

SPRINGER BRIEFS IN APPLIED SCIENCES AND
TECHNOLOGY · COMPUTATIONAL INTELLIGENCE

Jyotismita Chaki
Nilanjan Dey



Image Color Feature Extraction Techniques Fundamentals and Applications



Springer

SpringerBriefs in Applied Sciences and Technology

Computational Intelligence

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland

SpringerBriefs in Computational Intelligence are a series of slim high-quality publications encompassing the entire spectrum of Computational Intelligence. Featuring compact volumes of 50 to 125 pages (approximately 20,000-45,000 words), Briefs are shorter than a conventional book but longer than a journal article. Thus Briefs serve as timely, concise tools for students, researchers, and professionals.

More information about this subseries at <http://www.springer.com/series/10618>

Jyotismita Chaki · Nilanjan Dey

Image Color Feature Extraction Techniques

Fundamentals and Applications



Springer

Jyotismita Chaki
School of Information Technology
and Engineering
Vellore Institute of Technology
Vellore, India

Nilanjan Dey
Department of Information Technology
Techno India College of Technology
Kolkata, West Bengal, India

ISSN 2191-530X ISSN 2191-5318 (electronic)
SpringerBriefs in Applied Sciences and Technology
ISSN 2625-3704 ISSN 2625-3712 (electronic)
SpringerBriefs in Computational Intelligence
ISBN 978-981-15-5760-6 ISBN 978-981-15-5761-3 (eBook)
<https://doi.org/10.1007/978-981-15-5761-3>

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2021
This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721,
Singapore

Preface

Content-Based Image Retrieval (CBIR) is developed as a promising technique for retrieving images and browsing bulky image databases. Currently, CBIR is among the fast-rising topic. It is the main research area of Computer Science and Engineering disciplines too. It is the procedure of retrieving images from a gathering based on automatically extracted features. CBIR methods support in handling of the digital images by means of computers.

The common feature extraction techniques used in CBIR may be divided into the subsequent three groups: 1. shape feature extraction, 2. color feature extraction, 3. texture feature extraction. Color feature or color descriptor puts effort to compute image color in various ways that agree with human perception (or task-specific necessities). Good image retrieval accuracy needs an efficient color descriptor to be able to efficiently discover perceptually comparable image colors from a database.

Two main variables motivate the use of color in image processing. First, color is a strong descriptor often simplifying the recognition and retrieval of objects from a scene. Second, people can distinguish thousands of shades of color and intensity compared to just about two dozen shades of gray. Color is one of the most significant characteristics that enable us to recognize images. This book emphasizes on various image color feature extraction methods which are necessary for image recognition and classification. Color feature helps to determine if a specified color matches a model satisfactorily, or which of the representative class is the utmost comparable. Color features are also required to translate or transform one color model to another so that it best matches other image colors, completely or in part. Different approaches for extracting image color features are conferred in this book that will affect the image retrieval from a large database on how the procedures are employed.

The following areas that are covered are the strengths of this book: (1) details of the need of image color features in CBIR; (2) details of different properties of image color features; (3) details of different color spaces and conversion from one color space to another; (4) details of histogram-based color features; (5) details of MPEG-7 color features; (6) details of other color feature extraction techniques like

color coherence, color moments, and color correlogram; (7) different applications of image color feature in the field of content-based image retrieval; etc.

Each chapter is followed by very helpful summary or conclusion parts and a significant amount of references to the primary sources of data, many of which are related to latest literature in the field. The purpose of this book is not only to help beginners with a holistic approach toward understanding color feature extraction techniques but also to present to researchers new technological trends and design challenges they have to cope with, while designing image retrieval systems.

The book is organized as follows:

Chapter 1 gives the overview of image color feature. The need for image color in the field of image retrieval is discussed in this chapter. Color image processing includes pseudocolor and full-color or true-color processing. The purpose of pseudocolor processing is to color a grayscale image by assigning different colors in different intensity ranges of a gray-level image. In the full-color image, the actual color of the image is considered. In such type of images, the colors can be specified by using different color models like RGB (linear and non-linear), HSI, HSV, CMY, CMYK, CIE L*a*b*, YUV, YIQ, Munsell, HMMD, Opponent, etc. Algorithms for the conversion from one color space to another are also mentioned in this chapter. Different color quantization techniques such as scalar or uniform, vector quantization, octree, etc. are discussed in this chapter. The examples related to different statistical texture feature extraction techniques are illustrated through MATLAB examples.

Chapter 2 presents histogram-based image color features like histogram intersection, fuzzy histogram, and different distance measures that can be used to check the similarity between color image histograms. Several histogram distance measures like Histogram Minkowski distance, Histogram Euclidean Distance, Histogram Intersection Distance, Histogram Quadratic (Cross) Distance, Histogram Manhattan Distance, Histogram Chebyshev Distance, Histogram Cosine Distance, Histogram Canberra Distance, Histogram Kolmogorov-Smirnov Divergence Distance, Histogram Cramer-von Mises Distance, Histogram Chi-square Distance, Histogram Squared Chord Distance, Histogram Kullback-Leibler Divergence Distance, and Histogram Jeffrey Divergence Distance are discussed in this chapter. Also, several advantages and limitations of image color histogram are discussed in this chapter. The examples related to different statistical texture feature extraction techniques are illustrated through MATLAB examples.

In Chap. 3, some MPEG-7 color feature extraction techniques are discussed such as dominant color descriptor, scalable color descriptor, group of images/group of frames color descriptor, color layout descriptor, and color structure descriptor. Several advantages and limitations of MPEG-7 color feature are discussed in this chapter. The examples related to different statistical texture feature extraction techniques are illustrated through MATLAB examples.

Chapter 4 discusses different image color features like color coherence vector, color moments, color co-occurrence matrix, color contrast occurrence matrix, color correlogram, and reference color table method. Several advantages and limitations of mentioned color feature are discussed in this chapter. The examples related to different statistical texture feature extraction techniques are illustrated through MATLAB examples.

Finally, Chap. 5 provides an overview of various applications of color features in image recognition in the area of leaf recognition, fruit recognition, flower recognition, random image, etc. Also, the way to detect edge using color features is included in this chapter. The examples related to the application texture feature extraction techniques of color images are illustrated through MATLAB examples.

Vellore, India
Kolkata, India

Dr. Jyotismita Chaki
Dr. Nilanjan Dey

Contents

1	Introduction to Image Color Feature	1
1.1	Color Spaces or Color Models	3
1.1.1	The RGB Color Space	5
1.1.2	The CMY and CMYK Color Space	7
1.1.3	The HSV and HSL Color Space	9
1.1.4	The YIQ and YUV Color Space	12
1.1.5	The CIE L*a*b* Color Space	12
1.1.6	The Munsell Color Space	13
1.1.7	The Opponent Color Space	16
1.1.8	Hue-Max-Min-Difference Color Space	17
1.1.9	New Trends	18
1.2	Color Quantization	20
1.2.1	Scalar or Uniform Color Quantization	21
1.2.2	Vector Color Quantization	21
1.2.3	Octree and Other Color Quantization Algorithms	22
1.3	Pseudocolor Image Processing	23
1.4	Full-Color Image Processing	25
1.5	Summary	26
	References	27
2	Histogram-Based Image Color Features	29
2.1	Color Histogram	29
2.2	Histogram Intersection	30
2.3	Color Histogram Discrimination	30
2.3.1	Histogram Minkowski Distance	31
2.3.2	Histogram Manhattan Distance	32
2.3.3	Histogram Euclidean Distance	32
2.3.4	Histogram Chebyshev Distance	33
2.3.5	Histogram Intersection Distance	33
2.3.6	Histogram Quadratic (Cross) Distance	34

2.3.7	Histogram Cosine Distance	34
2.3.8	Histogram Canberra Distance	35
2.3.9	Histogram Kolmogorov-Smirnov Divergence Distance	35
2.3.10	Histogram Cramer-Von Mises Distance	35
2.3.11	Histogram Chi-Square Distance	36
2.3.12	Histogram Squared Chord Distance	36
2.3.13	Histogram Kullback-Leibler Divergence Distance	36
2.3.14	Histogram Jeffrey Divergence Distance	37
2.4	Fuzzy Color Histogram	37
2.5	Summary	39
	References	40
3	MPEG-7 Image Color Features	43
3.1	Dominant Color Descriptor	43
3.2	Scalable Color Descriptor	48
3.3	Group of Image/Group of Frames Color Descriptor	49
3.4	Color Layout Descriptor	50
3.5	Color Structure Descriptor	53
3.6	Summary	54
	References	55
4	Other Image Color Features	57
4.1	Color Coherence Vector	57
4.2	Color Moments	60
4.3	Color Co-occurrence Matrix	62
4.4	Color Contrast Occurrence Matrix	67
4.5	Color Correlogram	67
4.6	Reference Color Table Method	69
4.7	Summary	70
	References	71
5	Applications of Image Color Features	73
5.1	Leaf Recognition	73
5.2	Fruit Recognition	75
5.3	Flower Recognition	75
5.4	Random Image Recognition	79
5.5	Edge Detection of Color Image	80
5.6	Summary	82
	References	82

About the Authors



Dr. Jyotismita Chaki is an Assistant Professor in School of Information Technology and Engineering at Vellore Institute of Technology, Vellore, India. She has done her PhD (Engg) from Jadavpur University, Kolkata, India. Her research interests include: Computer Vision and Image Processing, Artificial Intelligence, Pattern Recognition, Medical Imaging, Soft computing, Data mining, Machine learning. She has authored many international conferences and journal papers. She is the author of the following books titled: (1) A Beginner's Guide to Image Preprocessing Techniques (CRC Press, Taylor and Francis), (2) A Beginner's Guide to Image Shape Feature Extraction Techniques (CRC Press, Taylor and Francis) and (3) Texture Feature Extraction Techniques for Image Recognition (Springer). She has served as a reviewer of Applied Soft Computing (Elsevier), Biosystem Engineering (Elsevier), Pattern Recognition Letters (Elsevier), Journal of Visual Communication and Image Representation (Elsevier), Signal Image and Video Processing (Springer) journal and also served as Program Committee member of many reputed conferences.



Dr. Nilanjan Dey is an Assistant Professor in Department of Information Technology at Techno International New Town (Formerly known as Techno India College of Technology), Kolkata, India. He is a visiting fellow of the University of Reading, UK. He is a Visiting Professor at Duy Tan University, Vietnam. He was an honorary Visiting Scientist at Global Biomedical Technologies Inc., CA, USA (2012-2015). He was awarded his PhD. from Jadavpur University in 2015.

He is the Editor-in-Chief of International Journal of Ambient Computing and Intelligence, IGI Global. He is the Series Co-Editor of Springer Tracts in Nature-Inspired Computing, Springer Nature, Series Co-Editor of Advances in Ubiquitous Sensing Applications for Healthcare, Elsevier, Series Editor of Computational Intelligence in Engineering Problem Solving and Intelligent Signal processing and data analysis, CRC. He has authored/edited more than 75 books with Springer, Elsevier, Wiley, CRC Press and published more than 300 peer-reviewed research papers.

His main research interests include Medical Imaging, Machine learning, Computer Aided Diagnosis, Data Mining etc. He is the Indian Ambassador of International Federation for Information Processing (IFIP) – Young ICT Group.

Chapter 1

Introduction to Image Color Feature



Two main factors motivate the need for color in image processing. First, color is a strong descriptor frequently simplifying the recognition and extraction of objects from a picture. Second, people can distinguish thousands of tones of color and intensity comparable to just about two dozen tones of gray [1–4]. Color is one of the most significant characteristics that enable people to recognize images. Color is a feature that relies on the reflection and processing of light to the eye in the brain. Color is utilized on a daily basis to say the distinction between objects, locations, and daytime. Comparing the image color content is obvious and therefore this is one of the popular choices to perform image recovery tasks. Color function is a solid descriptor which can often simplify the recognition and extraction of objects from a particular image.

The exact method through which the human brain controls and understands color sight is a complicated physiopsychological process. Nevertheless, the fundamental attributes of light itself can easily be expressed on the basis of experimental and theoretical findings. In 1666, Sir Isaac Newton noted that it spreads evenly into an incessant color band from violet to red when white light is transmitted via a prism. He also observed that no color ends sharply within this spectrum, but the transition from one color to the next is always smooth. Figure 1.1 demonstrates this process. Each color in the spectrum is a particular wavelength corresponding to a distinctive color. It is possible to mix different combinations of these wavelengths to create other colors too.

From the light reflected from the images, the colors experienced by the human visual system and some other creatures are defined [5, 6]. As shown in Fig. 1.2, only a tiny part of the entire electromagnetic spectrum is composed of visible light. Items that reflect all wavelengths of the light spectrum will appear white, while objects that favor the reflection of some wavelengths will appear to the observer as some color shade. For example, artifacts in the range of 420–470 nm that reflect wavelengths while consuming much of the others may appear as a blue color.

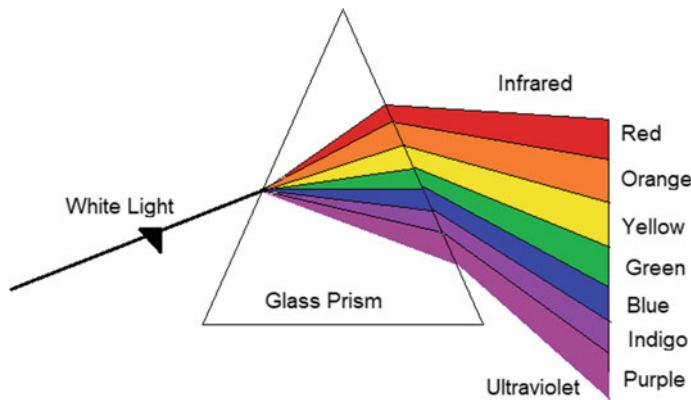


Fig. 1.1 The color spectrum obtained by transmitting white light via a prism

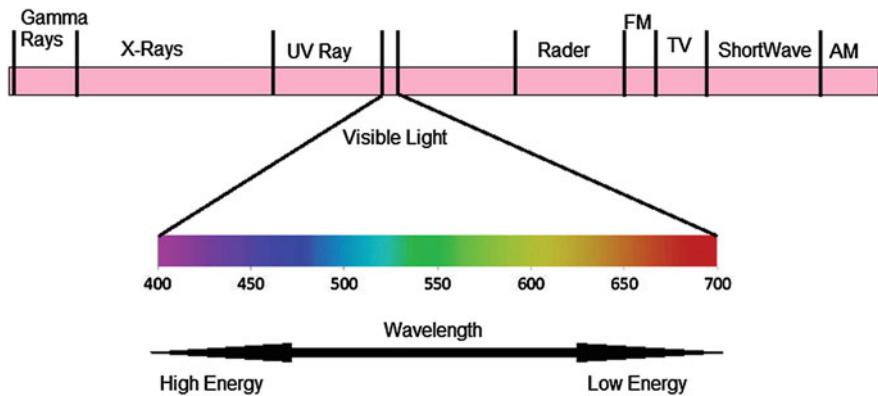


Fig. 1.2 Wavelength making up visible light or spectrum

Special sensor cell structures, widely known as cones, are in charge of color perception in human vision within the eye [7, 8]. Researchers have assessed, based on observational evidence, that six to seven million cones residing in the human eye can be segmented into three different groups of sensing approximately corresponding to the colors blue, green, and red. About only 2% of these cones belong to the blue group, 33% to the green group, and 65% to the red group. It is observed that these cluster ratios do not reflect the specificity of color directly; in fact, the blue cones are the most delicate, compensating for their deficiency of presence.

Figure 1.3 demonstrates the absorption of light by the red, green, and blue, cones as a meaning of the eye's wavelength and also the highest wavelengths for which they are most delicate on a standardized range. Colors are regarded as varying proportions of these primary colors because of these characteristics of the cone cells of the eye. Moreover, due to the continued existence of the visual spectrum and the varying cone

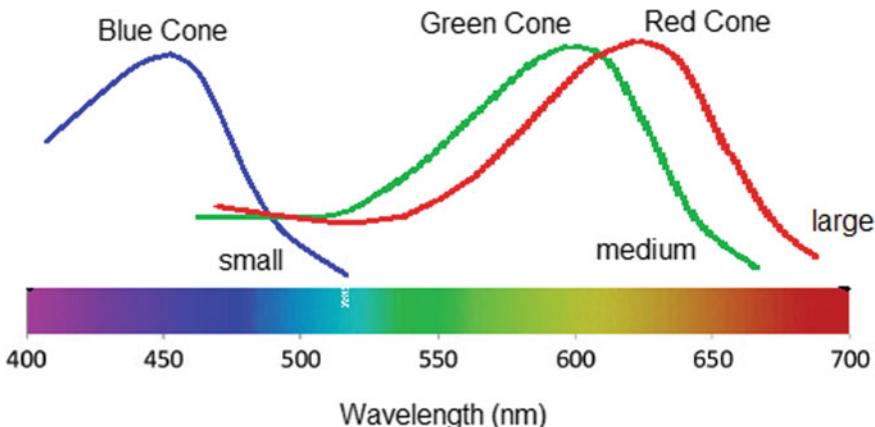


Fig. 1.3 Absorption of light by the human eye cones

sensitivity, no particular color may be classified as red, green, or blue. Mixtures of such primaries dependent on defined wavelengths cannot be utilized to generate all spectrum wavelengths; the ranges of blue, red, and green should be allowed to vary in order to generate all colors.

A color model is a mathematical model that in certain regular and approved ways facilitates color definition. The model, also recognized as color system or color space, designates a subspace and a coordinate system within that system of coordinates where every point conforms specifically to a single color [9]. Nowadays, most color models are designed to utilize either in computer hardware (monitor, printer, etc.) or in applications where the use of color is the main goal (i.e., graphic design). Presently there are countless color models in realistic usage merely because color science is a vast domain covering many applications.

1.1 Color Spaces or Color Models

A suitable technique for depicting the color signal is required to use color as a visual stimulus in image processing, multimedia, statistics, and machine vision applications. This need is addressed by the various color specification systems or color models (color spaces or solids). Color spaces offer a good rational technique for specifying, ordering, manipulating, and displaying the colors of the object taken into consideration [10]. A well-selected depiction retains vital data and gives insight into the required visual operation. The color model chosen should, therefore, be well adapted to tackle the declaration and answer to the problem. Selecting the finest image representation method includes understanding how to generate color patterns and what information these signals require. Despite the fact that color spaces impose restrictions on color perception and depiction, they also assist people to fulfill significant

tasks. In specific, color models can be utilized to specify colors, distinguish colors, assess color similarity, and identify image categories for a variety of applications. Color model literature can be discovered in the modern sciences such as engineering, physics, computer science, artificial intelligence, sociology, and philosophy.

In the literature, four fundamental color model families can be defined as follows:

1. **Colorimetric** color models based on spectral reactance physical measurements [11]. For such models, three primary colors are identified by their coordinates, like the CIE chromaticity diagram. To obtain other colors, the primary colors are mixed subtractively or additively.
2. **Psychophysical** color based on human color perception [12]. Such color spaces are either focused on subjective observation standards and relative references (e.g., Munsell color space) or are constructed through studies to meet human color perception (e.g., hue, saturation, and lightness model).
3. **Physiological** color representations are focused on the three main cone types as in the human retina [13]. The color space utilized in computer hardware by Red-Green-Blue (RGB) is a renowned example of a color model that is physiologically inspired.
4. **Opponent** color spaces or models relying on perception experimentations, primarily using main opponent colors, like the color pairs Yellow-Blue and Red-Green [14].

Alternatively, color models can be split into three classifications in applications for image processing, such as

1. **Device-oriented** color models that are correlated with devices including RGB, CMY, and YIQ input; processing; and output [15]. In modern applications, such spaces are of primary significance where color needs to be specified in a manner that is consistent with the hardware devices utilized to provide, manipulate, or obtain color image data.
2. **User-oriented** color models that are used as a link between hardware and human operators for manipulating color information such as HCV, HSL, HSB, HSV, CIE-LAB, MTM, and CIE-LUV [16]. These models enable the individual to define color through perceptual characteristics and can be regarded as a new imitation of human color perception.
3. **Device-independent** color models that were utilized to indicate color signals regardless of device or application characteristics [17]. These models are important in applications where color similarities are needed and perceptual information is transmitted through networks connecting various hardware platforms.

As frequently utilized, image formats like GIFs, BMPs, and JPEGs always store and display images in the RGB color model, an RGB color space-based image retrieval scheme will not require color space conversion and will, therefore, be facilitated. Though, due to the deficiency of the RGB color space not being perceptually uniform, the RGB color model may be converted into other models in color-based

image retrieval that enhances perceptual consistency. However, there is no proof to demonstrate which color space will always produce the highest results for image recovery.

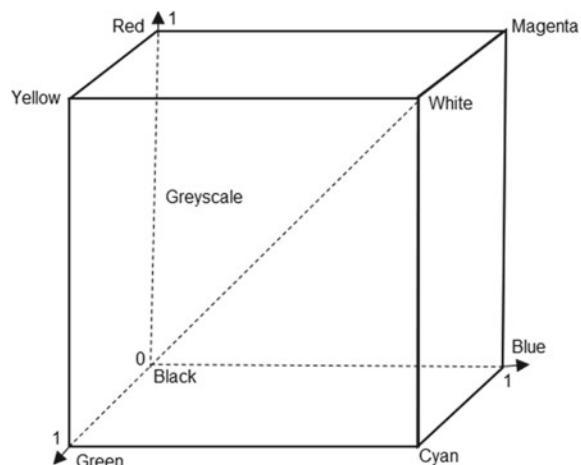
1.1.1 The RGB Color Space

Each color appears in a subspace of the conventional Cartesian coordinate system as a three-dimensional point within the RGB color space. Each axis signifies one of the three color components (red (R), green (G), and blue (B)) that will portray all colors in the model [18]. For simplicity, these values are sometimes standardized so that all values of R, G, and B fall in the range [0,1]. Figure 1.4 visually portrays the RGB subspace. The RGB color images are made up of three-channel images, also named as components, for each prime color. These channels congregate into an additive way to define the color at a specified pixel (by a mixture of red, green, and blue each pixel is colored). This directly translates into how a color monitor displays color, which is why it is sometimes the default color model utilized in most applications. One significant limitation of the RGB color model is that it is not visually standardized, which means that the computed range in the RGB model is not really proportional to the variation in visual color.

Linear RGB Color Model

The linear R component is proportional to the physical power intensity, radiated from an object around the visible spectrum's 700 nm band [19]. Similarly, the 546:1 nm range refers to a normal G component and the 435:8 nm band refers to a linear B component. As a consequence, the linear RGB space is device-independent and utilized to obtain color consistency throughout various devices in several color

Fig. 1.4 Normalized RGB coordinate color cube



management systems. Utilizing the following matrix conversion (Eq. 1.1), the linear RGB values in the range [0, 1] can be transformed into the subsequent CIE XYZ values in the range [0, 1].

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4125 & 0.3576 & 0.1804 \\ 0.2127 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9502 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1.1)$$

The conversion from CIE XYZ to RGB values in the range [0, 1] is demarcated by using Eq. (1.2).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2405 & -1.5372 & -0.4985 \\ -0.9693 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0573 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (1.2)$$

On the other hand, tristimulus XYZ values can be attained from the linear RGB values by using the subsequent matrix represented in Eq. (1.3).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.117 & 0.812 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1.3)$$

The linear RGB values are a physical depiction of a radiated object's chromatic light. Though, the human visual system's perceptual reaction to emit R, G, and B intensities is more complicated and non-linear. The linear RGB space is extremely non-uniform perceptually and not appropriate for mathematical perceptual attribute assessment. Therefore, it is very rare to utilize linear RGB values to depict an image. On the other hand, in image processing applications such as filtering, non-linear, R'G'B' values are generally utilized.

Non-linear RGB Color Model

The video camera is exposed to the linear light emitted from the object when an image is captured. Using gamma correction, the linear RGB intensity incident on the video camera is converted into non-linear RGB measurements. The conversion from linear RGB values to non-linear R'G'B' values in the range [0, 1] is demarcated by using Eq. (1.4).

$$R' = \begin{cases} 4.5R, & \text{if } R \leq 0.018 \\ 1.099 R^{\frac{1}{\gamma}} - 0.099, & \text{otherwise} \end{cases}$$

$$G' = \begin{cases} 4.5G, & \text{if } G \leq 0.018 \\ 1.099 G^{\frac{1}{\gamma}} - 0.099, & \text{otherwise} \end{cases}$$

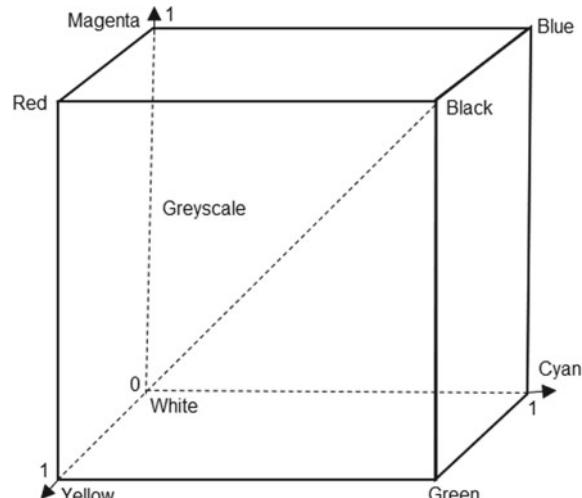
$$B' = \begin{cases} 4.5B, & \text{if } B \leq 0.018 \\ 1.099 B^{\frac{1}{\gamma}} - 0.099, & \text{otherwise} \end{cases} \quad (1.4)$$

where γ is regarded as the camera's or acquisition device's gamma factor. Normally the value of γ is 1/0.45 that is utilized in video cameras. In practical scanners and cameras, the linear section near small intensities reduces the impact of the noise of the sensor. Therefore, the digital image pixel values obtained from the object and placed in a scanner or camera are the values of $R'G'B'$, which are generally transformed to the interval of 0 to 255. The three parts R' , G' , and B' of a color image pixel with one byte for each element are then needed to depict three bytes. It is these non-linear values of $R'G'B'$ that are recorded in computers as image information records and utilized in applications for machine vision application. Generally, the RGB symbol utilized in machine vision literature relates to the values of $R'G'B'$ and thus attention must be given in the conversion of color models and other appropriate computations.

1.1.2 The CMY and CMYK Color Space

Like the RGB color model, the CMY color model is a subspace of conventional Cartesian three-dimensional space, considering the form of a cube unit. The fundamental secondary colors cyan, magenta, and yellow are represented by each axis [20]. However, unlike RGB, CMY is a subtractive model of color, which means that the source in RGB is pure black and the source in CMY is pure white. In other terms, growing CMY coordinate values shift to darker shades where growing RGB coordinate values shift to brighter colors (see Fig. 1.5). The transformation from RGB to CMY can be done utilizing Eq. (1.5).

Fig. 1.5 Normalized CMY coordinate color cube



$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1.5)$$

where all color values were assumed to be standardized to the interval [0,1]. Equation (1.5) reaffirms the CMY model's subtractive character. Although black should produce equivalent components of cyan, magenta, and yellow, it has been discovered that this contributes to muddy outcomes in printing applications. The fourth element of real black is introduced to produce the CMYK color space in printing applications. The use of this CMYK model relates to four-color printing. Like the RGB model, pixel ranges in the CMY color model do not conform to variations in perceptual color.

The method to transform RGB color to CMYK is stated below.

The R, G, B values are divided by 255 to change the range [0, 255] to [0, 1] as shown in Eq. (1.6).

$$\begin{aligned} \hat{R} &= R / 255 \\ \hat{G} &= G / 255 \\ \hat{B} &= B / 255 \end{aligned} \quad (1.6)$$

The black key (K) color is computed from the red (\hat{R}), green (\hat{G}), and blue (\hat{B}) colors as shown in Eq. (1.7).

$$K = 1 - \max(\hat{R}, \hat{G}, \hat{B}) \quad (1.7)$$

The cyan color (C) is computed from the red (\hat{R}) and black (K) colors as represented in Eq. (1.8).

$$C = \left(1 - \hat{R} - K \right) / (1 - K) \quad (1.8)$$

The magenta color (M) is computed from the green (\hat{G}) and black (K) colors as represented in Eq. (1.9).

$$M = \left(1 - \hat{G} - K \right) / (1 - K) \quad (1.9)$$

The yellow color (Y) is computed from the blue (\hat{B}) and black (K) colors as shown in Eq. (1.10).

$$Y = \left(1 - \hat{B} - K\right) / (1 - K) \quad (1.10)$$

The way to transform CMYK color to RGB is stated below.

The R, G, B values are specified in the range [0, 255]. The red (R) color is computed from the cyan (C) and black (K) colors as shown in Eq. (1.11).

$$R = 255 \times (1 - C) \times (1 - K) \quad (1.11)$$

The green color (G) is computed from the magenta (M) and black (K) colors as represented in Eq. (1.12).

$$G = 255 \times (1 - M) \times (1 - K) \quad (1.12)$$

The blue color (B) is computed from the yellow (Y) and black (K) colors as shown in Eq. (1.13).

$$B = 255 \times (1 - Y) \times (1 - K) \quad (1.13)$$

1.1.3 The HSV and HSL Color Space

The color models of HSV (hue, saturation, and value) and HSL (hue, saturation, and lightness) are very distinct from the earlier studied RGB and CMY/K color models as in that both systems separate a point's total intensity value from its chromaticity. The HSV color model can be viewed as a downward pointing hexacone in three dimensions [21]. The line that runs down the middle of the vertical axis of the cone signifies the value of intensity V. Hue is portrayed as the angle relative to the red axis residing perpendicular to the intensity axis on the plane. Saturation relates to the perpendicular distance of a point from the axis of intensity. Figure 1.6 shows this HSV color model's hexacone representation.

The group of equations below (Eqs. 1.14 –1.18) can be utilized to convert a point in the RGB coordinate system to the suitable value in the HSV space.

$$H' = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right\} \quad (1.14)$$

$$H = H', \text{ if } B \leq G \quad (1.15)$$

$$H = 360^\circ - H', \text{ if } B > G \quad (1.16)$$

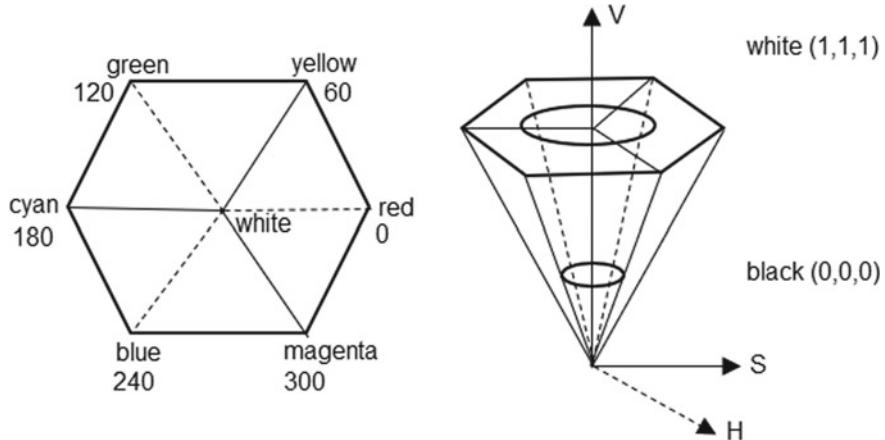


Fig. 1.6 Color hexagon for HSV depiction

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)} \quad (1.17)$$

$$V = \frac{\max(R, G, B)}{255} \quad (1.18)$$

Here the RGB value range is [0, 255].

The model of HSL color is very comparable to the HSV system. A double hexahedron is utilized to visualize the subspace in three dimensions, with two apexes at both pure black and pure white instead of just one at pure black. The saturation element in HSL [22] goes from a fully saturated color to the respective gray, while saturation in HSV, with V at its maximum, passes from a frilly transparent color to black. In addition, the color element in HSL always extends from black to white through the selected hue. In HSV, the component of intensity goes only from black to the selected hue. Because chromaticity is separated from the intensity in both HSV and HSL color spaces, intensity-based images can be processed only, leaving the original color data intact. This has resulted in the widespread utilization of HSL and HSV in computer vision research.

The HSI machine vision literature does not show whether the non-linear or linear RGB is being utilized in these transformations. Therefore, the non-linear ($R'G'B'$), which is inherent in conventional machine vision, is utilized. But the ambiguity must be observed. $R'G'B'$ (range [0, 1]) transformation to HSI (range [0, 1]) is extremely non-linear and complex as shown in Eqs. (1.19–1.21).

$$H = \cos^{-1} \left[\frac{\frac{1}{2}[(R' - G') + (R' - B')]}{\sqrt{[(R' - G')^2 + (R' - B')(G' - B')]}^{\frac{1}{2}}} \right] \quad (1.19)$$

$$S = 1 - \frac{3}{(R' + G' + B')} [\min(R', G', B')] \quad (1.20)$$

$$I = \frac{1}{3}(R' + G' + B') \quad (1.21)$$

where $H = 360^\circ - H$ if $(B'/I) > (G'/I)$. Hue is standardized to the range [0,1] by $H = H/360^\circ$. Hue (H) is not demarcated when the saturation (S) is zero. Likewise, saturation (S) is unspecified if the intensity (I) is zero.

To convert the HSI values to the R'G'B' in the range [0, 1], then the hue (H) value range [0, 1] first transformed back to the un-standardized in the range $[0^\circ, 360^\circ]$ by $H = 360^\circ(H)$. The R'G' conversion for $(0^\circ < H \leq 120^\circ)$ is shown in equation set (1.22).

$$\begin{aligned} B' &= I(1 - S) \\ R' &= I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \\ G' &= 3I - (R' + B') \end{aligned} \quad (1.22)$$

The G'B' conversion for $(120^\circ < H \leq 240^\circ)$ is shown in equation set (1.23).

$$\begin{aligned} H &= H - 120^\circ \\ R' &= I(1 - S) \\ G' &= I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \\ B' &= 3I - (R' + G') \end{aligned} \quad (1.23)$$

The B'R' conversion for $(240^\circ < H \leq 360^\circ)$ is shown in equation set (1.24).

$$\begin{aligned} H &= H - 240^\circ \\ G' &= I(1 - S) \\ B' &= I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \\ R' &= 3I - (G' + B') \end{aligned} \quad (1.24)$$

The vital benefits of the HSI color spaces over other color spaces are (1) good compatibility with human perception, (2) chromatic values separation from achromatic values, and (3) the prospect of utilizing one color feature, i.e., hue, only for segmentation purposes. Several image segmentation methods take benefit of this. Segmentation is generally done in one color feature (hue) as an alternative of three, permitting the utilization of much faster algorithms.

But, hue-oriented color models have some substantial limitations, like (1) singularities in the conversion, e.g., approximate hue for achromatic points, (2) sensitivity

to minor deviations of RGB values adjacent to singular points, and (3) mathematical unpredictability when working on hue because of the feature's angular nature.

1.1.4 The YIQ and YUV Color Space

The YIQ color system was created by and for the television industry as a result of a need to compress digital imagery streaming with as little image deterioration as needed [23]. The luminance value Y is divided from the chromaticity value I and Q, just like the HSV and HSL models. This enables engineers to represent the luminance value with more bits than the chromaticity attributes, as the human vision system is much more susceptible to the modifications of the intensity. Equation 1.25 provides an estimated linear transformation from a collection of RGB coordinates to the YIQ space.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.30 & 0.50 & 0.11 \\ 0.60 & -0.28 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1.25)$$

For television broadcasting along with the common compression procedures MPEG and JPEG, a comparable color model YUV is also utilized [24]. Equation 1.26 provides the estimated linear transformation from RGB coordinates to YUV space.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ -0.15 & -0.29 & -0.44 \\ 0.62 & -0.51 & 0.10 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1.26)$$

1.1.5 The CIE L*a*b* Color Space

The CIE $L^*a^*b^*$ color space or model was established to the perceptually uniform and possesses a Euclidean metric. This implies that there would be a strong correlation between Euclidean distance between two points (colors) and human visual interpretation [25]. CIE $L^*a^*b^*$ is totally based on the CIE XYZ color space, in which the components X, Y, and Z portray tristimulus that can convey any color that the average human observer can perceive. These primary colors are unreal, which means that the actual color stimuli cannot realize them. Because RGB is not an absolute color space, it is impossible to accurately convert RGB coordinates into CIE $L^*a^*b^*$ space and it cannot generate all humanly perceptible colors. Alternatively, by using Eq. 1.27, RGB coordinates can be projected into the CIE XYZ color space.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.41 & 0.36 & 0.18 \\ 0.24 & 0.72 & 0.07 \\ 0.02 & 0.12 & 0.95 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1.27)$$

where R, G, and B are on the range [0, 1]. From the XYZ model, it is now probable to project into CIE $L^*a^*b^*$ space using Eqs. (1.28–1.30).

$$L^* = 116 \left(\frac{Y}{Y_T} \right)^{1/3} - 16 \quad (1.28)$$

$$a^* = 500 \left[\left(\frac{X}{X_T} \right)^{1/3} - \left(\frac{Y}{Y_T} \right)^{1/3} \right] \quad (1.29)$$

$$b^* = 200 \left[\left(\frac{Y}{Y_T} \right)^{1/3} - \left(\frac{Z}{Z_T} \right)^{1/3} \right] \quad (1.30)$$

The values XT, YT, and ZT signify the tristimulus values of a reference white point. The back transformation to the XYZ model from the $L^*a^*b^*$ model is completed by first calculating the luminance Y, and subsequently the calculation of X and Z is represented in Eqs. (1.31–1.33).

$$Y = \left(\frac{L^* + 16}{116} \right)^3 Y_T \quad (1.31)$$

$$X = \left(\frac{a^*}{500} + \left(\frac{Y}{Y_T} \right)^{1/3} \right)^3 X_T \quad (1.32)$$

$$Z = \left(-\frac{b^*}{200} + \left(\frac{Y}{Y_T} \right)^{1/3} \right)^3 Z_T \quad (1.33)$$

1.1.6 The Munsell Color Space

The color space of Munsell reflects the previous effort to engage the perception of color into a color model. The area of Munsell is described as an artists' comparative reference [26]. Its overall shape is that of a three-dimensional cylindrical depiction of the perceived lightness, hue, and saturation. Furthermore, opposite to the HSI or HSV color models that are represented by the hue, saturation, and lightness of the color

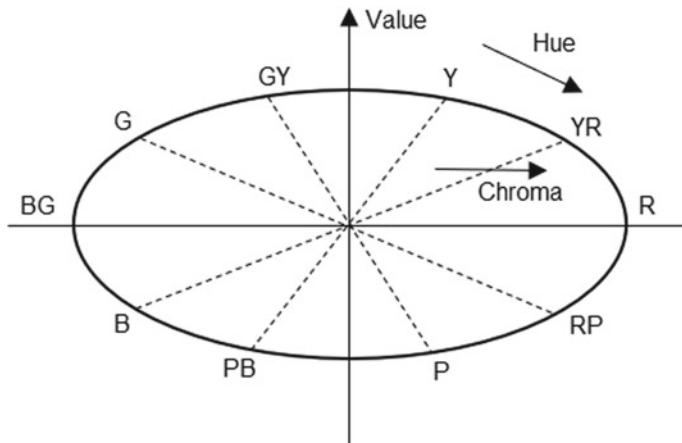


Fig. 1.7 The Munsell color space

solids, the Munsell model utilizes the color atlas method where the characteristics of perception are utilized for sampling.

The basic concept behind the color model of Munsell is that of equivalent visual spacing between each of the three characteristics. Hue is scaled to a color that is unique to define. A circular band split into ten parts to represent it. The sections are defined as red, yellow-red, yellow, green-yellow, green, blue-green, blue, purple-blue, purple, and red-purple.

If finer distinctions of hue are needed, each part can be further split into ten subsections. A chromatic hue is defined as one or two neighboring hues resembling it. Value in the color model of Munsell relates to the lightness or darkness of a color and is split into 11 parts counted from zero (black) to ten (white). The chroma describes the strength of the color. It is evaluated in consecutive steps beginning from one with low chroma values in weak colors. The highest chroma available relies on the utilization of the hue and the value. As shown in Fig. 1.7, the strong vertical axis of the Munsell color is the line of V observations from black to white. Hue shifts perpendicular to the vertical line along each circle. At last, on the V axis, chroma begins at zero and shifts along each circle's radius.

The space of Munsell consists of a collection of 1200 color chips each designated as a unique component of hue, value, and chroma. These chips are combined to create a three-dimensional structure that forms a warped sphere. The fundamental Munsell colors book has different editions, with various appearances (matte or glossy), various sample sizes, and various number of samples. The set of glossy finish collection shows color point chips organized on 40 charts of constant hue. The chips are organized in rows and columns on every constant-hue chart. The colors move from light at the top of each chart to very dark at the bottom by steps designed to be perceptually equal. They also move from achromatic colors like white and gray at the chart's inner edge to chromatic colors at the chart's outer end by steps that are also meant to be

perceptually equal. All graphs together form the color atlas, which is the Munsell system's solid color.

Even though the Munsell color book can be utilized to identify or label colors, it is not utilized in practice directly for applications for machine vision. Generally, stored image data is transformed to the Munsell coordinates, quite often in RGB format, either using closed formulas or lookup tables preceding to the actual application. Using the following mathematical algorithm, the transformation from the RGB components to the Munsell hue (M_H), value (M_V) correlating to the luminance, and chroma (M_C) correlating to the saturation can be attained.

$$\begin{aligned}x &= 0.620R + 0.178G + 0.204B \\y &= 0.299R + 0.587G + 0.144B \\z &= 0.056G + 0.942B\end{aligned}\tag{1.34}$$

A non-linear conversion is applied to the in-between values as shown in equation set (1.35).

$$\begin{aligned}a &= f(x) - f(y) \\b &= 0.4(f(z) - f(y))\end{aligned}\tag{1.35}$$

where $f(w) = 11.6w^{1/3} - 1.6$. Then the new variables are converted to equation set (1.36).

$$\begin{aligned}c &= (p + q \cos(\theta))a \\d &= (r + s \sin(\theta))b\end{aligned}\tag{1.36}$$

where $\theta = \tan^{-1}(a/b)$, $p = 8.880$, $q = 0.966$, $r = 8.025$, and $s = 2.558$. At last, (M_H), (M_V), and (M_C) are computed using Eqs. (1.37–1.39).

$$M_H = \arctan\left(\frac{c}{d}\right)\tag{1.37}$$

$$M_V = f(y)\tag{1.38}$$

$$M_C = \sqrt{(c^2 + d^2)}\tag{1.39}$$

Then again, the transformation from RGB, or other color models, to the Munsell color model can be attained by lookup tables.

1.1.7 The Opponent Color Space

This color space or model group is a collection of physiologically driven color models influenced by the human visual system's nature [27]. The human vision system can be defined in terms of opponent hues, blue and yellow on the one hand and red and green on the other hand, that suspend each other when overlaid, as per the color vision principle. The RGB signals are converted to three channels, two opponent color channels (RG, YB) and one achromatic channel (I) are obtained using Eqs. (1.40–1.42).

$$RG = R - G \quad (1.40)$$

$$YB = 2B - R - G \quad (1.41)$$

$$I = R + G + B \quad (1.42)$$

At the same time, systematic tests of region segmentation can derive a number of efficient color characteristics. The image that has the profound valleys on its histogram and has the greatest discriminating power to distinguish the image groups in a specified region does not need to be the R, G, and B color characteristics according to the segmentation method. As a characteristic is supposed to have great discriminating power if it has a large variance, the utilization of the Karhunen-Loeve (KL) conversion has obtained color characteristics with great discriminating strength. The calculation of the fresh color characteristics for the region pixels is performed by the KL transformation of R, G, and B signals at each phase of segmenting a region. Three color features can be determined that establish an efficient group of features for segmenting color images as represented in equation set (1.43).

$$\begin{aligned} I1 &= \frac{R + G + B}{3} \\ I2 &= R - G \\ I3 &= \frac{2G - R - B}{2} \end{aligned} \quad (1.43)$$

Color space hue could be coded in an opponent color in a circular format ranging from blue, green, yellow, red, and black to white. Saturation is described as distance in color classifications from the hue circle making hue and saturation specifiable. Thus, when opponent representation is often considered as a linear transformation of RGB space, opponent representation is much more appropriate for perceived image modeling than RGB is. Figure 1.8 shows the opponent color stage by the human visual system.

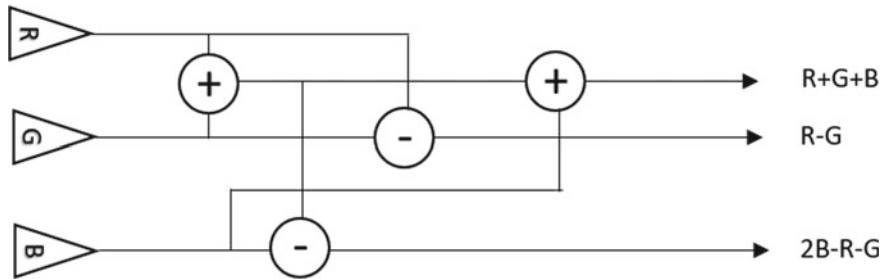


Fig. 1.8 The opponent color stage of human visual system

1.1.8 Hue-Max-Min-Difference Color Space

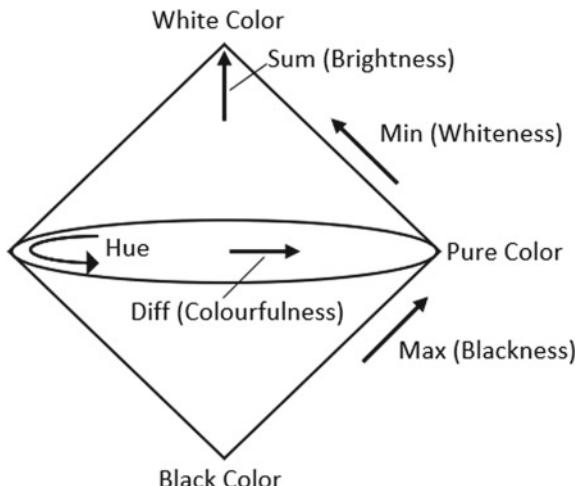
The color space of the Hue-Max-Min-Difference (HMMD) is identical to a space of perceptually uniform color. The double cone shape of this color space is shown in Fig. 1.9.

The elements are known as “Max,” “Min” and “Diff,” which are calculations between RGB and HMMD as per the transform shown in Eq. 1.44.

$$\begin{aligned} \text{Max} &= \max(R, G, B) \\ \text{Min} &= \min(R, G, B) \\ \text{Diff} &= \text{Max} - \text{Min} \end{aligned} \quad (1.44)$$

Although in the HMMD color space the four components are defined, one more component can be described as Sum as defined using Eq. (1.45).

Fig. 1.9 HMMD double cone color space



$$\text{Sum} = (\text{Max} + \text{Min})/2 \quad (1.45)$$

So in this color space, a total of five components are defined. A set of three elements, {H, Max, Min} or {H, Diff, Sum}, is however enough to construct the color space of the HMMD and define a color level. The definitions of each part are distinct. In HSV color space, Hue ($H = [0^\circ, 360^\circ]$) has the same property as Hue. Max ($= [0, 1]$) determines how much black color there is providing shadow or blackness flavor. Max has the same RGB-related transformation as HSV meaning, but HMMD varies from the true subspace. The definition is thus separate from Value. Min determines the amount of white color, providing the quality of whiteness or tint and the range is $[0, 1]$. Diff determines how similar color is to pure colors, offering a tonal or vibrant color in the range $[0, 1]$. It has a similar property in HSV as Saturation but again the valid subspace is different. Lastly, the color brightness is determined by the sum with range $[0, 1]$.

1.1.9 New Trends

The variety of available color models have problems with the application. Since most of them are intended to conduct well in a particular application, their efficiency under varying working circumstances is quickly deteriorating [28]. Thus, various (mainly device-dependent) color spaces need to be merged into a single standard model. The variations between monitor device-independent spaces and RGB space, like HSV and CIE $L^*a^*b^*$ spaces, create difficulties in applications, for example, face recognition, multimedia database recognition, etc. mainly because of the complication in the procedures required to support the color spaces reliant on the transformation from/to the device.

An innovative standardized image model created on a colorimetric RGB (sRGB) model is used to solve such issues and satisfy the requirements of network-centered applications and WWW-based image processing systems. The objective of the innovative color model is to complement the present color model management approaches by offering a straightforward yet reliable and cost-effective image processing technique in operating devices, device drivers, and the Web utilizing a straightforward, robust device-independent color concept.

As most computer monitors are comparable in their key color features and the RGB model is the most appropriate color model for devices that form a recent computer-based imaging system, the colorimetric RGB model seems like the best candidate for this standardized color model. Two factors are of primary significance in identifying a colorimetric space, such as (1) the observing environment parameters with their human visual system dependencies and (2) the normal computer space colorimetric definitions and conversions.

The descriptions of the viewing environment comprise all the transformations necessary to assist transitions between standard and target viewing environment. The colorimetric definitions, on the other hand, include the required transformations

to transform between the new sRGB and the color space of CIE-XYZ. The parameters of the reference viewing environment with the sRGB tristimulus values are computed from the CIE-XYZ values as shown in Eq. (1.46).

$$\begin{bmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{bmatrix} = \begin{bmatrix} 3.2410 & -1.5374 & -0.4986 \\ -0.9692 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (1.46)$$

Negative sRGB tristimulus values and sRGB attributes higher than 1 are not maintained in practical image processing schemes and are typically deleted using some type of clipping. The linear tristimulus values are converted as follows (equation set (1.47) and (1.48)) into non-linear sR'G'B's.

If $R_{sRGB}, G_{sRGB}, B_{sRGB} \leq 0.0034$, then

$$\begin{aligned} sR' &= 12.92R_{sRGB} \\ sG' &= 12.92G_{sRGB} \\ sB' &= 12.92B_{sRGB} \end{aligned} \quad (1.47)$$

If $R_{sRGB}, G_{sRGB}, B_{sRGB} > 0.0034$, then

$$\begin{aligned} sR' &= 1.055R_{sRGB}^{1.0/2.4} - 0.055 \\ sG' &= 1.055G_{sRGB}^{1.0/2.4} - 0.055 \\ sB' &= 1.055B_{sRGB}^{1.0/2.4} - 0.055 \end{aligned} \quad (1.48)$$

With a black digital count of 0 and a white digital count of 255 for 24-bit coding as depicted in equation set (1.49), the non-linear R'G'B' values are then transformed into digital values.

$$\begin{aligned} sR_e &= 255sR' \\ sG_e &= 255sG' \\ sB_e &= 255sB' \end{aligned} \quad (1.49)$$

The backward conversion is demarcated using equation set (1.50).

$$\begin{aligned} sR' &= sR_e + 255 \\ sG' &= sG_e + 255 \\ sB' &= sB_e + 255 \end{aligned} \quad (1.50)$$

If $R_{sRGB}, G_{sRGB}, B_{sRGB} \leq 0.03928$, then

$$\begin{aligned} R_{sRGB} &= sR' + 12.92 \\ G_{sRGB} &= sG' + 12.92 \\ B_{sRGB} &= sB' + 12.92 \end{aligned} \quad (1.51)$$

If $R_{sRGB}, G_{sRGB}, B_{sRGB} > 0.03928$, then

$$\begin{aligned} R_{sRGB} &= \left(\frac{sR' + 0.055}{1.055} \right)^{2.4} \\ G_{sRGB} &= \left(\frac{sG' + 0.055}{1.055} \right)^{2.4} \\ B_{sRGB} &= \left(\frac{sB' + 0.055}{1.055} \right)^{2.4} \end{aligned} \quad (1.51)$$

With,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{bmatrix} \quad (1.52)$$

Adding a new standardized color model that promotes Web-based imaging technologies, device drivers, monitors, and printers that complement current color management support can help both manufacturers and users by providing a definite path to an enhanced color management scheme.

1.2 Color Quantization

Color quantization is the method by which the number of colors used to depict an image is reduced. The image space and the segmentation (i.e., split up) of the color space used are determined by a quantization system [29]. Each axis is split into a set of components by implementing a conventional quantization system on color space. The number of colors (n) utilized to depict an image will be $n = p \times q \times r$ when the axes are divided into p , q , and r parts. Color space quantization into n colors is often referred to as a system for quantizing n -bins. Figure 1.10 shows the impact of the color image quantization. Each axis' segmentation relies on the color space used.

In the next section, different color spaces and their quantization methods will be described.



Fig. 1.10 Color quantization: **a** the original image using 256^3 colors, **b** quantized in 64 bins, **c** quantized in 8 bins, using RGB color space

1.2.1 *Scalar or Uniform Color Quantization*

Each part of the color space is handled separately in scalar quantization. Then each axis is split into segments of equivalent size. The planes that move through the division points perpendicular to the axis then define areas in the color space. The number of these areas depends on the color space division scheme used. One feasible strategy is to split the axis of red and green into 8 sections and the axis of blue into 4, producing 256 areas. Another option is to divide the red and blue into 6 and the green into 7 sections which produces 252 areas. Each one of these regions will generate a color for the color map.

After dividing the color space, each of the original colors is mapped to the region in which it falls. For each region, the representative colors are then averaged of all the colors mapped to that region. Since each region depicts an entry in the color map, it is possible to reiterate the same method for mapping the original colors to a region to map the original colors to the color map. While the implementation of this algorithm is fast and easy, it does not produce excellent results. There is often no mapping of colors in the color space, resulting in colormap entries to be wasted.

If the axis is broken on a logarithmic scale rather than linear, this algorithm can also be implemented in a non-uniform way. This will generate slightly improved outcomes as both dark and bright colors cannot be distinguished by the human eye.

1.2.2 *Vector Color Quantization*

This method does not treat individual components distinctly. Every color pixel is treated as an individual entity. The goal is to discover a set of m demonstrative color vectors and substitute each original color by one of the new color vectors. m is generally pre-determined. The resultant deviation should be nominal. This method

turns out to be an expensive global optimization problem. The following approaches thus used to compute local optima.

Popularity Algorithm

Another type of uniform quantization is the popularity algorithms. Rather than splitting the color space into 256 regions, these algorithms split the color model into much smaller areas, and therefore many more regions are created. One feasible implementation is to split the space into $4 \times 4 \times 4$ regions (262,144 regions) in size. The original colors are mapped again to the region in which they fall. For each region, the representative color is the average of the projected colors. The color map is chosen by getting the representative colors from the 256 most prominent regions (the regions with the most mapping of colors). If a non-empty region is not chosen for the color map its index in the color map (the index assigned to color mapping to that region) is the color map element nearest to its representative pixel (Euclidean distance).

These algorithms are also simple to execute and produce better outcomes than the standardized algorithm of quantization. However, they require a longer time to perform and may have a considerably higher storage demand based on the region size. This may not generate a good outcome based on the image features. This can be said about all standardized sub-division systems since any information about the image is used by the technique of dividing the color space.

Median-Cut Algorithm

In this algorithm, the classical technique is considered for color quantization. This technique is implemented in several applications. In this technique, the first color histogram of the original image is computed. Then recursively split RGB color space till the number of boxes is equal to the anticipated number of representative colors. At every step of recursion, a box with maximum pixels is split at the median of the longest of its three axes so that half pixels are left in each sub-box. In the final step, the mean color of all pixels in each sub-box is calculated and utilized as the representative color (each contained pixel is substituted by this mean). Since these algorithms utilize image information in splitting the color space this class of algorithms constantly generates good outcomes while consuming memory and time complexity no worse than popularity algorithms.

1.2.3 Octree and Other Color Quantization Algorithms

The octree algorithm's concept is to read sequentially in the image. Each color is then placed in an octree of volume 8 (a distinct color is represented by each leaf at depth 8). On the tree, a threshold of K (in this event $K = 256$) leaves is placed. Inserting a color into the tree can lead to two results.

1. If there are less than K nodes, the image will be filtered down the tree until either it hits some leaf nodes with a representative color associated with it or it gets the leaf node depicting its distinctive color.

2. If there are more than K leaves in the tree, some groups of leaves in the tree must be combined together (their average representative colors) and a new representative color is stored in the parent.

In the literature, there are two probable criteria to be utilized in the selection of leaves to be combined.

1. First, select reducible nodes with the largest depth in the tree. They depict colors nearest to each other.
2. If there is more than one leaves the group at the highest depth of the algorithm could
 - A. Combine the leaves that denote the least number of pixels. This will support to preserve the error small.
 - B. Decrease the leaves that signify the most pixels. In this case, big areas will be consistently filled in a little wrong color while preserving thorough shadings.

The color map comprises the representative colors of the leaf and clusters in the tree once the whole image has been handled in this way. The color map index will then be placed on that leaf, and the image quantization method will simply filter down the tree until a leaf is hit.

Other approaches utilize 10% of the pixels that are arbitrarily chosen as representative colors. Statistical and clustering methods can be used to choose the representative colors such as the k-means method.

1.3 Pseudocolor Image Processing

The purpose of pseudocolor processing is to color a grayscale image by assigning different colors in different intensity ranges of a gray-level image. Pseudocolor is also called false color as the colors are not originally present in the grayscale image [30]. The human eye can interpret near about two dozen of gray shades in a grayscale image whereas can interpret nearly 1000 variations of color in a color image. So given a grayscale image, if it is converted to a color image by using pseudocolor processing, then the interpretation of different intensities is much more convenient as compared to the ordinary grayscale image. Pseudocoloring can be done by intensity slicing method. Suppose there are L number of intensity values in a grayscale image $I(x, y)$ which varies from $0, \dots, (L - 1)$. In this case, l_0 represents black where $I(x, y) = 0$ and l_{L-1} represents white where $I(x, y) = L - 1$. Suppose there is P number of planes perpendicular to the intensity plane where $0 < P < L - 1$. These planes are placed to the intensity levels l_1, l_2, \dots, l_P . P number of planes divide the intensities to $P + 1$ number of intervals. So, the color C_k assigned to the gray-level intensity at position (x, y) can be denoted by $f(x, y) = C_k$ if $I(x, y) \in D_k$ where D_k is the intensity range between l_k and l_{k+1} . So, it can be said that P number of planes divide the intensities to $P + 1$ number of intervals denoted by D_1, D_2, \dots, D_{P+1} . By using

this concept, the gray-level intensity range can be divided into some intervals and for each interval, a particular color can be assigned. By this way, a grayscale image can be colored and this procedure is known as pseudocoloring. Figure 1.11 shows the pseudocoloring image of a grayscale image. By the pseudocolor image, we can visualize different intensities of the image region with different colors, which are almost flat in the grayscale image. So, using the pseudocolor image, intensities of the image are much more interpretable or distinguishable than the grayscale image. In the case of RGB image, colors are added to R, G, and B channels separately and the combination of R, G, and B channels gives the interpretation of the pseudocolor image.

Grayscale to color image conversion can be done by the transformations shown in Fig. 1.12.

In Fig. 1.12, $I(x, y)$ is the grayscale image which is transformed by three different transformations: RED transformation, GREEN transformation, and BLUE transformation [6]. RED, GREEN, and BLUE transformation gives the red, green, and blue plane output of the input grayscale image which is given by

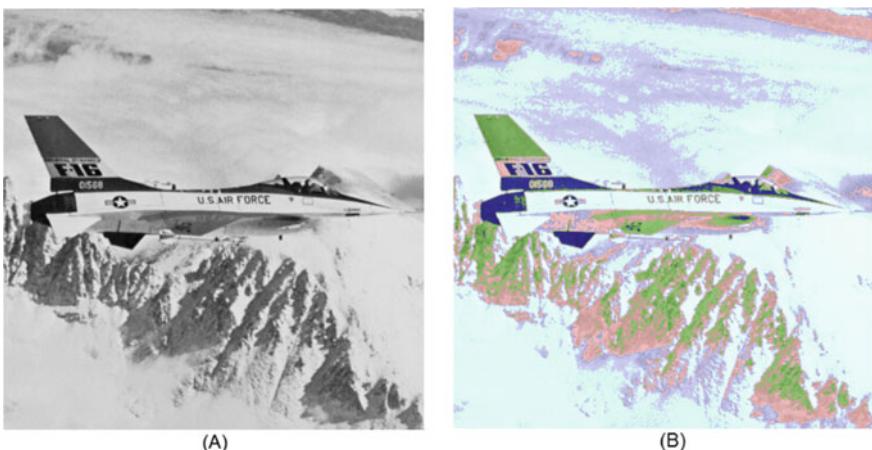


Fig. 1.11 **a** Grayscale image, **b** pseudocolor image



Fig. 1.12 Grayscale to color transformation



Fig. 1.13 **a** Grayscale image, **b** pseudocolor transformed image

$I_R(x, y)$, $I_G(x, y)$, and $I_B(x, y)$. When these three planes are combined together and displayed in a color display system then it is known as a pseudocolor image. For example, Eq. (1.52) denotes the transformation functions used to generate the color image and Fig. 1.13 shows the color transformation of a grayscale image by using Eq. (1.53).

$$\begin{aligned} I_R(x, y) &= I(x, y) \\ I_G(x, y) &= 0.33 \times I(x, y) \\ I_B(x, y) &= 0.11 \times I(x, y) \end{aligned} \quad (1.53)$$

In this example, to convert this grayscale image to color, the exact intensities of the grayscale image are copied to the red plane but the degraded version of intensities of the original grayscale image is used in the green and blue planes. The combination of this red, green, and blue planes is shown in Fig. 1.13.

1.4 Full-Color Image Processing

There are two major categories of full-color image processing as mentioned below:

1. Process each component of the image separately and then generate a composite processed color image. Each component can be processed utilizing grayscale processing methods.
2. Work with color pixels directly by considering each pixel as a vector.

$$c = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1.54)$$

Since each pixel is a function of coordinates $(x; y)$, Eq. (1.54) can be expressed as shown in Eq. (1.55).

$$c(x, y) = \begin{bmatrix} c_R(x, y) \\ c_G(x, y) \\ c_B(x, y) \end{bmatrix} = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix} \quad (1.55)$$

where each vector component is a spatial variable in x and y .

The two approaches may or may not generate equivalent outcomes. The procedure utilized should be valid to both scalars and vectors. The process of each component of the vector must be independent of the other components. Neighborhood processing will be an example where these two methods generate dissimilar outcomes. Averaging the images distinctly in individual planes and averaging the vectors will generate unalike outcomes. In theory, any conversion can be done in any color space. In practice, some conversions are better suited for specific color spaces. The cost of color space transformations must be measured.

1.5 Summary

The need for image color in the field of image retrieval is discussed in this chapter. Color image processing includes pseudocolor and full-color or true-color processing. The purpose of pseudocolor processing is to color a grayscale image by assigning different colors in different intensity ranges of a gray-level image. In the case of RGB image, colors are added to R, G, and B channels separately and the combination of R, G, and B channels gives the interpretation of the pseudocolor image. By the pseudocolor image, we can visualize different intensities of the image region with a different color, which are almost flat in the grayscale image. So, using the pseudocolor image, intensities of the image are much more interpretable or distinguishable than the grayscale image. In the full-color image, the actual color of the image is considered. In such type of images, the colors can be specified using different color models like RGB (linear and non-linear), HSI, HSV, CMY, CMYK, CIE $L^*a^*b^*$, YUV, YIQ, Munsell, HMMD, Opponent, etc. Algorithms for the conversion from one color space to another are also mentioned in this chapter. Different color quantization techniques such as scalar or uniform, vector quantization, octree, etc. are discussed in this chapter.

References

1. Rama Varior R, Wang G, Lu J, Liu T (2016) Learning invariant color features for person reidentification. *IEEE Trans Image Process* 25(7):3395–3410. <https://doi.org/10.1109/tip.2016.2531280>
2. Benitez-Quiroz F, Srinivasan R, Martinez AM, Discriminant functional learning of color features for the recognition of facial action units and their intensities. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/tpami.2018.2868952>
3. Tommoy TH, Hanif MA, Rahman HA, Khandaker N, Hossain I (2016) Error reduction in arsenic detection through color spectrum analysis. In: 2016 19th international conference on computer and information technology (ICCIT). Dhaka, pp 343–350. <https://doi.org/10.1109/iccitechn.2016.7860221>
4. Dey N, Ashour AS, Hassani AE (2017) Feature detectors and descriptors generations with numerous images and video applications: a recap. In: Feature detectors and motion detection in video processing, pp 36–65. <https://doi.org/10.4018/978-1-5225-1025-3.ch003>
5. Wang C, Li Z, Dey N, Li Z, Ashour AS, Fong SJ, Shi F (2018) Histogram of oriented gradient based plantar pressure image feature extraction and classification employing fuzzy support vector machine. *J Med Imaging Health Inform* 8(4):842–854. <https://doi.org/10.1166/jmihi.2018.2310>
6. Pi JK, Yang J, Zhong Q, Wu MB, Yang HC, Schwartzkopf M, Roth SV, Muller-Buschbaum P, Xu ZK (2019) Dual-layer nanofilms via mussel-inspiration and silication for non-iridescent structural color spectrum in flexible displays. *ACS Appl Nano Mater*. <https://doi.org/10.1021/acsnano.9b00909>
7. Devlin RC, Khorasaninejad M, Chen WT, Oh J, Capasso F (2016) Broadband high-efficiency dielectric metasurfaces for the visible spectrum. *Proc Natl Acad Sci* 113(38):10473–10478. <https://doi.org/10.1073/pnas.1611740113>
8. Morley CV, Fortney JJ, Marley MS, Zahngle K, Line M, Kempton E, Lewis N, Cahoy K (2015) Thermal emission and reflected light spectra of super Earths with flat transmission spectra. *Astrophys J* 815(2):110
9. Perlman I (2016) Absorption, light, spectra of for visual pigments. Encyclopedia of Ophthalmology, pp 1–2. https://doi.org/10.1007/978-3-642-35951-4_1036-1
10. Wang S et al (2015) Micro-expression recognition using color spaces. *IEEE Trans Image Process* 24(12):6034–6047. <https://doi.org/10.1109/TIP.2015.2496314>
11. Fedenko VS, Shemet SA, Landi M (2017) UV-vis spectroscopy and colorimetric models for detecting anthocyanin-metal complexes in plants: an overview of in vitro and in vivo techniques. *J Plant Physiol* 212:13–28. <https://doi.org/10.1016/j.jplph.2017.02.001>
12. Cyriac P, Bertalmio M, Kane D, Vazquez-Corral J (2015) A tone mapping operator based on neural and psychophysical models of visual perception. In: Human vision and electronic imaging. International Society for Optics and Photonics, vol 9394, p 93941I. <https://doi.org/10.1117/12.2081212>
13. Gao S, Yang K, Li C, Li Y (2015) Color constancy using double-opponency. *IEEE Trans Pattern Anal Mach Intell* 37(10):1973–1985. <https://doi.org/10.1109/tpami.2015.2396053>
14. Ganasa P, Kumar V, Prasad AD (2016) Performance evaluation of color models in the fusion of functional and anatomical images. *J Med Syst* 40(5):122. <https://doi.org/10.1007/s10916-016-0478-5>
15. Ganesan P, Sathish BS, Vasanth K, Sivakumar VG, Vadivel M, Ravi CN (2019) A comprehensive review of the impact of color space on image segmentation. In: 2019 5th international conference on advanced computing & communication systems (ICACCS). Coimbatore, India, pp 962–967. <https://doi.org/10.1109/icaccs.2019.8728392>
16. Ganesan P, Sajiv G (2017) User oriented color space for satellite image segmentation using fuzzy based techniques. In: 2017 international conference on innovations in information, embedded and communication systems (ICIIECS). Coimbatore, pp 1–6. <https://doi.org/10.1109/iciiecs.2017.8275977>

17. Zhang Z, Huang W, Li W, Tian J (2017) Illumination-based and device-independent imaging model and spectral response functions. In: 2017 IEEE 7th annual international conference on cyber technology in automation, control, and intelligent systems (CYBER). Honolulu, HI, pp 47–52. <https://doi.org/10.1109/cyber.2017.8446589>
18. Vaishnavi D, Subashini TS (2015) Robust and invisible image watermarking in RGB color space using SVD. Procedia Comput Sci 46:1770–1777. <https://doi.org/10.1016/j.procs.2015.02.130>
19. Kolkur S, Kalbande D, Shimpali P, Bapat C, Jatakia J (2017) Human skin detection using RGB, HSV and YCbCr color models. arXiv preprint [arXiv:1708.02694](https://arxiv.org/abs/1708.02694)
20. Bao X, Song W, Liu S (2017) Research on color space conversion model from CMYK to CIE-LAB based on GRNN. In: Pacific-Rim symposium on image and video technology. Springer, Cham, pp 252–261. https://doi.org/10.1007/978-3-319-75786-5_21
21. Shaik KB, Ganesan P, Kalist V, Sathish BS, Jenitha JMM (2015) Comparative study of skin color detection and segmentation in HSV and YCbCr color space. Procedia Comput Sci 57:41–48. <https://doi.org/10.1016/j.procs.2015.07.362>
22. Saravanan G, Yamuna G, Nandhini S (2016) Real time implementation of RGB to HSV/HSI/HSL and its reverse color space models. In: 2016 international conference on communication and signal processing (ICCSP). Melmaruvathur, pp 0462–0466. <https://doi.org/10.1109/iccsp.2016.7754179>
23. Ma J, Fan X, Yang SX, Zhang X, Zhu X (2018) Contrast limited adaptive histogram equalization-based fusion in YIQ and HSI color spaces for underwater image enhancement. Int J Pattern Recognit Artif Intell 32(07):1854018. <https://doi.org/10.1142/S0218001418540186>
24. Prema CE, Vinsley SS, Suresh S (2016) Multi feature analysis of smoke in YUV color space for early forest fire detection. Fire Technol 52(5):1319–1342. <https://doi.org/10.1007/s10694-016-0580-8>
25. del Mar Pérez M, Ghinea R, Rivas MJ, Yebra A, Ionescu AM, Paravina RD, Herrera LJ (2016) Development of a customized whiteness index for dentistry based on CIELAB color space. Dental Mater 32(3):461–467. <https://doi.org/10.1016/j.dental.2015.12.008>
26. Paramei GV, Griber YA, Mylonas D (2018) An online color naming experiment in Russian using Munsell color samples. Color Res Appl 43(3):358–374. <https://doi.org/10.1002/col.22190>
27. Gong J, Guo J (2016) Image copy-move forgery detection using SURF in opponent color space. Trans Tianjin Univ 22(2):151–157. <https://doi.org/10.1007/s12209-016-2705-z>
28. Sağı T, Çunkaş M (2015) Color image segmentation based on multiobjective artificial bee colony optimization. Appl Soft Comput 34:389–401. <https://doi.org/10.1016/j.asoc.2015.05.016>
29. Valenzuela G, Celebi ME, Schaefer G (2018) Color quantization using Coreset sampling. In: 2018 IEEE international conference on systems, man, and cybernetics (SMC). Miyazaki, Japan, pp 2096–2101. <https://doi.org/10.1109/smcc.2018.00361>
30. Jiang N, Wu W, Wang L, Zhao N (2015) Quantum image pseudocolor coding based on the density-stratified method. Quantum Inf Process 14(5):1735–1755. <https://doi.org/10.1007/s11128-015-0986-0>

Chapter 2

Histogram-Based Image Color Features



2.1 Color Histogram

A color histogram (CH) in photography and image processing is a depiction of color variation within an image. A color histogram for digital images portrays the pixel number of that have colors in every one of defined color range list, and that include the color model of the image and the collection of all probable colors [1, 2].

CH can be constructed for any type of color model, even though the term is utilized mostly for three-dimensional spaces such as HSV or RGB. The term intensity histogram may be utilized for monochromatic images. The color histogram is D -dimensional for multi-spectral images, in which every pixel is depicted by a random number of measurements (for instance, beyond the three RGB measurements), with D being the number of measurements taken. Every measurement has its own set of light spectrum wavelengths, some of which can be outside of the observable spectrum.

If the collection of probable color values is small enough, each of these colors can be positioned in a range on its own, and then the histogram is simply the pixel count with each color probable. The model is frequently splitted into a reasonable range number, frequently organized as a regular grid, each comprising several alike values of color. Also, CH can be portrayed and presented as a smooth function described over the color model approximating the pixel number.

CH, like other types of histograms, is a statistic that can be interpreted as an estimate of the underlying continuous color value ranges.

The histogram offers a compact overview of the data distribution in an image. An image's CH is fairly invariant with the orientation and translation around the viewing axis and only changes gradually with the viewing angle [3, 4]. CH is especially well suited for the issue of identifying an object of unknown location and orientation within a scene by equating histogram signatures of two images and matching the color of one image with another. Specifically, the conversion of an RGB image into

the illumination invariant RG-chromaticity space makes it possible for the histogram to work well at different light levels.

2.2 Histogram Intersection

The histogram intersection does not need the exact segregation of the object from its background as well as it is reliable to the occlusion of objects in the foreground. Histogram intersection is broadly used due to its capability to handle partial matches when the areas of the two histograms are dissimilar [5–8]. Assumed a pair of histograms, P and Q , comprising n bins, respectively, the histogram intersection is demarcated by using Eq. (2.1).

$$\sum_{i=1}^n \min(P_i, Q_i) \quad (2.1)$$

The result of a model histogram intersection with an image histogram is the pixel number from the model having equivalent pixels in the image of the same color. To get a normalized histogram intersection, it is divided by the pixel number in the histogram model. The similarity value can then be represented using Eq. (2.2).

$$HI(P, Q) = \frac{\sum_{i=1}^n \min(P_i, Q_i)}{\sum_{i=1}^n Q_i} \quad (2.2)$$

The normalized match value for the histogram intersection is not decreased by affecting the background pixels. The value of the histogram intersection match is enhanced by a pixel in the background if the pixel has the similar color as one of the model's colors, and the pixel number of that color in the object is lower than the pixel number of that model color.

Figure 2.1 illustrates the histogram intersection.

The 100 bins are on the x -axis while the number of elements inside each bin is represented on the y -axis. The intersection between the red graph and the green one is depicted by blue color. When the two distributions are close, the intersection is bigger.

2.3 Color Histogram Discrimination

The formulas for color distance come at a measure of similarity among images based on the interpretation of color information [9–15]. Several distance formulas

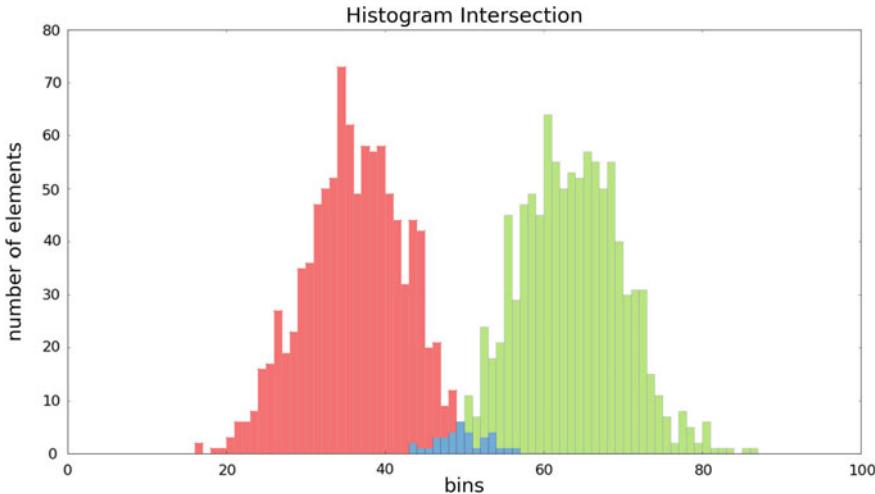


Fig. 2.1 Histogram intersection

have been utilized for image recognition which comprise Histogram Minkowski Distance, Histogram Manhattan Distance, Histogram Euclidean Distance, Histogram Chebyshev Distance, Histogram Intersection Distance, Histogram Quadratic (Cross) Distance, Histogram Cosine Distance, Histogram Canberra Distance, Histogram Kolmogorov-Smirnov Divergence Distance, Histogram Cramer-von Mises Distance, Histogram Chi-square Distance, Histogram Squared Chord Distance, Histogram Kullback-Leibler Divergence Distance, and Histogram Jeffrey Divergence Distance.

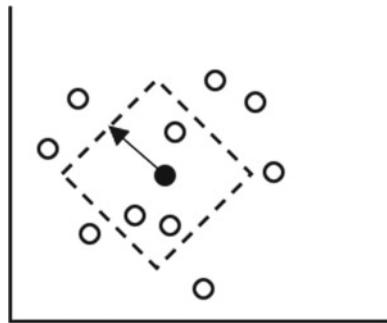
Let $P(i, j, k)$ and $Q(i, j, k)$ signify two-color histograms where i , j , and k are the number of colors in each color channel.

2.3.1 Histogram Minkowski Distance

Histogram Minkowski Distance (HMD) is calculated in normed vector space. A normed vector space is a vector space defining a norm. Presume V is a vector space then a norm on V is a valued function which fulfills the following conditions: (1) zero vector—this vector has zero length, (2) scalar factor—the direction of the vector will not change if it is multiplied with a positive number, and (3) triangle inequality—if the distance is a norm then the measured distance between two points is always a straight line.

As the norm vector has the abovementioned properties, it helps the HMD to be consistent and invariant in translation. HMD can be calculated using Eq. (2.3).

Fig. 2.2 Geometric interpretation of HMD



$$\text{HMD} = \left(\sum_i \sum_j \sum_k |P(i, j, k) - Q(i, j, k)|^r \right)^{1/r} \quad (2.3)$$

where r is a positive integer.

Minkowski distance is the generalized histogram distance measure. Generalized implies that this distance can be modified in various ways to measure the distance between the two data points: $r = 1$ means Manhattan distance or city block distance or taxicab geometry or L_1 distance, .. means Euclidean distance or L_2 distance, and $r = \infty$ means Chebyshev distance.

2.3.2 Histogram Manhattan Distance

Histogram Manhattan Distance (HMD) calculation is biased to the histogram resembling a reference image with the histogram focusing at the lower bits. HMD among color histograms can be calculated using an absolute sum of the difference between the histograms and can be expressed using Eq. (2.4).

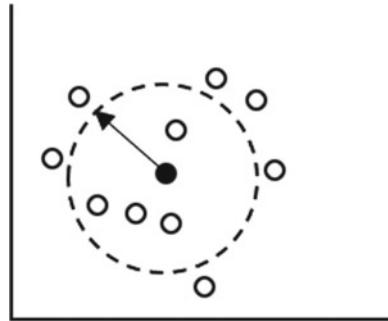
$$\text{HMD} = \sum_i \sum_j \sum_k |P(i, j, k) - Q(i, j, k)| \quad (2.4)$$

The geometric interpretation of the HMD is shown in Fig. 2.2.

2.3.3 Histogram Euclidean Distance

The Euclidean distance (HED) among the color histograms can be calculated using Eq. (2.5).

Fig. 2.3 Geometric interpretation of HED



$$\text{EHD}(P, Q) = \sqrt{\sum_i \sum_j \sum_k (P(i, j, k) - Q(i, j, k))^2} \quad (2.5)$$

The geometric interpretation of HED is shown in Fig. 2.3.

2.3.4 Histogram Chebyshev Distance

Chebyshev distance is also recognized as the maximum metric. Other than this, it is also known as the chessboard distance, as the number of steps required by a chess king to travel from one point to another can be shown on the real number plane. The Histogram Chebyshev Distance (HCD) can be calculated using Eq. (2.6).

$$\text{HCD} = \max_{i,j,k} |P(i, j, k) - Q(i, j, k)| \quad (2.6)$$

HCD value is the maximum of the difference of histograms of the image pair, which demonstrates the maximum variation of the two images.

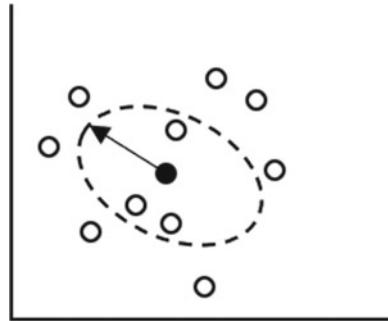
2.3.5 Histogram Intersection Distance

The color Histogram Intersection Distance (HID) is used for color image recognition. The intersection of histograms $P(i, j, k)$ and $Q(i, j, k)$ is depicted using Eq. (2.7).

$$\text{HID}(P, Q) = \frac{\sum_i \sum_j \sum_k \min(P(i, j, k), Q(i, j, k))}{\min(|P|, |Q|)} \quad (2.7)$$

The magnitude of each histogram is represented by $|P|$ and $|Q|$, which is equal to the number of samples. Colors that are not available in the query image of the

Fig. 2.4 Geometric interpretation of HQD



user do not contribute to the distance between the intersections. That decreases the background color contribution.

2.3.6 Histogram Quadratic (Cross) Distance

The color Histogram Quadratic Distance (HQD) was utilized by the QBIC system. It is also known as Mahalanobis distance. HQD can be represented using Eq. (2.8).

$$\text{HQD}(P, Q) = (P - Q)^t M (P - Q) \quad (2.8)$$

The cross-distance method calculates the cross-correlation among histogram bins based on the visual similarity of the bins depicted in the colors. And a matrix M , which is a similarity matrix, represents the collection of all cross-correlation values. And $m(i, j)$ -th element in the similarity matrix M for RGB space is represented by Eq. (2.9).

$$m_{ij} = 1 - \text{HQD}_{ij} / \max(\text{HQD}_{ij}) \quad (2.9)$$

where HQD_{ij} is the distance among the color i and j in the RGB space. HQD is used for color histogram similarity as it closely resembles human perception.

The geometric interpretation of the HQD is shown in Fig. 2.4.

2.3.7 Histogram Cosine Distance

The Histogram Cosine Distance (HCD) is small when the image points are in the same direction (i.e., are similar). HCD will be large when the image points are in different directions. HCD can be computed using Eq. (2.10).

$$\text{HCD} = 1 - \frac{\sum_i \sum_j \sum_k P(i, j, k) Q(i, j, k)}{\sqrt{\sum_i \sum_j \sum_k P(i, j, k)^2} \sqrt{\sum_i \sum_j \sum_k Q(i, j, k)^2}} \quad (2.10)$$

2.3.8 Histogram Canberra Distance

The Histogram Canberra Distance (HCaD) has a good effect on the data which is spread about the origin. The Manhattan distance gives a high-value output for the two similar images which create differences among similar images. Therefore, every histogram pair difference is normalized by dividing it by the sum of a pair of histograms. It has similarities with HMD. HCaD can be computed using Eq. (2.11).

$$\text{HCaD} = \sum_i \sum_j \sum_k \frac{|P(i, j, k) - Q(i, j, k)|}{\min(|P(i, j, k)|, |Q(i, j, k)|)} \quad (2.11)$$

2.3.9 Histogram Kolmogorov-Smirnov Divergence Distance

This is a non-symmetric distance measure. The Histogram Kolmogorov-Smirnov Divergence Distance (HKSD) can be computed using Eq. (2.12).

$$\text{HKSD} = \max_{i,j,k} |P(i, j, k) - Q(i, j, k)| \quad (2.12)$$

2.3.10 Histogram Cramer-Von Mises Distance

In this test, the aim is to add up the square differences among the contrasting cumulative distributions. Again, this distance is generally conceived as a test to equate an observed distribution with a presumed continuous distribution of probability by the parent. The algorithm can, therefore, be adapted to the comparison of two samples, and the comparison of two histograms. The Histogram Cramer-von Mises Distance (HCMD) can be calculated using Eq. (2.13).

$$\text{HCMD} = \sum_i \sum_j \sum_k (P(i, j, k) - Q(i, j, k))^2 \quad (2.13)$$

2.3.11 Histogram Chi-Square Distance

The Histogram Chi-square Distance (HCSD) calculates the underlying similarity of two histograms where differences are emphasized. It can be computed using Eq. (2.14).

$$\text{HCSD} = \sum_i \sum_j \sum_k \frac{[P(i, j, k) - Q(i, j, k)]^2}{P(i, j, k) + Q(i, j, k)} \quad (2.14)$$

The major limitation of HCSD is that it accounts only for the correspondence between bins with the same index and does not use information across bins.

2.3.12 Histogram Squared Chord Distance

The Histogram Squared Chord Distance (HSCD) can be calculated using Eq. (2.15).

$$\text{HSCD} = \sum_i \sum_j \sum_k \left(\sqrt{P(i, j, k)} - \sqrt{Q(i, j, k)} \right)^2 \quad (2.15)$$

2.3.13 Histogram Kullback-Leibler Divergence Distance

The Histogram Kullback-Leibler Divergence Distance (HKLD) considers histograms as distributions and calculates their similarity by measuring the relative entropy. It measures how far away one histogram is from the other. It is not symmetric. HKLD of Q from P can be computed using Eq. (2.16).

$$\text{HKLD} = \sum_i \sum_j \sum_k P(i, j, k) \log \frac{P(i, j, k)}{(P(i, j, k) + Q(i, j, k))/2} \quad (2.16)$$

The major limitation of this measure is that it does not inevitably match perceptual similarity well and is delicate to histogram binning.

2.3.14 Histogram Jeffrey Divergence Distance

The Histogram Jeffrey Divergence Distance (HJD) is the modification of HKLD that is symmetric, numerically stable, and robust regarding noise and the size of histogram bins. It can be calculated using Eq. (2.17).

$$\text{HJD} = \sum_i \sum_j \sum_k \left[P(i, j, k) \log \frac{P(i, j, k)}{(P(i, j, k) + Q(i, j, k))/2} + Q(i, j, k) \log \frac{Q(i, j, k)}{(P(i, j, k) + Q(i, j, k))/2} \right] \quad (2.17)$$

2.4 Fuzzy Color Histogram

Color histogram signifies the uneven distribution of colors in the image. Two similar colors will be measured as similar since they are assigned to the same histogram bin. In contrast, if they fall into two distinct bins, two colors will be considered completely distinct although they might be very identical to each other [16–22]. This makes color histogram resistant to noisy disturbances like variations in illumination and errors in quantization. This is the main need for the fuzzy color histogram. In addition to the resemblance of various colors from various bins, a fuzzy color histogram recognizes the dissimilarity of those colors belonging to the same bin. Fuzzy color histogram thus becomes more resistant to the noisy interference. The fuzzy color histogram of the image I can be articulated as $\text{FCH}(I) = [\text{FCH}_1, \text{FCH}_2, \dots, \text{FCH}_n]$, which is depicted using Eq. (2.7).

$$\text{FCH}_i = \sum_{j=1}^N \mu_{ij} S_j = \frac{1}{N} \sum_{j=1}^N \mu_{ij} \quad (2.7)$$

where S_j is the probability of a pixel chosen from image I being the j -th pixel, which is $1/N$, and $S_{i|j}$ is the conditional probability of the chosen j -th pixel belonging to the i -th color bin, and ij is the membership value of the j -th pixel in the i -th color bin.

To measure a color image's fuzzy color histogram, the membership values for each pixel are calculated concerning all existing color bins. Next, in RGB color space, the quantization is performed by mapping all pixel colors to the n histogram bins. These n colors are grouped into n clusters, each cluster describing a fuzzy bin of color.

Fuzzy C-Means Clustering (FCM) algorithm is utilized to categorize n colors to n cluster and attain membership values. Membership values are characterized as a membership matrix $U = [\mu_{ij}]_{n \times n'}$. Each element ij in U is the membership value of the j -th color bin in the i -th color bin. Membership matrix U is pre-computed only once and can be utilized to produce the fuzzy color histogram for each database image.

Algorithm of FCM is as follows

Step 1: Select the number of clusters c , the weighting exponent w , error tolerance ϵ , and maximum iteration I_{\max} .

Step 2: Initialize the cluster centroid r_i for $1 \leq i \leq c$.

Step 3: Input data $L = \{l_1, l_2, \dots, l_n\}$.

Step 4: Update the membership $U^{(I)}$ with

$$u_{ik}^{(I)} = \frac{1}{\sum_{j=1}^c \left[\left\| \frac{x_k - r_i}{x_k - r_j} \right\| \right]^{\frac{2}{w-1}}}$$

$$i = 1, 2, \dots, c$$

$$k = 1, 2, \dots, n$$

Step 5: Update all cluster center $R^{(I)}$ with

$$r_i^{(I)} = \frac{\sum_{k=1}^n (\mu_{ik})^w x_k}{\sum_{k=1}^n (\mu_{ik})^w}$$

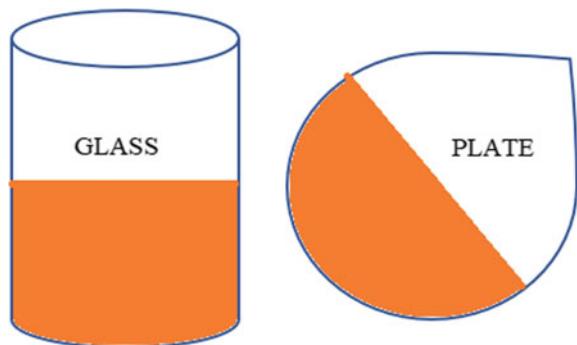
Step 6: if $\|R_{I+1} - R_I\| > \epsilon$ or $I < I_{\max}$, $I = I + 1$. Return to step 4, otherwise stop.

There are some advantages and limitations of color histogram image features. Some of them are discussed here.

Advantages: Histogram features can be produced very easily from the image histogram, and the comparison of these features is computationally quick and efficient. This uses basic image properties, as characteristics of the histogram. Histograms are invariant to rotation around the viewing axis and translation and alter gradually when the angle of view changes or size changes, and occlusion occurs. Since histograms alter gradually with the view, a small number of histograms, referring to a collection of canonical views, can properly represent a three-dimensional object.

Limitations: The first problem is the color histograms having a high dimensionality. This high dimensionality allows the implementation of feature reduction, hierarchical indexing, and prefiltering methods. The wide dimensionality also makes the distance equation more complex. Cross-distance functions that include the perceptual distance among histogram bins are particularly complicated. Another major drawback of the characterization of histograms is that the depiction is based on the color of the object being examined, overlooking its shape and texture. CH will theoretically be the same for two images with distinct properties of objects that share some information about colors. For example, there is no way to differentiate a red and

Fig. 2.5 Glass and plate with the same histogram



white glass from a red and white plate if the number of red and white pixels is the same in glass and plate (Fig. 2.5).

2.5 Summary

This chapter presents histogram-based image color features like histogram intersection, fuzzy histogram, and different distance measures that can be used to check the similarity between color image histogram. Several histogram distance measures like Histogram Minkowski distance, Histogram Euclidean Distance, Histogram Intersection Distance, Histogram Quadratic (Cross) Distance, Histogram Manhattan Distance, Histogram Chebyshev Distance, Histogram Cosine Distance, Histogram Canberra Distance, Histogram Kolmogorov-Smirnov Divergence Distance, Histogram Cramer-von Mises Distance, Histogram Chi-square Distance, Histogram Squared Chord Distance, Histogram Kullback-Leibler Divergence Distance, and Histogram Jeffrey Divergence Distance are discussed in this chapter. The most commonly used histogram distance measure is the Euclidean distance. This distance has a physical sense when the feature vector lies in two- or three-dimensional spaces: the straight line distance between two data points. When the feature vector lies in higher dimensional space, the Euclidean distance may be influenced by the elements of certain outliers. A small number of elements of outliers can dominate the gap in the feature vector. This is why in some cases the Manhattan distance is more robust than the Euclidean distance since the summation of the absolute difference in Manhattan is smoother than the Euclidean distance square operation. Also, several advantages and limitations of image color histogram are discussed in this chapter.

References

1. Erkut U, Bostancioğlu F, Erten M, Özbayoğlu AM, Solak E (2019) HSV color histogram based image retrieval with background elimination. In: 2019 IEEE 1st international informatics and software engineering conference (UBMYK). Ankara, pp 1–5. <https://doi.org/10.1109/ubmyk48245.2019.8965513>
2. Ali H, Lali MI, Nawaz MZ, Sharif M, Saleem BA (2017) Symptom based automated detection of citrus diseases using color histogram and textural descriptors. Comput Electron Agric 138:92–104. <https://doi.org/10.1016/j.compag.2017.04.008>
3. Huang X, Zhang R, Jia K, Wang Z, Nie W (2018) Taxi detection based on the sliding color histogram matching. In: 2018 IEEE 3rd international conference on image, vision and computing (ICIVC). Chongqing, pp 86–90. <https://doi.org/10.1109/icivc.2018.8492826>
4. Borjigin S, Sahoo PK (2019) Color image segmentation based on multi-level Tsallis–Havrda–Charvát entropy and 2D histogram using PSO algorithms. Pattern Recogn 92:107–118. <https://doi.org/10.1016/j.patcog.2019.03.011>
5. Wu Y, Huang TS (2002) Nonstationary color tracking for vision-based human-computer interaction. IEEE Trans Neural Networks 13(4):948–960. <https://doi.org/10.1109/TNN.2002.1021895>
6. Lee NK, Kapanidis AN, Koh HR, Korlann Y, Ho SO, Kim Y, Gassman N, Kim SK, Weiss S (2007) Three-color alternating-laser excitation of single molecules: monitoring multiple interactions and distances. Biophys J 92(1):303–312. <https://doi.org/10.1529/biophysj.106.093211>
7. Liew AC, Leung SH, Lau WH (2003) Segmentation of color lip images by spatial fuzzy clustering. IEEE Trans Fuzzy Syst 11(4):542–549. <https://doi.org/10.1109/TFUZZ.2003.814843>
8. Lee JH, Lee WH, Jeong DS (2003) Object tracking method using back-projection of multiple color histogram models. In: Proceedings of the 2003 international symposium on circuits and systems, 2003. ISCAS'03, Bangkok, vol 2, pp II–II. <https://doi.org/10.1109/iscas.2003.1206062>
9. Hafner J, Sawhney HS, Equitz W, Flickner M, Niblack W (1995) Efficient color histogram indexing for quadratic form distance functions. IEEE Trans Pattern Anal Mach Intell 17(7):729–736. <https://doi.org/10.1109/34.391417>
10. Soni P, Lamba VK, Jangra S, Kumar S (2019) Effective CBIR system using color histogram and distance measures. J Web Eng Technol 6(1):11–14
11. Forero MG, Arias-Rubio C, González BT (2018) Analytical comparison of histogram distance measures. In: Iberoamerican congress on pattern recognition. Springer, Cham, pp 81–90. https://doi.org/10.1007/978-3-030-13469-3_10
12. Fan J, Liang RZ (2018) Stochastic learning of multi-instance dictionary for earth mover's distance-based histogram comparison. Neural Comput Appl 29(10):733–743. <https://doi.org/10.1007/s00521-016-2603-2>
13. Mistry Y, Ingole DT, Ingole MD (2018) Content based image retrieval using hybrid features and various distance metric. J Electr Syst Inf Technol 5(3):874–888. <https://doi.org/10.1016/j.jesit.2016.12.009>
14. Suhasini PS, Krishna KSR, Krishna IM (2017) Content based image retrieval based on different global and local color histogram methods: a survey. J Inst Eng (India): Ser B 98(1):129–135. <https://doi.org/10.1007/s40031-016-0223-y>
15. Irawan C, Listyaningsih W, Sari CA, Rachmawanto EH (2018) CBIR for herbs root using color histogram and GLCM based on K-nearest neighbor. In: 2018 international seminar on application for technology of information and communication. Semarang, pp 509–514. <https://doi.org/10.1109/isematic.2018.8549779>
16. Gharde ND, Thounaojam DM, Soni B, Biswas SK (2018) Robust perceptual image hashing using fuzzy color histogram. Multimedia Tools Appl 77(23):30815–30840. <https://doi.org/10.1007/s11042-018-6115-1>

17. Almubarak HA, Stanley RJ, Stoecker WV, Moss RH (2017) Fuzzy color clustering for melanoma diagnosis in dermoscopy images. *Information* 8(3):89. <https://doi.org/10.3390/inf08030089>
18. Zhao J, Xie G (2016) A modified fuzzy color histogram using vision perception variation of pixels at different location. *Multimedia Tools Appl* 75(2):1261–1284. <https://doi.org/10.1007/s11042-014-2367-6>
19. Akimoto S, Takahashi T, Suzuki M, Arai Y, Aoyagi S (2016) Human detection by fourier descriptors and fuzzy color histograms with fuzzy c-Means method. *J Robot Mechatron* 28(4):491–499. <https://doi.org/10.20965/jrm.2016.p0491>
20. Forcén JI, Pagola M, Bustince H, Soto-Hidalgo JM, Chamorro-Martínez J (2017) Adding fuzzy color information for image classification. In: 2017 IEEE international conference on fuzzy systems (FUZZ-IEEE). Naples, pp 1–6. <https://doi.org/10.1109/fuzz-ieee.2017.8015622>
21. Liu H, Zhao F, Chaudhary V (2018) Pareto-based interval type-2 fuzzy c-means with multi-scale JND color histogram for image segmentation. *Digit Sig Process* 76:75–83. <https://doi.org/10.1016/j.dsp.2018.02.005>
22. Veluchamy M, Subramani B (2020) Fuzzy dissimilarity color histogram equalization for contrast enhancement and color correction. *Appl Soft Comput* 89:106077. <https://doi.org/10.1016/j.asoc.2020.106077>

Chapter 3

MPEG-7 Image Color Features



This chapter delivers an outline of MPEG-7 color descriptors. The choice of these color descriptors is influenced by various factors [1–3]. These include (a) their capacity to classify the likeness of perceptual colors, assessed by descriptor performance in matching video segments and images using color features, (b) minimum complexity of the related feature extraction and matching procedures, because MPEG-7 systems must be capable of handling recovery tasks through broad multi-media repositories, or small mobile devices with minimal computational power, (c) coded description sizes that have a significant role in indexing and distributing descriptors through limited bandwidth networks, and (d) the descriptors' scalability and interoperability.

3.1 Dominant Color Descriptor

This descriptor represents a small number of dominant color attributes and also their statistical characteristics such as variance and distribution. This descriptor gives a compact depiction of the color set in a region of image [4–7]. The objective is to deliver a powerful, portable, and understandable depiction of colors present in an area of the image.

Its key applications are image database resemblance extraction and image database surfing based on one or multiple colors. Contrasting from the conventional histogram-based features, the demonstrative colors are calculated from every image rather than being static in the color model, thereby permitting for reliable and compact color representation.

The Dominant Color Descriptor (DCD) is demarcated using Eq. (3.1).

$$\text{DCD} = \{\{CVal_j, Per_j, CVar_j\}, SC\}, (j = 1, 2, \dots, M) \quad (3.1)$$



Fig. 3.1 The feature extraction process of DCD

where M is the number of dominant colors. Every dominant color value $CVal_j$ is a vector with the consequent values of the color space element. The percentage Per_j (standardized to a value in range $[0 \ 1]$) is the pixel fraction equivalent to the color $CVal_j$ in the image or image region and $\sum_j Per_j = 1$. The color variance $CVar_j$ defines the variability of the pixel colors in a group across the relevant color set. The spatial coherence s is a single number representing the complete spatial uniformity of the image's dominant colors. The number of dominant colors M can differ from image to image, and it was observed that a sum of eight dominant colors is adequate to reflect an image or area of the image. The color space quantization relies on the color space parameters specified for the whole database and does not require to be listed with every descriptor.

The dominant color extraction technique uses the generalized Lloyd algorithm to segment the range of pixel values in the color space belonging to a specified image area into clusters. The algorithm reduces a global measurement of distortions GD described in Eq. (3.2).

$$GD = \sum_{p=1}^P \|CV_p - CC_j(p)\|^2 \quad (3.2)$$

where P is the pixel number, CV_p is the p -th color vector, and $CC_j(p)$ is the p -th pixel cluster or group center (color set).

DCD feature extraction procedure (Fig. 3.1) comprises three phases: a conversion of color space, a clustering technique, and a computation of each centroid's percentages.

The first phase is a conversion of color space suggested by the MPEG-7 specification, to execute the clustering process in a standardized color space including the CIE LUV. This color space conversion had to be done because of the artifacts utilized during this experiment on the RGB color space.

The color model CIE LUV is implemented by the International Commission on Illumination (CIE). It is widely utilized for computer graphics that deal with colored lights.

Its depiction is illustrated in Fig. 3.2.

Initially, the RGB values are transformed into the CIE XYZ color space as shown in Eq. (3.3).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.24 & -1.53 & -0.49 \\ -0.97 & 1.87 & 0.04 \\ 0.05 & -0.21 & 1.06 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.3)$$

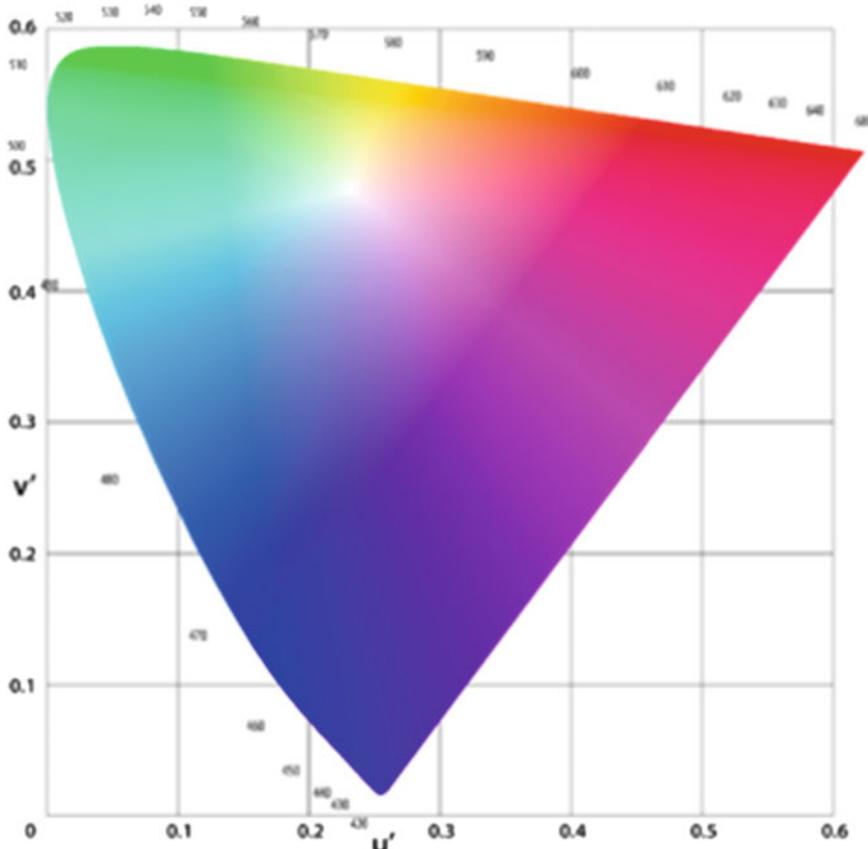


Fig. 3.2 Representation of CIE LUV color space

When this procedure has completed, the conversion to the CIE LUV color space is executed as shown in Eq. (3.4).

$$\begin{aligned}L^* &= 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16 \\u^* &= 13L^*(u' - u'_n) \\v^* &= 13L^*(v' - v'_n)\end{aligned}\quad (3.4)$$

where u'_n and v'_n denote the reference light source or the white point. u' and v' can be represented using Eq. (3.5).

$$u' = \frac{4X}{X + 15Y + 3Z} = \frac{4x}{-2x + 12y + 3}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} = \frac{9y}{-2x + 12y + 3} \quad (3.5)$$

Once the input image is converted to CIE LUV, a clustering algorithm is used to find the dominant image colors. The method is started with a single cluster composed of the entire pixels and one symbolic color determined as the cluster's centroid (center of the mass). Any technique of clustering can be used, but the K-means algorithm is the one selected for this chapter. K-means clustering is a cluster assessment technique that attempts to divide m findings into k clusters, where each observation belongs to the closest mean cluster. One instance of the clustering is shown in Fig. 3.3. It is expected to form two clusters from m items. Two clusters are acquired by applying K-means, Cluster 1 in red and Cluster 2 in blue. The respective centroids are also computed.

In the case of image sorting, the m pixels of the image into k clusters are necessary using K-means to produce pixels of similar colors. This method implements a series of centroid computation and clustering phases until the required number of clusters is achieved.

Consequently, the following diagram is an example of the clustering of colors utilizing K-means to derive an input image from DCD. As the standard suggests three or four dominant colors to be used for the DCD, four colors have been chosen

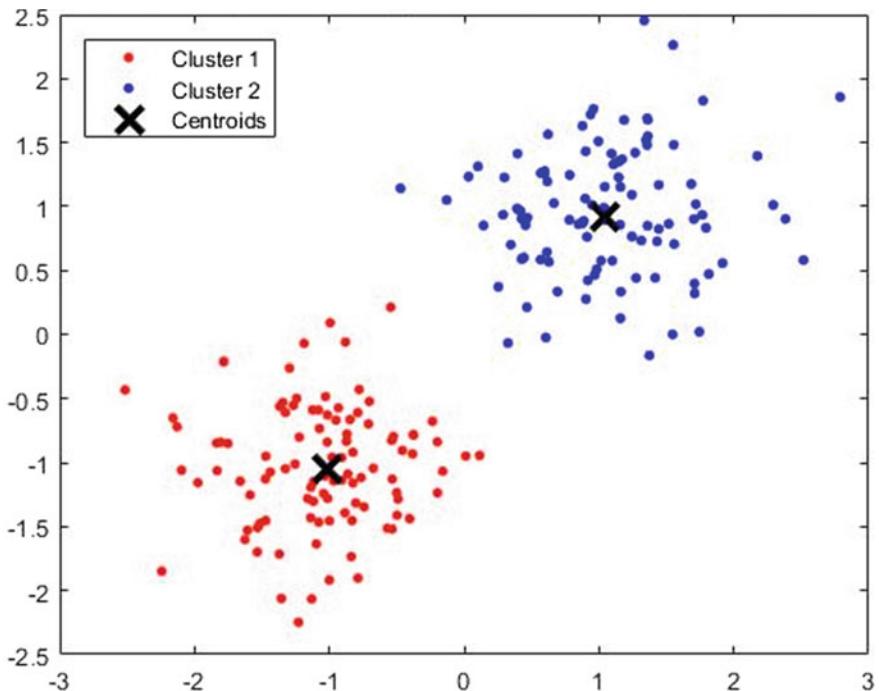


Fig. 3.3 Cluster assignments and centroids

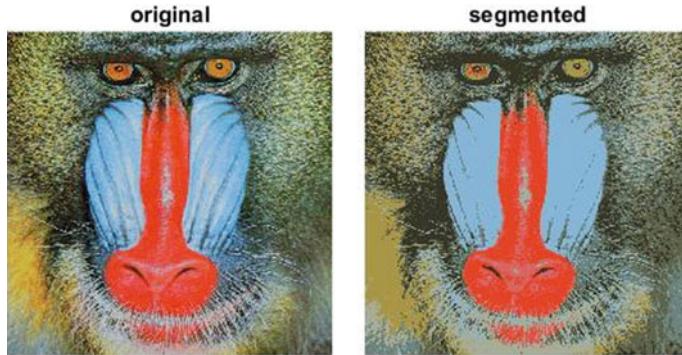


Fig. 3.4 K-means: RGB clustering with eight clusters

in this specific example. In Fig. 3.4, the image is in RGB color space when the K-means algorithm is implemented to enhance its comprehension.

It can be noted that only eight dominant colors were picked from the original image which relates to the right K-means cluster centroids.

The algorithm then performs a series of stages of centroid approximation and clustering till it reaches a stopping condition (lowest possible distortion or highest possible iteration number). The largest distortion clusters are splitted by applying disruption vectors to their centroids till the distortion drops beneath a predetermined level or the maximum cluster number has been produced.

The spatial coherence is determined by choosing every color set and measuring the coherence per group as the total pixel connectivity within the group, and then calculating a weighted average of these measurements utilizing percentages as weights.

The similarity checking function is based on the components exist in the target and query descriptors. The basic similarity checking function M_{DC} utilizes only the color values and the similarity coefficient and is demarcated using Eq. (3.6).

$$M_{DC} = \sum_{i=1}^{P_1} Q_i^2 + \sum_{j=1}^{P_2} T_j^2 - \sum_{i=1}^{P_1} \sum_{j=1}^{P_2} 2SC_{ij}Q_iT_j \quad (3.6)$$

where T and Q are the target and query descriptor, respectively and SC_{ij} is the similarity coefficient between two colors.

There are several advantages and limitations of this descriptor. Some of them are mentioned here.

Advantages: A concise summary of the symbolic image color or image area color is provided by DCD. This permits a small number of dominant color values to be specific and also their mathematical characteristics such as variance distribution. Its aim is to deliver an efficient, compressed, and straightforward color depiction in an image area present.

Limitations: DCD provides the absence of spatial information of the description.

3.2 Scalable Color Descriptor

The Scalable Color Descriptor (SCD) is extracted from a color histogram specified in the color space Hue-Saturation-Value (HSV) with a fixed color model quantization. This utilizes a Haar transform coefficient encoding, facilitating scalable information presentation, and also complexity scalability of the extraction and matching processes of features [8–11].

SCD can be represented as a Haar-based encoding scheme implemented in the HSV color space through the values of a color histogram. The histogram values are derived, transformed, and represented in an integer representation in a non-linear manner, which gives deeper meaning to the small values. The Haar transform is extended across the histogram bins to the integer values. The transform's basic unit comprises a difference and a sum operation (Fig. 3.5), which relates to primitive high-pass filters and low-pass filters. Adjacent bin sum pairs are comparable to the computation of a histogram with half the number of bins. From the aggregation of each two neighboring Hue bin values from the 256-bin histogram, a depiction of a 128-bin histogram of 8 levels can be obtained in H, 4 levels in S, and 4 levels in V. If this method is replicated, the corresponding 64, 32, or 16 sum coefficients from the Haar representation are equal to 64, 32, or 16 bins histogram. If an implementation does not need full resolution, a limited number of Haar coefficients can be derived from a histogram of 32, 64, or 128 bins; this will still ensure compatibility with one more depiction where all coefficients have been extracted and the accuracy of the coefficients usable in both depictions. As all divisions in the original color space quantization are powers of 2, the mixture with the Haar transformation indicates to be very normal.

The Haar transform's high-pass (difference) coefficients convey the details in finer resolution levels (with a higher number of bins). Natural image signal histograms generally present a strong correlation between neighboring histogram bins. This can be stated by the color “imperfection” (minor variation) due to different impacts of shadowing and illumination. Therefore, the high-pass coefficients conveying distinctions between adjacent histogram bins can generally be predicted to have only limited values. Taking advantage of this feature, the high-pass coefficients can be truncated to an integer depiction with only a small number of bits.

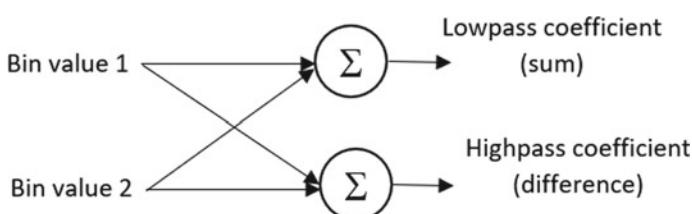


Fig. 3.5 The basic unit of Haar transform

The default matching function (D_{SC}) for scalable color in the Haar transform space is created on the Manhattan distance metric and is specified by Eq. (3.7).

$$D_{SC} = \sum_{i=1}^M |H_T[i] - H_Q| \quad (3.7)$$

where M is the number of training, H_T and H_Q are the training and query image, respectively. After reversing a Haar transform, the similarity can be measured either in the original color histogram space or in the Haar transform space. For both cases, the recovery efficiency is very similar, usually slightly higher if the histogram is recreated.

There are several advantages and limitations of this descriptor. Some of these are discussed here.

Advantages: SCD is compact, robust, perceptual meaning.

Limitations: Need post-processing for spatial information.

3.3 Group of Image/Group of Frames Color Descriptor

The group of images/group of frames color descriptor is an expansion of the SCD to a selection of frames in a video or image series. Such descriptor is focused on combining the color attributes of the discrete images or video frames.

The color classification using Group of Image/Group of frame (GoI/GoF) is utilized for the combined description of color-based attributes of frames in a video segment or multiple images. This descriptor can be utilized to depict a group of images or a group of adjacent or non-adjacent video frames [12–15]. A key frame or a key image is usually selected from such a selection for a group of frames or images, and the color-related characteristics of the whole collection are defined by those of the selected sample. These approaches rely heavily on the consistency of the representative sample size and may lead to inaccurate tests. GoI/GoF are color descriptors based on histograms that efficiently capture the color content of multiple video frames or images.

The construction of the GoI/GoF is the same as that of scalable color with the only difference of one added field, aggregation, which depicts how the color pixels from various images/frames are combined before the color histogram extraction. The probable values are median, average, and intersection.

The median histogram $GoIF_{MH}$ is attained by computing the median value from every histogram bin over the images/frames and conveying this value to the resultant histogram bin. The median histogram can be computed using Eq. (3.8).

$$GoIF_{MH}[i] = \text{median}(H_0[i], \dots, H_{M-1}[i]) \quad (3.8)$$

where i is the number of the histogram of each image/frame and M is the number of images/frames.

By the use of the median, the consequence of outliers in the image/frame groups can be eliminated. Thus, $GoIF_{MH}$ enhances the recovery performance in cases of alteration of light, occlusion, etc.

The average histogram $GoIF_{AV}$ is attained as the color histogram average for all the image/frames in the group and can be calculated using Eq. (3.9).

$$GoIF_{AV}[i] = \frac{1}{M} \sum_{c=1}^M H_c[i] \quad (3.9)$$

where c is the number of images/frames and i is the number of histogram bins. $GoIF_{AV}$ usually provides the recovery performance for representative video clips.

The intersection histogram $GoIF_{IN}$ is attained by computing the smallest value of each histogram bin over the images/frames and conveying this value to the resultant histogram bin. $GoIF_{IN}$ is calculated using Eq. (3.10).

$$GoIF_{IN}[i] = \min_c(H_c[i]) \quad (3.10)$$

where c is the number of images/frames and i is the number of histogram bins. $GoIF_{IN}$ discovers the least mutual colors in the images/frames and can consequently be utilized in applications that need the exposure of a high-level correlation in the color.

As mentioned earlier, the GoI/GoF is an extension of the SCD. The depiction for the GoI/GoF color descriptor is similar to the SCD with an added characteristic “aggregation.” The three different probable approaches of aggregation are signified by this feature. This is trailed by the related SCD descriptor.

3.4 Color Layout Descriptor

The Color Layout Descriptor (CLD) depicts the spatial layout on a grid overlaid on a region or image of the dominant colors. The depiction is focused on the Discrete Cosine Transform (DCT) coefficients [16–19]. It is a very compact descriptor that is extremely effective in quick browsing and search applications. It may be extended to both still images and video clips.

Figure 3.6 demonstrates how the descriptor extracts from an image.

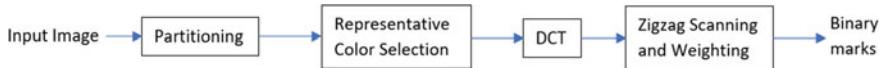


Fig. 3.6 CLD extraction procedure from an image

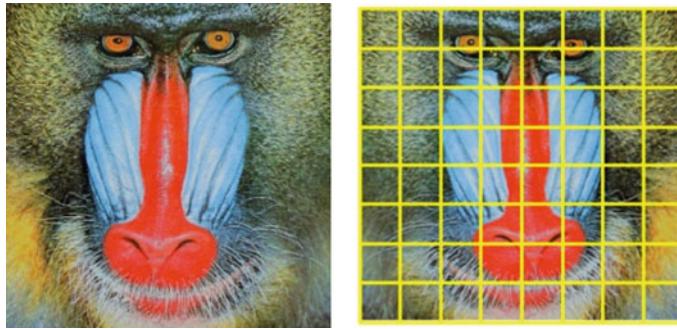


Fig. 3.7 Partitioning

This comprises four steps such as image partitioning, representative color selection, DCT transformation, and zigzag-scanned coefficient quantization. In the first phase, an input image is partitioned into 64 blocks to ensure an invariance in resolution or scale. Figure 3.7 demonstrates the first phase.

In the second step, each block selects a single dominant color. Any technique of choosing a representative color can be implemented, but it is preferred to utilize the average pixel color as the representative color because it is easiest and the consistency of the representation is usually enough. The selection results in an 8×8 size image icon. Figure 3.8 shows the representative color selection.

Once the image symbol is produced, the translation of the color model is done between RGB and YC_bC_r . The conversion of color space can be performed at any level through the MPEG-7 standard advised to continue with the conversion at this point to minimize the computing load involved in this process. This conversion is demarcated by a linear transformation of the RGB color model as shown in Eq. (3.11).

$$Y(\text{Luminance}) = 0.299 \times R + 0.587 \times G + 0.114 \times B - 128$$

$$C_b(\text{Blue_Chrominance}) = 0.169 \times R - 0.331 \times G + 0.500 \times B$$

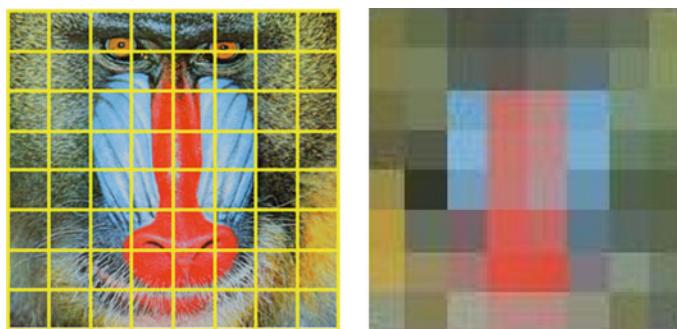


Fig. 3.8 Representative color selection

$$C_r(\text{Red_Chrominance}) = 0.500 \times R - 0.419 \times G - 0.081 \times B \quad (3.11)$$

In the third phase, the luminance (Y) and the red and blue chrominance (C_r and C_b) are converted by 8×8 DCT, thus obtaining three sets of 64 DCT coefficients. To compute the DCT in a 2D image array of size $P \times Q$, the following formula (Eq. 3.12) is utilized.

$$I(u, v) = \sqrt{2/P} \sqrt{2/Q} \sum_{i=0}^{P-1} \sum_{j=0}^{Q-1} \Lambda(i) \cdot \Lambda(j) \cdot \cos\left[\frac{\pi \cdot u}{2 \cdot P}(2i+1)\right] \cos\left[\frac{\pi \cdot v}{2 \cdot Q}(2j+1)\right] \cdot I(i, j)$$

where,

$$\Lambda(x) = \begin{cases} \frac{1}{\sqrt{2}}, & x = 0 \\ 1, & \text{otherwise} \end{cases} \quad (3.12)$$

They are scanned in a zigzag manner, and the first few coefficients are quantized non-linearly (using 32 and 64 levels, respectively, for AC and DC coefficients). Figure 3.9 illustrates the zigzag scanning method.

The zigzag scan aims to cluster the low-frequency coefficients of the 8×8 matrix. The standard permits the function to be interpreted scalably by adjusting the number of embedded coefficients. For YC_bC_r color images, it is suggested to utilize a minimum of 12 coefficients: 6 for luminance and 3 for C_b as well as C_r . Nevertheless, it may also be considered another choice to utilize a minimum of 18 coefficients: 6 for both chrominance and luminance, for high-quality still pictures.

This descriptor's key features are comparing video-clip-to-video-clip, image-to-image as well as matching image/video-clip sketch. A definition of the color scheme can be gained by using the MPEG-7 Grid Scheme data type and the DCD. Though this combination would involve a significantly higher number of bits, matching will

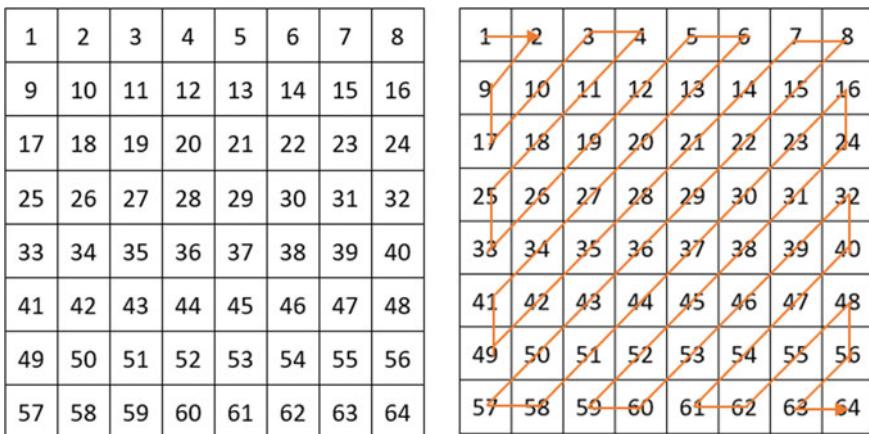


Fig. 3.9 Zigzag scanning

be more complicated and costly. CLD delivers more accurate and quicker recovery utilizing a more compact definition.

There are several advantages and limitations of this descriptor. Some of them are mentioned here.

Advantages: CLD is very compact with an invariant resolution color depiction for high-speed image recovery. It is intended to represent spatial color representation effectively. CLD can be utilized for a wide range of similarity-based recovery, visualization, and information selection. It is particularly useful for applications focused on spatial-structure extraction, such as sketch-based recovery and recognition of video segments. The sketch-based recovery is regarded as a very useful feature as it can provide very user-friendly environments, particularly when the search is sufficiently fast.

Limitations: Difficult to recognize spatial information.

3.5 Color Structure Descriptor

The Color Structure Descriptor (CSD) is based on color histograms but intends to use a small structuring window to locate localized color distributions [20–23]. The color configuration descriptor is connected to the Hue-Min-Max-Difference (HMMD) color model to ensure interoperability.

CSD describes an image by both the image's color distribution and the color's regional spatial structure. The additional color structure knowledge makes the descriptor responsive to specific image characteristics that are blind to the color histogram. Figure 3.10 demonstrates this with image pair, each consisting of two iso-colored planes, one black and one gray. The gray iso-colored plane on the left is heavily structured while the one on the right is less structured. An iso-color plane structure is a degree to which its pixels are grouped relative to the size of a similar structuring feature.

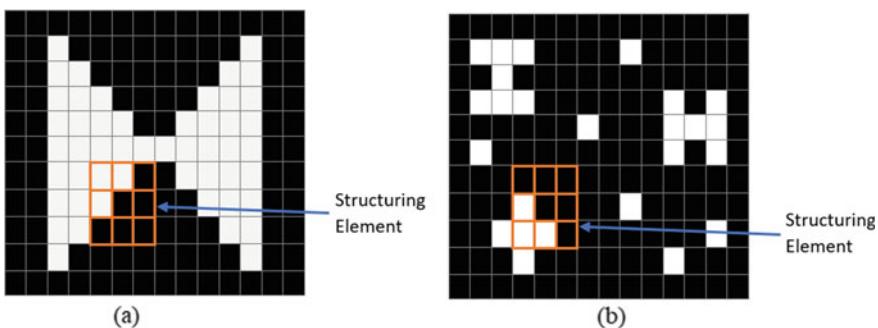


Fig. 3.10 The iso-colored plane, **a** heavily structured, **b** less structured

The CSD is similar in form to a color histogram as shown in Eq. 3.13, although different semantically. In particular, the CSD is a single-dimensional array of 8-bit quantum values.

$$\text{CSD} = \bar{h}_s(p), p \in \{1, \dots, P\} \quad (3.13)$$

where P is selected from the set $\{32, 64, 128, 256\}$ and s is the scale of the related square structuring component in the instance of, $s = 3^2$. The P bins (array elements) of \bar{h}_s are linked in an injective method to the P cells of the non-uniformly quantized HMMD color model.

The CSD is better described in terms of the histogram of the color structure. There is a three-step process to extract a CSD:

- i. A 256-bin CS histogram (i.e., collected) is derived from an image depicted in the color space of the 256 cells quantized HMMD. If the image is in a different color model then it has to be translated to HMMD and re-quantized before extraction.
- ii. When $N < 256$ is required, bins will be combined to get a N -bin CS histogram.
- iii. By the color occurrence statistics in typical consumer images, the amplitudes of each of the N bins are non-linear quantized.

There are several advantages and limitations of this method. Some of them are discussed here.

Advantages: Spatial information preserved.

Limitations: Sensitive to noise, orientation, and scale.

3.6 Summary

In this chapter, some MPEG-7 color feature extraction techniques are discussed such as dominant color descriptor, scalable color descriptor, group of images/group of frames color descriptor, color layout descriptor, and color structure descriptor. In the HSV color space, the scalable color descriptor is a color histogram represented by a Haar transform. It is beneficial for the image-to-image matching and restoring depending on the color feature. The group of images/group of frames descriptor is an expansion of the scalable color descriptor to a group of frames in a video or image array. This descriptor is based on combining the color characteristics of the images or video frames in the query. The color layout descriptor describes the spatial layout on a grid overlaid on an area or image of the representative colors. Representation is based upon the discrete cosine transform coefficients. The color structure descriptor is also built on color histograms but aims to use a small structuring window to determine the localized color distribution. The color structure descriptor is linked to the HMMD color space to ensure interoperability.

References

1. Babu C, Shanthi M (2012) An efficient image compression technique by extracting features in the image. *Bonfring Int J Ad Image Process* 2:95–99
2. Histograms CII (2013) Bi-level classification of color indexed image histograms for content based image retrieval. *J Comput Sci* 9(3):343–349. <https://doi.org/10.3844/jcssp.2013.343.349>
3. Mohapatra DP, Patnaik S (eds) (2013). Intelligent computing, networking, and informatics: proceedings of the international conference on advanced computing, networking, and informatics. Springer Science & Business Media, India, vol 243, June 2013
4. Pavithra LK, Sharmila TS (2019) An efficient seed points selection approach in dominant color descriptors (DCD). *Cluster Comput* 22(4):1225–1240. <https://doi.org/10.1007/s10586-019-02907-3>
5. Czapiewski P, Forczmański P, Okarma K, Frejlichowski D, Hofman R (2016) Clothing similarity estimation using dominant color descriptor and SSIM index. In: Proceedings of the 9th international conference on computer recognition systems CORES 2015. Springer, Cham, pp 491–500
6. Rejeb IB, Ouni S, Zagrouba E (2017) Image retrieval using spatial dominant color descriptor. In: 2017 IEEE/ACS 14th international conference on computer systems and applications (AICCSA). Hammamet, pp 788–795. <https://doi.org/10.1109/aiccsa.2017.127>
7. Algur SP, Ayachit NH (2016) A channelized binning method for extraction of dominant color pixel value. arXiv preprint [arXiv:1605.08856](https://arxiv.org/abs/1605.08856)
8. Georgescu FA, Răducanu D, Datcu M (2017) New MPEG-7 scalable color descriptor based on polar coordinates for multispectral earth observation image analysis. *IEEE Geosci Remote Sens Lett* 14(7):987–991. <https://doi.org/10.1109/lgrs.2017.2656822>
9. Chantharinthon P, Panthuwadeethorn S, Phimoltares S (2017) Robust video editing detection using scalable color and color layout descriptors. In: 2017 14th international joint conference on computer science and software engineering (JCSSE). Nakhon Si Thammarat, pp 1–6. <https://doi.org/10.1109/jcsse.2017.8025923>
10. Imran M, Hashim R, Irtaz A, Mahmood A, Abdullah U (2016) Class wise image retrieval through scalable color descriptor and edge histogram descriptor. *Int J Adv Appl Sci* 3(12):32–36. <https://doi.org/10.21833/ijaas.2016.12005>
11. Dewi AF, Arnia F, Muharar R (2017) Effectiveness of MPEG-7 color features in clothing retrieval. *Bull Electr Eng Inf* 6(2):166–173. <https://doi.org/10.11591/eei.v6i2.619>
12. Aghamaleki JA, Behrad A (2018) Detecting double compressed MPEG videos with the same quantization matrix and synchronized group of pictures structure. *J Electron Imaging* 27(1):013031. <https://doi.org/10.1117/1.JEI.27.1.013031>
13. Sikora T (2018) MPEG digital video coding standards. In: Compressed video over networks. CRC Press, pp 45–88
14. Busari KA, Adamu ID, Ali MH, Busari TA, Afolabi MA, Kashimbila MM (2018) Simulation of the effect of bit assignment to multi-hypothesis pictures per group for motion compensated video compression (MCVC). *Bayero J Pure Appl Sci* 11(1):1–7. <https://doi.org/10.4314/bajopas.v1i1.1S>
15. Pandey S, Dwivedy P, Meena S, Potnis A (2017) A survey on key frame extraction methods of a MPEG video. In: 2017 international conference on computing, communication and automation (ICCCA). Greater Noida, pp 1192–1196. <https://doi.org/10.1109/cca.2017.8229979>
16. Chantharinthon P, Panthuwadeethorn S, Phimoltares S (2017) Robust video editing detection using scalable color and color layout descriptors. In: 2017 14th IEEE international joint conference on computer science and software engineering (JCSSE). Nakhon Si Thammarat, pp 1–6. <https://doi.org/10.1109/jcsse.2017.8025923>
17. Tsai HH, Chang BM, Lo PS, Peng JY (2016) On the design of a color image retrieval method based on combined color descriptors and features. In: 2016 first IEEE international conference on computer communication and the internet (ICCCI). Wuhan, pp 392–395. <https://doi.org/10.1109/cci.2016.7778950>

18. Watcharasing J, Thiralertphanich T, Panthuwadeethorn S, Phimoltares S (2019) Classification of fruit in a box (FIB) using hybridization of color and texture features. In: 2019 16th international joint conference on computer science and software engineering (JCSSE). Chonburi, pp 303–308. <https://doi.org/10.1109/jcsse.2019.8864164>
19. Lee YH, Bang SI (2019) Improved image retrieval and classification with combined invariant features and color descriptor. *J Ambient Intell Humanized Comput* 10(6):2255–2264. <https://doi.org/10.1007/s12652-018-0817-0>
20. Abdelali AB, Hannachi M, Touil L, Mtibaa A (2017) Adequation and hardware implementation of the color structure descriptor for real-time temporal video segmentation. *J Real-Time Image Proc* 13(4):739–758. <https://doi.org/10.1007/s11554-014-0438-6>
21. Florea C, Toca C, Gieseke F (2017) Artistic movement recognition by boosted fusion of color structure and topographic description. In: 2017 IEEE winter conference on applications of computer vision (WACV). Santa Rosa, pp 569–577. <https://doi.org/10.1109/wacv.2017.69>
22. Siswantoro J, Arwoko H, Widiasri M (2019) Image based indonesian fruit recognition using MPEG-7 color structure descriptor and k-Nearest neighbor. In: International conference on informatics, technology, and engineering 2019. The Anvaya Resort in Denpasar
23. Yu H, Yang W, Xia GS, Liu G (2016) A color-texture-structure descriptor for high-resolution satellite image classification. *Remote Sens* 8(3):259. <https://doi.org/10.3390/rs8030259>

Chapter 4

Other Image Color Features



4.1 Color Coherence Vector

Color Coherence Vector (CCV) is a more complicated technique than color histogram. Conceptually, the coherence of color can be described as the degree to which pixels of that color are part of large regions of a similar color [1–3]. These essential regions can be termed as coherent regions, and they are efficient in image characterization.

The images shown in Fig. 4.1 have alike color histograms, regardless of the difference in their appearances. In both images, the number of green colors is roughly the same. The green pixels are broadly distributed in Fig. 4.1a, whereas the green pixels construct a single coherent region in the Fig. 4.1b.

The coherence measure categorizes pixels as either incoherent or coherent. Coherent pixels are a part of some substantial connecting region, whereas incoherent pixels are not. For each color in the image, CCV signifies this classification. CCV's avert coherent pixels matching with incoherent pixels in an image region. This permits fine discrepancies that cannot be achieved with color histograms.

The initial phase of measuring a CCV is close to a color histogram estimation. First, blur the image moderately by replacing the adjacent pixel values in a small local area with the average value. It removes slight variations among adjacent pixels. Next discrete the color space, so that the image includes only n distinct colors.

The next phase is to categorize the pixels either as coherent or incoherent in a specified bucket of color. A coherent pixel, though not an incoherent pixel, is part of a large group of pixels of the same color. The groups of pixels are identified by evaluating similar components. A Connected Component (CC) is a maximal pixel set so that for any two pixels $P_1, P_2 \in CC$, there is a pathway in CC between P_1 and P_2 . Note that CC is computed within a specified discretized bucket of color. This efficiently segments the image constructed on the discretized color space.

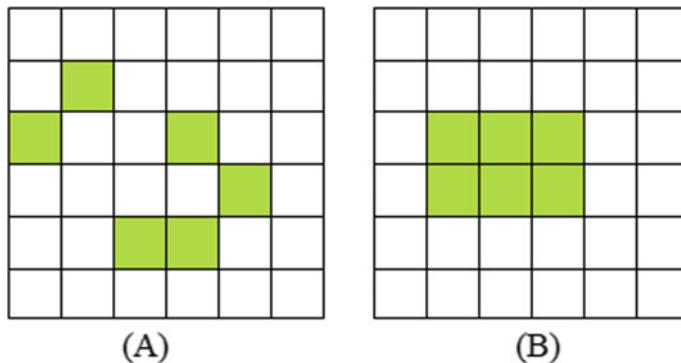


Fig. 4.1 Two images having the same color histogram

When this is finished, each pixel should form precisely one CC. Pixels are either categorized as coherent or incoherent, based on the size of the CC in pixels. A pixel is consistent if a fixed value λ equals the size of its CC; otherwise, the pixel is incoherent.

Some of the pixels with a specified discretized color will be coherent, and some will be incoherent. Assume that CO_i and ICO_i are the number of coherent and pixel incoherents of the i -th discretized color. So, the total number of pixels with that color is $CO_i + ICO_i$, and the color histogram can be represented as follows:

$$\langle CO_1 + ICO_1, \dots, CO_n + ICO_n \rangle$$

But to compute the color coherence, for each discretized color, the coherence pair is calculated as $(CO_i + ICO_i)$.

The color coherence vector for the image can be represented as follows:

$$\langle (CO_1, ICO_1), \dots, (CO_n, ICO_n) \rangle$$

An example of the computation of a CCV is demonstrated here. Assuming $\lambda = 10$ and the image contains 36 colors.

21	18	27	24	11	15
22	28	11	25	12	16
23	19	12	14	13	35
24	29	23	15	32	36
25	21	16	30	33	37
26	22	17	31	34	38

The color space is discretized so that bin 1 contains color intensities of 10–19, bin 2 contains 20–29, and bin 3 contains 30–39. The output of discretization is as follows:

2	1	2	2	1	1
2	2	1	2	1	1
2	1	1	1	1	3
2	2	2	1	3	3
2	2	1	3	3	3
2	2	1	3	3	3

Then CC is computed. Components are labeled with P , Q , and R . The output of CC is as follows:

Q	P	Q	Q	P	P
Q	Q	P	Q	P	P
Q	P	P	P	P	R
Q	Q	Q	P	R	R
Q	Q	P	R	R	R
Q	Q	P	R	R	R

The CC table will be as follows:

Label	P	Q	R
Color	1	2	3
Size	14	14	9

The components P and Q have more than λ pixels and the components R have less than λ pixels. Thus, the pixels in P and Q will be classified as coherent, whereas the pixels in R will be classified as incoherent.

This technique has several advantages and limitations. Some of them are discussed here.

Advantages: This method considers the color spatial distribution information between pixels in its coherent component.

Limitations: Coherent pixels depict the pixels that are in the connected components within the image. But if all these components are merged into one component, then the generated component will have the same number of pixels as in the connected components.

For example, assuming λ equals to 8. Figure 4.2 illustrates the limitation. Even though the appearance of these two images (4.2a, b) is different, they have the same CCV.

Another limitation of this method is that the locations of these connected components are relative to each other. For example, in Fig. 4.3a–c, images have the same CCV with a different appearance.

Additionally, CCV suffers from high dimensionality and high computational cost.

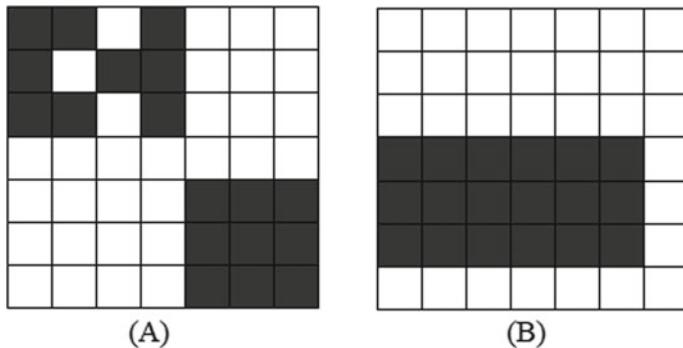


Fig. 4.2 Images with same CVV and different appearance

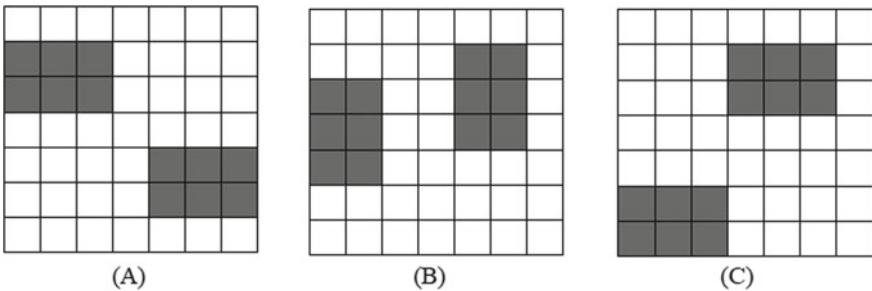


Fig. 4.3 Image with the same CVV with different locations of the connected component

4.2 Color Moments

Color moments can be utilized to distinguish images according to their color characteristics. Those moments once measured, deliver a color similarity comparison between images. Then, these similarity values can be correlated with image values stored in a database for activities such as image characterization [4–6].

The foundation of color moments lies in the concept that color distribution in an image can be viewed as a distribution of probability. Several unique moments characterize probability distributions. Consequently, if the color of an image matches a certain distribution of probability, then moments of that distribution may be utilized as color characteristics of the image.

A color can be demarcated by 3 or more values dependent on the color channels. Moments are computed for each of these channels in an image. Let, the i -th color channel at the j -th image pixel is represented as P_{ij} and N is the total number of pixels in the image. The color moments are defined using Eqs. 4.1 to 4.5.

- a. **Mean:** Mean or moment1 is the average of the color value in an image.

$$M_i = \sum_{j=1}^N \frac{1}{N} P_{ij} \quad (4.1)$$

- b. **Standard Deviation:** Standard deviation or moment2 is the square root of the variance of the color distribution.

$$SD_i = \sqrt{\frac{1}{N} \sum_{j=1}^N (P_{ij} - M_i)^2} \quad (4.2)$$

- c. **Skewness:** Skewness or moment3 is the degree of asymmetry in the color distribution.

$$SK_i = \sqrt[3]{\frac{1}{N} \sum_{j=1}^N (P_{ij} - M_i)^3} \quad (4.3)$$

- d. **Kurtosis:** Kurtosis or moment4 delivers information about the shape of the color distribution.

$$K_i = \sqrt[4]{\frac{1}{N} \sum_{j=1}^N (P_{ij} - M_i)^4} \quad (4.4)$$

- e. **Higher order color moments:** These moments are not usually the part of color moments in an image retrieval task but can deliver meaningful information in the retrieval task. A function can be formed by using moment1, moment2, and moment3 which is the sum of the weighted difference between the moments as shown in Eq. (4.5).

$$R_{(I,T)} = \sum_{i=1}^c w_{i1} |M_i^I - M_i^T| + w_{i2} |SD_i^I - SD_i^T| + w_{i3} |SK_i^I - SK_i^T| \quad (4.5)$$

where I and T are the two images to be compared, c is the number of image color channels, and w_i is the weight of each moment and is user-specified. Image pairs can be ranked based on their R values. Images with smaller values are ranked higher and considered more similar than those with a lower rank and higher R values.

It should be recalled that the R value is a function of resemblance and not a metric. A comparison of two different pairs of distributions is possibly the result of the same R value. In reality, this contributes to the collection of false positives along with, potentially, truly similar images.

There are some advantages and limitations of this technique. Some of them are discussed here.

Advantages: No need to store complete color information. Thus, fast image recovery can be expected by using this method. Also, the first three-color moments can be compared to each other as they have the same units.

Limitations: This method is not enough to describe all colors. Additionally, it does not provide any spatial information of the color distribution.

4.3 Color Co-occurrence Matrix

Color Co-occurrence Matrix (CCM) calculates the spatial color distribution features in an image, instead of just the co-occurrence details of the gray pixel values. CCM is simply an adaptation of the gray-level concurrence matrix, which is widely utilized to extract texture features from a color image [7–9]. Let $p(p = C_1, C_2, C_3)$ and $q(q = C_1, C_2, C_3)$ be two of the three color channels, and $\text{CCM}_{p,q}$ is the CCM between the color channels p and q of the color image. $\text{CCM}_{p,q}(i, j)$ is utilized to measure the number of times every pixel value of color channel p is equal to i and color channel q is equal to j at the same position and can be represented using Eq. (4.6).

$$\text{CCM}_{p,q}(i, j) = \sum_x \sum_y \begin{cases} 1 & p(x, y) = i \text{ and } q(x, y) = j \\ 0 & \text{otherwise} \end{cases} \quad (4.6)$$

Despite the human vision system's strong cognitive ability, simulating human being's visual perception mechanism is useful for achieving good visual processing outcomes. As stated, there are certain limitations to the visual quality of humans. For example, only the object's most salient features are essential for visual perception. Therefore, the HSI color space must be quantized to a limited number of color levels to enable the corresponding extraction of the texture feature, while preserving some distinctive color details and eliminating the insignificant information. It shows that for the following visual attribute extraction, this quantization process is most analogous to the human visual paradigm and can significantly reduce the variance of the color.

From an HSI color image, six CCMs ($\text{CCM}_{H,H}$, $\text{CCM}_{H,S}$, $\text{CCM}_{H,I}$, $\text{CCM}_{S,S}$, $\text{CCM}_{S,I}$, $\text{CCM}_{I,I}$) can be derived from the collective combination of different color channels. For example, one of the CCMs in the HSI color model ($\text{CCM}_{S,I}$) is shown in Fig. 4.4.

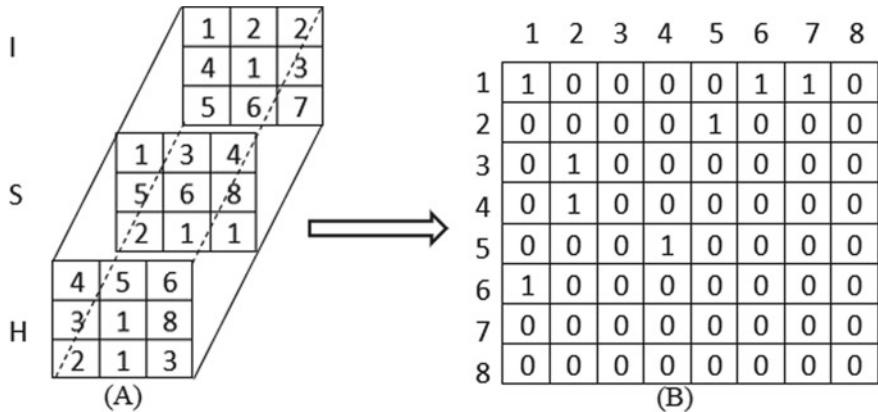


Fig. 4.4 a HSI pixel value representation, b corresponding CCMs presentation

Since CCMs are very reactive to substantial differences in spatial resolution when calculating the co-occurrence of the pixels, it is, therefore, important to normalize these matrices to decrease this sensitivity (each CCM is divided by the total of its elements, so that the amount of the elements of each CCM is 1).

A series of second-order statistics depicting the image texture features can be measured based on CCM. Haralick presented 14 second-order statistics from the normalized CCMs to characterize texture features. They are angular second moment, contrast, correlation, variance, inverse difference moment, entropy, sum average, sum variance, sum entropy, difference entropy, difference variance, information measures of correlation 1 (IMC1), information measures of correlation 2 (IMC2), and maximum correlation coefficient (MCC).

Here the following notations are used: $P(i, j)$ is the element (i, j) of a normalized CCM; G is the total number of color component levels used, μ is the mean value of ..; i and j represent the number of rows and columns of P ; and μ_x , μ_y , σ_x and σ_y are the means and standard deviations of P_x and P_y . $P_x(i)$ is the i -th entry in the marginal-probability matrix attained by summing the rows of $P(i, j)$:

$$P_x(i) = \sum_{j=0}^{G-1} P(i, j) \quad (4.7)$$

$$P_y(j) = \sum_{i=0}^{G-1} P(i, j) \quad (4.8)$$

$$\mu_x = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} i P(i, j) = \sum_{i=0}^{G-1} i P_x(i) \quad (4.9)$$

$$\mu_y = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} j P(i, j) = \sum_{j=0}^{G-1} j P_y(j) \quad (4.10)$$

$$\sigma_x^2 = \sum_{i=0}^{G-1} (i - \mu_x)^2 \sum_{j=0}^{G-1} P(i, j) = \sum_{i=0}^{G-1} (P_x(i) - \mu_x(i))^2 \quad (4.11)$$

$$\sigma_y^2 = \sum_{j=0}^{G-1} (j - \mu_y)^2 \sum_{i=0}^{G-1} P(i, j) = \sum_{j=0}^{G-1} (P_y(j) - \mu_y(j))^2 \quad (4.12)$$

- a. **Angular Second Moment (ASM):** ASM or energy is used to measure the uniformity of an image. ASM is high if the image has homogeneous pixel intensity distribution or if the pixels are very comparable as in homogeneous image matrix very few pixels have larger magnitudes. In contrary, if the image matrix comprises many small magnitudes, then the ASM will have a smaller value. This feature can be denoted by Eq. (4.13).

$$\text{ASM} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{P(i, j)\}^2 \quad (4.13)$$

- b. **Contrast:** This feature is used to measure the variation in the local pixel intensities over the entire image. For a flat image, the value of this measure is 0. This feature can be signified by Eq. (4.14).

$$\text{Contrast} = \sum_{i=1}^G \sum_{j=1}^G |i - j|^2 P(i, j) \quad (4.14)$$

- c. **Correlation:** This is a measure of linear dependency between pixels in an image. The value is -1 and 1 depending on a fully negative or positive correlated image. If the image is a continuous image, the correlation value is NaN. This feature can be denoted by Eq. (4.15).

$$\text{Correlation} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{\{i \times j\} \times P(i, j) - \{\mu_x \times \mu_y\}}{\sigma_x \times \sigma_y} \quad (4.15)$$

- d. **Variance or Sum of Squares:** This measure depicts the pixel value dispersion with respect to mean. This feature can be signified by Eq. (4.16).

$$\text{Variance} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - \mu)^2 P(i, j) \quad (4.16)$$

- e. **Inverse Difference Moment or Local Homogeneity (IDM):** This feature measures the homogeneity or smoothness of the image pixel distribution. It is inversely related with the contrast feature. If contrast is large then IDM will be small and vice versa. This feature can be signified by Eq. (4.17).

$$\text{IDM} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1 + (i - j)^2} P(i, j) \quad (4.17)$$

- f. **Entropy:** This indicates the degree of disorder among pixels in the image. It is inversely related with the ASM. Inhomogeneous scenes have low entropy, whereas a homogeneous scene has high entropy. This feature can be denoted by Eq. (4.18).

$$\text{Entropy} = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i, j) \times \log(P(i, j)) \quad (4.18)$$

Other co-occurrence features that can be obtained from CCM are denoted from Eqs. (4.19) to (4.26).

$$\text{Sum_Average} = \sum_{f=0}^{2G-2} f P_{x+y}(f)$$

Where,

$$P_{x+y}(f) = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i, j) \quad (4.19)$$

$$\text{Sum_Variance} = \sum_{f=0}^{2G-2} (f - \text{Sum_Average})^2 P_{x+y}(f) \quad (4.20)$$

$$\text{Sum_Entropy} = - \sum_{f=0}^{2G-2} P_{x+y}(f) \log(P_{x+y}(f)) \quad (4.21)$$

$$\text{Difference_Entropy} = - \sum_{f=0}^{G-1} P_{x+y}(f) \log(P_{x+y}(f)) \quad (4.22)$$

$$\text{Difference_Variance} = \left[\sum_{f=0}^{G-1} f^2 P_{x-y}(f) \right] - \text{Difference_Average}^2$$

Where,

$$P_{x-y}(f) = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i, j) \quad (4.23)$$

$$\text{IMC1} = \frac{\text{Entropy} - HXY1}{\max\{HX, HY\}}$$

Where,

$$HXY1 = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i, j) \log\{P_x(i)P_y(j)\}$$

$$HXY2 = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P_x(i)P_y(j) \log\{P_x(i)P_y(j)\}$$

$$HX = \text{Entropy}(P_x)$$

$$HY = \text{Entropy}(P_y) \quad (4.24)$$

$$\text{IMC2} = \sqrt{1 - \exp(-2.0(HXY2 - \text{Entropy}))} \quad (4.25)$$

$$\text{MCC} = \sqrt{\text{Second_Largest_Eigenvalue_of_}A}$$

Where,

$$A(i, j) = \sum_{f=0}^{G-1} \frac{P(i, f)P(j, f)}{P_x(i)P_y(j)} \quad (4.26)$$

There are some advantages and limitations of this technique. Some of them are discussed here.

Advantages: CCM is really simple to implement. This method has a great performance in terms of complexity and time of processing. For historical record images, including a simple text font and graphics, the features offered by the CCM system would be a good option since the tool is easy to use and fast.

Limitations: One of the limitations of the CCM method is the large matrix dimensionality and strong correlation. Because of their large dimensionality, the CCMs are very sensitive to the image texture sample sizes that are analyzed. That is why the number of gray levels is also high. It needs a large amount of memory for processing. However, the capabilities of the CCM are not suitable for historical record images with a significant quantity of noise. Lastly, for the purposes of image detection, distance d selection can be crucial. The meaning of d must be broad enough to provide an image surface pattern, but still low enough to maintain the local character of spatial dependency.

4.4 Color Contrast Occurrence Matrix

The color contrast occurrence matrix is focused on the measurement of color variation. The color disparity is represented using a visual variance presented in CIELab and two angles describing the chromaticity and the darker or brighter direction to be related to the human perception [10].

The feature reflects the likelihood of having a specific color difference separated by a spatial vector between two pixels. This spatial vector is classically described by a distance and orientation. The color variation is represented by a visual interval and two angles in CIELab color space. The first angle on the ab plane is specified which is the color difference's chromatic orientation. The second angle is specified between the vector of color difference and the ab plane characterizing a darker or lighter difference represented by the vector of difference.

There are some advantages and limitations of this technique. Some of them are discussed here.

Advantages: Regarding angular spatial relationships and distance, the co-occurring pairs of pixels can be connected spatially in different orientations, as in observing the association between two pixels at a time. As a consequence, the combination of intensities of pixels and their locations is portrayed.

Limitations: The matrix is prone to the rotation. Defining pixel relationships by altering displacement vectors (distance to the neighboring pixel: 1, 2, 3, ...) and directions (orientation angle of an offset: 0°, 45°, 90°, 135°) results in varying co-occurrence distributions from the same sample image.

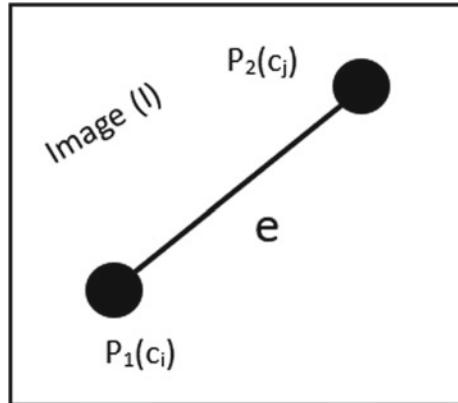
4.5 Color Correlogram

The color correlogram has been suggested to classify not only the pixel color distributions but also the spatial correlation of color pairs. Color correlogram is a color-dependent function that describes how distance varies in the spatial correlation of color pairs. Figure 4.5 shows the concept of color correlogram [11–13].

Let I is an image of size $m \times m$. The image colors are quantized into m distinct colors, c_1, c_2, \dots, c_m and two considering image pixels are represented as $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. A color correlogram is a table defined by pairs of colors, where the e -th entry for (c_i, c_j) determines the probability of having a color pixel c_j at a distance e from a color pixel c_i in the image. Let a distance d between two considering pixel is computed by utilizing L_∞ -norm distance as shown in Eq. (4.27).

$$|P_1 - P_2| = \max\{|x_1 - x_2|, |y_1 - y_2|\} \quad (4.27)$$

Fig. 4.5 Concept of the color correlogram



Then, the color correlogram of image I is defined as shown in Eq. (4.28).

$$CR_{c_i, c_j}^e = \Pr_{P_1 \in I_{c_i}, P_2 \in I} [P_2 \in I_{c_j} | |P_1 - P_2| = e] \quad (4.28)$$

where $|P_1 - P_2|$ is the distance between pixels P_1 and P_2 , $(c_i, c_j) \in \{c_1, \dots, c_m\}$ and $e \in \{1, \dots, d\}$.

If all the probable combinations of color pairs are considered, the size of the color correlogram will be very large ($O(m^2d)$); hence, a shortened version of the feature named the color auto-correlogram is frequently utilized in its place. The color auto-correlogram only depicts the spatial correlation between alike colors and therefore decreases the dimension to ($O(md)$) and is demarcated as shown in Eq. (4.29).

$$CAC_i^e = CR_{c_i, c_j}^e \quad (4.29)$$

A large value of d will result in costly computing requiring huge volume. A small d may undermine the feature's efficiency, but the primary concern is the quantized m colors. Usually, m is selected from 64 to 256 quantized colors, which for correlogram is a pretty large number. The color auto-correlogram offers the best performance in retrieval compared to the color histogram but is also the costliest in computing because of its dimensionality. In order to minimize computing costs, a color correlogram is therefore suggested; else, auto-correlogram can be used for faster microprocessors.

There are some advantages and limitations of this technique. Some of them are discussed here.

Advantages: This method is robust in accepting large alterations in the scene appearance produced by variations in observing locations, partial occlusions, variations in the background scene, and enlargement that cause radical alterations in shape. As this the spatial correlation of colors in an image, it can be captured by this method and it is efficient in discriminating images. It, therefore, corrects the

major limitations of the classical histogram technique. The correlogram can also be calculated efficiently.

Limitation: This method suffers from computation complexity.

4.6 Reference Color Table Method

A collection (list) of reference colors is specified in the reference color table method. A range of colors is chosen in such a manner that all of the needed colors are perceptually covered up. The histogram for that smaller set of colors is determined for each image. For this reason, the nearest color defined in the table is assigned to each pixel in the color image. Then the pixel histogram is determined with the currently allocated colors. If the chosen reference color table is appropriate, the “new image” generated by selecting the nearest color from the table will be similar to the original image [14–16].

This procedure can be viewed as an alternative solution to the distance approach and the histogram intersection. This method eliminates the complexity of histogram matching and preserves the distance method’s speed and robustness.

In this technique, a group (table) of reference colors are defined. The selection of the group of colors is based on the colors in the database images so that the database image colors are almost covered perceptually. Table 4.1 shows the example of a reference color table for a particular database.

For every image in the database, a histogram for this set of colors is computed. For this reason, the color table classifies every pixel in the color image against the colors and assigns the closest color. The closest color in the reference table is determined by using a basic distance measure method. The histogram of the pixels is then measured with the newly allocated colors. If the selected color table is perfect, then the newly created image would perceptually be similar to the original image after assigning the

Table 4.1 Reference color table

Color	R	G	B
Black	0	0	0
Dark blue	0	0	128
Blue	0	0	255
Dark green	0	128	0
Green	0	0	255
Turquoise	0	128	128
Sky blue	0	128	255
Spring green	0	255	128
White	255	255	255
Red	255	0	0
Gray	128	128	128

closest color from the table. So, the color feature specified for this approach is this reduced color histogram dependent on reference table colors.

The reference color histogram feature mentioned is first computed for a particular query image. The image is then compared against all images in the database to get the measure of the similarity with each image in the database. The query image is assigned to the dataset class with the maximum similarity.

There are some advantages and limitations of this technique. Some of them are discussed here.

Advantages: Efficient in image retrieval. Also, computationally fast as compared to the traditional histograms.

Limitations: This technique needs a predetermined collection of reference colors that can roughly cover all of the application's colors. While in some cases this criterion may be fulfilled. But where there are continuing additions/deletions to the database images and where information of colors in the images is not accessible in advance, such a methodology will not yield good results. This method needs a representative sample of all dataset images to select the color table of reference. Such a priori knowledge cannot be obtained in many cases such as a trademark dataset.

4.7 Summary

This chapter presents different image color features like color coherence vector, color moments, color co-occurrence matrix, color contrast occurrence matrix, color correlogram, and reference color table method. The performance of these features depends on the color distribution of images. The color coherence vector offers a better comparison rate for images that have widely scattered colors. Any color distribution can be handled by using color moments. Additionally, as most of the information is determined on the low-order moments, only the first, second, and third moments (mean, variance, and skewness) are enough color feature representation for many applications. The color co-occurrence matrix can compute the texture of the color image. As co-occurrence matrices are usually sparse and large, several metrics of the matrix are frequently taken to obtain a more beneficial set of features. A color correlogram (henceforth correlogram) shows how the spatial correlation of pairs of colors varies with distance. The reference color table method is very efficient in many applications where there is no need to update the dataset images. But if continuous modification is needed in the dataset images then it is better to select other color feature extraction techniques for image classification.

References

1. Thoopsamut P, Limthanmaphon B (2019) Handwritten signature authentication using color coherence vector and signing behavior. In: Proceedings of the 2019 2nd international conference on information science and systems, pp 38–42. <https://doi.org/10.1145/3322645.3322682>
2. Li Y, Liu M (2018) Aerial image classification using color coherence vectors and rotation & uniform invariant LBP descriptors. In: 2018 IEEE 3rd advanced information technology, electronic and automation control conference (IAEAC), pp 653–656. <https://doi.org/10.1109/iaeac.2018.8577881>
3. Moslehi M, de Barros FP (2019) Using color coherence vectors to evaluate the performance of hydrologic data assimilation. *Water Resour Res* 55(2):1717–1729. <https://doi.org/10.1029/2018WR023533>
4. Mohseni SA, Wu HR (2019) A color moments-based system for recognition of emotions induced by color images. In: 2019 international conference on image and vision computing New Zealand (IVCNZ), pp 1–6. <https://doi.org/10.1109/ivcnz48456.2019.8961020>
5. Deng Y, Yu Y (2019) Self-feedback image retrieval algorithm based on annular color moments. *EURASIP J Image Video Process* 2019(1):1–13. <https://doi.org/10.1186/s13640-018-0400-9>
6. Ghosh A, Sarkar A, Ashour AS, Balas-Timar D, Dey N, Balas VE (2015) Grid color moment features in glaucoma classification. *Int J Adv Comput Sci Appl* 6(9):1–14
7. Kotyk T, Dey N, Ashour AS, Drugarin CVA, Gaber T, Hassani AE, Snasel V (2016) Detection of dead stained microscopic cells based on color intensity and contrast. In: The 1st international conference on advanced intelligent system and informatics (AISI2015), November 28–30, 2015. Beni Suef, Egypt, pp 57–68. https://doi.org/10.1007/978-3-319-26690-9_6
8. Souza L, Ebigo A, Probst A, Messmann H, Papa JP, Mendel R, Palm C (2018) Barrett's Esophagus identification using color co-occurrence matrices. In: 2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI), pp 166–173. <https://doi.org/10.1109/sibgrapi.2018.00028>
9. Vanitha J, SenthilMurugan M (2017) An efficient content based image retrieval using block color histogram and color co-occurrence matrix. *Int J Appl Eng Res* 12(24):15966–15971
10. Jebali H, Richard N, Chatoux H, Naouai M (2018) Relocated colour contrast occurrence matrix and adapted similarity measure for colour texture retrieval. In: International conference on advanced concepts for intelligent vision systems, pp 609–619. https://doi.org/10.1007/978-3-030-01449-0_51
11. Malviya AV, Ladhae SA (2016) Pixel based image forensic technique for copy-move forgery detection using auto color correlogram. *Procedia Comput Sci* 79:383–390. <https://doi.org/10.1016/j.procs.2016.03.050>
12. Maheen JB, Aneesh RP (2019) Machine learning algorithm for fire detection using color correlogram. In: 2019 2nd international conference on intelligent computing, instrumentation and control technologies (ICICICT), vol 1, pp 1411–1418. <https://doi.org/10.1109/icicict46008.2019.8993246>
13. Liu Y, Li Y (2016) Particle filer tracking algorithm based on fusion of color correlogram and SIFT features. *Comput Digit Eng* 45(7)
14. Li Z, Tan Z, Cao L, Chen H, Jiao L, Zhong Y (2019) Directive local color transfer based on dynamic look-up table. *Sig Process Image Commun* 79:1–12. <https://doi.org/10.1016/j.image.2019.06.010>
15. Hogervorst MA, Toet A (2017) Improved color mapping methods for multiband nighttime image fusion. *J Imaging* 3(3):36. <https://doi.org/10.3390/jimaging3030036>
16. Chen X, Zhang Q, Lin M, Yang G, He C (2019) No-reference color image quality assessment: from entropy to perceptual quality. *EURASIP J Image Video Process* 2019(1):77. <https://doi.org/10.1186/s13640-019-0479-7>

Chapter 5

Applications of Image Color Features



Color is an important and the vital visual feature for image recognition. The use of image color is one of the most interesting issues in creation efficient content-based image retrieval. Color feature cannot be defined exactly as defining the likeness among color feature is difficult [1]. Therefore, two steps are needed in color-based image recovery, i.e., image color feature extraction and similarity computation among the extracted features. Some of the applications of image retrieval using color are described in this chapter.

5.1 Leaf Recognition

Although most leaves are green in color, there are some species whose striking non-green hues provide a cue to their quick identification. Plant leaves have three primary class pigments such as chlorophyll, carotenoids, and anthocyanins. The typical color of plant leaf is green due to the dominance of green pigments like chlorophyll a and chlorophyll b. Chlorophyll absorbs the red and blue light from the sunlight and reflects back the green light. However, pigments like carotenoids absorb the blue-green and blue light from the sunlight and the leaf appears yellow or yellow-orange to our eyes. A third class of pigments anthocyanins absorb blue, blue-green, and green light and the corresponding leaves appear red or purple to our eyes. Thus, although a large percentage of the leaves are green, there can be occasionally non-green leaves which can provide quick markers for their identification [2, 3]. Leaf color can range from true black to bright red and every shade of purple, maroon, and brown in between. There are many methods to recognize leaf images [4, 5]. One of the leaf recognition algorithms using color feature is as follows:



Fig. 5.1 Leaf sample images

- (1) First, different types of digital leaves are collected which differ in color. Figure 5.1 displays some samples of green and non-green leaf images which are collected from [6].
- (2) Then the collected images are splitted into testing and training sets.
- (3) Color histogram is used for the classification of leaf images. In the current study, a color image is split into R, G, B color channels and 256 bin histograms are computed from each channel. The color image is thereafter represented by a 256-element color feature vector generated by summing up all the three histograms [7]. Figure 5.2 illustrates the histograms of a sample leaf image.

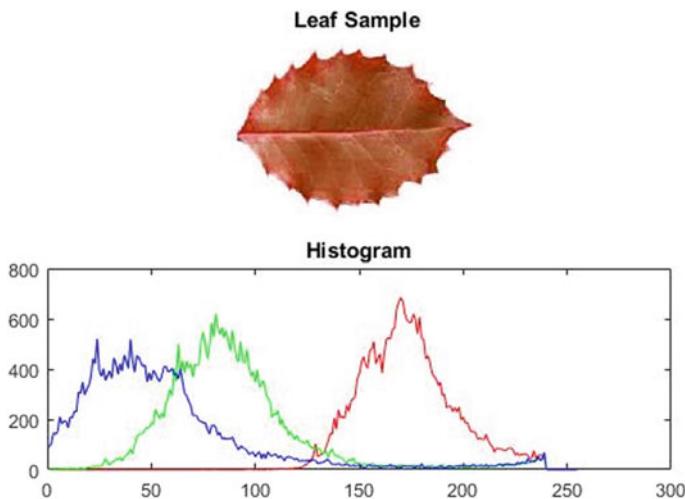


Fig. 5.2 Histogram of a sample image



Fig. 5.3 Sample fruit images

- (4) Euclidean distance method is used to compare training and testing images color histograms.
- (5) The test leaf image is characterized by the minimum Euclidean distance with the training images.

5.2 Fruit Recognition

There are several techniques to classify fruits utilizing color feature [8–11]. One of the fruit recognition algorithms utilizing color descriptor is as follows:

- (1) First, different types of fruit images are collected which differ in color. Figure 5.3 displays some sample images of fruit which is collected from [12].
- (2) Then the collected images are splitted into testing and training sets.
- (3) Color correlogram is used to extract color feature for fruit image recognition. Figure 5.4 shows some correlogram plots of some sample fruits.
- (4) Euclidean distance method is used to compare the correlogram vector between training and testing images.
- (5) The test fruit image is classified by the minimum Euclidean distance with the training images.

5.3 Flower Recognition

There are several flower recognition techniques using color feature [13–16]. One of the flower recognition algorithms utilizing color descriptor is as follows:

- (1) First flower images are collected which differ in color. Figure 5.5 displays some sample images of flower which are collected from [17].
- (2) Then the collected images are splitted into testing and training sets.
- (3) Then some pre-processing is done to extract only the flower portion from the image. Figure 5.6 shows one sample flower image taken for this demonstration. Steps of pre-processing are described as follows.

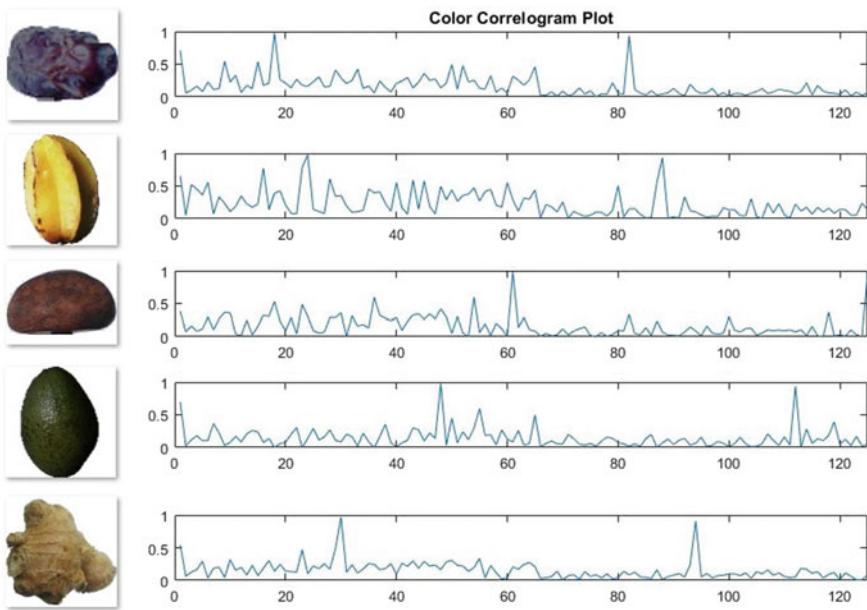


Fig. 5.4 Color correlogram plot of sample fruit images



Fig. 5.5 Sample flower images

- First, the image is converted to HSV color space from the RGB color space. Figure 5.7 shows the converted image.
- Then, the V (Value) channel (Fig. 5.8) data is extracted to create the mask for segmentation.
- The threshold value (T) for binarization is calculated from this V channel data using Eq. (5.1).

$$T = \text{mean}(V) + \text{std}(V) \quad (5.1)$$

Fig. 5.6 Sample flower image



Fig. 5.7 Image in HSV color space

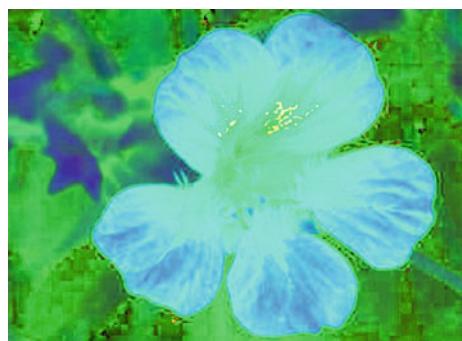


Fig. 5.8 V channel data



Mean (V) and Std (V) are the average and standard deviation of V channel data. Figure 5.9 shows the thresholded (binarized) image obtained from Fig. 5.8 using the threshold T .

- d. After that the holes of the binarized image are filled (Fig. 5.10).
- e. The largest connected component part is extracted from Fig. 5.10 to create the mask as shown in Fig. 5.11.
- f. The extracted flower portion using this mask is shown in Fig. 5.12.

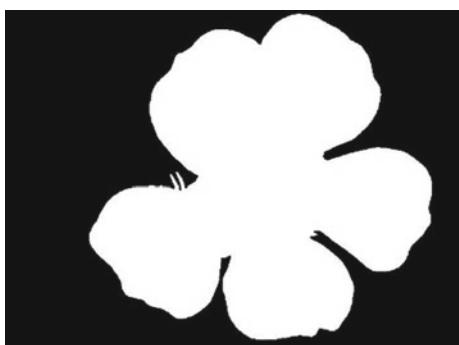
Fig. 5.9 Binarized image



Fig. 5.10 Output of hole filling



Fig. 5.11 Extracted largest connected component



- (4) Then color moments are extracted from every training and testing image.
- (5) Euclidean distance method is used to compare color moments between training and testing images.
- (6) The test flower image is classified by the minimum Euclidean distance with the training images.

Fig. 5.12 Extracted flower portion



5.4 Random Image Recognition

There are several techniques to identify random images utilizing color feature [18–21]. One of the random image recognition algorithms utilizing color descriptor is as follows:

- (1) First, some random images are collected which varies in color. Figure 5.13 displays some sample images that are collected from [22].
- (2) Then the collected images are splitted into testing and training sets.
- (3) The RGB images are converted to HSV color space and the hue channel is used for further processing as it contains the color information of the image.
- (4) Fuzzy color histogram is used to extract similar images from the dataset. For that, trapezoidal membership function is used to fuzzify the input data. Fifteen bins are used for this experimentation: (i) red (R), (ii) red-orange (RO), (iii) orange-brown (OB), (iv) orange-yellow (OY), (v) yellow (Y), (vi) yellow-green (YG), (vii) green (G), (viii) green-cyan (GC), (ix) cyan (C), (x) cyan-blue (CB), (xi) blue (B), (xii) blue-magenta (BM), (xiii) magenta (M), (xiv) magenta-pink (MP), and (xv) pink-red (PR). Figure 5.14 shows the structure of the membership function used.
- (5) Fuzzy color histogram is generated from each training and testing image. Figure 5.15 illustrates the generated histogram from some sample images.

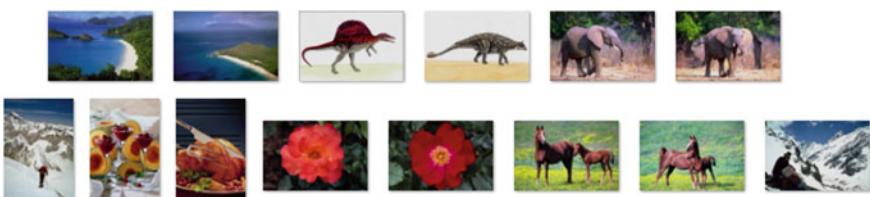


Fig. 5.13 Samples of random images

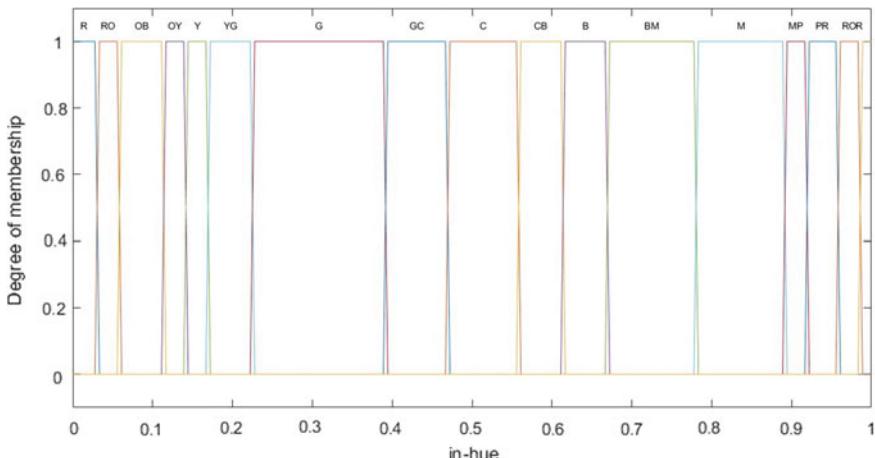


Fig. 5.14 Input membership function structure

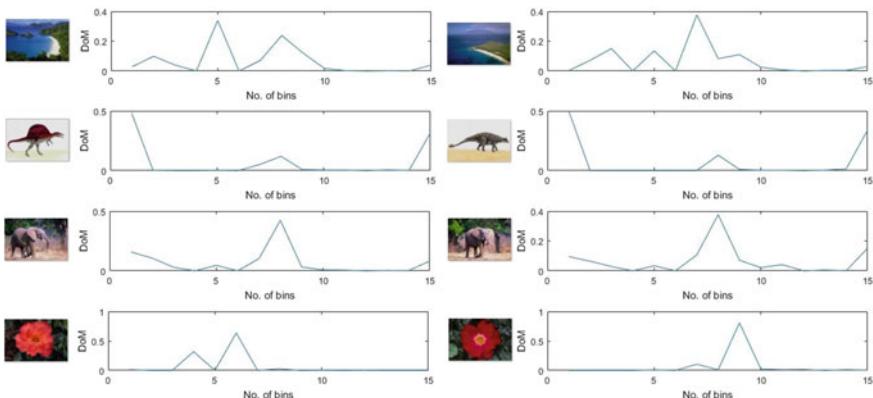


Fig. 5.15 Fuzzy color histogram of some sample images

- (6) Euclidean distance is used to check the likeness between the training and testing fuzzy histograms.
- (7) The test image is classified by the minimum Euclidean distance with the training images.

5.5 Edge Detection of Color Image

There are many ways to detect edge from color image [23–26]. One of the algorithms for edge detection of color image is as follows:

- (1) First some images are collected. For this experiment, RGB images are used. Figure 5.16 displays the sample of color image that is collected from [27].
- (2) Extract red, green, and blue channels from the RGB image. Figure 5.17 shows the pictorial depiction of three abovesaid channels.
- (3) Four directional convolutional masks are applied to red, green, and blue channels to extract pixels in the direction of 0° , 45° , 90° , and 135° , respectively. Then, the three masked outputs (obtained from three channels) are combined to get 0° , 45° , 90° , and 135° edge images. Figure 5.18 illustrates the four masked output.

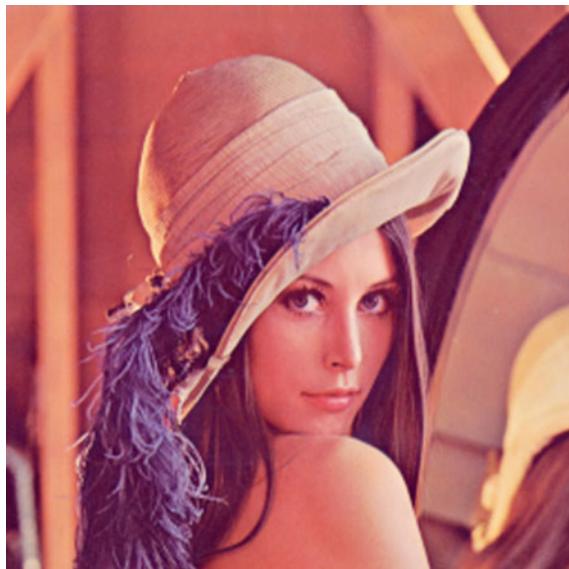


Fig. 5.16 Sample color image



Fig. 5.17 Three channels

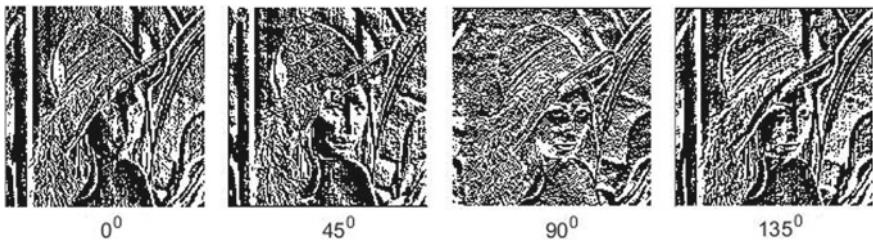


Fig. 5.18 Masked edge output

5.6 Summary

The color feature extraction according to human observation is a difficult task. Since human observation and vision are tough system, it is a bliss to hope that computer vision achieves brilliantly with negligible complications. Additionally, the choice of suitable color features for an image classification system must identify the feature types that are suitable for the task. There is no complete color feature that works best for each category of image.

References

- Guo P, Stanley RJ, Cole JG, Hagerty JR, Stoecker WV (2017) Color feature-based pillbox image color recognition. In: VISIGRAPP (4: VISAPP), pp. 188–194. <https://doi.org/10.5220/0006136001880194>
- Chaki J, Dey N, Moraru L, Shi F (2019) Fragmented plant leaf recognition: bag-of-features, fuzzy-color and edge-texture histogram descriptors with multi-layer perceptron. Optik 181:639–650. <https://doi.org/10.1016/j.jleo.2018.12.107>
- Chaki J, Parekh R, Bhattacharya S (2016) Plant leaf recognition using a layered approach. In: 2016 IEEE international conference on microelectronics, computing and communications (MicroCom). Durgapur, pp 1–6. <https://doi.org/10.1109/microcom.2016.7522541>
- Tharwat A, Gaber T, Awad YM, Dey N, Hassanien AE (2015) Plants identification using feature fusion technique and bagging classifier. In: The 1st international conference on advanced intelligent system and informatics (AISI2015). Beni Suef, Egypt, pp 461–471
- AlShahrani AM, Al-Abadi MA, Al-Malki AS, Ashour AS, Dey N (2018) Automated system for crops recognition and classification. In: Computer vision: concepts, methodologies, tools, and applications, pp. 1208–1223
- Leaf classification. <https://www.kaggle.com/c/leaf-classification>
- George J, Raj SG (2017) Leaf recognition using multi-layer perceptron. In: 2017 IEEE international conference on energy, communication, data analytics and soft computing (ICECDS). Chennai, pp 2216–2221. <https://doi.org/10.1109/icecds.2017.8389846>
- San M, Aung MM, Khaing PP (2019) Fruit recognition using color and morphological features fusion. I.J. Image, Graphics Signal Process 10:8–15. <https://doi.org/10.5815/ijigsp.2019.10.02>
- Jana S, Basak S, Parekh R (2017) Automatic fruit recognition from natural images using color and texture features. In: 2017 IEEE devices for integrated circuit (DevIC). Kalyani, pp 620–624. <https://doi.org/10.1109/devic.2017.8074025>

10. De Goma JC, Quilas CAM, Valerio MAB, Young JJP, Sauli Z (2018) Fruit recognition using surface and geometric information. *J Telecommun Electron Comput Eng (JTEC)* 10(1–15):39–42
11. Fachrurrozi M, Fiqih A, Saputra BR, Algani R, Primanita A (2017) Content based image retrieval for multi-objects fruits recognition using k-means and k-nearest neighbour. In: 2017 IEEE international conference on data and software engineering (ICoDSE). Palembang, pp 1–6. <https://doi.org/10.1109/icodse.2017.8285855>
12. Fruits 360 dataset. <https://www.kaggle.com/moltean/fruits>
13. Lodh A, Parekh R (2017) Flower recognition system based on color and GIST features. In: 2017 IEEE devices for integrated circuit (DevIC). Kalyani, pp 790–794. <https://doi.org/10.1109/devic.2017.8074061>
14. Galphade AV, Walse KH (2019) Supervised learning approach for flower images using color, shape and texture features. *Int Res J Eng Technol* 6(05)
15. Bulbul MF, Happy KS, Azme Z (2018) Flower image retrieval using color and shape features and multiple distance functions. *Adv Image Video Process* 6(6):08–08. <https://doi.org/10.14738/aivp.66.5618>
16. Rosyani P, Taufik M, Waskita AA, Apriyanti DH (2018) Comparison of color model for flower recognition. In: 2018 3rd IEEE international conference on information technology, information system and electrical engineering (ICITISEE). Yogyakarta, pp 10–14. <https://doi.org/10.1109/icitisee.2018.8721026>
17. Category flower dataset. <https://www.robots.ox.ac.uk/~vgg/data/flowers/102/>
18. Youssef A, Albani D, Nardi D, Bloisi DD (2016) Fast traffic sign recognition using color segmentation and deep convolutional networks. In: International conference on advanced concepts for intelligent vision systems, pp 205–216. https://doi.org/10.1007/978-3-319-48680-2_19
19. Shih HC (2018) Hand gesture recognition using color-depth association for smart home. In: 2018 IEEE 1st international cognitive cities conference (IC3). Okinawa, pp 195–197. <https://doi.org/10.1109/ic3.2018.00-27>
20. Păvăloiu I, Ignat A (2016) Iris recognition using color and texture features. In: International workshop soft computing applications, pp 483–497. https://doi.org/10.1007/978-3-319-62524-9_36
21. Ma J, Du K, Zhang L, Zheng F, Chu J, Sun Z (2017) A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing. *Comput Electron Agric* 142:110–117. <https://doi.org/10.1016/j.compag.2017.08.023>
22. CIFAR-10. <https://www.kaggle.com/c/cifar-10>
23. Jothiaruna N, Sundar KJA (2020) A segmentation method for comprehensive color feature with color-to-grayscale conversion using SVD and region-growing method. In: First international conference on sustainable technologies for computational intelligence, pp 303–310. https://doi.org/10.1007/978-981-15-0029-9_24
24. Nakate SM (2018) Effective object boundary detection using color double opponency. *Int J Eng Sci* 8(2):16024–16027
25. Liu GH (2016) Salient areas detection using color volume. In: 2016 IEEE advanced information management, communicates, electronic and automation control conference (IMCEC). Xi'an, pp 474–478. <https://doi.org/10.1109/imcec.2016.7867257>
26. Sengupta S, Mittal N, Modi M (2020) Improved skin lesions detection using color space and artificial intelligence techniques. *J Dermatol Treat* 1–27. <https://doi.org/10.1080/09546634.2019.1708239>
27. Standard test images. <https://www.ece.rice.edu/~wakin/images/>