polynomial degree 5	Polynomial degree 9

Exact	Interpolation with 4 levels	Interpolation with 8 levels
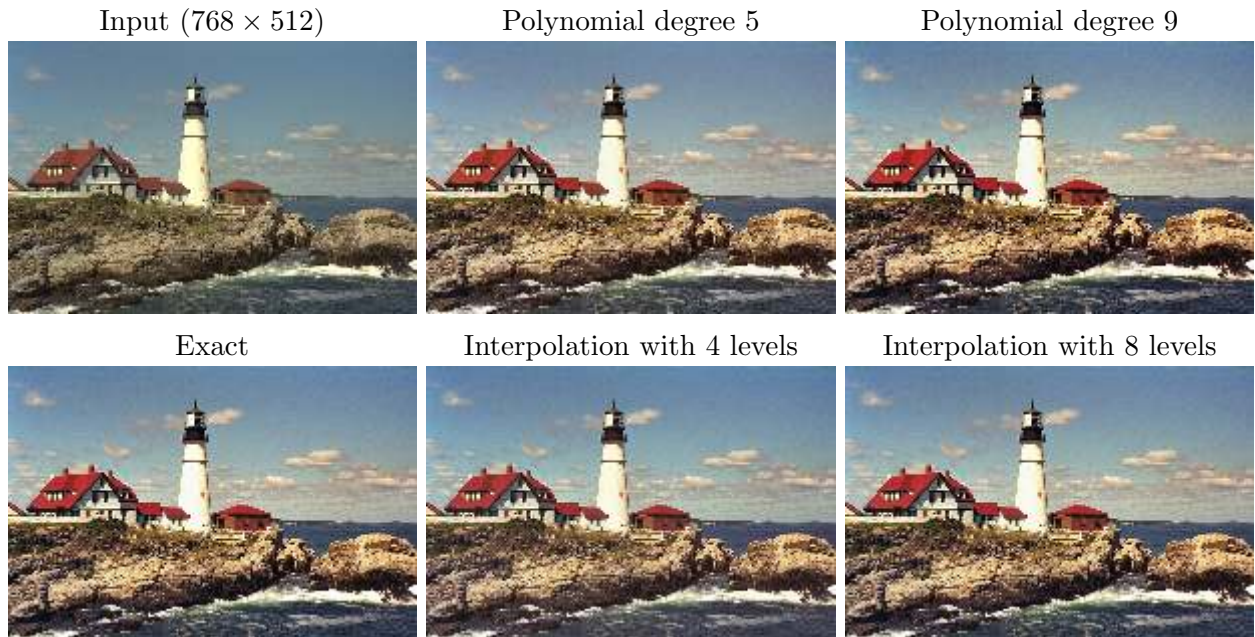
6.4 Enhancement

ACE and histogram equalization are applied here to improve image contrast. Enhancement can also be used to correct images with poor exposure.

Input (331×248)	ACE, $\alpha = 5$ (291 ms)	HE (1 ms)

Input (480×480)	ACE, $\alpha = 5$ (230 ms)	HE (3 ms)
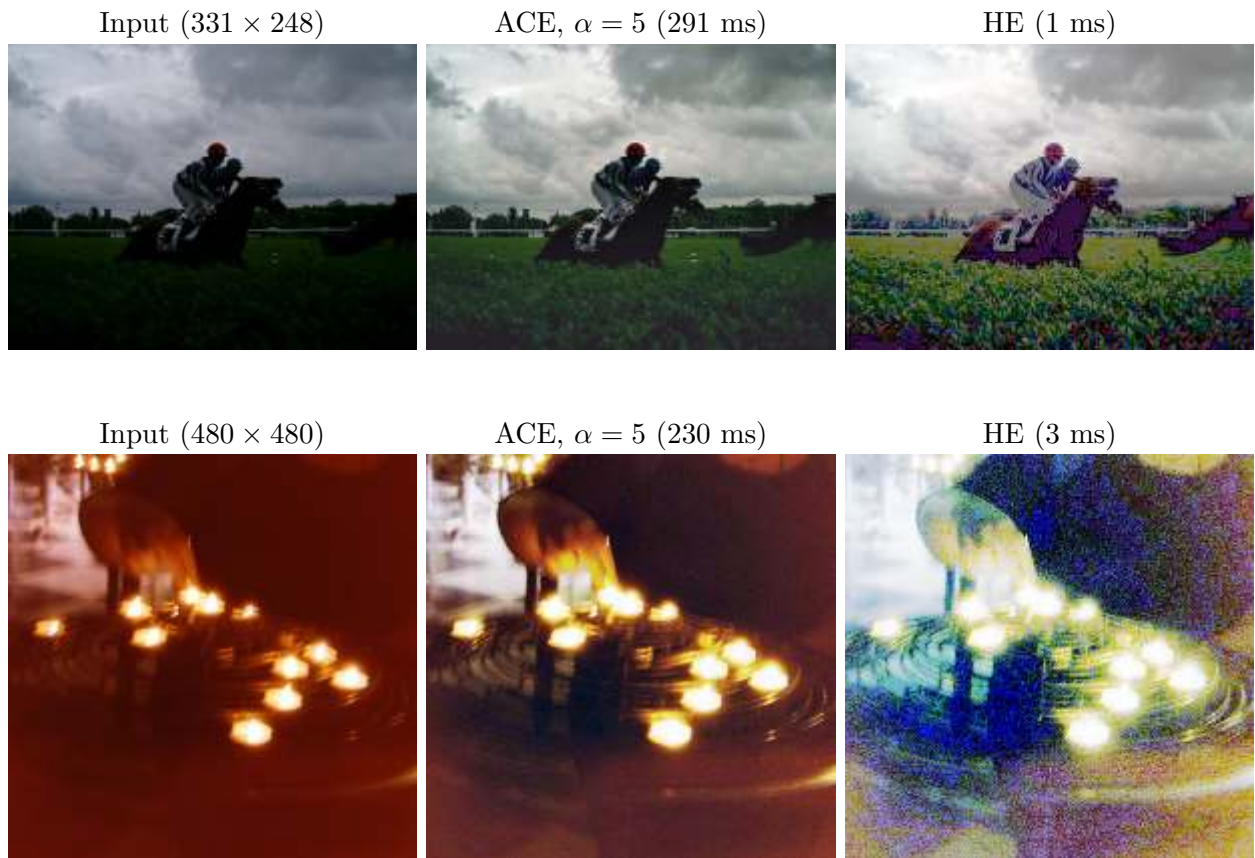
Histogram equalization is effective but sometimes too harsh, producing artificial colors in the dark foreground. ACE is more moderate and avoids these artifacts.

6.5 Color Correction

ACE can also be used to correct film that has discolored and faded from aging [8, 14]. The examples in this section demonstrate restoration of images scanned from old Kodachrome and Ektachrome film. These images, originally created by the photographer Erwin Blumenfeld, are provided courtesy of his estate.

"Untitled, circa 1950"	ACE, $\alpha = 5$	HE

"Self-portrait, NY, circa 1960"	ACE, $\alpha = 5$	HE

"New-York Venus, 1952"	ACE, $\alpha = 5$	HE
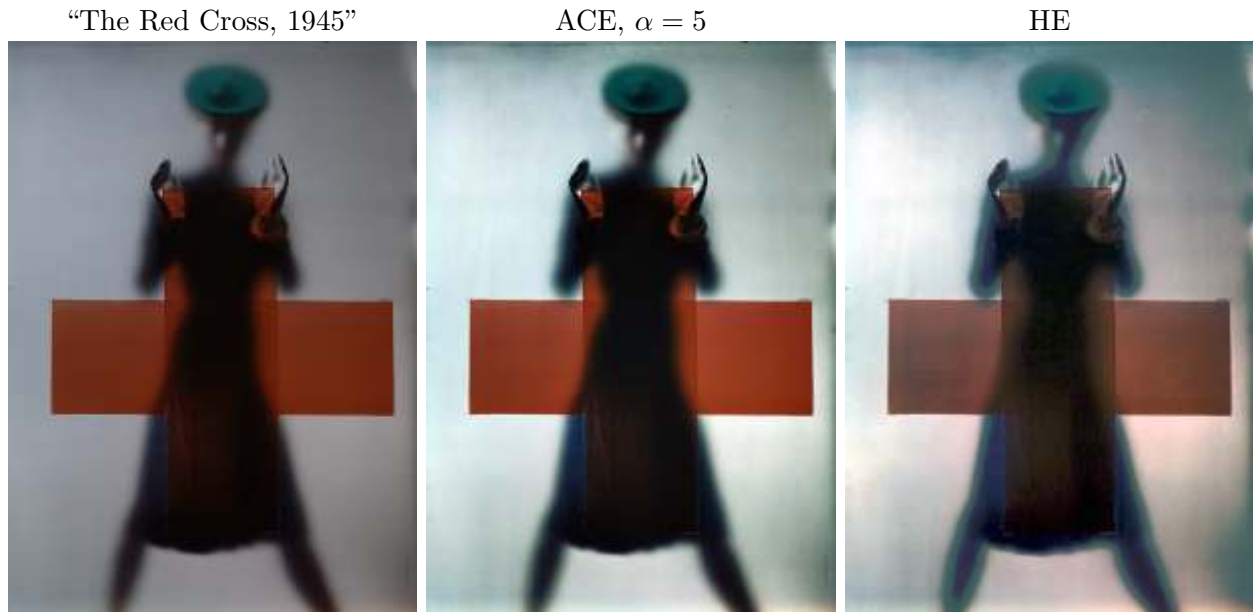
"The Red Cross, 1945"	ACE, $\alpha = 5$	HE

Acknowledgments

Image Credits

NASA (http://dragon.larc.nasa.gov/retinex/)[4]

Alan Fink, Kodak Image Suite #21 [5]

Ana Belén Petro, CC-BY-NC-SA

Adrian Boliston (http://www.flickr.com/photos/boliston/3936248103/), CC-BY

©The Estate of Erwin Blumenfeld, CC-BY-NC-SA[6]

References

[1] E. Ya. Remez, "Sur la détermination des polynômes d'approximation de degré donnée," Communications de la Societé Mathématique de Kharkov 10, 41, 1934.

[2] H.K. Hartline, H.G. Wagner, F. Ratcliff, "Inhibition in the eye of limulus," Journal of General Physiology, vol. 39, no. 5, pp. 651–673, 1956.

---

[4]Copyright information: http://www.nasa.gov/audience/formedia/features/MP_Photo_Guidelines.html.
[5]Copyright information: http://r0k.us/graphics/kodak/PhotoCD_credits.txt.
[6]This license only applies to the low-resolution versions included in this article.

[3] J. von Kries, "Sources of color science," In: MacAdam, David L. (Ed.), Chromatic Adaptation, MIT Press, Cambridge, MA, pp. 109–119, 1970.

[4] E.H. Land, J.J. McCann, "Lightness and Retinex Theory," Journal of the Optical Society of America 61, 1–11, 1971. http://dx.doi.org/10.1364/JOSA.61.000001

[5] G. Buchsbaum, "A spatial processor model for object color perception," Journal of the Franklin Institute vol. 310, no. 1, pp. 1–26, 1980.

[6] J.J. McCann, "Local/global mechanisms for color constancy," Die Farbe vol. 34, pp. 275–283, 1987.

[7] S. Martucci, "Symmetric convolution and the discrete sine and cosine transforms," IEEE Transactions in Signal Processing SP-42, pp. 1038–1051, 1994. http://dx.doi.org/10.1109/78.295213

[8] R. Gschwind, F.S. Frey, "Digital Reconstruction of Faded Color Photographs," *Revue Informatique et Statistique dans les Sciences humaines*, vol. 33, no. 1–4, pp. 253–274, 1997.

[9] C. Gatta, A. Rizzi, D. Marini, "ACE: An automatic color equalization algorithm," Proceedings of the First European Conference on Color in Graphics Image and Vision (CGIV02), 2002.

[10] A. Rizzi, C. Gatta, D. Marini, "A new algorithm for unsupervised global and local color correction," Pattern Recognition Letters, vol. 124, pp. 1663–1677, 2003. http://dx.doi.org/10.1016/S0167-8655(02)00323-9

[11] A. Rizzi, C. Gatta, D. Marini, "From Retinex to automatic color equalization: Issues in developing a new algorithm for unsupervised color equalization," Journal of Electronic Imaging, vol. 13, no. 1, pp. 75–84, 2004. http://dx.doi.org/10.1117/1.1635366

[12] M. Grant, S. Boyd, Y. Ye, "Disciplined convex programming," in *Global Optimization: from Theory to Implementation, Nonconvex Optimization and Its Applications*, L. Liberti and N. Maculan, eds., Springer, vol. 84, Part II, pp. 155–210, 2006. http://dx.doi.org/10.1007/0-387-30528-9_7 (Library homepage: http://cvxr.com/cvx)

[13] M. Bertalmío, V. Caselles, E. Provenzi, A. Rizzi, "Perceptual Color Correction Through Variational Techniques," IEEE Transactions in Image Processing, vol. 16, no. 4, pp. 1058–1072, 2007. http://dx.doi.org/10.1109/TIP.2007.891777

[14] D. Nikitenko, M. Wirth, K. Trudel, "Applicability of white-balancing algorithms to restoring faded colour slides: An empirical evaluation," *Journal of Multimedia*, vol. 3, no. 5, 2008.

[15] R. Palma-Amestoy, E. Provenzi, M. Bertalmío, V. Caselles, "A Perceptually Inspired Variational Framework for Color Enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 458–474, 2009. http://dx.doi.org/10.1109/TPAMI.2008.86

[16] M. Bertalmío, V. Caselles, E. Provenzi, "Issues about Retinex Theory and Contrast Enhancement," *International Journal of Computer Vision*, vol. 83, no. 1, pp. 101–119, 2009. http://dx.doi.org/10.1007/s11263-009-0221-5

[17] P. Getreuer, "Total Variation Deconvolution using Split Bregman," *Image Processing On Line*, 2012. http://dx.doi.org/10.5201/ipol.2012.g-tvdc