

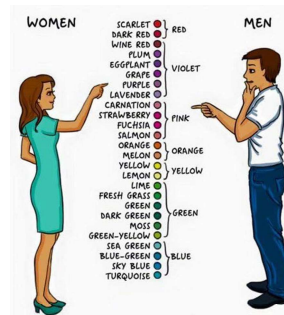
Lecture notes



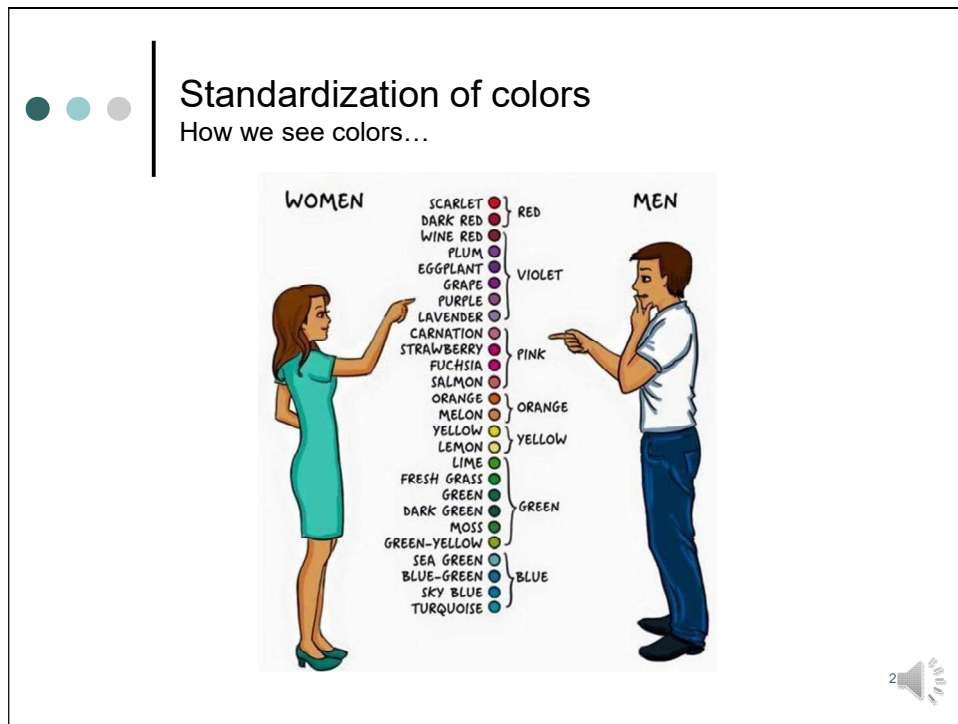
Image Processing II

Color images

Part 2 – Standardization of colors



Slide #2



This cartoon is a good start for the second part of our lecture, which is dedicated to the standardization of colors. There is much more truth in this cartoon than you might think in the first instance. It not only nicely characterizes the behavior of men and women.

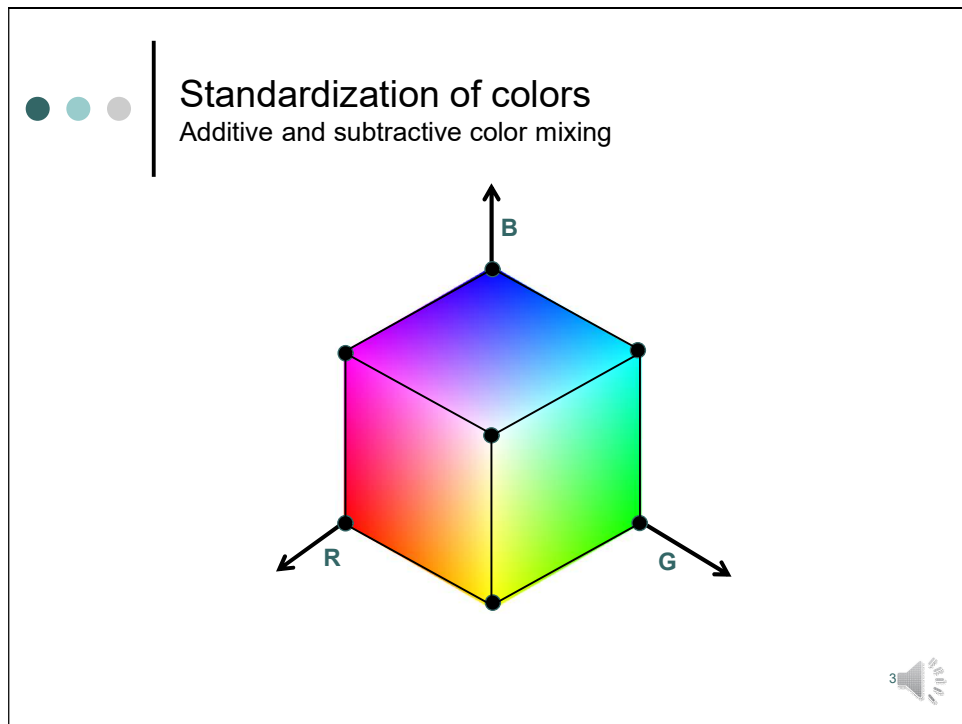
Well, I should not generalize too much. At least the woman in this cartoon carefully combined her lime green dress with dark green shoes to have a perfectly outfit. I personally feel like the man in this cartoon, and I hardly care, if my T-shirt is blue, green or red. So why should I discriminate between “turquoise”, “sky blue” or “navy blue”.

But, we not only interpret colors in slightly different ways. However, as we discussed in the last video, also polymorphisms for the opsin genes exist, which result in different absorption properties. This means that although most of us enjoy color vision, we might perceive the surrounding world differently. It is not only that due to different spectral sensitivities colored objects might draw different attention to you or me. As we will see later in more detail, colored objects that look different to you may look identical to me, and things that are alike to you may look distinctly different to me.

It is interesting and fascinating to see that a difference of a single nucleotide in our X chromosome can place people in distinct perceptual worlds. Moreover, women who are heterozygous for opsin genes have the potential for enhanced color vision.

But, how can we quantify the way we perceive colors? This is done by colorimetry.

Slide #3




In our real life subtractive color mixing dominates. The colors we see around us are created by the light which is reflected from the objects around us. Only when we watch TV or work in front of a monitor the perception of colors is created by additive color mixing.

Basic colorimetry, however, is built on the results of additive color mixing experiments, because the laws of additive color mixing are much more simpler than those of subtractive color mixing. Moreover, when we consider subtractive color mixing we have to take much more factors, like the light source, into account. Apart from this, subtractive color mixing always contains as a last step additive color mixing, because the parts of the spectrum which are reflected create a color sensation by stimulation of the photoreceptors. This is in essence additive color mixing.

Thus, in an attempt to standardize colors, we should focus first on additive color mixing.

Slide #4



Standardization of colors


Graßmann's laws

*IV. Zur Theorie der Farbenmischung;
von H. Graßmann, Professor in Stettin.*

Im 87. Bande dieses Journals theilt Hr. Helmholtz eine Reihe zum Theil neuer und sinureicher Beobachtungen mit, aus welchen er den Schlufs zieht, dafs die seit Newton allgemein angenommene Theorie der Farbennischung in den wesentlichsten Punkten irrig sey, und es namentlich nur zwei prismatische Farben gebe, nämlich Gelb und Indigo, welche vermischt Weiße liefern. Daher möchte es nicht überflüssig seyn, zu zeigen, wie die Newton'sche Theorie der Farbenmischung bis zu einem gewissen Punkte hin, und namentlich der Satz, dafs jede Farbe ihre Complementarfarbe hat, welche mit ihr vermischt Weiß liefert, aus unbestreitbaren Thatsachen mit mathematischer Evidenz hervorgeht, so dafs dieser Satz als einer der wohlbegründetsten in der Physik angesehen werden muß. Ich werde dann zeigen, wie die von Helmholtz angestellten *positiven* Beobachtungen, statt gegen diese Theorie zu zeugen, vielmehr dazu dienen können, dieselbe theils zu bestätigen, theils zu ergänzen.

Hierbei wird es nützlich seyn, den Farbendruck der


Graßmann, Ann Phys Chem 165, 69–84 (1853).



Important empirical laws of additive color mixing were already formulated in 1853 by the mathematician Hermann Graßmann (1809 – 1877).

Possibly you will run across his name also in the lectures of Advanced Mathematics, because Graßman worked quite a lot on vectors and tensors. But what we are interested in are his empirical laws of color mixing, which were published in the article, which is reproduced on this slide. During this time many scientific results were published in German. This makes reading this text for the German students among you very easy. But for the others I will translate his essential findings.

Slide #5




Standardization of colors

Graßmann's laws

mit farblosem Lichte, so wird der Farbeindruck durch diese Beimischung abgeschwächt. Die populäre Sprache ist reich an Bezeichnungen, welche diese Differenz bezeichnen sollen; die Bestimmungen: gesättigt, tief, blaß, fahl, matt, weißlich, welche man den Farbennamen hinzufügt, sollen dies Verhältniß darstellen. Die wissenschaftliche Bezeichnung, welche dieser populären Nomenklatur substituirt werden muß, ergibt sich aus dem Obigen von selbst, indem jeder Farbeindruck der genannten Art sich in drei mathematisch bestimmbare Momente zerlegt: den *Farbenton*, die *Intensität der Farbe*, und die *Intensität des beigemischten Weißs*. Die verschiedenen Farbtöne bilden eine stetige Reihe von der Art, daß sich, wenn man von

Graßmann, Ann Phys Chem 165, 69–84 (1853).

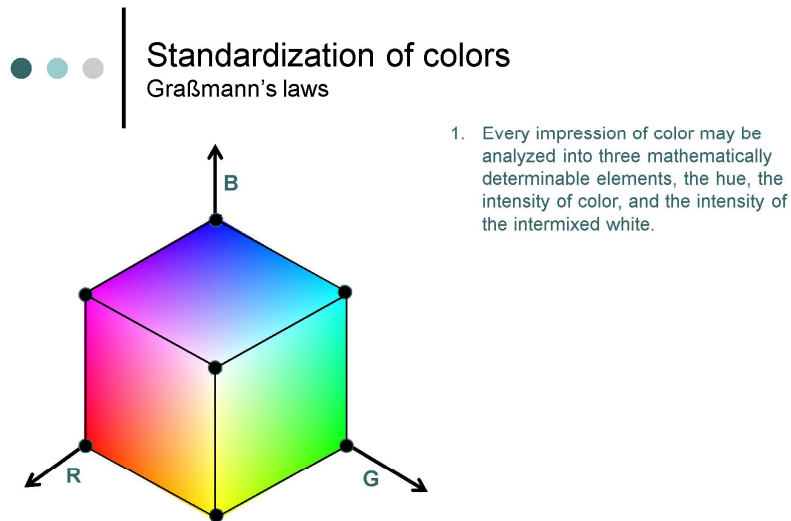


In his paper, Graßmann states that in our popular language we have a multitude of terms for different shades of color like saturated, deep, pale, wan, matt, whitish to better characterize the shades of colors.

Scientifically, however, every impression of color may be analyzed into three mathematically determinable elements, which are the hue, the brightness of color, and the brightness of the intermixed white.

Let's have a look at the color cube to understand better what he says.

Slide #6




Here's now the translation of Graßmann's law. The pure colors can be found at the periphery of the cube: red, yellow, green, cyan, blue and purple. This is the hue.

Here, the cube is shown in its brightest colors, but of course we can dim the intensity of the colors.

The saturation of the color is decreased, when we mix it with other colors. For example, when we take this red color and add blue and green components, then we move along the diagonal. The saturation decreases.

In brief, to specify a color, three elements are necessary and sufficient: the hue, the intensity (luminance), and the intensity (luminance) of the intermixed white, which defines the saturation.

Slide #7

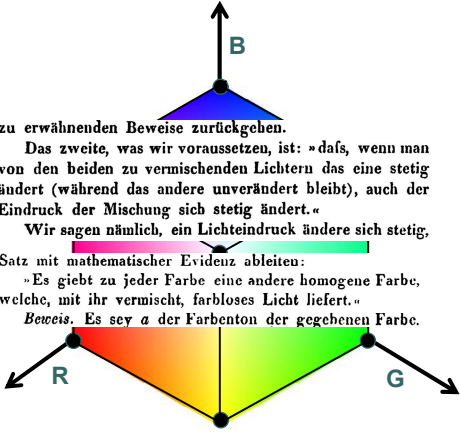


Standardization of colors

Graßmann's laws

zu erwähnenden Beweise zurückgeben.
Das zweite, was wir voraussetzen, ist: »dafs, wenn man von den beiden zu vermischenden Lichtern das eine stetig ändert (während das andere unverändert bleibt), auch der Eindruck der Mischung sich stetig ändert.«
Wir sagen nämlich, ein Lichteindruck ändere sich stetig,
Satz mit mathematischer Evidenz ableiten:
»Es giebt zu jeder Farbe eine andere homogene Farbe, welche, mit ihr vermischt, farbloses Licht liefert.«
Beweis. Es sey a der Farbenton der gegebenen Farbe.


1. Every impression of color may be analyzed into three mathematically determinable elements, the hue, the intensity of color, and the intensity of the intermixed white.
2. If one of two mingling lights is continuously altered, while the other remains unchanged, the impression of the mixed light is also continuously changed.



Also Graßmann's second law is easy to understand:

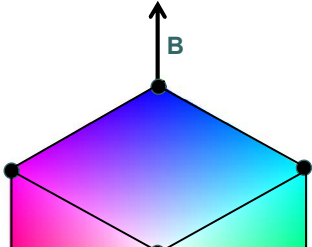
If the contribution of one light source is altered, while the other is constant, the impression of the mixed light is altered. Let's take a primary, like red. If we add another primary, like green, we change the hue towards yellow. If we take a secondary like yellow and add the complementary primary blue, we change the saturation. For every color, there is complementary color, which, when mixed together, produce a colorless gray.

Slide #8



Standardization of colors


Graßmann's laws



aussetzungen auszureichen. Ich werde jetzt, um den Hauptsatz der Farbenmischung abzuleiten, noch zu den bisherigen beiden Voraussetzungen eine dritte hinzufügen, nämlich die:

»dafs zwei Farben, deren jede constanten Farbenton, constante Farbenintensität und constante Intensität des beigemischten Weißs hat, auch constante Farbenmischung geben, gleich viel aus welchen homogenen Farben jene zusammengesetzt seyen.«

1. Every impression of color may be analyzed into three mathematically determinable elements, the hue, the intensity of color, and the intensity of the intermixed white.
2. If one of two mingling lights is continuously altered, while the other remains unchanged, the impression of the mixed light is also continuously changed.
3. Two colors, both of which have the same hue and the same proportion of intermixed white, also give identical mixed colors, no matter what homogeneous colors they may be composed of.




The third law is very important for the experiments, which are going to come:

Two lights with the same hue and saturation, when mixed, produce another color with identical hue and saturation, independent of their power spectra.

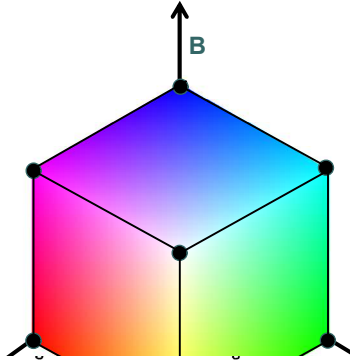
Or in other words: If two colors are indistinguishable (metamers), you can add any other color to them and they will still be metamers. This essentially says that our color perception is linear.

Slide #9




Standardization of colors

Graßmann's laws



Was endlich die Beimischung des farblosen Lichtes betrifft, so ist dazu noch eine Voraussetzung erforderlich. Am einfachsten ist es, anzunehmen:
»dafs die gesammte Lichtintensität der Mischung die Summe sey aus den Intensitäten der gemischten Lichter.«

1. Every impression of color may be analyzed into three mathematically determinable elements, the hue, the intensity of color, and the intensity of the intermixed white.
2. If one of two mingling lights is continuously altered, while the other remains unchanged, the impression of the mixed light is also continuously changed.
3. Two colors, both of which have the same hue and the same proportion of intermixed white, also give identical mixed colors, no matter what homogeneous colors they may be composed of.
4. The total intensity of any mixture is the sum of the intensities of the lights mixed.

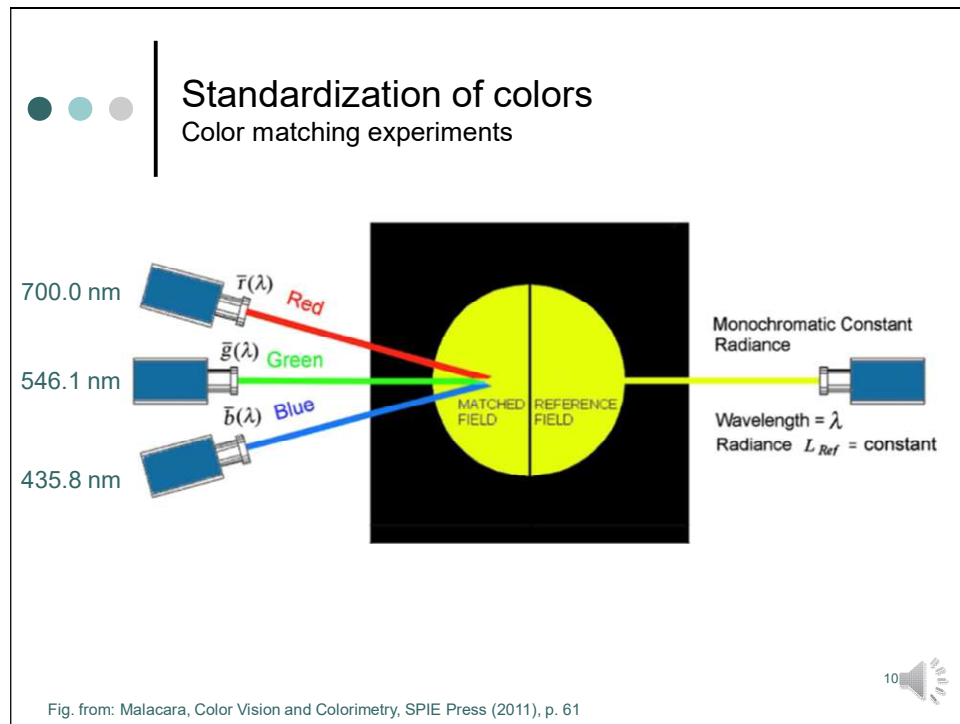


The total intensity of any mixture of light is the sum of the intensities of the lights being mixed. These laws are the basics of all mathematical procedures later established in colorimetry.

Now let's put this into practice. But how can we measure the response of the eye? Unfortunately, the response of the eye cannot be determined by a physical measurement. Instead only a brightness sensation can be determined, which is difficult to quantify. But what we can do is to compare the brightness of a test light, with that of a reference light.

By means of such matching experiments, the spectral responsivity of the eye can be determined.

Slide #10

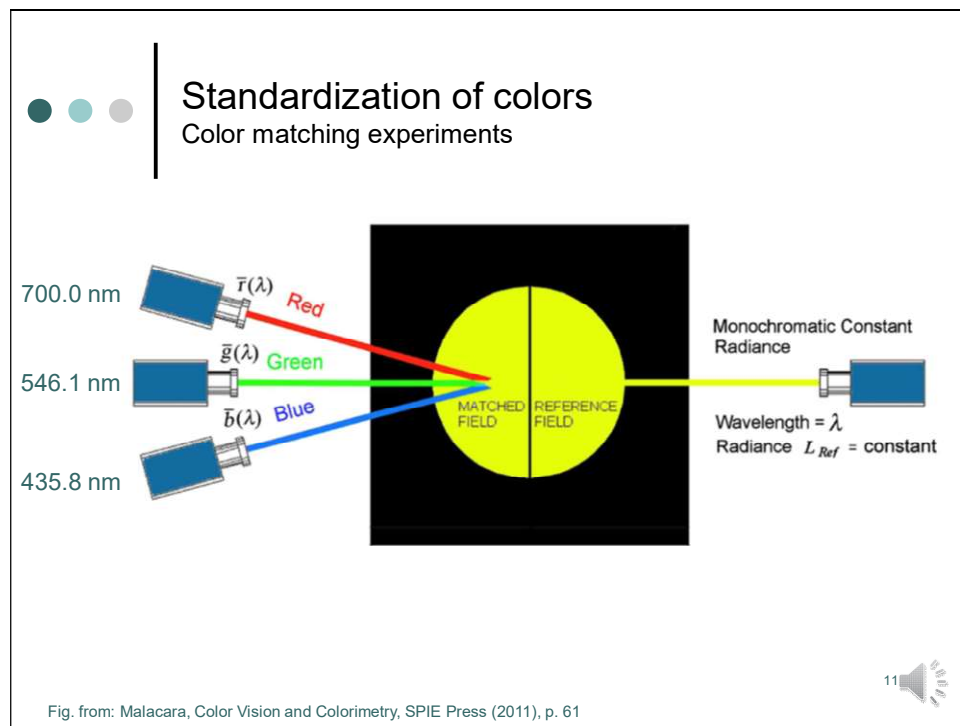


Early color matching experiments have been already performed by Hermann Graßmann (1809 – 1877) and John William Strutt, the later Baron Rayleigh.

The idea of the experiment is quite simple. A spectrometer selects a given wavelength and generates a color impression on the retina of a test person. The test person has to adjust the intensities of three light sources (primaries) such that the color impression matches the color selected by the spectrometer.

Standard primaries used by the Commission International de l'Eclairage (CIE) (1931) were 700nm (red) and the emission lines of the mercury lamp at 546.1nm (green) and 435.8 nm (blue). To avoid stimulating rods outside the fovea centralis, the field, in which the matching colors are determined must be small. CIE conventions initially defined a field of 2-deg (CIE, 1932). Since the 2-deg field is too small for many practical purposes, a 10-deg field was adopted in later CIE conventions (CIE, 1964).

Slide #11



To begin the color-matching procedure, the spectrally pure reference field is set to have any desired radiance L_{Ref} , with a wavelength of 700 nm (red color). A perfect color match is achieved when the red light in the matched field is adjusted to the same brightness as the reference field, while the green and blue beams are set to zero luminance. The procedure continues by decreasing the wavelength for the reference field in constant steps $\Delta\lambda$ until all of the visible range is covered. When shifting the wavelength, the radiance, which is defined as the radiant flux or radiant power per unit angle and unit area is kept constant.

Notably it is the power and not the brightness, which is kept constant. The brightness or to be more precise the perceived luminance of the monochromatic reference light source depends on the relative photopic luminous efficiency of the eye for different wavelengths.

For any monochromatic light source with wavelength λ , the perceived luminance and the radiance in the reference field are related by

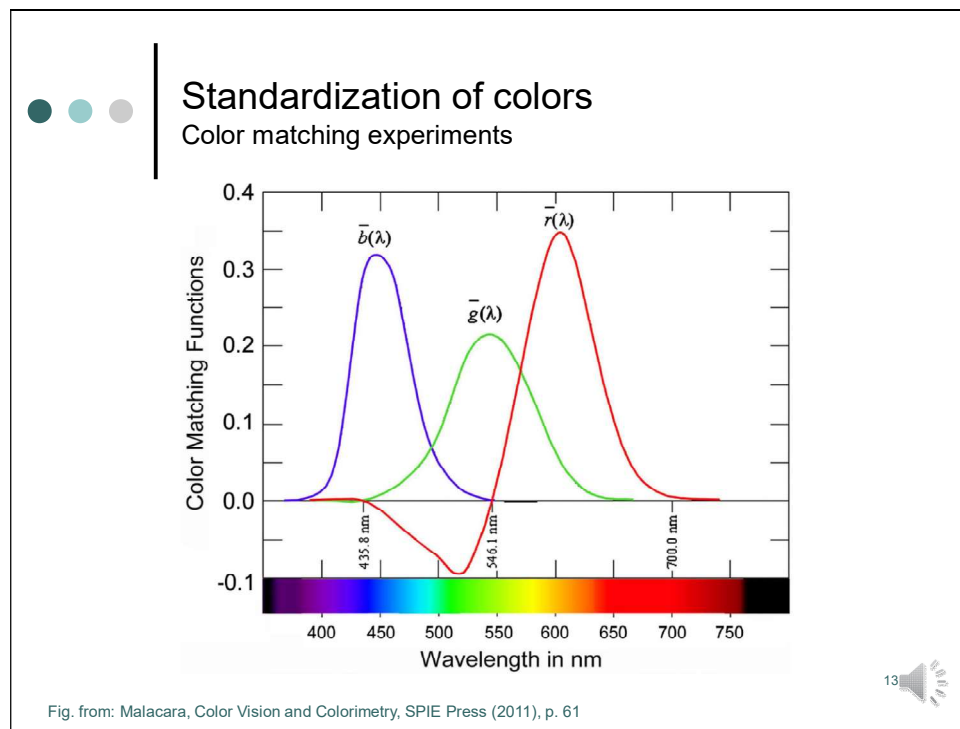
$$L_{VRef}(\lambda) = 683V(\lambda)L_{Ref}(\lambda)$$

where $V(\lambda)$ is the relative photopic luminous efficiency of the eye.

At every wavelength step, the test person has to adjust the three colors (red, green, and blue) until both fields, the matched field and the reference field, look identical. This means both fields must have the same hue, saturation, and perceived luminance. Matching of the two fields is obtained by careful trial and error. As a result, for each wavelength in the reference field, three radiances $L_R(\lambda)$, $L_G(\lambda)$, and $L_B(\lambda)$ will be obtained in the matched field. Please note, that although the perceived luminances need to be equal in the matched field and the reference field, the sum of the radiances in the matched field is not necessarily equal to the radiance L_{Ref} in the reference field.

The next slide shows the results of such a systematic investigation, which was done independently by Wright and Guild.

Slide #12



The plot shows the intensities of red, green and blue primaries, needed to match the light of the pure wavelength λ . First of all we see that at wavelengths of the primaries no contributions from other light sources are necessary. This is, of course, what we should expect. At 495 nm, which corresponds to a cyan color, we have equal contributions of the blue and green primary. At a wavelength of 570 nm, which corresponds to yellow color, we have equal contributions from the green and red primaries. This is also, what we should expect.


But, while discussing the curve you should have noticed something weird about the red curve, which is less than zero in some places. How is this possible? Negative light intensities do not exist.

Wright and Guild found that some colors could not be matched by mixing red, green and blue, for example, yellow with $\lambda=580$. Only when red was mixed to the reference light, test field and reference field appeared to be identical.

Graßmann's third law says that, if two colors are indistinguishable (metamers), you can add any other color to them and they will still be metamers. Thus, if we have to add red to the reference light to make it look similar to the sum of the blue and green primary, this is the same like subtracting the red from the test field.

Negative intensities are great mathematically, but not so practical for real world applications. You cannot have an RGB monitor with negative red intensities. Nevertheless, it would still be useful to have a good representation of the total amount of color we can see, and to know which of those colors can be reproduced by a given RGB basis.

Slide #13



Standardization of colors

Transformation to CIE chromaticity coordinates


No three monochromatic colors exist that can be mixed in positive amounts to match the full spectrum.

Instead only sums of monochromatic colors can form a basis set, which can be mixed in positive amounts to reproduce the full spectrum.

The transformation chosen by the Commission International de l'Eclairage (CIE) is:

$$\begin{bmatrix} X(\lambda) \\ Y(\lambda) \\ Z(\lambda) \end{bmatrix} = \begin{bmatrix} 2.76888 & 1.75175 & 1.13016 \\ 1 & 4.59070 & 0.06010 \\ 0 & 0.05651 & 5.59427 \end{bmatrix} * \begin{bmatrix} r(\lambda) \\ g(\lambda) \\ b(\lambda) \end{bmatrix}$$

The functions $X(\lambda)$, $Y(\lambda)$ and $Z(\lambda)$ are called the tristimulus curves or CIE curves.

 14

The International Illumination commission (CIE) realized that no three monochromatic colors could be mixed in positive amounts to match the full spectrum.

Instead only sums of monochromatic colors can form a basis set, which can be mixed in positive amounts to reproduce the full spectrum.

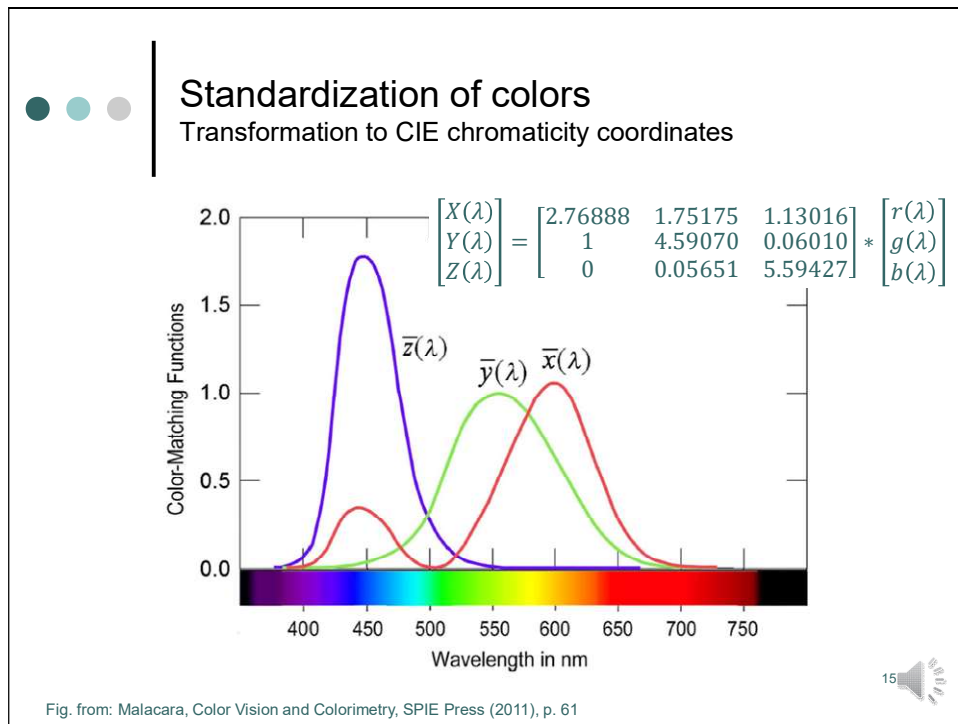
The commission defined a new basis XYZ, and this equation shows how the new coordinates can be calculated. Thus by simple matrix multiplication we can obtain the respective values. Multiplying the first row with the rgb vector yields X.

Multiplication of the second row yields Y, and multiplications of the third row yields Z.

Here, I do not want to go too much into detail, why these specific values have been chosen. But, note that the contribution of the primaries to Y corresponds to the relative luminance (per stimulus unit) of the primaries.

The next figure shows a plot of these data.

Slide #14




So here is a plot of the new coordinates XYZ, which have been obtained by multiplying the color matching functions of the original primaries with this transformation matrix.

Z looks pretty similar to the color matching function of the blue primary, which is not so astonishing, when we look at the transformation matrix. To obtain z the lower row needs to be multiplied with the rgb vector. Here the contribution of blue is greatest. The other contributions are almost negligible.

Y corresponds to the relative perceived luminance of the primaries as we have discussed it before. This means Y resembles the color matching function for green g(r), since green contributes most to the overall perceived luminance. But, it's not identical, since the contributions from red and blue to the luminance have to be taken into account.

X has contributions from all three primaries. The curve has a second smaller bump in the short wavelength part of the spectrum.

Slide #15



Standardization of colors

Transformation to CIE chromaticity coordinates

The transformation chosen by the Commission International de l'Eclairage (CIE) is:

$$\begin{bmatrix} X(\lambda) \\ Y(\lambda) \\ Z(\lambda) \end{bmatrix} = \begin{bmatrix} 2.76888 & 1.75175 & 1.13016 \\ 1 & 4.59070 & 0.06010 \\ 0 & 0.05651 & 5.59427 \end{bmatrix} * \begin{bmatrix} r(\lambda) \\ g(\lambda) \\ b(\lambda) \end{bmatrix}$$


The next step consists in normalizing the intensities.

$$x(\lambda) = \frac{X(\lambda)}{X(\lambda) + Y(\lambda) + Z(\lambda)} \quad y(\lambda) = \frac{Y(\lambda)}{X(\lambda) + Y(\lambda) + Z(\lambda)}$$

The remaining variable z is not independent:

$$z(\lambda) = \frac{Z(\lambda)}{X(\lambda) + Y(\lambda) + Z(\lambda)} = 1 - x(\lambda) - y(\lambda)$$

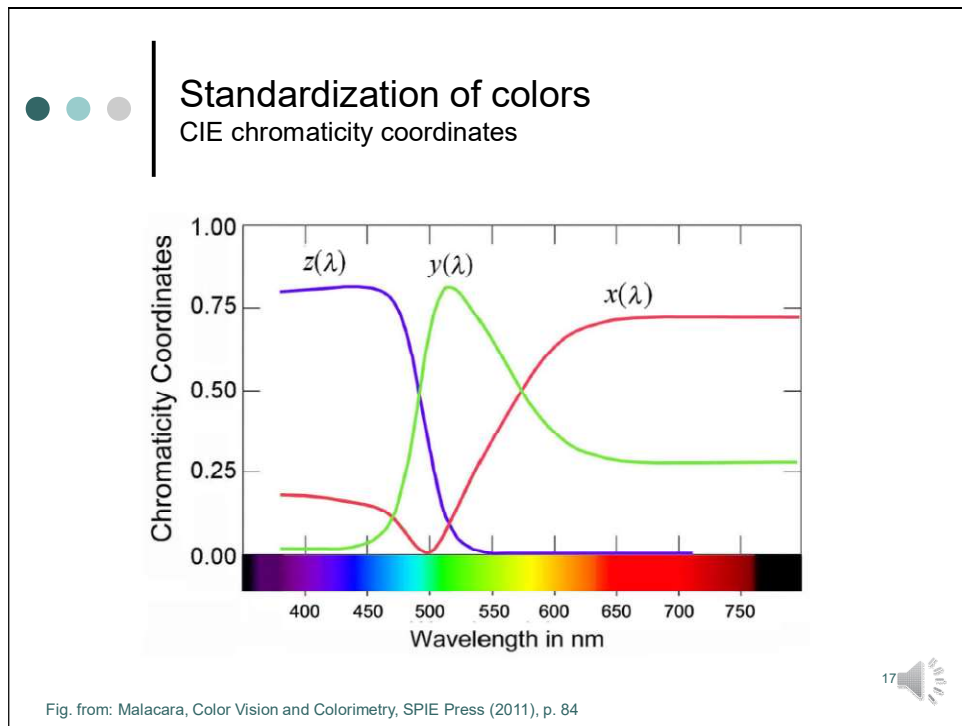
Thus we need only two variables to describe all colors we can see.



To calculate the relative contribution we need to normalize the color matching functions for the respective wavelength.

Note, that only two of the normalized variables are independent. As a result we need only two variables to describe all colors we can see. The next plot shows the normalized variables.

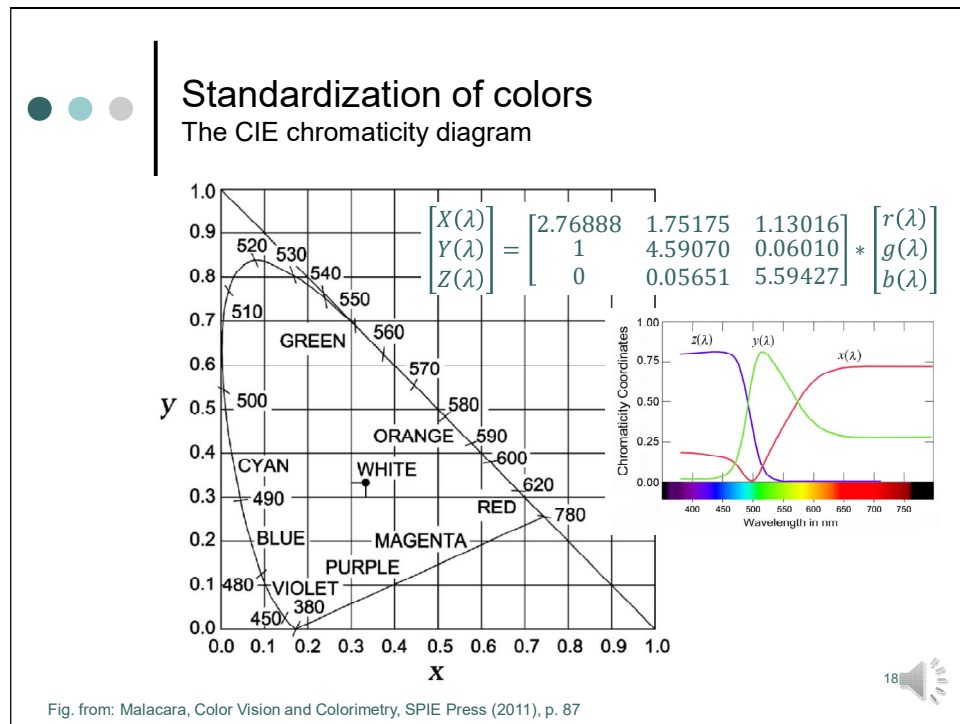
Slide #16



What you see here is the result of the normalization procedure.

In the next step we can plot two independent variables, like x and y , against each other. The plot yields a contour.

Slide #17



This plot of y versus x is known as the CIE chromaticity diagram. All spectrally pure colors appear along the horseshoe-like curve.

When interpreting this diagram we have to keep in mind, how the chromaticity coordinates have been calculated. Here's again the matrix equation we used for the transformation and the plot of the relative contributions.

Now, let's have a detailed look at the plot and start with the short wavelength edge of the spectrum. 380 nm corresponds to the coordinates $x = 0.1741$ and $y = 0.0050$. Accordingly z is 0.8209. z has almost its maximal value. Thus the blue component dominates, but to get this color sensation, we also need to have a contribution of x . Thus the red and green components contribute also to the color.

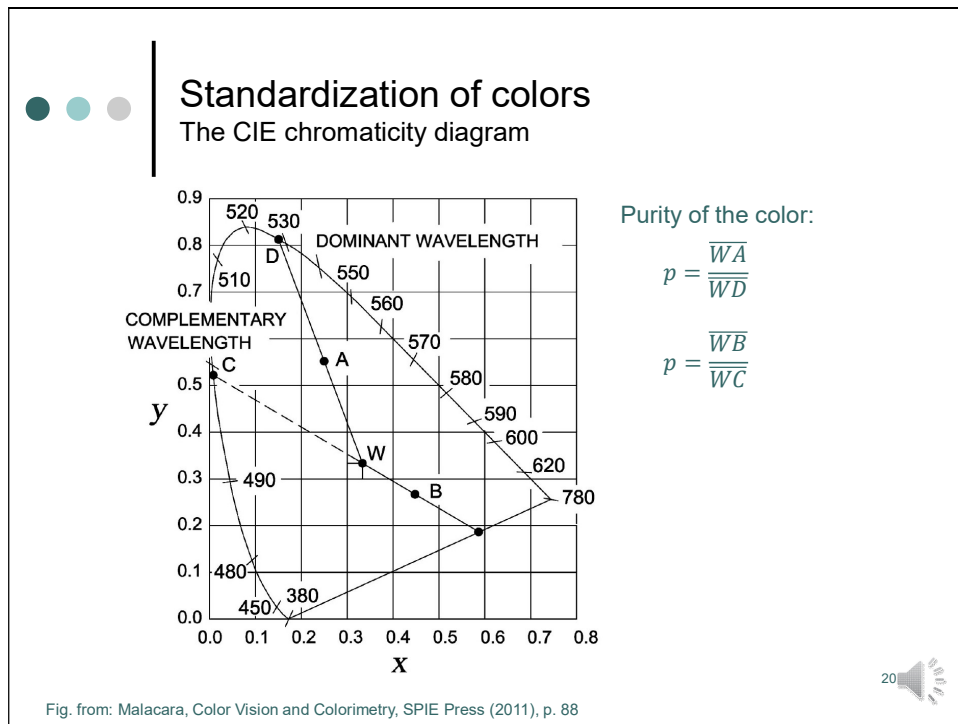
Around 495 nm x is minimal y and z have equal values. Y is maximal at 520 nm ($x=0.0743$, $y=0.8338$, $z=0.0919$). Here the green component dominates.

At 545 nm the contribution of z becomes negligible, and X and Y add up to unity. That's why the chromaticity diagram now follows the diagonal. From 650 nm on the contributions of x and y do not vary. This is why we cannot distinguish shades of red beyond 650 nm. At 780 nm we have reached the end of the visible spectrum. A straight line joining the extremes of the curve represents the purple colors, formed by a combination of red with blue in different proportions. So much for the hue of the color.

The saturation changes when we move to the inside of the plot. White is located at the coordinates $x = 0.33$, $y = 0.33$ (and $z = 0.33$). As we move from the border toward this point, saturation decreases.

The next slide shows this in more detail.

Slide #18



How can we characterize non-saturated colors?

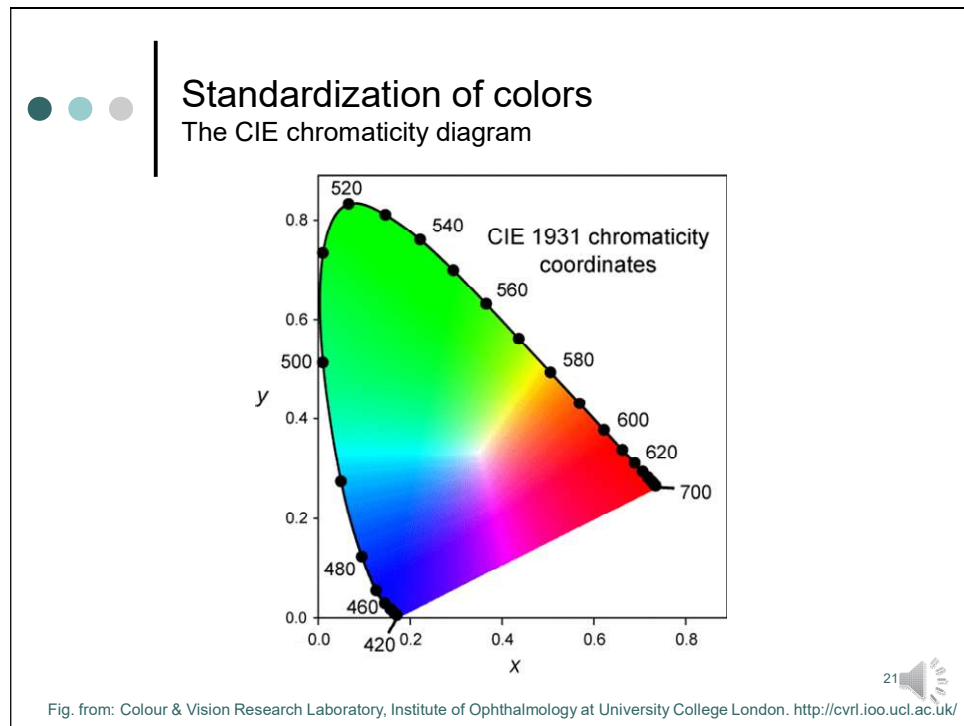
Let us to this aim consider a point A in the CIE diagram. Now, let us trace a line, which originates at the white point W, passes through the point A, and crosses the curve representing the spectrally pure colors. The crossing point D between this line and the curve indicates the dominant wavelength of color A. Purity or saturation can be defined by the ratio of the length of the line WA to the length of the line WD.

Now, let us now consider a point B on the chromaticity diagram. Now, the line from the white point W to point B does not pass through the curve of spectrally pure colors, but through the line representing purple colors. This means, that there does not exist any real wavelength in nature which can elicit a sensation to that similar in point B or any other colors in this sector.

Here we cannot define a dominant wavelength, but only a complimentary wavelength at the crossing point C of this line with the spectrally pure colors curve. Again, in this case the purity of color B is given by the ratio of the length of these lines. So much for the saturation or purity of a color.

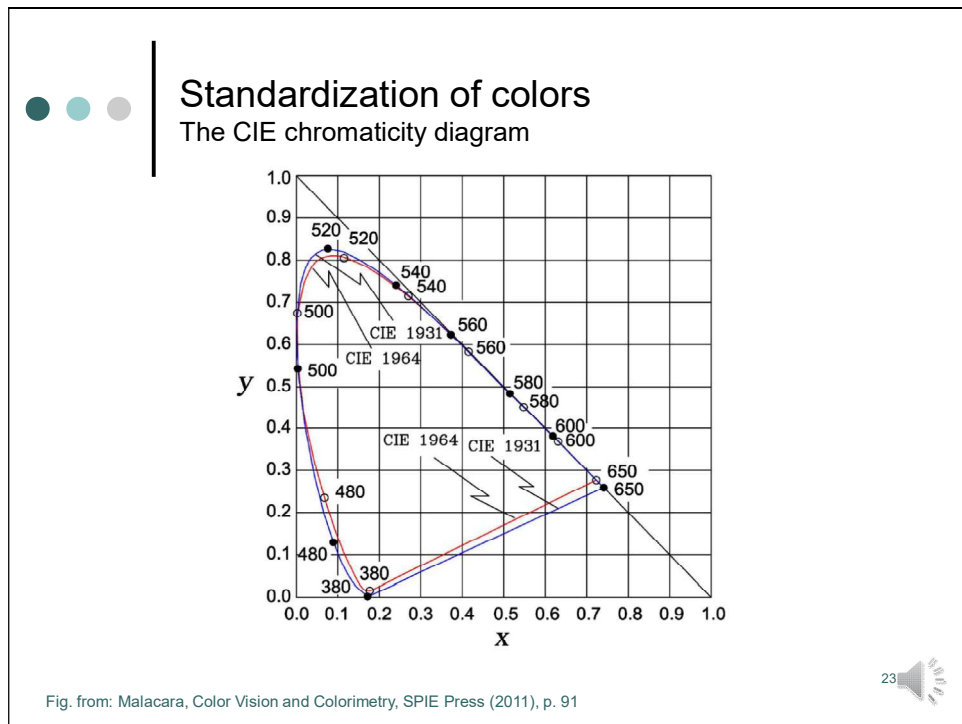
Luminosity, however, cannot be represented in this diagram. So, the point W (with coordinates $x = 0.33$, $y = 0.33$) represents not only the white color, but also all shades of gray and black. In the same manner, any other point can have any luminosity. A complete specification for any given color would be given by the chromaticity coordinates x and y and the luminosity Y .

Slide #19



Many textbooks show a color representation of the chromaticity diagram like the one in this figure. Such a color representation, however, cannot be made perfectly faithful, neither on a computer display nor in print. The reason is that the primary colors used to produce the figure cannot generate all colors represented by the CIE diagram. We will discuss this later, when we talk about the color gamut, i.e. the subset of colors of some color models

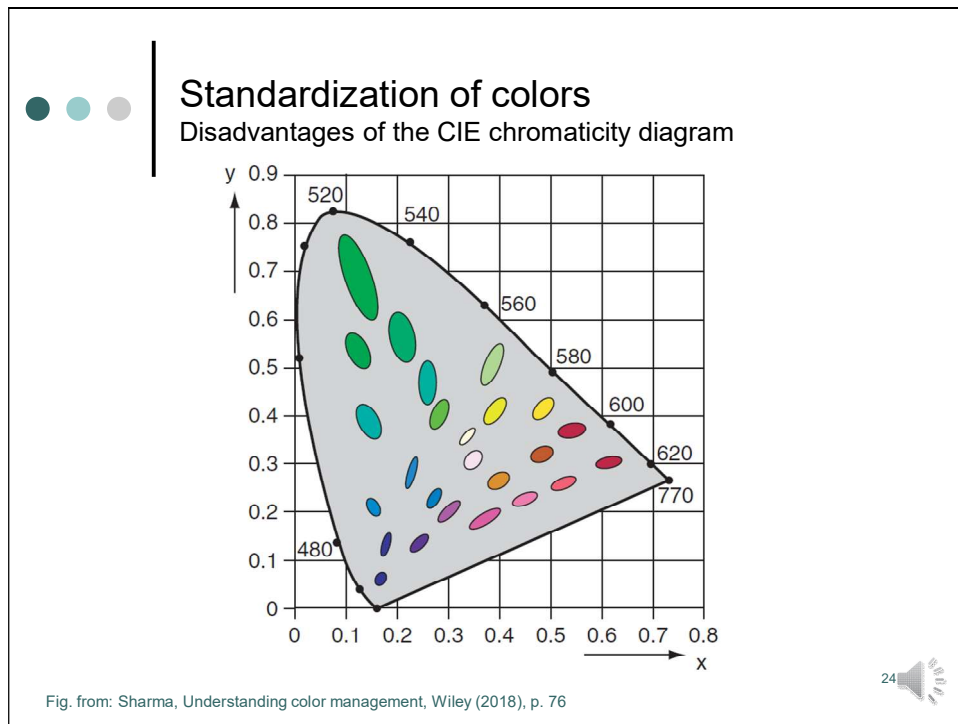
Slide #20



Just one additional remark:

The CIE diagrams, which we discussed by now, were originally obtained in the years before 1931 for a field diameter of 2 degrees. Later, in 1964, new data in a field of 10 degrees have been used for a new definition. This figure shows that the two diagrams are quite similar, so that everything derived for the old standards also applies for newer standards.

Slide #21



Although it is widely used, the CIE chromaticity diagram has several disadvantages.

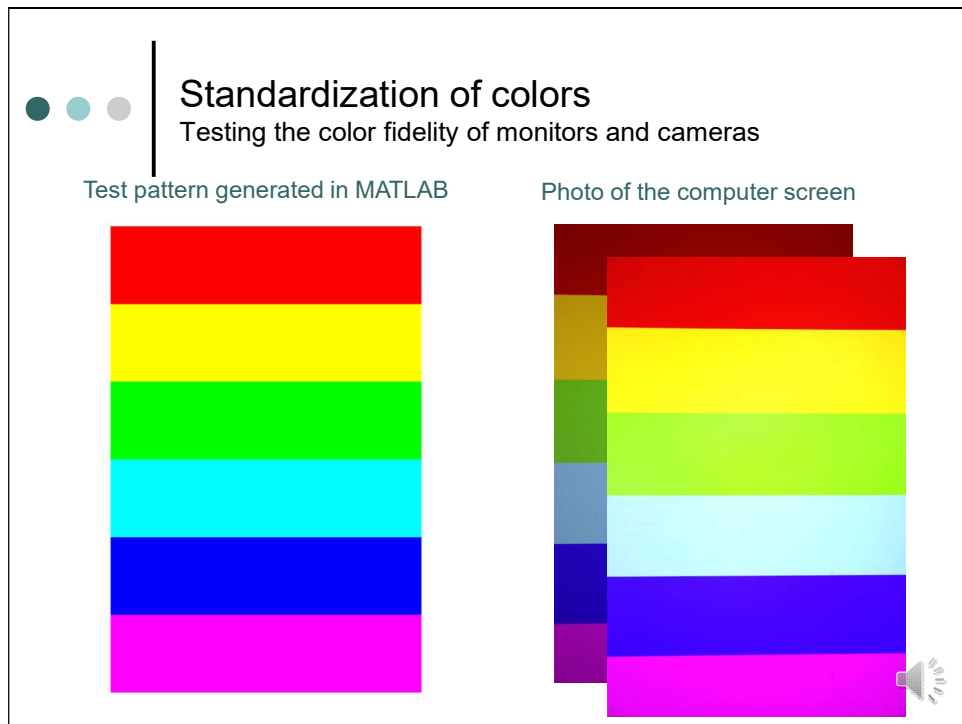
A significant problem is that it is *perceptually non-uniform*. Colors with an equal perceptual difference should be equally spaced in the diagram.

That this is not case shows already a short glance at the wavelength tick marks around the diagram, which are crunched together at the lower edges of the horseshoe.

If you examine the top part of the diagram, however, you will see that two colors can be very far apart in the green part of the diagram before they show any perceivable change. This can be nicely illustrated by MacAdam ellipses, which indicate areas of just-noticeable color difference. In a perceptual uniform color space theoretically all ellipses should be of the same size.

Another problem with the CIE xy color system is the absence of the luminance axis. The xy color space is a two-dimensional color space and Y must be described separately. Later we will discuss some other ways to represent colors.

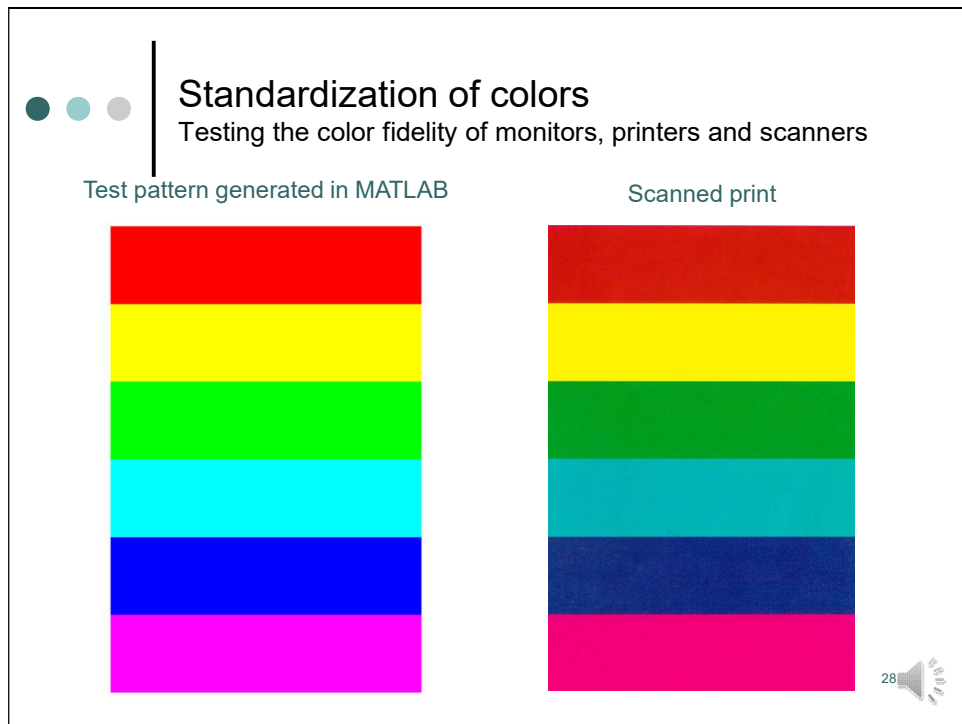
Slide #23



Just take this test pattern, I generated in MATLAB. It contains the three pure primary colors red, green and blue and the secondary colors which have been obtained by mixing equal amounts of the adjacent primaries.

Guess, how the photo looks like, when we display it on the screen? Well, it looks different, right? Since it looks a little bit darker, we might think of adjusting the brightness. At the end the colors look similar. But the colors are not as saturated, as in the original...

Slide #24



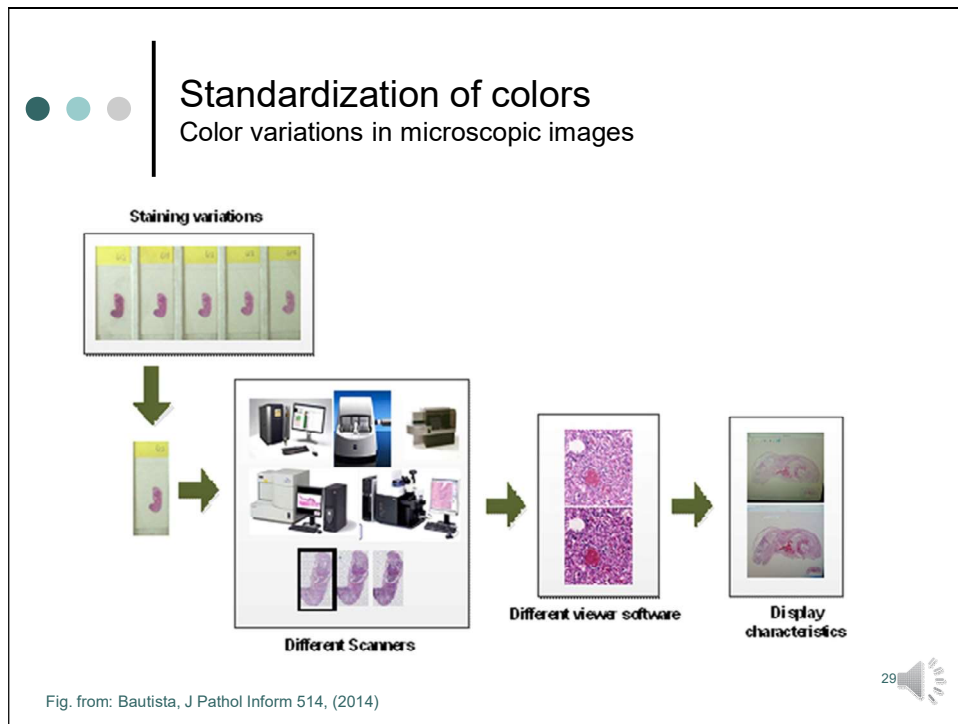
Now, let's test the reproducibility of the colors, when they are printed. Now, I printed the test pattern and scanned it again.

The result shows, that not only the saturation, but also the hue of the colors appears to be affected. The magenta in the original looks like red in the scanned printout.

Check it out by yourself. Take a photo of the test pattern with your mobile and display it on the screen? Are the colors the same?

In our daily life this might not be so important. Even though the colors are not perfectly reproduced, we might enjoy the photo anyway.

Slide #25



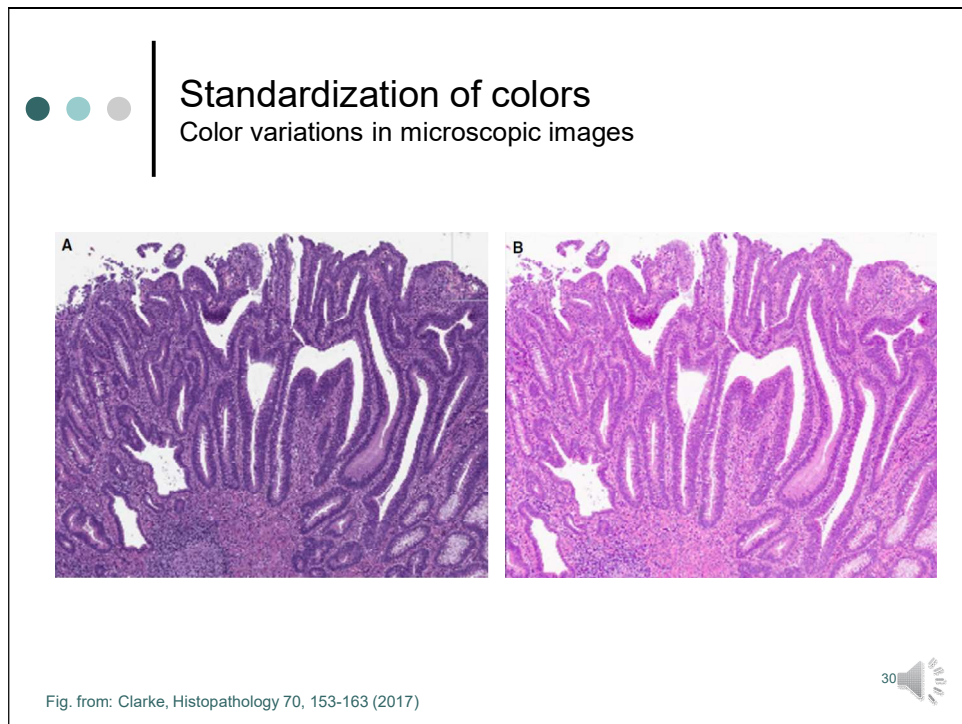
But now think of some medical applications. Color is central in the clinical practice. Pathologists use stains to highlight and identify cells and subcellular structures.

During recent years, new technologies have emerged to enable pathologists to carry out their diagnostic work digitally, rather than with the analogue optical microscope. This technology is often referred to as a 'digital microscope' in which whole slide images of tissue slides are scanned, transmitted and displayed on a computer monitor. We, too, used such an approach in the course of Human Biology to look at the microscopic anatomy of several organs.

To be able to make a reliable diagnosis color variations should not occur in histological images. However there are many possible causes for color variations. One of them, of course is the staining protocol. The other factors involve the scanner, the viewer software and the display. All of them require a good color management.

The following slides show an example.

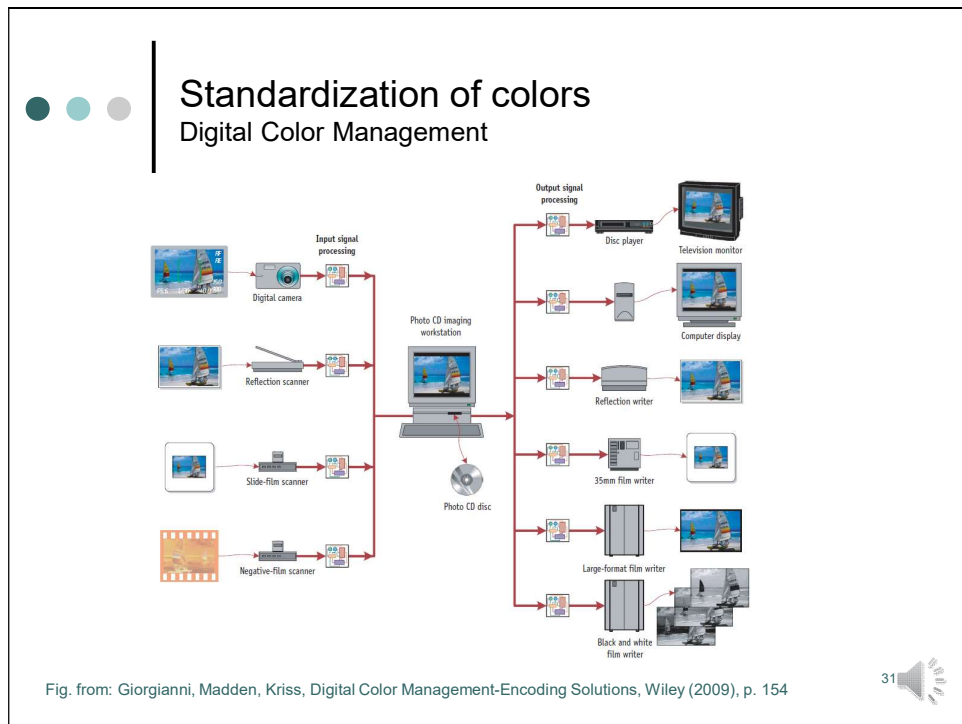
Slide #26



This slide shows a microscopic images of a tubulovillous adenoma with high-grade dysplasia and pseudoinvasion using the same display.

Image A was produced without any color management applied. The reproduced colors look a little bit strange for an usual hematoxylin-eosin stain. However, with color management, colors close to the true colors, as we are used to see them in the transmission microscope, can be retrieved. This is shown in the right image.


Slide #27



Color management means that the conversion between the color representations – or color spaces - of various devices, such as digital cameras, scanners, TV screens, monitors, film printers, computer printers, offset presses, and corresponding media is well coordinated.

The primary goal of color management is to obtain a good match across color devices. Thus we need first to get an idea which of the colors we can see with our eyes can be handled by a specific device. This subset of colors which can be accurately represented by a device is called gamut or color gamut. But let's get some systematic overview of some color spaces.

Slide #28



Color Spaces Overview

Hardware dependent models

Additive color models


- RGB (red, green, blue) models
 - sRGB (1996)
 - Adobe RGB (1998)
- HSI (hue, saturation, intensity) models
 - HSI (hue, saturation, intensity)
 - HSV (hue, saturation, value) (1983)
 - NTSC (luminance (Y), hue (I), saturation (Q))

Subtractive color models

- CMY (cyan, magenta, yellow) (1983)
- CMYK (cyan, magenta, yellow, black) (1983)

Hardware independent models

- CIE XYZ (1931), CIE (1964)
- CIE L*a*b (1976)
- CIE L*C*h



The purpose of a color model (or color system) is to facilitate the specification of colors in some standard way. In essence, a color model is a specification of a coordinate system, and a subspace within that system, such that each color in the model is represented by a single point contained in that subspace.

Most color models in use today can be grouped into hardware-dependent (such as for color monitors and printers) and hardware independent models.

The hardware-oriented models can be subgrouped into additive and subtractive models.

Additive models are

- RGB (red, green, blue) models for cameras or color monitors
- HSI (hue, saturation, intensity) models decouple the color and gray-scale information in an image. This corresponds better to the way humans describe and interpret color.

Subtractive models are used for color printing. Here, two prominent models are the

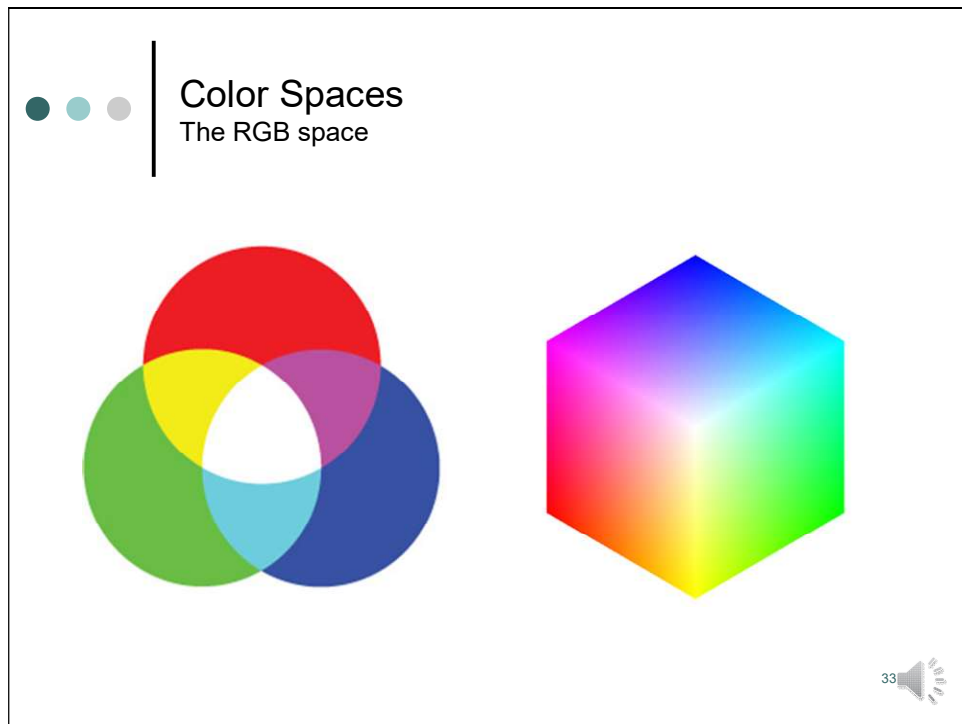
- CMY (cyan, magenta, yellow) and
- CMYK (cyan, magenta, yellow, black) model

Hardware independent models are the

- CIE XYZ system we talked already about
- various successors like CIE L*a*b

There are many more color models in use today than those listed here. In the following slides, however, we will focus our attention on these few models that are somehow representative for many other models used in image processing.

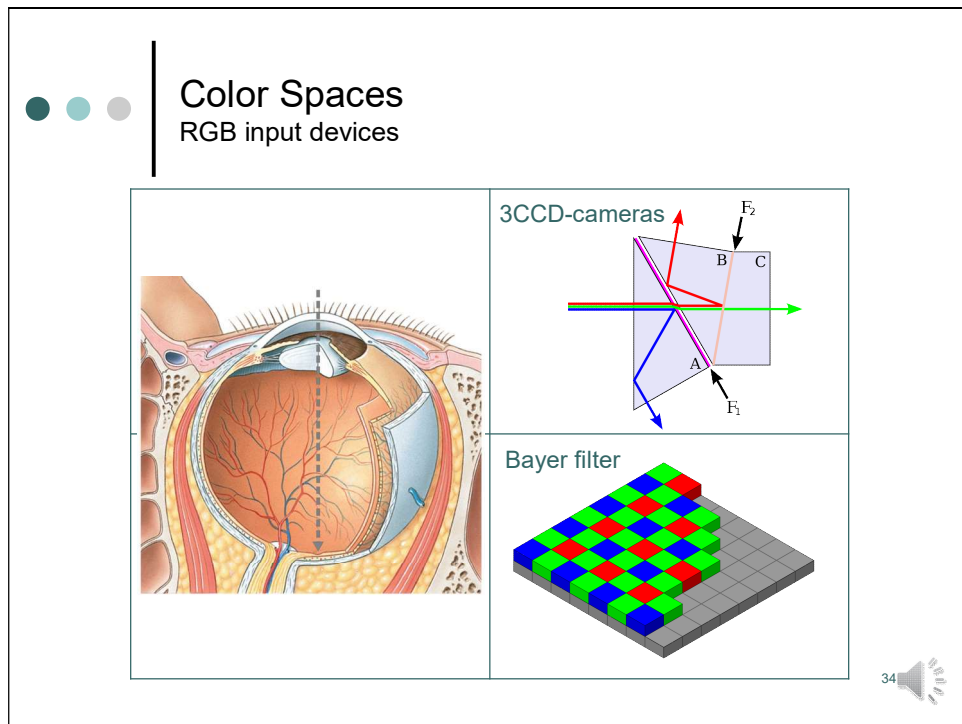
Slide #29



Let's start with the RGB model, about which we have talked already when we talked about additive color mixing. Here we should only briefly summarize what we have learned:

The **RGB color model** is an additive color model, in which the primaries red, green, and blue are added together. The name of the model comes from the initials of the primaries.

Slide #30

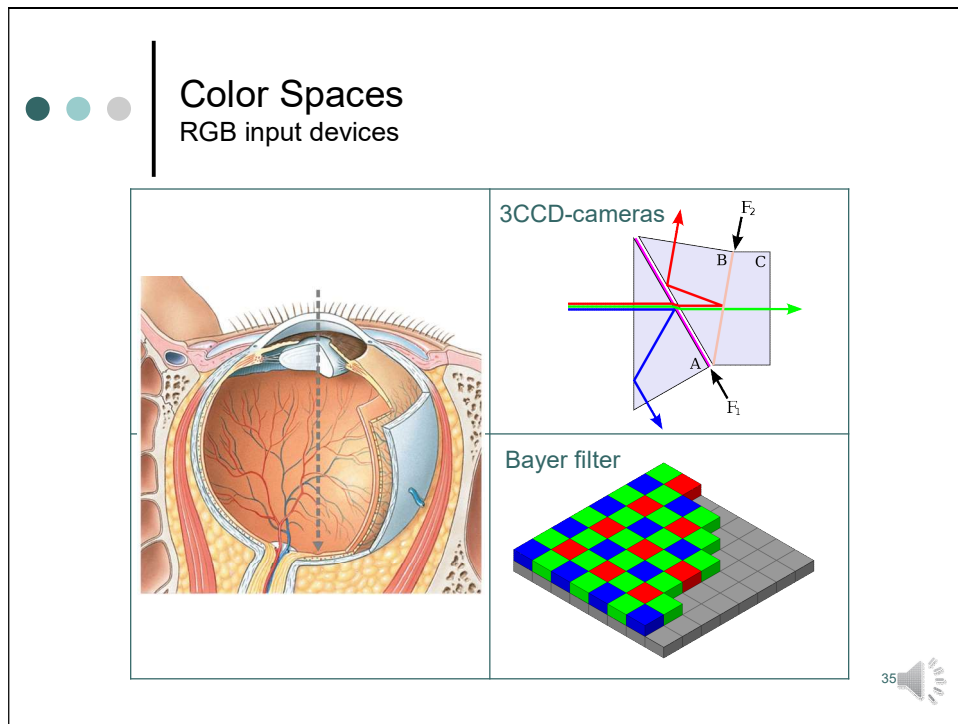


On this slide you see some RGB-input devices. About the prototype, the eye, we talked already in the first lecture.

Unlike the photoreceptors of the eye all pixels of a technical sensor are equally sensitive to the visual spectrum. To convey color sensitivity to the sensor elements, spectral filters have to be used.

One way is realized in three-CCD (3CCD) cameras. Here, incoming light coming is split by a prism into three beams, which are then filtered. Each wavelength band is then recorded by separate sensors. Since three sensors are needed, one receiving the filtered red, one for the filtered green, and one for the filtered blue, such cameras are expensive. Therefore such sensors are only employed in high quality cameras.

Slide #31

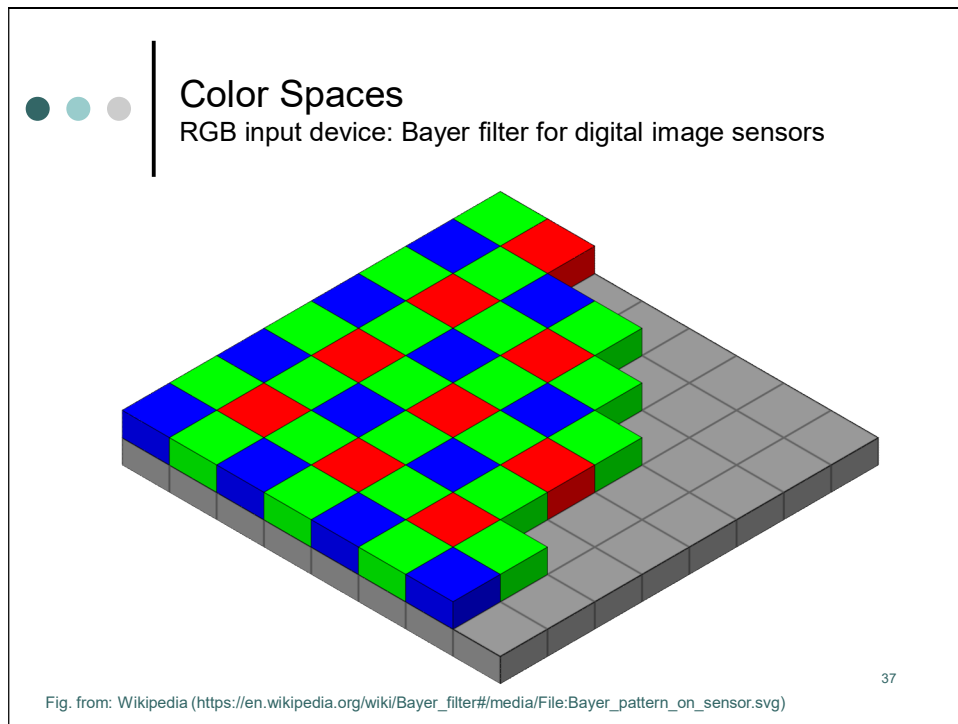


A cheaper way consists in using color filter arrays, so that pixels of the underlying sensor array are covered with either a red, green, or blue filter. One of the most commonly used arrangements is the Bayer pattern, which is named after Bryce Bayer, who worked at Eastman Kodak and filed a patent on the process in 1975. In the Bayer mask twice as many green elements as red or blue are used to mimic the physiologic response of the human eye. The green photosensors serve mostly to record the intensity. They are the luminance-sensitive elements. The red and blue sensors are used to determine the color. They are the chrominance-sensitive elements. In one of the next lectures, which will be devoted to sensors, we will talk about this device in more detail.

The image from the sensor has information about the scene, but it is not directly usable as some of the pixels represent red, green and blue values. In a process which is called demosaicing the separate red, green and blue sub-images must be recovered.

So much for some of the most important RGB input devices. Of course there are many more techniques to record color images. But I think we should not overdo it. The important message here is that different devices detect colors differently. RGB values recorded by one technique can be totally different from RGB values recorded by other techniques. Moreover, sensor elements and their response to the individual R, G, and B levels may vary from manufacturer to manufacturer, or even in the same device over time.

Slide #32




To record digital images two-dimensional sensor arrays, such as CCDs (Charge-Coupled Devices) or CMOS are usually used. Unlike the photosensors of the eye all pixels of a sensor are equally sensitive to the visual spectrum. To convey color sensitivity to the single elements, color filter arrays can be used. Each pixel of the underlying sensor array is covered with either a red, green, or blue filter, representing the fundamental method of RGB additive color synthesis.

Different manufacturers may choose a different arrangement and configuration for the color filter array. One of the most commonly used arrangements is the Bayer pattern, which is named after Bryce Bayer, who worked at Eastman Kodak and filed a patent on the process in 1975. In the Bayer mask twice as many green elements as red or blue are used to mimic the physiologic response of the human eye. The green photosensors serve mostly to record the intensity. They are the *luminance-sensitive elements*. The red and blue sensors are used to determine the color. They are the *chrominance-sensitive elements*. In the next lecture which will be devoted entirely to sensors we will talk about this device in more detail.

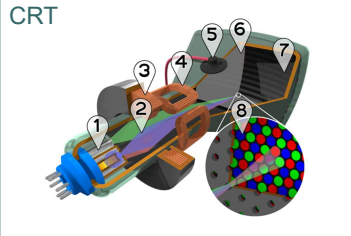
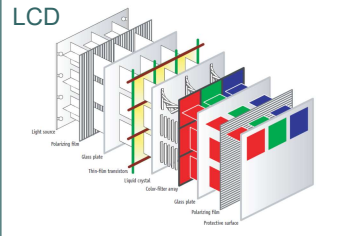
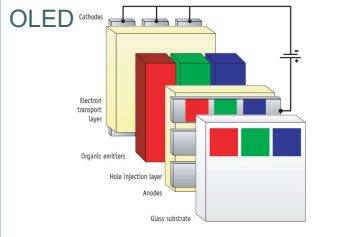
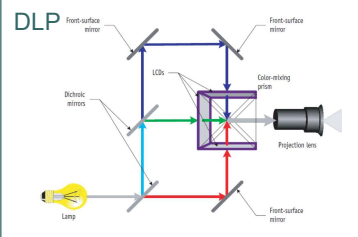
The image from the sensor has information about the scene, but it is not directly usable as some of the pixels represent red, green and blue values. In a process which is called *de-mosaicing* the separate red, green and blue sub-images must be recovered. In addition to de-mosaicking manufacturers employ a number of camera correction operations before saving the image in common image formats. This can include dark noise correction, white balance and color rendering processes.

Slide #33




Color Spaces

RGB output devices

CRT 	LCD 
OLED 	DLP 

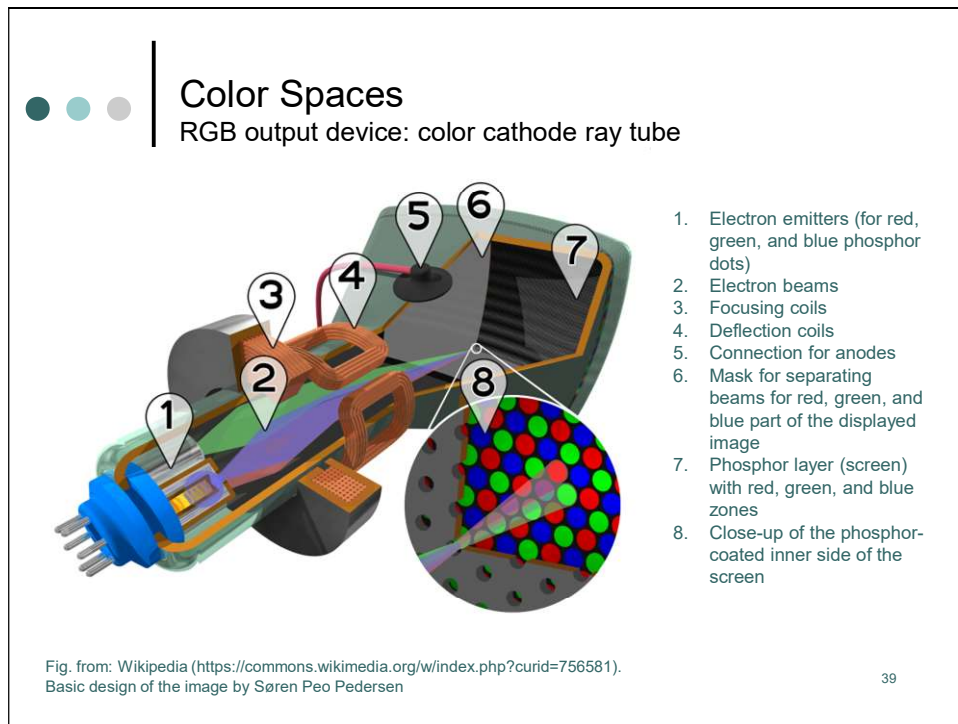
Figures from: Wikipedia and Giorgianni, Madden, Kriss, Digital Color Management-Encoding Solutions, Wiley (2009).

38 

The same applies for RGB output devices. The emission spectrum of the phosphors of a cathode ray tubes differs from that of the organic dyes used in OLEDs. And of course, the filters used in LCDs or DLP systems produce light of different wavelengths.

As a consequence the light reproduced for a given RGB will differ from device to device, and from manufacturer to manufacturer. Since dyes and phosphors bleach emitted light might even change in the same device over time.

Slide #33a (Additional slide, not part of the video)

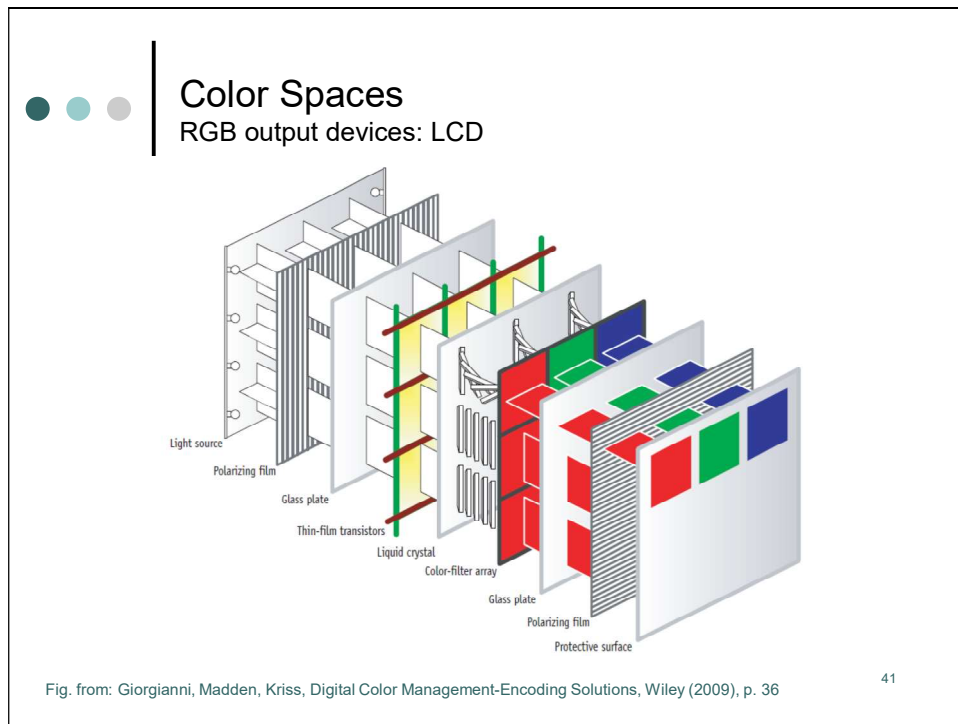


A prototype of an RGB output device is the cathode ray tube. Cathode-ray-tube monitors display images using three different phosphors which emit red, green, and blue light respectively. The phosphors are packed together in stripes or clusters, like in the example shown here (number 8). Most color CRTs have three electron guns, one for each primary color (number 1). Electrons are emitted from the guns at the back of the monitor and are accelerated toward the front panel. On their way the electron beams (number 2) are focused on the screen by focusing coils (number 3). In all “modern” CRT monitors and televisions, the beams are bent by magnetic deflection (number 4). In this way the electron beams are swept across the phosphor dots, making them glow and thereby producing light.

Since the electron beam is swept fast across the screen and the decay rate of the phosphors is long enough the scanning cannot be perceived with the human eye. From normal viewing distances the dots merge to form a continuous image.

An LCD panel consists of a backlight that shines through tiny, liquid crystals and then through red, green and blue filters, From normal viewing distances the color filters are not visible and merge to create a continuous image.

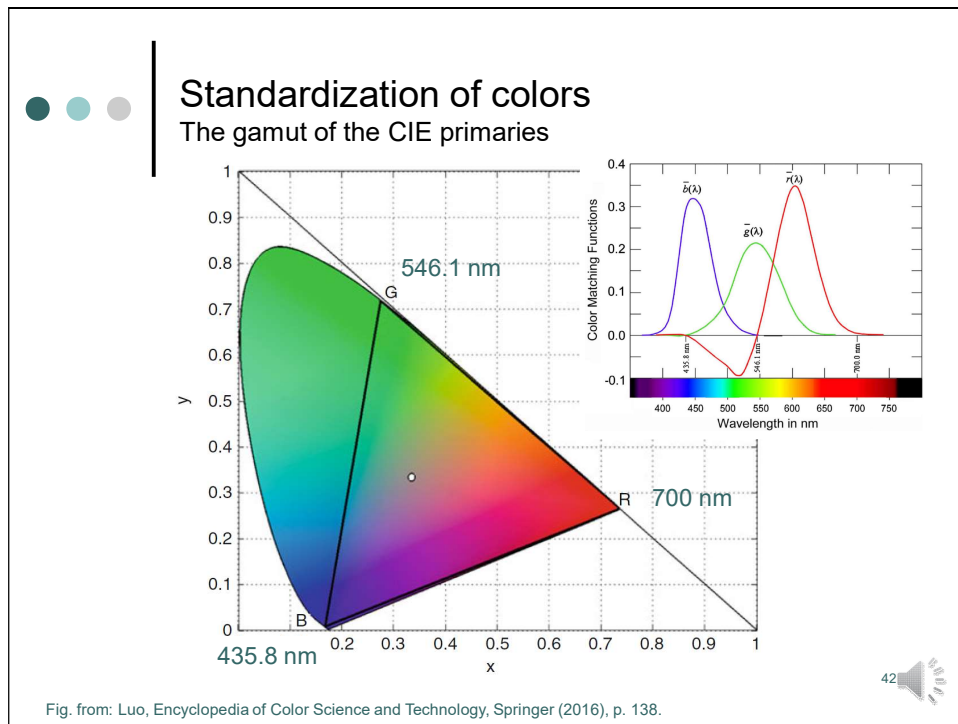
Slide #33b (Additional slide, not part of the video)



In an LCD-based display, *RGB* device code values ultimately determine the intensities of displayed red, green, and blue light by controlling voltages applied across the liquid crystal layer in each pixel

The colorimetric characteristics of the display are determined by the light source used to back-light the display (typically a fluorescent or LED-based source) and the colored filters used on the individual pixels.

Slide #34

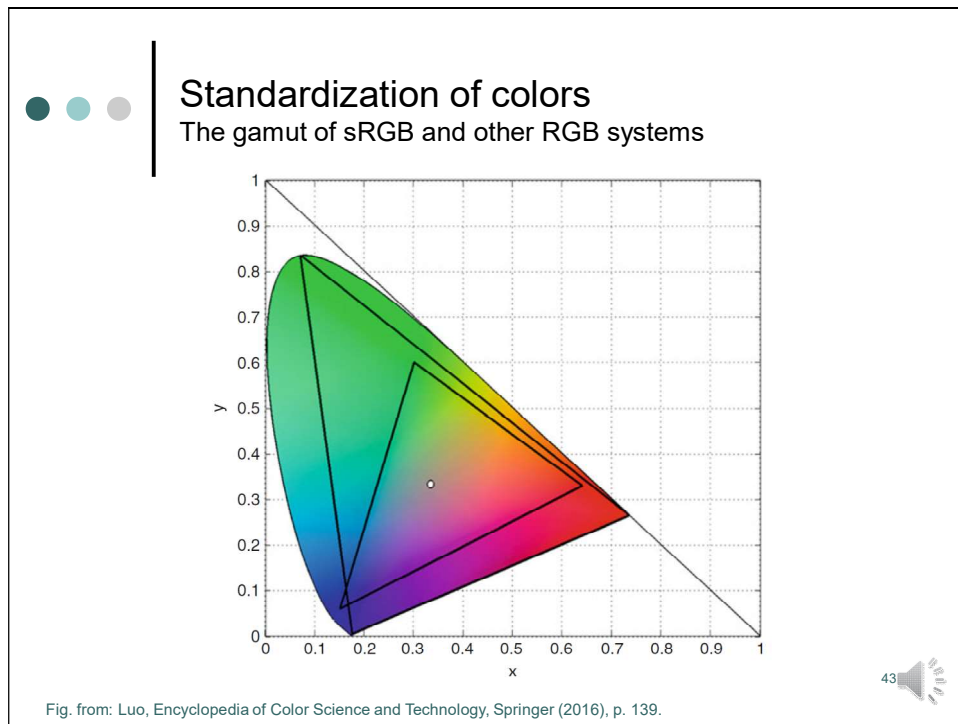


To demonstrate which subset of colors can be represented with RGB systems, we can resort to the CIE diagram, we discussed before. Please, note that the colors shown in this slide are just used for illustration. They do not accurately denote the true color, which is actually impossible, since an RGB monitor can only reproduce a subset of colors as we will see in a minute.

But, let's start with the RGB primaries used by Wright and Guild that formed the basis of the CIE system of 1931. The R was at 700 nm, the G was at 546.1 nm, and the B was 435.8 nm. If two lights are represented in the chromaticity diagram by two points, then chromaticities of the additive mixtures of the two lights will be represented by the straight line that joins the two points. Thus, in this figure all mixtures of the B and G primaries would lie on the straight line joining the chromaticities of the B and G primaries. The same applies for mixtures of G and R or R and B. When three primaries are used all mixtures of the primaries will lie within the triangle, whose corners are defined by the wavelengths of the primaries.

The range of colors that can be matched by a set of primaries is sometimes referred to as the gamut. Accordingly the gamut of an RGB system using the classical primaries would be equal to the triangle shown in this figure. Light in the wavelength range from 435 nm up to 546 nm is not within the gamut of the RGB model using the classical primaries, meaning that the perception of wavelengths within this spectral range cannot be reproduced by these primaries. This makes totally sense when you recall the color matching functions of the three primaries. To have a match between the field of the mixed primaries and the reference field, the test persons needed to add red to the reference field, which is equal to subtracting red from the matched field. This is mathematically possible and resulted in the negative contributions in the color matching functions. But it's impossible in the real world, so that wavelengths in the range where red is negative cannot be reproduced by these primaries. As a consequence the gamut of colors of an RGB system using the classical primaries is considerably smaller.


Slide #35



The gamut of the CIE RGB primaries was already small, but the gamut of the standard RGB system (sRGB) created by Hewlett Packard and Microsoft in 1996 to be used on monitors, printers, and the Web is still smaller. The gamut is represented by the small triangle in this figure. This RGB space is still today the "default" color space for images that contain no color space information.

The largest gamut that can be achieved by any RGB system can be obtained by using primaries whose chromaticities are located at the most distant points of the horseshoe, like 400, 520, and 700 nm. But this would still leave parts of physically realizable colors outside of gamut of the RGB triangle. No matter how carefully three primaries are selected, the gamut (represented by a triangle in the chromaticity diagram) will always be a subset of all possible colors.


Slide #36



Color Image Processing in MATLAB

Review: Image types in MATLAB

Image Type	Description
Binary Image	Image data are stored as an m -by- n logical array. Array values of 0 and 1 are interpreted as black and white, respectively.
Grayscale / intensity	Image data are stored as an m -by- n numeric array whose elements specify intensity values. <ul style="list-style-type: none">• For single or double arrays, values range from [0, 1].• For uint8 arrays, values range from [0,255].• For uint16, values range from [0, 65535].• For int16, values range from [-32768, 32767].
RGB Image	Image data are stored as an m -by- n -by-3 numeric array whose elements specify the intensity values of one of the three color channels. For RGB images, the three channels represent the red, green, and blue signals of the image. <ul style="list-style-type: none">• For single or double arrays, RGB values range from [0, 1].• For uint8 arrays, RGB values range from [0,255].• For uint16, RGB values range from [0, 65535].
Indexed Image	Image data are stored as an m -by- n numeric matrix whose elements are direct indices into a color map. Each row of the color map specifies the red, green, and blue components of a single color. <ul style="list-style-type: none">• For single or double arrays, integer values range from [1, p].• For logical, uint8, or uint16 arrays, values range from [0, p-1]. The colormap is a c -by-3 array of class double.

 46

Now, let's discuss how images are handled in MATLAB.

In MATLAB images are characterized by their type and their class. MATLAB supports four image types: Binary, Gray-scale, RGB images and indexed images. Class refers to the class of the pixels. An image is an array of pixels or elements of the same class.

Last semester we have applied most image processing operations on binary or gray-scale images, so our initial focus was on these two image types. Let's recap, what we have talked about:

- A binary image is a logical array of 0s and 1s. (Thus, an array of 0s and 1s whose values are of data class, say, uint8, is not considered a binary image in MATLAB). A numeric array is converted to binary using function logical.
- A gray-scale image is a data matrix whose values represent shades of gray. When the elements of a gray-scale image are of class uint8 or uint16, they have integer values in the range [0, 255] or [0, 65535], respectively. If the image is of class single or double the values are floating-point numbers usually being in the range between 0 and 1.
- For the RGB images the same applies as for the grayscale images, except for the fact that we have a layer for each color. The next slide shows this in more detail.
- Indexed images will be discussed in one of the next lectures.

Slide #37

Color Image Processing in MATLAB
RGB image format

Pixel values can be retrieved in the following way:
R = (row, column, 1)
G = (row, column, 2)
B = (row, column, 3)

Pixel position:
(row, column)

Blue component image
Green component image
Red component image

47


Fig. from: Gonzales, Woods, Eddins, Digital Image Processing using MATLAB, 2nd ed, Mac Graw Hill, Chennai (2010)

In MATLAB an RGB color image is an $M \cdot N \cdot 3$ array of color pixels.

In MATLAB an RGB image may be viewed as a "stack" of three gray-scale images that, when fed into the red, green, and blue inputs of a color monitor, produce a color image on the screen.

Like in grayscale images, the pixel position corresponds to the row and the column of the matrix. The color is defined by the layer. RGB values can be retrieved, by indexing the array in the usual way.

Slide #38



Color Spaces


RGB images in MATLAB

A layer can be extracted by indexing the array:

```
redLayer = rgbImage(:,:,1);  
greenLayer = rgbImage(:,:,2);  
blueLayer = rgbImage(:,:,3);
```

Individual layers can be merged to an RGB image by using the concatenate command:


```
rgbImage = cat(3, redLayer, greenLayer, blueLayer);
```



In this way also layers can be extracted. To this aim we have to use the colon operator. In this way we refer to all rows, all columns, and just specify the number of the layer.

The opposite operation, this means joining individual layers to an RGB image, can be achieved by using the concatenate command.


Slide #39



Color Spaces

Displaying images in MATLAB

Syntax	Description
<code>image(img);</code>	Displays the image stored in the variable <code>img</code> . If <code>img</code> is a vector or a matrix, the elements of <code>img</code> are displayed in a color which is defined in the colormap of the associated axes. If <code>img</code> is a 3-D array of RGB triplets, the image is displayed as an rgb image.
<code>imshow(img)</code>	displays the image <code>img</code> in a figure. <code>imshow</code> handles binary, grayscale, rgb and indexed images. <code>imshow</code> uses the default display range for the image data type and optimizes figure, axes, and image object properties for image display.
	See MATLAB help page for further information




Matlab has two commands that can be used to display images.

image is part of the MATLAB standard package, ***imshow*** comes with the image processing toolbox.

In principle *imshow* is the most versatile command. It can be used to display binary, grayscale, rgb and indexed images.

image works on RGB images as well. When it is applied on a grayscale image the image will be displayed in a color which is defined by the colormap of the associated axes. How colormaps work, is still something we need to discuss in some future lectures.


Slide #40



Color Spaces

Loading and saving images in MATLAB

Syntax	Description
Loading images	
<code>img = imread(filename);</code>	reads the image from the file specified by <code>filename</code> , guessing the format of the file from its contents. If <code>filename</code> is a multi-image file, then <code>imread</code> reads the first image in the file
<code>img = imread(filename,fmt);</code>	specifies the format of the file with the standard file extension indicated by <code>fmt</code>
<code>img = imread(filename,idx);</code>	reads the specified image or images from a multi-image file (GIF, CUR, ICO, TIF, and HDF4)
	See MATLAB help page for further information




To load images into the memory the command ***imread*** can be used.

filename specifies the name of the file, and **fmt** specifies the format.

If no format is specified MATLAB guesses the format from its content.

If more than one image is contained in a file, which can be the case with multi-image files, individual images can be specified by their index. If no index is specified the first image is extracted.


Slide #41



Color Spaces

Loading and saving images in MATLAB

Syntax	Description
<code>imwrite(img, filename)</code>	writes image data stored in the variable <code>img</code> to the file specified by <code>filename</code> , inferring the file format from the extension. <code>imwrite</code> creates the new file in your current folder. The bit depth of the output image depends on the data type of <code>img</code> and the file format. For most formats: If <code>img</code> is of data type <code>uint8</code> , then <code>imwrite</code> outputs 8-bit values. If <code>img</code> of data type <code>uint16</code> and the output file format supports 16 bit data (JPEG, PNG, and TIFF), then <code>imwrite</code> outputs 16 bit values. If the output file format does not support 16-bit data, then <code>imwrite</code> returns an error. If <code>img</code> is a grayscale or RGB color image of data type <code>double</code> or <code>single</code> , then <code>imwrite</code> assumes that the dynamic range is <code>[0,1]</code> and automatically scales the data by 255 before writing it to the file as 8-bit values. If the data in <code>img</code> is <code>single</code> , then the user has to convert <code>img</code> to <code>double</code> before writing to a GIF or TIFF file. If <code>img</code> is of data type <code>logical</code> , then <code>imwrite</code> assumes that the data is a binary image and writes it to the file with a bit depth of 1, if the format allows it. Valid formats are BMP, PNG, or TIFF

See MATLAB help page for further information 

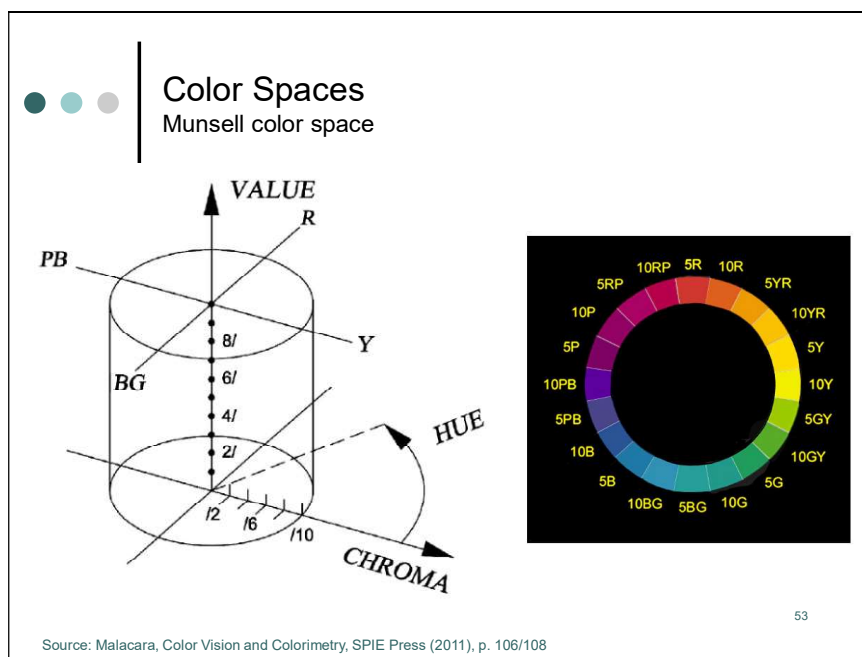
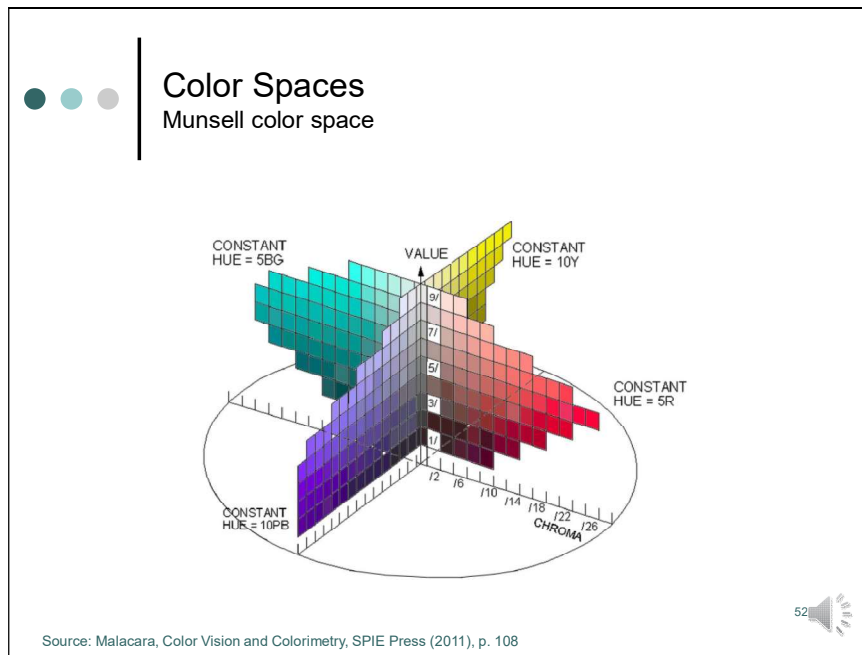
To save an image the command *imwrite* can be used.

img specifies the variable, in which the image data are stored. *filename* specifies the name of the file to which the data should be saved. The image format is specified by the extension of the filename.

Depending on the image format some specific rules might apply, which are listed here and also in more detail in the MATLAB help pages.

So much for the RGB images and how they can be handled in MATLAB.

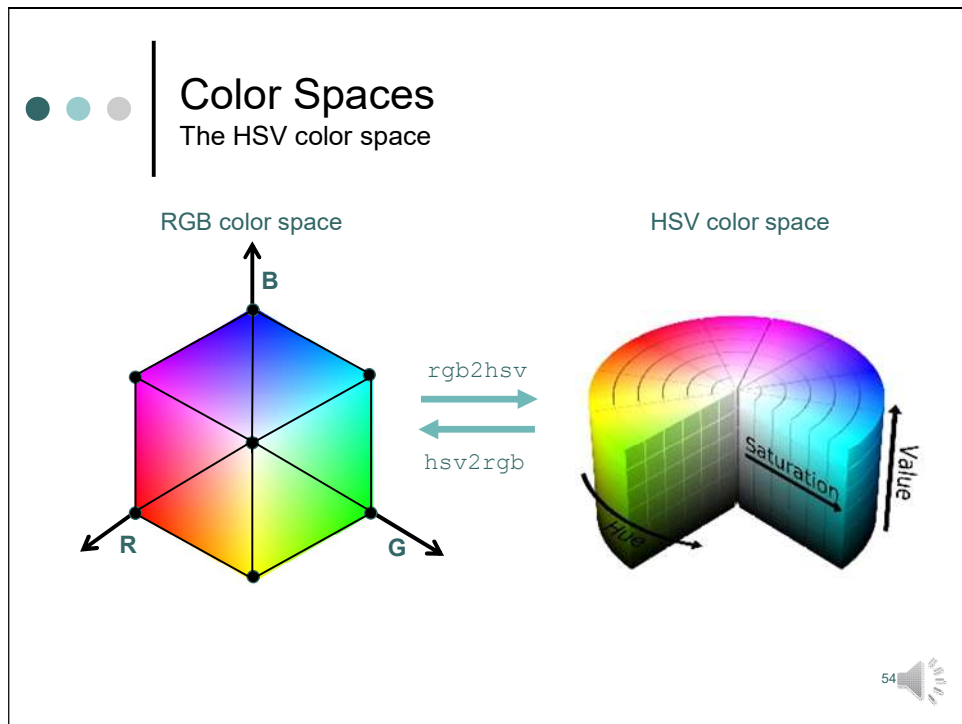
Slide #42



Historically, also artists have been interested in creating some systematic, ordered arrangement of colors. The painter Albert Munsell started in 1905 the process for the “quantification” of colors. The Munsell Color System treats color in accordance with the basic attributes of color, i.e. hue, saturation and lightness/luminance.

Munsell’s aim was to have a system of specifying color that was both visual and numerical. A fundamental principle of his system was equal spacing of colors. He took considerable effort in arranging the patches so that the difference between any two adjacent patches is always visually the same. Thus two side-by-side patches in the red region create the same visual color difference as two side-by-side patches in the green region. Furthermore, adjacent horizontal and vertical chips create a similar visual color difference.

Slide #43



Many color spaces follow this idea. One example is the HSV color space (Hue, Saturation, Intensity).

In the HSV system colors are defined by three values: hue, saturation value. As hue varies from 0 to 1.0, the corresponding colors vary from red through yellow, green, cyan, blue, magenta, and back to red, so that there are actually red values both at 0 and 1.0. As saturation varies from 0 to 1.0, the corresponding colors vary from unsaturated (shades of gray) to fully saturated (with no white component). As value, or brightness, varies from 0 to 1.0, the corresponding colors become increasingly brighter.

The HSV system is closer than the RGB system to the way in which humans experience and describe color sensations. In artists' terminology, hue, saturation, and value refer approximately to tint, shade, and tone.

The transition to the HSV color space can be imagined by looking at the RGB color cube along the gray axis (i.e. the axis joining the black and white vertices), which results in a hexagonally shaped color palette.

Hue is expressed as an angle around a color hexagon, saturation is expressed by the distance to the gray axis, and the value component is measured along the central axis.

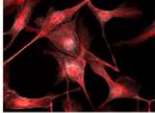
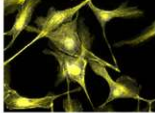
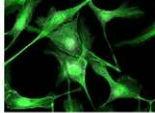
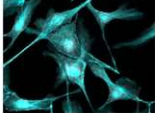
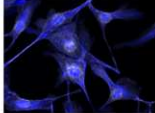
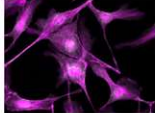
The functions `rgb2hsv` and `hsv2rgb` convert images between the RGB and HSV color spaces. The HSI color space (Hue, Saturation, Intensity) is similar to the HSV color system. However, conversion to HSI is not implemented in MATLAB.


Slide #44

● ● ●

Color Image Processing in MATLAB


Effect of varying the hue in the HSV color space

hue = 0.00 	hue = 0.17 	hue = 0.33 
hue = 0.50 	hue = 0.67 	hue = 0.83 

55 

The hsv system can be useful to change the hue, saturation or brightness of pixels directly. This example shows how the hue can be changed easily by converting an rgb image to the the hsv color space, setting the hue to specific values and converting the image back to the rgb color space.

Slide #45



Color Spaces

The YIQ color space

The YIQ color system was established for color television sets by the National Television Systems Committee (NTSC) in the United States.


Color image data consist of three components:

- luminance (Y),
- hue (I) and
- saturation (Q).

The YIQ components can be obtained from the RGB components of an image using the following linear transformation:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{rgb2ntsc}$$

Similarly, the RGB components can be obtained from the YIQ components by multiplication with the inverse matrix:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.621 \\ 1 & -0.272 & -0.647 \\ 1 & -1.106 & 1.703 \end{bmatrix} * \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad \text{ntsc2rgb}$$


The YIQ color system was established for color television sets by the National Television Systems Committee (NTSC) in the United States. Since color cathode ray tubes use red, green and blue phosphors, it seems (at least at the first sight odd) that the RGB system was not used. However, switching to the RGB system would have meant that old black&white TVs would not have been able to receive signals broadcasted for color televisions.

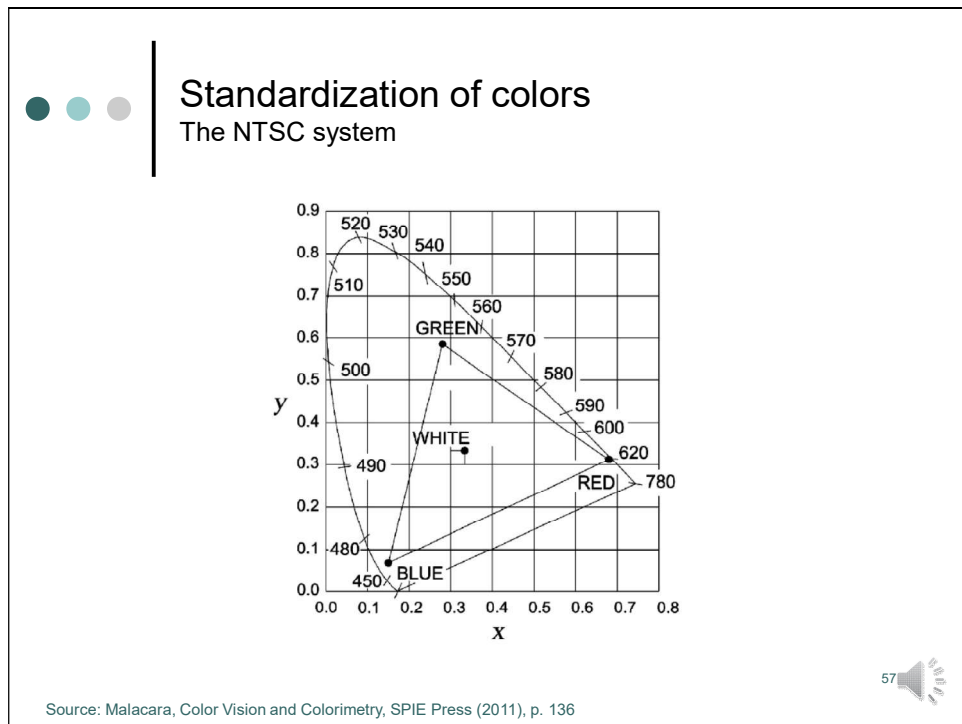
One of the main advantages of the YIQ color space is that grayscale information is separated from color data, so the same signal can be used for both color and black and white sets.

In the NTSC color space, image data consists of three components: luminance (Y), hue (I), and saturation (Q). The first component, *luminance*, represents grayscale information, while the last two components make up *chrominance (color information)*.

The YIQ components can be obtained from the RGB components of an image using the following linear transformation. Note that the elements in the first row sum to 1. They indicate the contribution of the red, green and blue component to the overall luminance. The elements of the 2nd and 3rd row sum up to zero. This is expected, since in a grayscale image R,G, and B are equal, and I and Q should be 0.

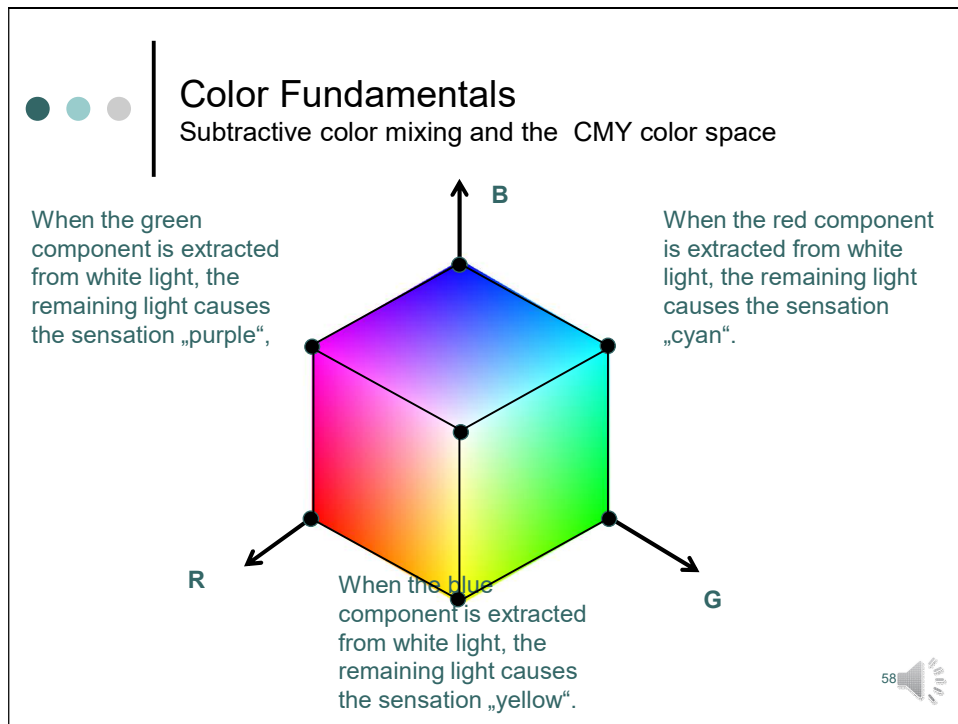
The function `rgb2ntsc` converts `colormaps` or RGB images to the NTSC color space. `ntsc2rgb` performs the reverse operation. Because luminance is one of the components of the NTSC format, the RGB to NTSC conversion is also useful for isolating the gray level information in an image. In fact, the toolbox functions `rgb2gray` and `ind2gray` use the `rgb2ntsc` function to extract the grayscale information from a color image.

Slide #46



This slide shows how the gamut of the NTSC system maps to the CIE chromaticity diagram. As expected not all of the colors which can be seen with our eye can be fully reproduced by color television sets.

Slide #47



So much for color models based on additive color mixing.

For color-printing processes, however, subtractive color mixing comes into play. The primary colors mostly used for subtractive color mixing are cyan, magenta, and yellow.

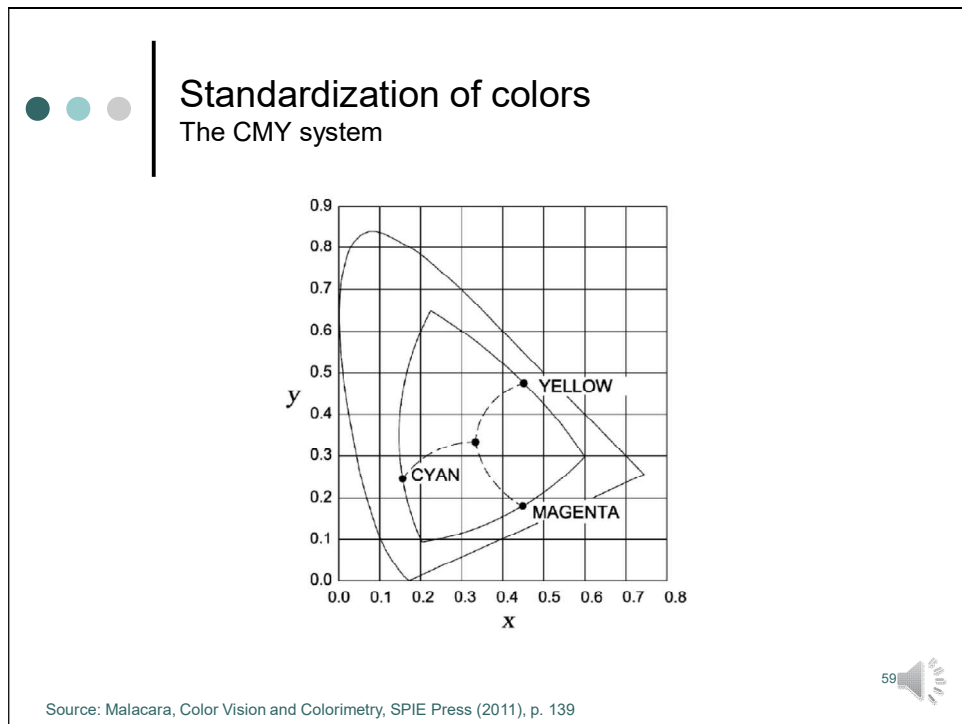
As previously discussed

- a cyan filter extracts red light from the white light and leaves the blue and green part of the spectrum,
- a yellow filter extracts blue from white light, and leaves the red and green part of the spectrum.
- Magenta extracts green and leaves blue and red.

With these colors we have what is known as the cyan/magenta/yellow (CMY) system. Ideally, cyan, magenta, and yellow are sufficient to generate a wide range of colors by the subtractive process.

In principle, equal amounts of cyan, yellow and magenta should produce black. But, since none of the filters is perfect, a dark brown color is generated. For this reason, a fourth real black ink is added in many printing processes to obtain a true black color. Black is designated as “K”, and the color system is called accordingly the CMYK system.

Slide #48




The process of subtractive color mixing can also be nicely demonstrated in the CIE diagram.

- A cyan filter extracts red light from the white light and leaves the blue and green part of the spectrum,
- A yellow filter extracts blue from white light, and leaves the red and green part of the spectrum.
- Magenta extracts green and leaves blue and red.

By mixing these primaries a color gamut represented by the triangle can be obtained.

Slide #49



Color Spaces Overview

Hardware dependent models

Additive color models


- RGB (red, green, blue) models
 - sRGB (1996)
 - Adobe RGB (1998)
- HSI (hue, saturation, intensity) models
 - HSI (hue, saturation, intensity)
 - HSV (hue, saturation, value) (1983)
 - NTSC (luminance (Y), hue (I), saturation (Q))

Subtractive color models

- CMY (cyan, magenta, yellow) (1983)
- CMYK (cyan, magenta, yellow, black) (1983)

Hardware independent models

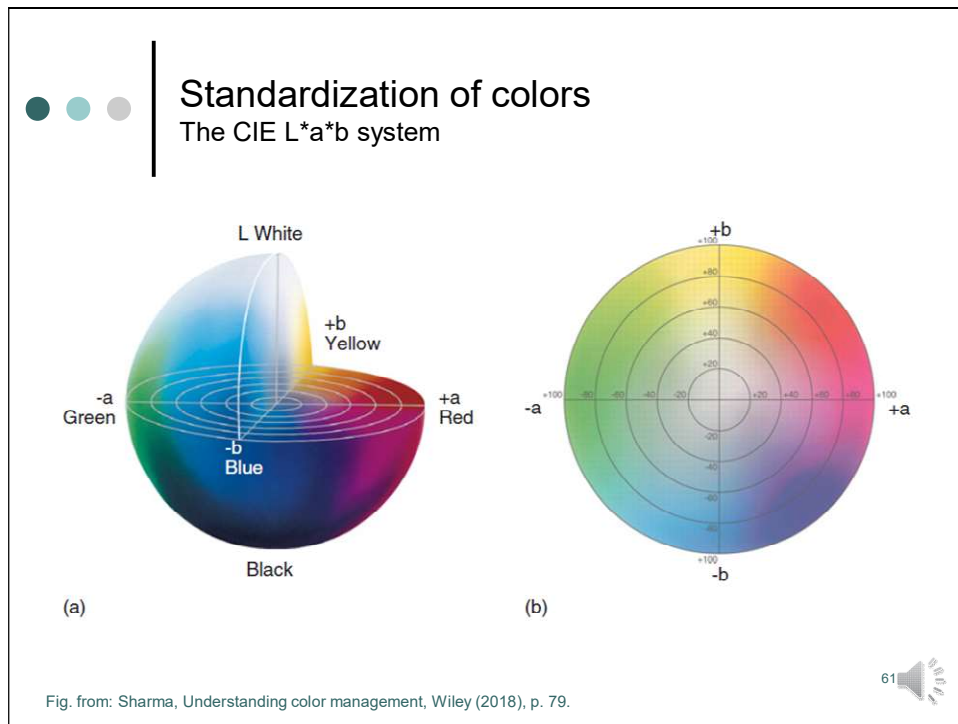
- CIE XYZ (1931), CIE (1964)
- CIE L*a*b (1976)
- CIE L*C*h



By now we have discussed all the hardware dependent models on our list.

Now let's jump to the hardware independent models. The first one listed here is the CIE XYZ system we have talked about in great detail, so that there is no need to repeat it here. Instead we should have a look at some newer models, which also have been established by the Commission International d'Eclairage.

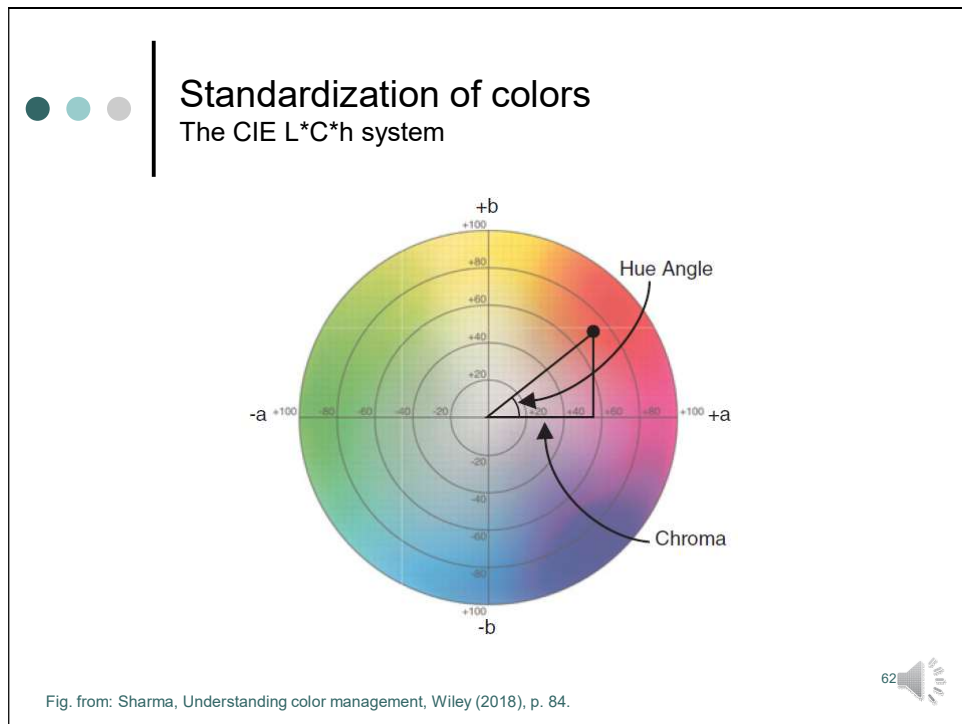
Slide #50



The **CIELAB color space** (also known as **CIE L*a*b*** or sometimes abbreviated as simply "Lab" color space) was defined by the International Commission on Illumination (CIE) in 1976. It expresses color as three values: L^* for the lightness from black (0) to white (100), a^* from green (–) to red (+), and b^* from blue (–) to yellow (+). CIELAB was designed such, that the same amount of numerical change corresponds to roughly the same amount of visually perceived change.

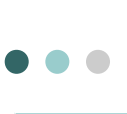
With respect to a given white point, the CIELAB model is device-independent. This means, it defines colors independently of how they are created or displayed. Since the CIELAB gamut includes both the gamuts of the RGB and the CMYK color models, the CIELAB color space is also often when graphics have to be converted from the RGB to CMYK system, and vice versa.

Slide #51



The CIE LCh color space is similar to the CIE Lab color space. Here, instead of the Cartesian coordinates a^* and b^* , the cylindrical coordinates C^* (chroma or relative saturation) and h (for the angle of the hue in the color wheel) are specified. The lightness L^* remains unchanged.

Slide #52



Color Image Processing in MATLAB

Conversion between color spaces

Color space	Application	Conversion between color spaces
HSV (hue (H), saturation (S), value (V))	Easy selection of colors	<code>hsv = rgb2hsv(rgb)</code> converts RGB images or colormaps to the HSV color space. <code>rgb = hsv2rgb(hsv)</code> converts HSV values to the RGB color space.
YIQ (luminance (Y), hue (I), and saturation (Q))	The YIQ color space is used in televisions in the United States. The standard is defined by the National Television Systems Committee (NTSC)	<code>YIQ = rgb2ntsc(rgb)</code> converts RGB images or colormaps to the NTSC color space. <code>rgb = ntsc2rgb(YIQ)</code> performs the reverse operation.
YCbCr (luminance (Y), chrominance (Cb,Cr))	The YCbCr color space is widely used for digital video. (Cb represents the difference between the blue component and a reference value. Cr represents the difference between the red component and a reference value)	<code>video = rgb2ycbcr(rgb)</code> converts RGB images or colormaps to the YCbCr color space. <code>rgb = ycbcr2rgb(video)</code> performs the reverse operation.
CIE XYZ	Device independent, widely used color space, created by the CIE in 1931	<code>XYZ = rgb2xyz(rgb)</code> converts RGB images to CIE 1931 XYZ coordinates. <code>rgb = xyz2rgb(xyz)</code> performs the reverse operation
CIE L*a*b	Device independent, perceptually uniform color space, created by the CIE in 1976.	<code>lab = rgb2lab(rgb)</code> converts RGB values to CIE 1976 L*a*b* values. <code>rgb = lab2rgb(lab)</code> converts RGB values to CIE 1976 L*a*b* values. An RGB color is out of gamut when any of its component values are less than 0 or greater than 1.

As discussed before MATLAB handles colors as RGB values.

To display image data that have been processed in other color spaces, they must be transformed to the RGB system.

The MATLAB image processing toolbox provides a variety conversion functions from the RGB to other color spaces. This includes device dependent color spaces such as the HSV colorspace, the YIQ color space used for televisions or the YCbCr color space for videos.

Also conversion to device independent color spaces like the CIE standards is possible.