

# Decolorize: fast, contrast enhancing, color to grayscale conversion

Mark Grundland\*, Neil A. Dodgson  
Computer Laboratory, University of Cambridge

## Abstract

We present a new contrast enhancing color to grayscale conversion algorithm which works in real-time. It incorporates novel techniques for image sampling and dimensionality reduction, sampling color differences by Gaussian pairing and analyzing color differences by predominant component analysis. In addition to its speed and simplicity, the algorithm has the advantages of continuous mapping, global consistency, and grayscale preservation, as well as predictable luminance, saturation, and hue ordering properties.

**Keywords:** Color image processing; Color to grayscale conversion; Contrast enhancement; Image sampling; Dimensionality reduction.

## 1 Introduction

We present a new contrast enhancing color to grayscale conversion algorithm which works in real-time. Compared with standard color to grayscale conversion, our approach better preserves equiluminant chromatic contrasts in grayscale. Alternatively, compared with standard principal component color projection, our approach better preserves achromatic luminance contrasts in grayscale. Finally, compared with recent optimization methods [1,2] for contrast enhancing color to grayscale conversion, this approach proves much more computationally efficient at producing visually similar results with stronger mathematical guarantees on its performance.

Many legacy pattern recognition systems and algorithms only operate on the luminance channel and entirely ignore chromatic information. By ignoring important chromatic features, such as edges between isoluminant regions, they risk reduced recognition rates. Although luminance predominates human visual processing, on its own it may not capture the perceived color contrasts. On the other hand, maximizing variation by principal component analysis can result in high contrasts, but it may not prove visually meaningful. Our algorithm is designed to enhance contrast in a visually meaningful way by augmenting luminance to reflect chromatic differences (Figure 1). It features novel, generally applicable strategies for image sampling and dimensionality reduction as well as task specific methods for color representation and dynamic range scaling. It performs global color to grayscale conversion by expressing grayscale as a continuous, image dependent, piecewise linear mapping of the RGB color primaries and their saturation. The degree of contrast enhancement, the scale of contrast features, and the need for noise suppression can easily be adjusted to suit its application. The enhanced grayscale image can take the place of the luminance image in existing systems for displaying, analyzing, and recognizing images. This can be especially useful when printing color images in grayscale. By rendering color contrasts in grayscale with less detail loss, we offer a more informative picture for visual inspection and interpretation.

---

\* Corresponding author: Mark Grundland (Mark@eyemaginary.com).

To obtain this paper with color illustrations, please visit: <http://www.eyemaginary.com/Portfolio/Publications.html>

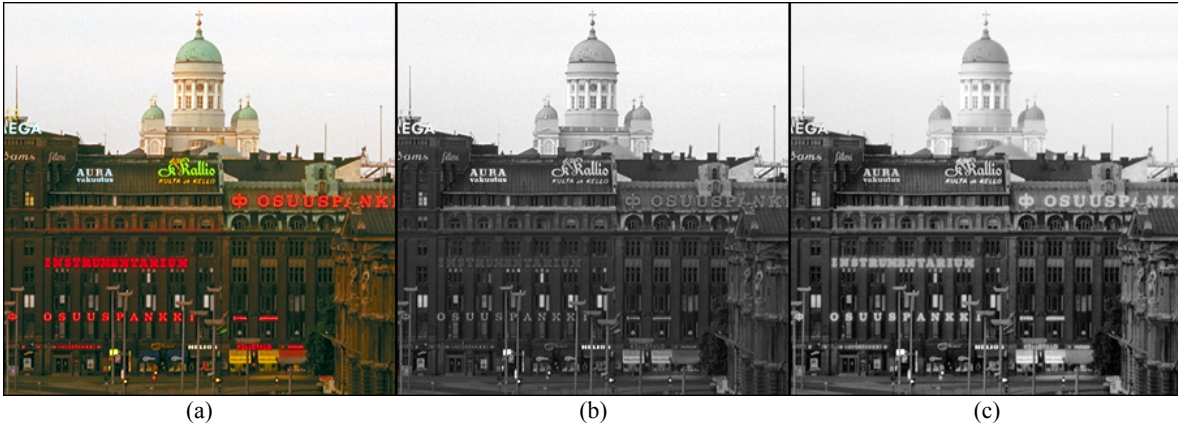


Fig. 1. Recovering chromatic contrasts in grayscale: (a) color image, (b) luminance channel, (c) our enhanced grayscale.

## 2 Previous work

In 2005, there were two significant proposals [1, 2] for treating color to grayscale conversion as an optimization problem involving either local contrasts between pixels [1] or global contrasts between colors [2]. However, unconstrained optimization over thousands of variables [1] or constrained optimization over hundreds of variables [2] is a slow process prone to local minima. It can prove impractical in applications that demand real-time, automated performance with mathematical guarantees on the results. Our algorithm satisfies these requirements without resorting to numerical optimization. Hence, our algorithm could be implemented in hardware or embedded in the driver software that renders color images on a grayscale printer. Our running time is linear in the number of pixels, unlike the Gooch et al. algorithm [1], and is independent of the number of colors in the image, unlike the Rasche et al. algorithm [2]. Owing to their complexity, the optimization algorithms [1,2] offer few guarantees about their results. For instance, they can fail to map pure white in the color image to pure white in the grayscale image, causing potential problems for pictures with white backgrounds. While our algorithm does not purport to model human visual adaptation, it ensures that converting color to grayscale leaves achromatic pixels unchanged.

The closest precedent for our approach to global color to grayscale conversion is the classic work of Strickland et al. [3] on saturation feedback in unsharp masking, a local color image enhancement technique used to sharpen images. However, our method is designed to take advantage of both hue and saturation differences when augmenting the luminance channel.

## 3 Design objectives

In designing our decolorize algorithm, we aimed for simplicity, speed, and scalability, without overly sacrificing image quality. We set the following visual goals:

- **Contrast Magnitude:** The magnitude of the grayscale contrasts should visibly reflect the magnitude of the color contrasts.
- **Contrast Polarity:** The positive or negative polarity of gray level change in the grayscale contrasts should visibly correspond to the polarity of luminance change in the color contrasts.
- **Dynamic Range:** The dynamic range of the gray levels in the grayscale image should visibly accord with the dynamic range of luminance values in the color image.

Table 1. Comparing the mathematical properties of contrast enhancing color to grayscale conversion algorithms.

Property	Gooch et al. [1]	Rasche et al. [2]	Strickland et al. [3]	Our method
Continuous mapping	–	–	+	+
Global consistency	–	+	–	+
Grayscale preservation	–	–	–	+
Luminance ordering	–	±	–	+
Saturation ordering	–	–	–	+
Hue ordering	–	–	–	+
Real-time performance	–	–	+	+

Legend: + satisfied; – not satisfied; ± approximated.

The design also satisfies the following logical constraints (Table 1):

- **Continuous mapping:** The transformation from color to grayscale is a continuous function. This constraint reduces image artifacts, such as false contours in homogeneous image regions.
- **Global consistency:** When two pixels have the same color in the color image, they will have the same gray level in the grayscale image. This constraint assists in image interpretation by allowing the ordering of gray levels to induce a global ordering relation on image colors.
- **Grayscale preservation:** When a pixel in the color image is gray, it will have the same gray level in the grayscale image. This constraint assists in image interpretation by enforcing the usual relationship between gray level and luminance value.
- **Luminance ordering:** When a sequence of pixels of increasing luminance in the color image share the same hue and saturation, they will have increasing gray levels in the grayscale image. This constraint reduces image artifacts, such as local reversals of image polarity.

As a result of its design, our algorithm has the following additional properties, further reducing image artifacts when rendering smooth color gradients:

- **Saturation ordering:** When a sequence of pixels having the same luminance and hue in the color image has a monotonic sequence of saturation values, its sequence of gray levels in the grayscale image will be a concatenation of at most two monotonic sequences.
- **Hue ordering:** When a sequence of pixels having the same luminance and saturation in the color image has a monotonic sequence of hue angles that lie on the same half of the color circle, its sequence of gray levels in the grayscale image will be a concatenation of at most two monotonic sequences.

## 4 Algorithm

We take an RGB color image as input  $(R_i, G_i, B_i) \in [0, 1]^3$  and produce a grayscale image as output  $T_i \in [0, 1]$ . Avoiding gamma correction issues, we assume linear color and grayscale values. The process is controlled by three parameters: the degree of image enhancement,  $\lambda$ ; the typical size of relevant image features in pixels,  $\sigma$ ; and the proportion of image pixels assumed to be outliers,  $\eta$ . Our examples all use  $\lambda = 0.5$ ,  $\sigma = 25$ , and  $\eta = 0.001$  on  $300 \times 300$  images. While  $\lambda = 0.5$  is quite high, it illustrates how well our method can improve contrast; in practice, a more subtle  $\lambda = 0.3$  may be preferable.

## 4.1 Color representation by YPQ color space

The RGB color values  $(R_i, G_i, B_i)$  of each pixel are converted to color values  $(Y_i, P_i, Q_i)$  in our linear YPQ color space:

$$\begin{bmatrix} Y_i \\ P_i \\ Q_i \end{bmatrix} = \begin{bmatrix} 0.2989 & 0.5870 & 0.1140 \\ 0.5000 & 0.5000 & -1.0000 \\ 1.0000 & -1.0000 & 0.0000 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix}. \quad (1)$$

Similar to the IHS color space, the YPQ color channels are analogous to the color channels in human visual processing. They consist of an achromatic luminance channel  $Y_i \in [Y_{\min}, Y_{\max}] = [0, 1]$  and a pair of chromatic opponent-color channels: yellow-blue  $P_i \in [-P_{\max}, P_{\max}] = [-1, 1]$  and red-green  $Q_i \in [-Q_{\max}, Q_{\max}] = [-1, 1]$ . The perpendicular chromatic axes support easy calculation of hue  $H_i \in [-H_{\max}, H_{\max}] = [-1, 1]$  and saturation  $S_i \in [S_{\min}, S_{\max}] = [0, 1.1180]$ :

$$H_i = \frac{1}{\pi} \tan^{-1}\left(\frac{Q_i}{P_i}\right) \text{ and } S_i = \sqrt{P_i^2 + Q_i^2}. \quad (2)$$

The luminance channel  $Y_i$  supplies the default color to grayscale mapping. It conforms to the NTSC-Rec.601 standard luminance axis, which has a length of  $Y_{\text{axis}} = 0.6686$ . The YPQ color components have approximately the same absolute scale,  $S_{\max} \approx H_{\max} = Y_{\max} = P_{\max} = Q_{\max} = 1$ , and therefore they may be readily combined in color calculations. The other benefit of the YPQ color space is its computational simplicity.

## 4.2 Image sampling by Gaussian pairing

To analyze the distribution of color contrasts between image features, we need to consider color differences between pixels. Each pixel color could be compared to the average color of its immediate neighborhood [3], to all pixels in its surrounding region [1], or to all quantized colors present in the image [2]. The former method risks blurring color contrasts; the latter methods are too computationally expensive. Nearby pixels tend to have highly similar colors representing the same image feature while faraway pixels tend to have unrelated dissimilar colors representing disparate image features. A fixed sized neighborhood is unable to capture contrasts belonging to image features of varying sizes.

We address these shortcomings using a novel randomized scheme, sampling by Gaussian pairing. Each image pixel  $X_i$  is paired with a pixel  $X'_i$  chosen randomly according to a displacement vector from an isotropic bivariate Gaussian distribution. The horizontal and vertical components of the displacement are each drawn from a univariate Gaussian distribution with mean zero and variance  $\frac{2}{\pi}\sigma^2$ . Hence, the expected distance between paired pixels is  $\sigma$ . In this way, we can account for image resolution by specifying the expected spatial scale of image features relevant for contrast enhancement. The displacement is carried out under symmetric boundary conditions. Each pixel contributes to at least one color contrast. For better performance, a subset of pixel pairings can be analyzed. For better accuracy, multiple pairings for each pixel can be considered. Repeatable results can be guaranteed by initializing the random number generator with a fixed seed.

### 4.3 Dimensionality reduction by predominant component analysis

To find the color axis that best represents the chromatic contrasts lost when the luminance channel supplies the color to grayscale mapping, we introduce a new dimensionality reduction strategy, predominant component analysis. We start by calculating the color differences:

$$\Delta R_i = R_i - R'_i, \Delta G_i = G_i - G'_i, \Delta B_i = B_i - B'_i, \quad (3)$$

$$\Delta Y_i = Y_i - Y'_i, \Delta P_i = P_i - P'_i, \Delta Q_i = Q_i - Q'_i, \quad (4)$$

$$\text{and } \Delta D_i = \sqrt{\Delta R_i^2 + \Delta G_i^2 + \Delta B_i^2}. \quad (5)$$

To ascertain the relative importance of enhancing color contrasts, the contrast loss ratio  $c_i \in [0, 1]$  measures the relative loss of contrast incurred when luminance differences  $\Delta Y_i$  are used to represent the RGB color differences  $\Delta D_i$  (when  $\Delta D_i = 0$ , we take  $c_i = 0$ ):

$$c_i = \frac{\Delta D_i - \frac{1}{Y_{\text{axis}}} |\Delta Y_i|}{\Delta D_i}. \quad (6)$$

The ordering of the luminance values decides the orientation  $o_i \in \{-1, 0, 1\}$  of each color contrast:

$$o_i = \text{sign}(\Delta Y_i). \quad (7)$$

In the PQ chrominance plane, we determine the predominant axis  $(\Delta p, \Delta q)$  of chromatic contrast through a weighted sum of the oriented chromatic contrasts  $(\Delta P_i, \Delta Q_i)$  of the paired pixels:

$$\Delta p = \sum o_i c_i \Delta P_i \text{ and } \Delta q = \sum o_i c_i \Delta Q_i. \quad (8)$$

Unlike principal component analysis which optimizes the variability of observations, predominant component analysis optimizes the differences between observations. The predominant chromatic axis aims to capture, with a single chromatic coordinate, the color contrast information that is lost in the luminance channel. For best contrast enhancement effect, this axis should be parallel to the prevailing chromatic contrasts. For the preservation of luminance polarity, its orientation should promote the positive correlation of chromatic contrasts with luminance contrasts. The direction of the predominant chromatic axis is optimal in the sense that it maximizes the covariance between chromatic contrasts and the weighted polarity of the luminance contrasts (see Appendix). We acknowledge that a single chromatic axis is an inherently limited tool, as it cannot be used to distinguish color contrasts that are perpendicular to it. Nevertheless, for most color images, a single well chosen chromatic axis suffices for effective contrast enhancement.

### 4.4 Image fusion of luminance and chrominance

Next, we combine luminance and chrominance information. The predominant chromatic data values  $K_i$  are obtained by projecting the chromatic data  $(P_i, Q_i)$  onto the predominant chromatic axis  $(\Delta p, \Delta q)$ :

$$K_i = P_i \Delta p + Q_i \Delta q. \quad (9)$$

To appropriately scale the dynamic range of the predominant chromatic channel  $C_i$ , our algorithm ignores the extreme values due to the level  $\eta$  of image noise. To detect outliers, a linear time selection algorithm,  $\Psi_\eta$  and  $\Psi_{1-\eta}$ , can be used to calculate the outlying quantiles,  $\eta$  and  $1-\eta$ , of the image data. It is applied to discount the extreme absolute values of the projected chromatic data  $K_i$  when scaling the predominant chromatic channel  $C_i$ :

$$C_i = \frac{K_i}{\Psi_{1-\eta}(|K|)}. \quad (10)$$

The predominant chromatic channel  $C_i$  is combined with the luminance channel  $Y_i$  to produce the desired degree  $\lambda$  of contrast enhancement:

$$U_i = Y_i + \lambda C_i. \quad (11)$$

At this intermediate stage of processing, the enhanced luminance  $U_i$  is an image dependent linear combination of the original colors  $(R_i, G_i, B_i)$ , mapping linear color gradients to linear luminance gradients. Though the luminance and chrominance channels could always be further enhanced by standard grayscale techniques, any nonlinear image fusion operator would have greater difficulty reproducing linear color gradients and thus could be more prone to false edge artifacts.

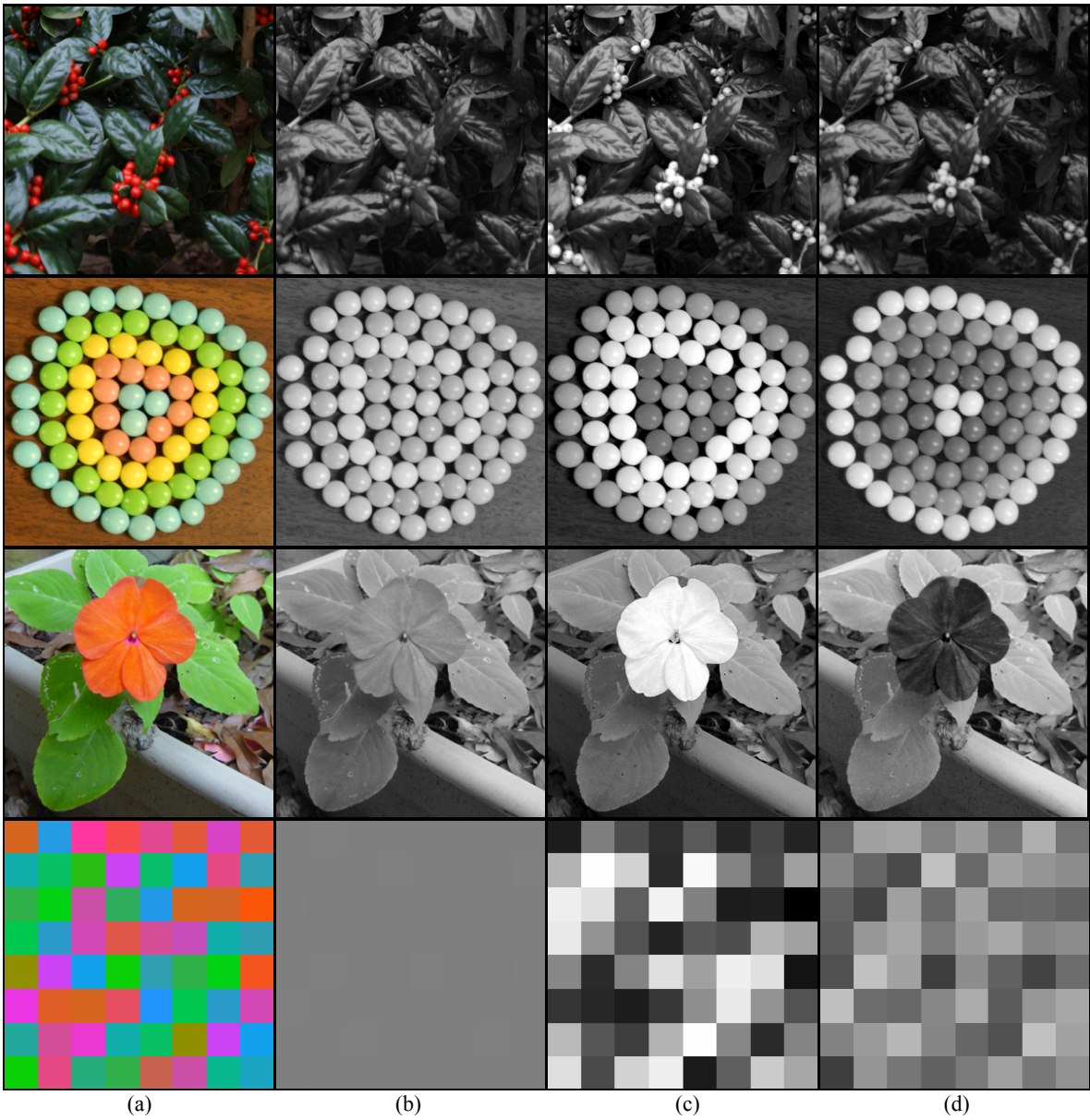


Fig. 2. Color to gray mapping: (a) color image [2], (b) luminance channel, (c) Rasche et al. method [2], (d) our method.

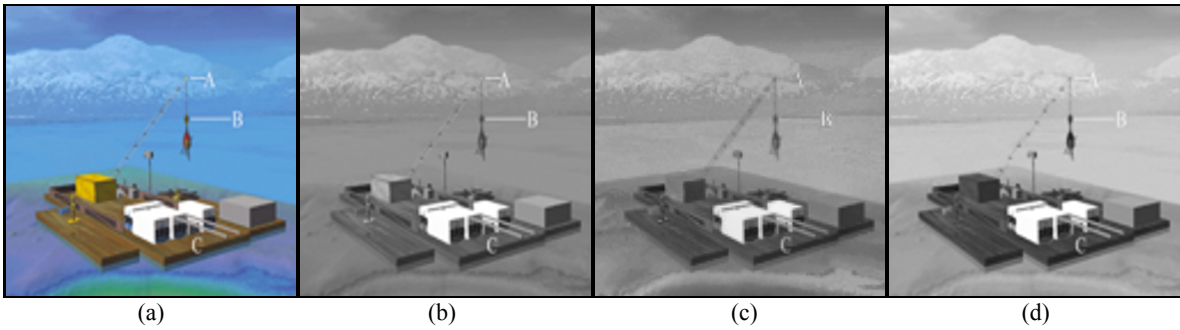


Fig. 3. Color to gray mapping: (a) color image [1], (b) luminance channel, (c) Gooch et al. method [1], (d) our method. Image enhancement (c-d) improves the overall contrast of the diagram at the cost of making the lettered labels less legible.

## 4.5 Saturation dependent dynamic range adjustment

The final step uses saturation to calibrate luminance while adjusting its dynamic range and compensating for image noise. When we evaluate the dynamic range  $[U_{\min}, U_{\max}]$  of the enhanced luminance  $U_i$ , we take care to exclude the effects of image noise:

$$U_{\min} = \Psi_{\eta}(U) \text{ and } U_{\max} = \Psi_{1-\eta}(U). \quad (12)$$

The corrected dynamic range  $[V_{\min}, V_{\max}]$  suitably expands the luminance channel's original dynamic range to better accommodate the desired degree  $\lambda$  of contrast enhancement:

$$V_{\min} = \lambda Y_{\min} + (1-\lambda)\Psi_{\eta}(Y) \text{ and } V_{\max} = \lambda Y_{\max} + (1-\lambda)\Psi_{1-\eta}(Y). \quad (13)$$

The corrected luminance  $V_i$  is obtained by linearly rescaling the enhanced luminance  $U_i$  to fit the corrected dynamic range  $[V_{\min}, V_{\max}]$ :

$$V_i = V_{\min} + \frac{(V_{\max} - V_{\min})}{(U_{\max} - U_{\min})}(U_i - U_{\min}). \quad (14)$$

We use the saturation channel  $S_i$  to derive the bounds  $[E_i, F_i]$  on the distortion permitted in achieving the desired degree  $\lambda$  of contrast enhancement:

$$E_i = \max(Y_{\min}, Y_i - \lambda \frac{Y_{\max}}{S_{\max}} S_i) \text{ and } F_i = \min(Y_{\max}, Y_i + \lambda \frac{Y_{\max}}{S_{\max}} S_i). \quad (15)$$

To be valid, the output gray levels  $T_i$  must respect the allowable dynamic range  $[Y_{\min}, Y_{\max}]$  of luminance values. To ensure that achromatic pixels retain their luminance as their gray level, the discrepancy between luminance and gray level needs to be suitably bounded,  $|T_i - Y_i| \leq \lambda \frac{Y_{\max}}{S_{\max}} S_i$ . The output gray levels  $T_i \in [E_i, F_i] \subseteq [Y_{\min}, Y_{\max}]$  are obtained by clipping the corrected luminance  $V_i$  to conform to the saturation dependent bounds:

$$T_i = \text{median}(E_i, V_i, F_i). \quad (16)$$

The resulting transformation of color and saturation values  $(R_i, G_i, B_i, S_i)$  to gray levels  $T_i$  can be expressed as a continuous, piecewise linear mapping:

$$T_i = r_{j_i} R_i + g_{j_i} G_i + b_{j_i} B_i + s_{j_i} S_i + t_{j_i} \text{ for } j_i \in \{1, 2, 3, 4, 5\}. \quad (17)$$

To produce its output image  $T_i$ , the algorithm linearly combines the luminance channel  $Y_i$  with feedback from either the predominant chromatic channel  $C_i$  or the saturation channel  $S_i$ . Hence, the enhanced contrast originates from either a linear or a polar representation of chromatic data. In both cases, the polarity of the predominant chromatic channel serves to determine whether the feedback is positive or negative. Moreover, the predominant chromatic channel acts as a bridge between positive and negative saturation feedback. The contrast enhancement parameter  $\lambda$  moderates the predominant chromatic channel feedback, the saturation channel feedback, and the dynamic range expansion. If more user control is desired, a different parameter may be assigned to each of these three roles.

## 5 Results

Our decolorize algorithm is effective at enhancing contrast (Figure 2). Of course, our approximation method with our default parameter settings will not match in every case the published results [1,2] of optimization algorithms with parameters tuned individually to suit each image. However, when tested on the previously published image sets [1,2], our algorithm consistently produces results that deliver most of the benefits of the optimization algorithms [1,2] without overly degrading the visual appearance of the images. Our algorithm avoids the noise [1], contouring [2], and halo [3] artifacts that can occasionally afflict the previous approaches. Naturally, some loss of detail may be inevitable when representing color by grayscale. Current contrast enhancement algorithms are still unable to emphasize semantically important features over superfluous details (Figure 3), which is a task for future computer vision techniques.

## 6 Discussion

Our decolorize algorithm is an efficient and effective method for contrast enhancing color to grayscale conversion with attractive mathematical properties. Instead of relying on computationally expensive numerical optimization [1,2], we use novel strategies for image sampling and dimensionality reduction.

As a way of modeling differences between spatial observations, sampling by Gaussian pairing and predominant component analysis may have wider applications, such as visualization of multispectral images. For local image and video processing, they could be applied to analyze local neighborhoods in a spatio-temporally coherent manner. For local image enhancement [1,3], our global method could be applied locally using a partially overlapped sub-block transformation. In addition to image enhancement for visual inspection, our method can help existing implementations of grayscale algorithms to take better account of isoluminant image regions when performing pattern recognition. Cheng and Hsia [4] pursued related goals and devised color algorithms for edge detection, image sharpening, and image compression relying on an approximation to principal component analysis to determine color ordering. The global consistency of our color to grayscale mapping imposes an image dependent color ordering relation that could prove useful in extending mathematical morphology to color image processing, as discussed by Hanbury and Serra [5]. Similarly, it could enable the median filter and other order statistic filters to take advantage of chromatic contrasts. These prospects comprise our research agenda for our future work.

## 7 Appendix

We briefly describe how predominant component analysis optimizes covariance. Consider a set of  $n$  pairs of identically distributed, multivariate observations  $X_k, X'_k \in \mathbb{R}^m$  which are positively or negatively related by weights  $A_k \in \mathbb{R}$ . The predominant component  $Z_k = X_k \cdot z$  is the projection of the data that scales and orients the projected differences  $E_k = D_k \cdot z$  to reflect the observed differences  $D_k = X_k - X'_k$  according to the polarity and magnitude of their associated weights  $A_k$ . The predominant axis  $z$  is a unit vector chosen to maximize the weighted average of the projected differences. As a result, the predominant axis  $z$  is the direction  $d$  of the weighted average of the observed differences. Moreover, the predominant axis  $z$  maximizes the covariance between the projected differences and their associated weights:

$$\Omega = \max_{\|z\|=1} \text{Cov}[A, E] = \max_{\|z\|=1} E[AE] = \max_{\|z\|=1} \frac{1}{n} \sum_{k=1}^n A_k (D_k \cdot z). \quad (18)$$

Observe that  $E[E] = E[D] \cdot z = 0$  and  $\text{Cov}[A, E] = E[AE] - E[A]E[E] = E[AD] \cdot z$ , hence:

$$\Omega = \|d\| \text{ for } z = \frac{d}{\|d\|} \text{ when } d = E[AD] = \frac{1}{n} \sum_{k=1}^n A_k D_k. \quad (19)$$

## References

1. A. A. Gooch, S. C. Olsen, J. Tumblin, B. Gooch, Color2Gray: salience-preserving color removal, *ACM Trans. Graphics* **24** (2005) 634–639 (Proceedings of SIGGRAPH).
2. K. Rasche, R. Geist, J. Westall, Re-coloring images for gamuts of lower dimension, *Computer Graphics Forum* **24** (2005) 423–432 (Proceedings of EUROGRAPHICS).
3. R. N. Strickland, C. S. Kim, W. F. McDonnell, Digital color image enhancement based on the saturation component, *Optical Engineering* **26** (1987) 609–616.
4. S.-C. Cheng, S.-C. Hsia, Fast algorithms for color image processing by principal component analysis, *J. Vis. Comm. & Image Representation* **14** (2003) 184–203.
5. A. G. Hanbury, J. Serra, Morphological operators on the unit circle, *IEEE Trans. Image Proc.* **10** (2001) 1842–1850.