

1-1-1995

Histogram Equalization Of 24-Bit Color Images In The Color-Difference (C-Y) Color Space

Arthur R. Weeks
University of Central Florida

G. Eric Hague
University of Central Florida

Harley R. Myler
University of Central Florida

Find similar works at: <https://stars.library.ucf.edu/facultybib1990>
University of Central Florida Libraries <http://library.ucf.edu>

This Article is brought to you for free and open access by the Faculty Bibliography at STARS. It has been accepted for inclusion in Faculty Bibliography 1990s by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

Recommended Citation

Weeks, Arthur R.; Hague, G. Eric; and Myler, Harley R., "Histogram Equalization Of 24-Bit Color Images In The Color-Difference (C-Y) Color Space" (1995). *Faculty Bibliography 1990s*. 1501.
<https://stars.library.ucf.edu/facultybib1990/1501>

Histogram equalization of 24-bit color images in the color difference (C-Y) color space

Arthur R. Weeks

G. Eric Hague

Harley R. Myler

University of Central Florida

Electrical and Computer Engineering Department

P.O. Box 25000

Orlando, Florida 32816

Abstract. We describe a method of enhancing color images by applying histogram equalization to the saturation component in the color difference (C-Y) color space. When histogram equalization is applied to the saturation component of a 24-bit image, the transform often leads to red, green, and blue components that exceed the realizable RGB intensities. The histogram equalization algorithm presented reduces this problem by taking into account the relationship that exists between luminance and saturation and how the luminance value limits the range of possible saturations. This method also retains a more uniform distribution of color saturation once the components are transformed back into the RGB space. This is important for images that contain high-luminance, low-saturation features.

1 Introduction

Histogram equalization is a well-established tool for enhancing gray-scale images.¹ It redistributes pixels across gray levels so as to yield a near-uniform distribution, improving the overall contrast and brightness in the equalized image and therefore enhancing its features. In applications such as feature extraction of geological imagery, color images are used instead of gray-scale images, due to their ability to convey more information. For example, diseased foliage is more easily distinguished using color images. In addition, Strickland, Kim, and McDonnell were able to show the im-

portance of color in the extraction of edge features from an image.² This method enhances the edges that are detectable in the luminance image with information from the saturation image.

Algorithms have been developed to apply gray-scale histogram equalization to color images.^{3,4} This extension from gray scale to color leads to two problems. First, the pixels in a gray-scale image are defined only by gray level, so gray-scale histogram equalization is a 1-D process. The pixels in color images are defined by red, green, and blue values, which implies that color equalization is a 3-D process. Note that if histogram equalization is applied to each of these colors independently, changes in the relative percentage of red, green, and blue for each pixel may occur. This modifies the hue of those pixels, producing color artifacts. Consequently, color equalization must process all three components at once, which makes for a complex histogram and a complex equalization process.

The second problem is that the RGB color space does not correspond well to the human interpretation of color. Faugeras has shown that in order for image processing routines to produce the least amount of detectable color distortion, the processed quantities should be mapped as closely as possible to those that are perceptually important.⁵ He developed a 3-D model for the human visual system (HVS) based on a single achromatic and two chromatic components. His visual perception model corresponds with the hue, saturation, and intensity (HSI) model, the luminance, in-phase, and quadrature phase (YIQ) model, and the color difference (C-Y)

Paper 94-003 received Feb. 17, 1994; revised manuscript received Sep. 9, 1994; accepted for publication Sep. 26, 1994.
1017-9909/95/\$6.00. © 1995 SPIE and IS&T.

color model that are commonly used. The C-Y color model is the same as the YIQ model rotated by 33 deg (Ref. 6).

Currently, the most common method of equalizing color images is to process only the luminance component.^{1,7} Although this does improve contrast, it basically converts the color image to a gray-scale image and ignores the color information. Genuine color equalization must include equalization of the color saturation component.

Trahanias and Venetsanopoulos developed a method of color equalization involving 3-D histogram equalization in the RGB color space.³ Their method computes the original 3-D histogram for the image and “stretches” it into a uniform distribution. This is accomplished by comparing the cumulative probability density for each point in the original histogram with its associated point in a uniform distribution. When the image’s original histogram does not correspond to the uniform histogram, the original RGB values for that pixel are successively and repeatedly adjusted by one in each dimension until the desired cumulative probability density is reached. Because this method processes all three dimensions at once and maintains the basic ratio between red, green, and blue, it does not produce the color artifacts related to the independent equalization of each color. However, this method does not scale well to 24-bit color images that have 16.7 million possible colors. A 640×480 image would require 64 Mbyte (4 bytes/cell) for the 3-D input histogram. Clearly, the memory requirements of this algorithm limit its application. Furthermore, adjustment of the RGB components in this way does not truly equalize the saturation of the pixels within a given hue.

Bockstein introduced a method of color equalization that is based on the saturation and luminance components.⁴ He transformed the RGB color space into an HSI color triangle divided into 96 hue regions, and applied histogram equalization to the saturation component of the pixels within each of those regions. However, the algorithm is incomplete when determining the maximum possible saturation to use for each transformation. Although Bockstein recognized that maximum saturation is a function of hue, he did not address the fact that it is also a function of luminance. The problem is that high- and low-luminance pixels cannot achieve all the possible values of saturation. When this fact is ignored, the resulting transformation produces pixels that contain non-realizable RGB intensities. Bockstein proposed that these points be scaled back to the border of the color space. This alters the uniform distribution created by the equalization process, resulting in an oversaturated image. Unfortunately, his paper does not include any results, although he does describe them as becoming “visually multicolored.”

A method presented by Inoue and Tajima enhances the sharpness, contrast, and saturation of color images using the HVS color space.⁸ In their method, the saturation component is adjusted by a scaling factor that depends on the low spatial frequency component of the saturation and a parameter chosen by subjective experimentation. Again, the main difficulty, as illustrated in Fig. 1(a) of their paper, is that no coupling between the luminance and saturation component is achieved. The algorithm used to correct the saturation component can easily produce colors that are nonrealizable in the RGB color space.

The approach presented here uses the color difference color space in which the achromatic information is repre-

sented by luminance (Y) and the chromatic information is represented by R-Y and B-Y, which can easily be converted to hue and saturation. By transforming from RGB to C-Y, the original 3-D equalization method proposed by Trahanias and Venetsanopoulos can be converted to a 1-D equalization process manipulating only the saturation component, similar to Bockstein’s algorithm. However, as noted before, special attention must be given to the range of saturations attainable for a given hue and luminance. These maximum saturation values are computed and then stored in a table for use by the equalization algorithm.

This paper also introduces a visualization technique that uses the C-Y color space to evaluate the effectiveness of different color equalization methods. Section 2 describes the C-Y color space and the color equalization algorithm presented in this paper. Section 3 presents the results for two sample images, with comparisons to Bockstein’s algorithm.

2 Color Saturation Equalization Algorithm

We chose to implement the color equalization algorithm in the C-Y color space rather than the HSI color space for two reasons. First, the RGB to C-Y transformation and its inverse are simpler to implement than the corresponding RGB to HSI transformations.¹ Second, the C-Y color space is the system used for color television.⁶ This makes a real-time hardware implementation of the algorithm feasible, due to the enormous amount of color video hardware available.

In a 24-bit color image, the red, green, and blue components of each pixel have 256 possible values. The RGB color space is shown by the cube in Fig. 1, with the appropriate colors indicated at each corner. Each of the 16.7 million possible colors lies somewhere within this cube. Gray-scale images use only the points located along the line connecting the origin and the “white” corner. Figure 2(a) shows the C-Y color space, including the definition of hue and saturation. The transformation from RGB to C-Y scales the RGB cube and rotates it so that the “white” corner is located along the origin of the C-Y space, as shown in Fig. 2(b). This transformation is computed as follows:

$$\begin{aligned} Y &= 0.299 * R + 0.587 * G + 0.114 * B, \\ R - Y &= 0.701 * R - 0.587 * G - 0.114 * B, \\ B - Y &= -0.299 * R - 0.587 * G + 0.886 * B, \end{aligned} \quad (1)$$

where R, G, and B are the red, green, and blue components of the RGB image. The inverse transformation to obtain the

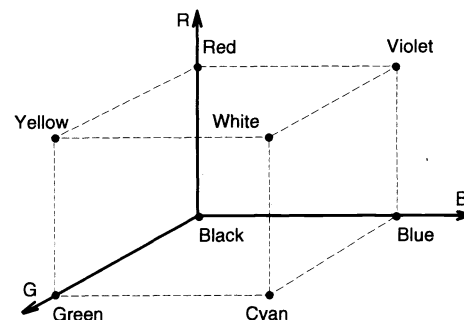


Fig. 1 The RGB color space.

red, green, and blue components adds the Y value back to R-Y, B-Y, and G-Y using Eq. (1) and

$$G-Y = -0.51 \cdot R-Y - 0.19 \cdot B-Y \quad (2)$$

Within the C-Y space, conversion between R-Y/B-Y and hue/saturation is only a conversion between rectangular and polar coordinates, as shown in Fig. 2(a). The equations are

$$\theta = \tan^{-1} \left(\frac{R-Y}{B-Y} \right), \quad (3)$$

$$S = (R-Y^2 + B-Y^2)^{1/2}, \quad (4)$$

where the saturation S is the length of the line from the origin to the color of interest, and hue θ is the angle between the B-Y axis and the saturation line.

The C-Y color space is still 3-D. Figure 2(b) shows how each of the corners of the RGB space maps into the C-Y space. However, the third dimension represents only changes in luminance for constant values of hue and saturation. This normally means that the hue and saturation components have been decoupled from the luminance component. Unfortunately, 24-bit RGB images have a finite range of intensities that creates a dependence between the maximum realizable saturation and the luminance component. Figure 3 shows the C-Y color space sliced into several planes of equal luminance. The planes' strange shape comes from the unequal weighting of R, G, and B in Eq. (1), combined with the cubic shape of the original RGB color space. Normally, each plane of the C-Y space is depicted as circular, where the maximum saturation is the same for all hues.⁶ Because we restrict ourselves to colors that exist within the RGB cube, the maximum obtainable saturation is now a function of hue, as shown in each of the planes in Fig. 3. With or without this limitation, however, the maximum value of saturation must decrease as the luminance increases toward white or decreases toward black. In Fig. 3, the oblique planes become smaller at the luminance extremes, just as the circular planes would become smaller in the normal C-Y space. The net result is that the maximum realizable saturation is a function of both hue and luminance.

The goal of color equalization is to create a uniform distribution of color within the image. This implies that each hue has a uniform distribution of saturation levels. To achieve this, separate saturation equalizations must be performed within each hue. The approach used here is similar to the one presented by Bockstein. The C-Y color space is divided into angular pie-shaped regions. Each of the hue regions is a 3-D space. Figure 4 illustrates one of these hue regions, showing the dependence of the maximum saturation on the luminance component. Bockstein's algorithm performs a single equalization for each hue region and distributes the pixels uniformly across that region without regard to luminance. To avoid a saturation transform that yields colors outside the RGB space, separate equalizations must be performed for the different luminance values within a given hue region.

The equalization process begins by segmenting the C-Y color space into N hue regions and K luminance regions. Next, the values for the maximum realizable saturation $S_{\max}(n,k)$ for $n = 1, 2, 3, \dots, N$ and $k = 1, 2, 3, \dots, K$ are computed by searching all the possible RGB combinations and selecting

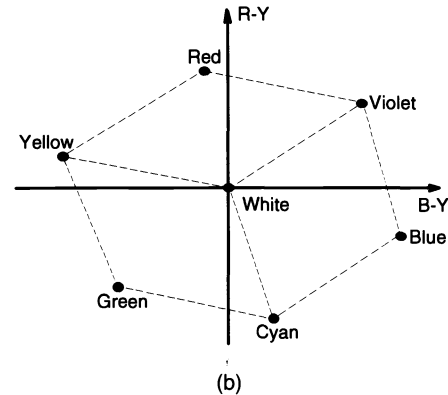
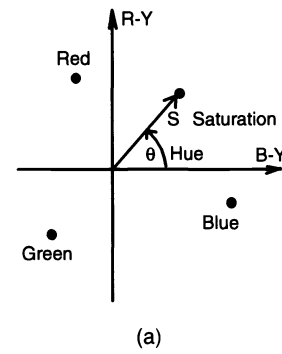


Fig. 2 (a) The C-Y color space and (b) the corresponding RGB cube colors in the C-Y color space.

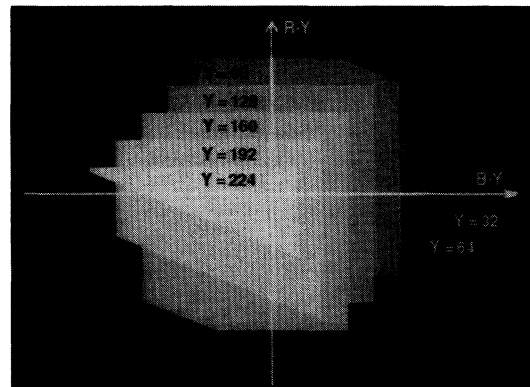


Fig. 3 The C-Y color space sliced into planes of constant luminance showing realizable colors.

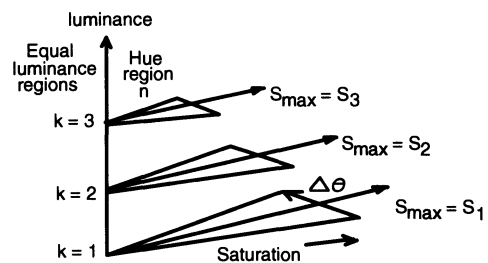


Fig. 4 A diagram of a hue region showing the maximum realizable saturation for different luminance values.

the corresponding maximum saturation value for the hue-luminance combination. This computation is performed once for the desired number of regions, and then the results are stored in a table where they can be used repeatedly by the equalization algorithm. This approach is much faster than using an equation to compute the maximum saturation values every time the equalization process is performed.

The histogram equalization performed on the saturation component in each $n = 1, 2, 3, \dots, N$ hue region and $k = 1, 2, 3, \dots, K$ luminance region yields saturation transformations of

$$S'_{nk}(s) = S_{\max}(n, k) \sum_{i=0}^s \frac{s_{ink}}{s_{ink}}, \quad (5)$$

where s_{ink} is the number of pixels in the i 'th bin of the n 'th hue and k 'th luminance region, and s_{nk} is the total number of pixels in the n 'th hue and the k 'th luminance region. Given an input saturation level s , Eq. (5) gives the new output saturation level $S'_{nk}(s)$ for that region.

Complete color equalization requires an equalization of both the saturation and luminance components. Once the saturation equalization is complete, histogram equalization is performed on the luminance component of the image using standard gray-level histogram equalization. This process equalizes the entire luminance image and results in a single transform

$$Y'(y) = Y_{\max} \sum_{i=0}^y \frac{n_i}{n_i}, \quad (6)$$

where Y_{\max} is the maximum possible luminance, n_i is the number of pixels with luminance i (the i 'th bin), and n_i is the total number of pixels in the image. For a 24-bit color image, Y_{\max} is 255. Equation (6) gives the new luminance values $Y'(y)$ as a function of input luminance y . Even though the realizable luminance values depend on the saturation component, ignoring this dependence in the equalization of the luminance is not a major disadvantage. Some colors will be pushed outside the realizable RGB color space, but they can be brought back by normalizing each component using the following:

$$\frac{255}{\max(R, G, B)}. \quad (7)$$

This sets the largest RGB component to the maximum allowable luminance of 255 for a 24-bit color image. Even though this method does change the luminance and saturation of colors outside the realizable RGB color space, it leaves the hue unchanged.

3 Results

The color equalization algorithm was implemented on a MS-DOS compatible 33-MHz 80486 computer running Microsoft Windows 3.1 and using the C programming language. The images were obtained by scanning 5- \times 4-in. color photographs using a 400-dpi, 24-bit color scanner. The resolution of the scanner was adjusted to obtain 640 \times 480 pixel images. The only preprocessing applied to the images was to adjust the white balance to correct for the poor color balance in the original photographs. Color equalization was performed us-

ing 96 hue regions and 64 luminance regions. The 96 hue regions were chosen to correspond to the number of hue regions proposed by Bockstein. Although 96 regions yielded the best results, the use of fewer hue regions only slightly degraded the resulting images. Tests were conducted using as few as 4 hue regions. The 64 luminance regions provided enough regions to adequately include the effect of luminance on the maximum saturation value, while still yielding enough points per region to adequately perform histogram equalization. A 96 \times 64 array was used to store the maximum saturation values used to determine the saturation equalization transforms.

The complete color equalization process affects both saturation and luminance components. For the reader to easily discern the result of the C-Y saturation equalization algorithm presented here, two resulting images are shown. The first illustrates the impact of saturation-only equalization, and the second illustrates the full effect of both saturation and luminance equalization on image brightness and contrast.

Bockstein's algorithm, which also equalizes both saturation and luminance, is independent of the order in which the two components are equalized. This order independence is a result of not including the luminance effects on the maximum saturation value, and therefore not using it when equalizing saturation. In the C-Y algorithm presented here, saturation equalization does use the luminance component. This leads to two methods for equalizing both components. The first equalizes the luminance and then uses the new luminance values when equalizing the saturation component. The second method equalizes the saturation component using the original luminance values, followed by equalization of the luminance component. This difference in ordering affects which luminance value is used when determining the saturation limits for the saturation equalization process. The first method will yield images that have higher contrast and brightness, but lower saturation. This is due to increasing the overall range of luminance values, which then limits the maximum realizable saturation values. The second method tends to produce images that have higher saturation, but lower brightness and contrast. This is because highly saturated colors have a small range of allowed intensities in the RGB space. After testing our algorithm on a number of sample images, the second method tended to produce images that contained more saturation and, hence, looked more visually pleasing.

Two different images were chosen to illustrate the C-Y color equalization algorithm. The first image is shown in Fig. 5(a), which shows the ceiling inside an Austrian church. This image was chosen for its high spatial detail and its abundance of low-saturation colors. A particular feature of importance is the cross in the upper center of the image. Figure 5(b) is an image of the corresponding C-Y color space distribution. This image was obtained by plotting the RGB color at its appropriate B-Y (horizontal) and R-Y (vertical) coordinates. The center of the image represents the origin of the C-Y color space. Whenever more than one point occupies the same B-Y and R-Y coordinates, the highest luminance value is plotted. Figure 5(b) shows that most of the colors in the image lie near the origin, indicating a low-saturation color image. It also shows the distribution of hues present in the image. Although all hues are present, only red and blue occur with any degree of saturation. This can be confirmed from the original image shown in Fig. 5(a).

Figures 5(c) and 5(d) show the results of the C-Y saturation equalization of the image in Fig. 5(a). The resulting image in Fig. 5(c) shows much greater saturation, with the color green now visible in parts of the image. Notice also that the cross in the upper center of the image is still visible. The corresponding C-Y distribution in Fig. 5(d) shows that the algorithm has uniformly distributed the pixels throughout the color space. On the other hand, Figs. 5(e) and 5(f) show the results of using Bockstein's algorithm. Notice that the cross has become completely obscured. Because Bockstein's algorithm does not take into account the luminance limits on saturation, these high-luminance points are equalized outside the allowable color space. When they are mapped back into the RGB color space, they take on the maximum allowable saturation for that hue. This can be seen in the C-Y color space shown in Fig. 5(f). In comparing this C-Y color space image to the C-Y color space image of Fig. 5(d), the distribution of points is no longer uniformly distributed; the points bunch up near the outer surface of the color space. This is easily seen in the grouping of points near the outer edge of the red region. The effect of ignoring the luminance component can also be seen in the blue and green portions of Figs. 5(d) and 5(f). Bockstein's method was unable to equalize colors uniformly within this region. Most likely this is due to Bockstein's method of combining both low- and high-luminance colors in the same equalization process. If these colors dominate, they will bias the equalization process, preventing medium-luminance colors from being properly equalized out to the edge of the color space.

Figure 5(g) is the final enhanced image of saturation equalization followed by luminance equalization using the algorithm presented here, and Fig. 5(h) is the saturation and luminance equalized image using Bockstein's algorithm. The addition of luminance equalization has improved the overall contrast and brightness of both images, although more noise is observable using Bockstein's method. This is a result of ignoring the relationship that exists between the luminance and saturation components in a color image.

The second sample image is shown in Fig. 6(a). It is a photograph of the Swiss Alps taken in the presence of early morning fog, which partially obscures the mountain range in the background. It was chosen to illustrate the ability of the equalization algorithm to enhance geographical features. Color histogram equalization should be able to enhance both the colors present in the mountain regions, as well as the foliage present in the foreground. Figure 6(b) is the corresponding C-Y color space image of Fig. 6(a). Like the first sample image, this image is composed of predominantly low-saturation colors.

Figures 6(c) and 6(d) are the saturated equalized image using the C-Y equalized algorithm and the corresponding C-Y color space image, respectively. In comparing the original and the enhanced C-Y color space image, it is easily observed that the saturation component is now more uniformly distributed. Saturation equalization has enhanced the features on the mountain face as shown in Fig. 6(c), while preserving the white sky. Figure 6(e) was obtained using Bockstein's algorithm on only the saturation component. Since Bockstein's algorithm ignores the saturation limits on high-luminance pixels, his method will overequalize the saturation component of those pixels. This can be seen in the distortion of the white sky area of the image. Figure 6(f) is

Bockstein's corresponding C-Y color space image. Like the previous example, comparing Figs. 6(d) and 6(f), this image shows that Bockstein's method does not result in a set of uniformly distributed saturation values.

Figure 6(g) is the final equalized image obtained by saturation equalization followed by luminance equalization using the C-Y algorithm. Histogram equalization on the luminance component enhanced the details in the trees as well as the clouds in the sky. Figure 6(h) is the luminance and saturation equalized image using Bockstein's algorithm. This image has lost its photographic qualities and appears more like a graphic drawing. This is due to the overequalization of the saturation component by Bockstein's method.

The processing time required to execute the algorithm presented here as well as Bockstein's algorithm was on the order of 5 min. The time varied slightly depending on the number of saturation regions found to contain zero pixels. The memory requirement for our algorithm was on the order of 3 Mbytes for the saturation histogram, and 1.8 Mbytes for the input and output images. This memory requirement is far below the 64 Mbytes required by Trahanias and Venetsanopoulos.³ Experimental results have shown that we can easily reduce the number of regions from 96 to 16 without much deterioration in the resulting images. This reduces the required memory from 4.8 Mbytes to less than 1 Mbyte.

4 Conclusion

Regardless of the color space chosen, equalization of the saturation component must include the limits of finite luminance associated with 24-bit RGB images. Ignoring the relationship between the luminance and the saturation components results in images that contain colors that no longer represent the original scene. A choice has to be made when equalizing both the luminance and saturation components. Because the luminance and the saturation components are no longer independent, the sequence of the equalization determines the type of enhancement. We found that first equalizing the saturation component and then equalizing the luminance component yielded the best images when attempting to obtain the most saturation.

Additional comparisons of the results presented here were made with images that were obtained by equalization of only the luminance component. Histogram equalization of the luminance component alone yielded images that contained much more contrast and brightness but appeared to have a much lower color saturation. From these comparisons, we suggest that equalization of the luminance component alone does not yield reasonable results for color images.

Acknowledgments

This research has been jointly supported through the University of Central Florida's Electrical and Computer Engineering Department and the Center for Research and Education in Optics and Lasers (CREOL) under support by the Strategic Defense Initiative Organization, Innovative Science and Technology Office through the Office of Naval Research under contract N00014-92-C-0123.

References

1. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, pp. 137-180, 227-229, 247-248, Addison-Wesley, Reading, MA (1992).

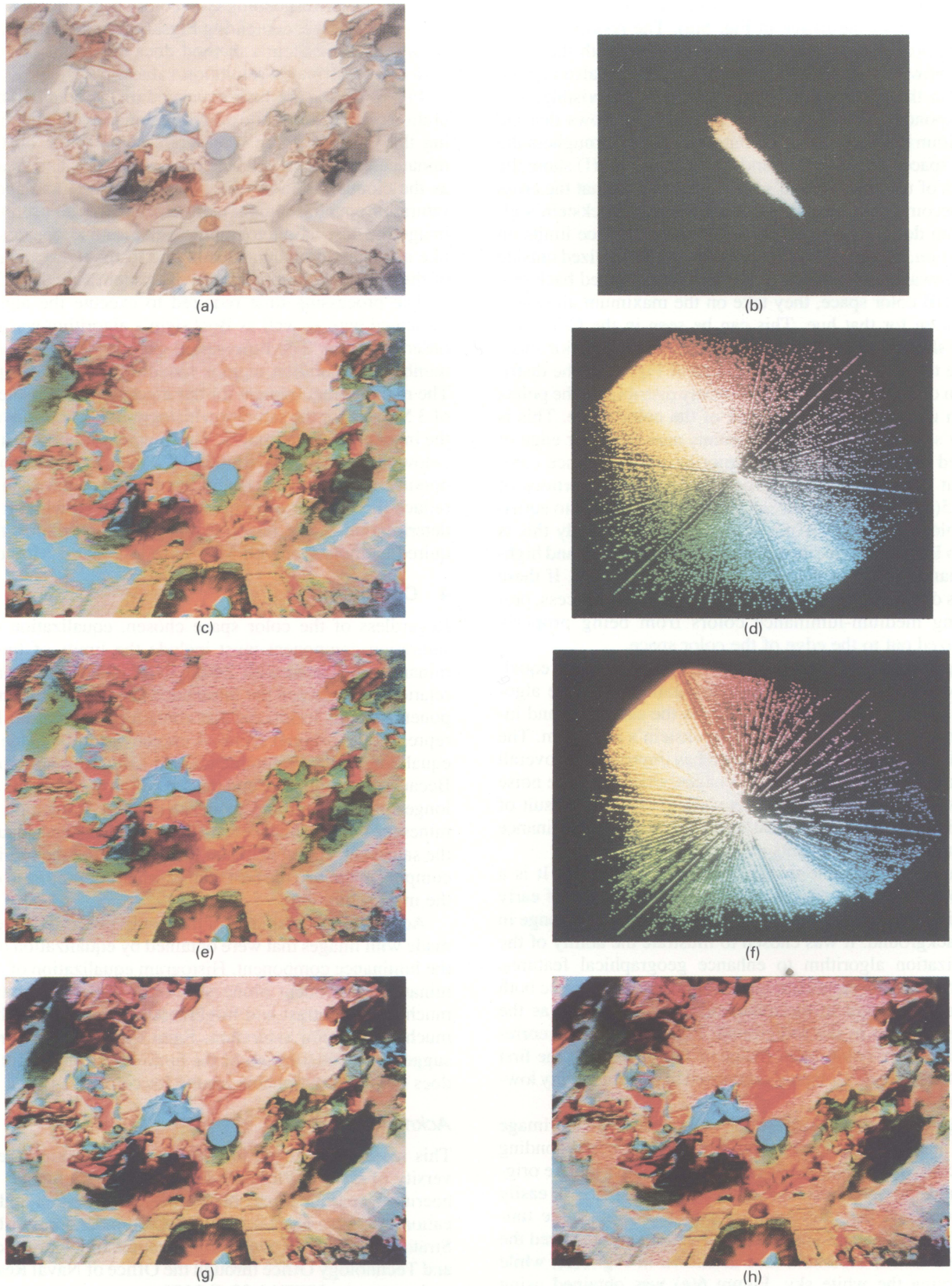


Fig. 5 (a) The original image of the ceiling of an Austrian church; (b) the C-Y color space plot of part (a); (c) equalization of only the saturation component using the C-Y algorithm; (d) the C-Y color space of part (c); (e) equalization of only the saturation component using Bockstein's algorithm; (f) the C-Y color space of part (e); (g) equalization of both the saturation and the luminance components using the C-Y algorithm; and (h) equalization of both the saturation and the luminance components using Bockstein's algorithm.

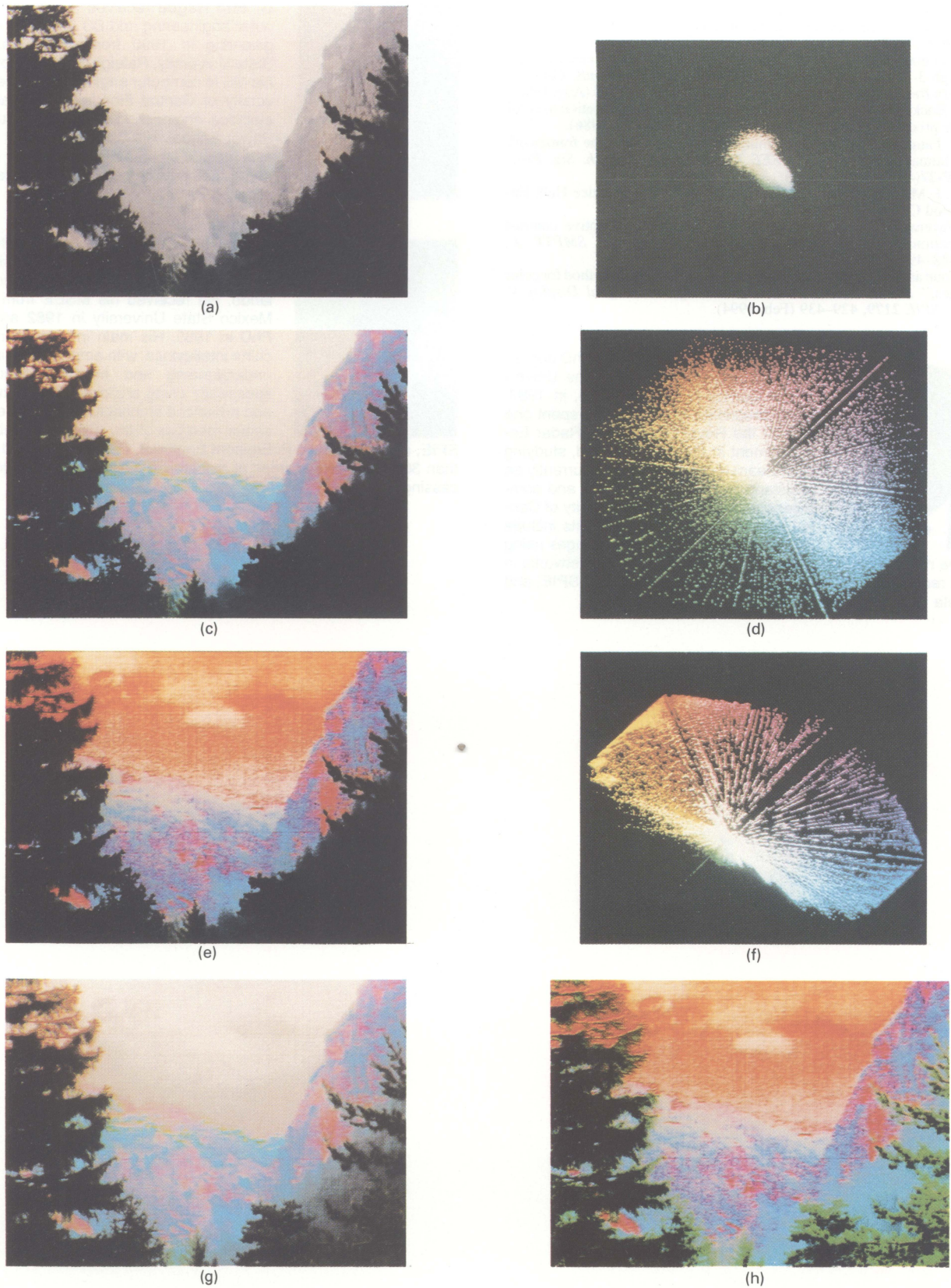
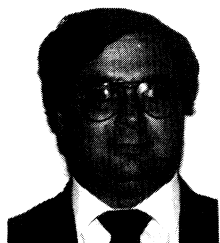


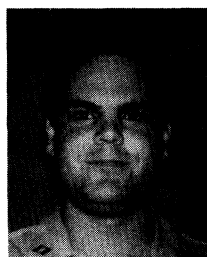
Fig. 6 (a) The original image of a mountain scene; (b) the C-Y color space plot of part (a); (c) equalization of only the saturation component using the C-Y algorithm; (d) the C-Y color space of part (c); (e) equalization of only the saturation component using Bockstein's algorithm; (f) the C-Y color space of part (e); (g) equalization of both the saturation and the luminance components using the C-Y algorithm; and (h) equalization of both the saturation and the luminance components using Bockstein's algorithm.

2. R. N. Strickland, C. S. Kim, and W. F. McDonnel, "Digital color image enhancement based on the saturation component," *Opt. Eng.* **26**(7), 609-616 (1987).
3. P. E. Trahanias and A. N. Venetsanopoulos, "Color image enhancement through 3-D histogram equalization," *Proc. 15th IAPR Intl. Conf. on Pattern Recognition*, The Hague, Netherlands, pp. 545-548 (Aug. 1992).
4. I. M. Bockstein, "Color equalization method and its application to color image processing," *J. Opt. Soc. Am. A* **3**(5), 735-737 (1986).
5. O. D. Faugeras, "Digital color image processing within the framework of a human visual model," *IEEE Trans. Acoust., Speech, Sig. Proc.* **ASSP-27**(4), 380-393 (1979).
6. A. V. J. Martin, *Technical Television*, pp. 498-500, Prentice-Hall, Englewood Cliffs, NJ (1962).
7. M. Faverau, S. Soca, J. Balon, and M. Cattoen, "Adaptive contrast correction using real-time histogram modification," *SMPTE J.*, pp. 488-491 (May 1984).
8. A. Inoue and J. Tajima, "Adaptive quality improvement method for color images," *Human Vision, Visual Processing, and Digital Display V, Proc. SPIE 2179*, 429-439 (Feb. 1994).

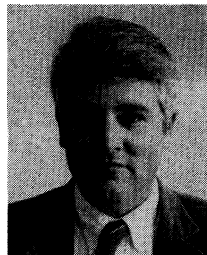


Arthur R. Weeks received his PhD degree in electrical engineering from the University of Central Florida, Orlando, in 1987. After completion of his PhD, he spent one year at the Royal Signals and Radar Establishment in Malvern, England, studying laser beam propagation. He is currently an assistant professor of electrical and computer engineering at the University of Central Florida. Dr. Weeks's interests include the reduction of noise within images using

adaptive nonlinear filters and the use of artificial neural networks in pattern recognition. Dr. Weeks is a member of the IEEE, SPIE, and Tau Beta Pi.



G. Eric Hague received his BS in computer engineering and BS in electrical engineering in 1990 from North Carolina State University, Raleigh. He is completing his MS in computer engineering at the University of Central Florida, while teaching electrical engineering at the Naval Nuclear Power School. His current research interests include color image processing using human visual perception and adaptive nonlinear filters.



Harley R. Myler is an associate professor of electrical and computer engineering at the University of Central Florida in Orlando. He received his MSEE from New Mexico State University in 1982 and the PhD in 1985. His main interest is in machine intelligence, with emphasis on image understanding and machine cognition, specifically using artificial neural networks and advanced architectures. Dr. Myler is a senior member of the IEEE, a member of

SPIE, and a Tau Beta Pi Eminent Engineer. He has published more than 30 papers in the areas of machine intelligence and image processing.