

Zusammenfassung

In der folgenden Arbeit werden einige Farbmodelle für digitale Bilder vorgestellt und erklärt, wofür sie verwendet werden. Außerdem wird über deren Konvertierungen und mögliche Probleme, die dabei auftreten können, berichtet.

1 Einleitung

Mit der Erstellung digitaler Bilder wird versucht, die reale Welt so originalgetreu wie möglich abzubilden. Dabei stößt man farblich sehr schnell an seine Grenzen, da es viele Geräte gibt, die die Originalfarben eines Objekts o. Ä. nicht darstellen können. Aufgrund dessen haben sich Organisationen wie z. B. die Internationale Beleuchtungskommission (CIE¹) Farbmodelle überlegt, die einen bestimmten Farbbereich abdecken und sich beispielsweise durch Veränderung der Helligkeitswerte gegenseitig erweitern. Im Folgenden werde ich kurz allgemein über Farbmodelle berichten, dann genauer auf die Modelle RGB, HSV und YUV, YPbPr und YCbCr eingehen und erklären, wie eine Farbraumkonvertierung berechnet und mithilfe des Programms „Inviwo“² umgesetzt wird. Abschließend gehe ich auf dabei entstehende Probleme und mögliche Limitierungen ein und fasse das Ergebnis kurz zusammen.

2 Grundlagen

Bevor man sich mit dem Thema Farbkonvertierung näher beschäftigen kann, müssen einige wichtige Begriffe erklärt werden. Abb. 1: CIE-Normvalenzsystem

In einem Farbmodell werden die Farben digitaler Bilder nach einer bestimmten Methode beschrieben. Jeder Farbe ist ein Farbort zugeordnet, welcher ein Punkt mit dazugehörigem Vektor in einem drei- oder vierdimensionalen Farbraum ist und die Farbe eindeutig beschreibt. Der Betrag dieses Vektors wird als Farbwert bezeichnet. Ein Farbraum, auch Gamut genannt, wird über einem Farbmodell aufgespannt und umfasst eine Teilmenge der Farben, die der Mensch wahrnehmen kann. Daher kann ein Farbmodell mehrere Farbräume enthalten. Jedes Gerät, das zur Bildverarbeitung genutzt wird, beispielsweise Drucker oder Digitalkameras, besitzt einen eigenen Farbraum. Mit diesem können sie nur die gewünschten Farben darstellen, die in ihrem Farbraum enthalten sind, wodurch sich die Farben eines Bildes beim Übertragen von einem Gerät zum Anderen wesentlich ändern können. Farbräume werden häufig in Form von Diagrammen (meist

¹CIE = Commission internationale de l'éclairage

²Interactive Visualization Workshop

im CIE-Normvalenzsystem) oder verschiedenen Farbkörpern, z. B. Würfel oder Kegeln, dargestellt. Als Beispiel sieht man in Abbildung 1 eine Darstellung des Standard-RGB-Farbraums (sRGB) im CIE-Normvalenzsystem, wobei das eingezeichnete Dreieck den Farbraum beschreibt. Das CIE-Normvalenzsystem stellt den XYZ-Farbraum dar, der alle vom Menschen registrierbaren Farben umfasst. Dementsprechend werden andere Farbräume meist in den XYZ-Farbraum gezeichnet.

Abb. 2: Prinzip der additiven Farbmischung

Farbräume werden außerdem in additive und subtraktive Farbmischung unterteilt. Bei der additiven Farbmischung wird rotes, grünes und blaues Licht, die Primärfarben des Lichts, auf einen Punkt gebündelt, wodurch weißes Licht reflektiert wird (s. Abb. 2). Entfernt man alle Lichter, erhält man Schwarz. Um andere Farbtöne, sogenannte Mischfarben, zu erzeugen, werden zum weißen Licht zwei der Primärfarben hinzuaddiert. Beispielsweise erhält man durch Addition von Rot und Grün als Reflexion Gelb. Die additive Farbmischung entspricht der Farbwahrnehmung des Menschen und wird bei vielen Farbmodellen verwendet, u. a. beim RGB-Farbmodell.

3 Farbmodell, Konvertierung und Implementierung

3.1 Vorstellung der Farbmodelle

3.1.1 Das Farbmodell RGB

Das menschliche Auge nimmt Farben mittels Rezeptoren wahr, die die sichtbaren Farben in Rot, Grün und Blau aufteilen. Daher stellt das Rot-Grün-Blau-Farbmodell (RGB-Farbmodell) die Farben des menschlichen Sehens am genauesten dar. Zum RGB-Modell gibt es verschiedene Farbräume, die alle den sRGB-Farbraum erweitern (für Genaueres siehe [?]). Die einzelnen Werte der RGB-Komponenten liegen zwischen 0 und 255, wobei, wenn alle Werte 0 sind, schwarz und wenn alle Werte 255 sind, weiß erzeugt wird (s. Abb. 2). Dementsprechend erhält man die reine Form einer der drei Farben, indem die gewünschte Farbe auf 255 und die anderen beiden auf 0 gesetzt werden. Wie hoch die Werte sind, hängt von den verwendeten Geräten ab. Am Computer wird meistens das RGB-Modell verwendet, da der Großteil der Computer RGB-Farbräume unterstützen. Durch die Aufteilung auf die Farben Rot-Grün-Blau und die Multiplikation aller möglicher Farbwerte ist es theoretisch möglich, 16,7 Millionen verschiedene Farben zu erzeugen.

3.1.2 Das Farbmodell HSV

Das Hue-Saturation-Value-Farbmodell, kurz HSV-Modell, ist ein Farbmodell, das dem Benutzer eine intuitive Farbauswahl ermöglichen soll. Dabei werden die Farben so gut wie möglich der menschlichen

Abb. 3: HSV-Farbkörper

Beschreibung von Farben nachempfunden, wobei sie in Farbton (hue), Sättigung (saturation) und Helligkeit (value) eingeteilt werden und damit das RGB-Modell erweitern. Der dazugehörige Farbkörper wird oft als um 180 Grad gedrehter Kegel dargestellt (s. Abb. 3): Die Grundfläche beschreibt die Farben des RGB-Farbraums, die in Winkeln angegeben werden (rot ist 0 Grad), die Tiefe oder Höhe des Kegels legt die Helligkeit fest (die Spitze ist schwarz, der Mittelpunkt der Grundfläche weiß) und der Radius von innen (0) nach außen (1) die Sättigung. Da beim HSV-Modell die Farben durch das Mischen von Primärfarben mit anderen Farbteilen entstehen und den vom Menschen wahrgenommenen Farben sehr nahe kommen, wird das HSV-Modell oft bei Grafikdesign verwendet.

3.1.3 Die Farbmodelle YUV, YPbPr und YCbCr

Das YUV-Farbmodell wurde zur Bildübertragung beim analogen Farbfernsehen nach den Normen des NTSC³ und des PAL-Verfahrens⁴ entwickelt und basiert auf dem RGB-Farbmodell. Die Darstellung einer Farbe im YUV-Modell setzt sich aus der Luminanz (Y), d.h. dem Helligkeitsanteil, und der Chrominanz (UV), der Farbinformation, der Farbe zusammen. Die Chrominanz unterteilt sich hierbei in die Farbdifferenzen zwischen der Helligkeit und dem Blau- ($U = B - Y$) bzw. Rotanteil ($V = R - Y$). Die Farbdifferenz beim Grünanteil ($G - Y$) wird aus dem YUV-Wert berechnet. Da das menschliche Auge Helligkeit etwa zehnmal intensiver wahrnimmt als Farbe, erfolgt die Bildaufbereitung durch Kolorierung: Farbinformationen in niedriger Auflösung werden über ein voll aufgelöstes Schwarz-Weiß-Bild gelegt. Dadurch benötigt man zur Bildübertragung im RGB-Format nur noch einen Kanal für die Luminanzinformationen.

Das YPbPr-Modell gründet sich ebenfalls auf das RGB-Modell und wird bei der analogen Bildübertragung verwendet, wobei die YPbPr-Werte mit den RGB-Werten vor der Gammakorrektur⁵ berechnet werden. Dabei sind die Pb- und Pr-Werte nicht wie beim YUV-Modell die Differenzen zwischen Helligkeit und Farbanteil, sondern die Abweichung von Grau zu den Farbanteilen: Pb ist der Wert der Differenz zwischen Grau und Blau (+0,5) und Gelb (-0,5), Pr gibt den Unterschied zu Rot (+0,5) und Cyan (-0,5) an. Gelb und Cyan sind hierbei die Komplementärfarben.

Äquivalent zum YPbPr-Modell ist bei der digitalen Bildübertragung das YCbCr-Modell: Der Cb-Wert entspricht dem des Pb-Wertes (Grau-Abweichung zu Blau und Gelb) und der Cr-Wert dem des Pr-Wertes (Grau-Abweichung zu Rot und Cyan). Das YCbCr-Modell wird mittlerweile zusätzlich zur Fernsehbildübertragung auch für Videoaufzeichnungen (DVDs o. Ä.) und Einzelbilder, z. B. im JPEG-Format, verwendet.

³für Details siehe [?]

⁴für Details siehe [?]

⁵für Details siehe [?]

3.2 Umrechnung und Implementierung

3.2.1 Allgemein

Umgesetzt wurden folgende Farbmodell-Konvertierungen: RGB nach HSV und RGB nach YCbCr. Da Inwivo die eingelesenen Bilder nach der Konvertierung wieder im RGB-Farbraum anzeigt, wurden zur Überprüfung auch die Rück-Konvertierungen HSV nach RGB und YCbCr nach RGB implementiert. Zur genauen Betrachtung der einzelnen Komponenten des Bildes im YCbCr-Farbraum gibt es zusätzlich die Möglichkeit, das importierte Bild so auszugeben, dass alle Komponenten des Vektors gleich sind (Y-, Cb- oder Cr-Wert). Die einzelnen Konvertierungen wurden zur Übersicht jeweils in eigene Methoden über die main-Methode in der Shader-Datei ausgelagert und in der main-Methode aufgerufen. Außerdem sind die Abfragen verschiedener Fälle, die eintreten können (refRGB nach HSV und umgekehrt), als if-Abfragen formuliert, da man in der Shader-Datei keine switch-cases verwenden kann. Die bei den HSV-Rechnungen verwendeten Datentypen sind überwiegend floats, bei der YCbCr-Matrizenmultiplikation Matrizen und Vektoren des dreidimensionalen Vektorraums der reellen Zahlen; über- und zurückgegebene Parameter sind ebenfalls dreidimensionale Vektoren (vec3), aus denen die RGB-Werte herausgelesen werden. Alle Variablen werden mit 0.0 initialisiert, damit Fehler, bei denen Parameter nicht existieren oder falsch berechnet werden, sofort erkannt werden können. In Inwivo gibt es für jede Konvertierung bzw. Bilddarstellung eine Auswahloption im Dropdownmenü. Öffnet man die Datei, sind daher sieben Prozessoren mit Bildausgaben (Canvas) eingerichtet, wobei jedem Prozessor eine Option zugeordnet ist.

3.2.2 RGB nach HSV

Die RGB-Werte r, g und b erhält man durch die Funktionen VektorName.r bzw. VektorName.g und VektorName.b als Werte zwischen 0 und 1. Von diesen bestimmt man für die weiteren Rechnungen das Maximum max_value und Minimum min_value mit der entsprechenden Funktion. Dabei muss man zwei Funktionen ineinander verschachteln, da die Funktionen nur das Maximum bzw. Minimum von zwei Zahlen bestimmen können. Mit dem Ergebnis erhält man direkt den value-Wert, der auf das Maximum der RGB-Werte gesetzt wird. Der hue-Wert wird in Abhängigkeit von dem zuvor bestimmten Minimum und Maximum berechnet und in Grad zwischen 0 und 360 angegeben:

$$hue := \begin{cases} 0.0, & \text{for } max_value = min_value \\ 60.0 \cdot \left(0.0 + \frac{g-b}{max_value-min_value}\right), & \text{for } max_value = r \\ 60.0 \cdot \left(2.0 + \frac{b-r}{max_value-min_value}\right), & \text{for } max_value = g \\ 60.0 \cdot \left(4.0 + \frac{r-g}{max_value-min_value}\right), & \text{for } max_value = b \end{cases}$$

Durch diese Rechnungen könnte der Fall eintreten, dass hue kleiner als 0 wird, was durch eine if-Abfrage abgefangen und in der zu hue 360.0 hinzuaddiert wird. Anschließend teilt man den hue-Wert durch 360.0, damit man wieder einen Wert zwischen 0 und 1

bekommt. Der saturation-Wert ist ebenfalls von den Maximum- und Minimumwerten abhängig:

$$satVal := \begin{cases} 0.0, & \text{for } max_value = 0.0 \\ \frac{max_value - min_value}{max_value}, & \text{otherwise} \end{cases}$$

Die Methode `rgb2hsv(vec3 rgbVal)` bekommt also einen dreidimensionalen Vektor, der die RGB-Werte als Einträge hat, und gibt einen dreidimensionalen Vektor mit den HSV-Werten zurück, die dann als RGB-Werte interpretiert und dargestellt werden:

Abb. 4: Originalbild

Abb. 5: Bild nach der Konvertierung

3.2.3 HSV nach RGB

Die HSV-Werte `hue`, `sat` und `val` erhält man, wie oben erklärt, durch den übergebenen Vektor als Werte zwischen 0 und 1. Da der `hue`-Wert in bei den Umrechnungen in Grad angegeben wird, multipliziert man ihn mit 360.0. Falls der `saturation`-Wert gleich 0 ist, sind die RGB-Werte gleich dem `value`-Wert und die Methode gibt einen dreidimensionalen Vektor aus, bei dem jede Komponente gleich dem `value`-Wert ist. Falls nicht, wird der `hue`-Wert durch 60.0 geteilt und als Hilfsvariable ein Integer „`i`“ eingeführt, das mit dem abgerundeten Wert von `hue` initialisiert wird. Dabei wird die `floor()`-Funktion verwendet, die äquivalent zu den Gaußklammern nach unten ist. Zusätzlich zu `i` werden vier weitere Hilfsvariablen definiert:

$$f := hue - i, \quad p := val \cdot (1.0 - sat), \quad q := val \cdot (1.0 - (sat \cdot f))$$

und

$$t := val \cdot (1.0 - sat \cdot (1.0 - f))$$

. Die Bestimmung der RGB-Werte erfolgt nun anhand einer `if`-Abfrage in Abhängigkeit von `i`, die den Vektor mit den RGB-Werten direkt zurückgibt:

$$(r, g, b)^T := \begin{cases} (val, t, p)^T, & \text{for } i = 0 \\ (q, val, p)^T, & \text{for } i = 1 \\ (p, val, t)^T, & \text{for } i = 2 \\ (p, q, val)^T, & \text{for } i = 3 \\ (t, p, val)^T, & \text{for } i = 4 \\ (val, p, q)^T, & \text{for } i = 5 \end{cases}$$

Die Methode `hsv2rgb(vec3 hsvVal)` bekommt also den dreidimensionalen Vektor, der die zuvor berechneten HSV-Werte enthält, und gibt einen dreidimensionalen Vektor mit den zurückgerechneten RGB-Werten zurück. So erhält man wieder das Originalbild:

Abb. 6: Bild im HSV-Farbraum

Abb. 7: Bild nach der Konvertierung

3.2.4 RGB nach full-range YCbCr

Die Konvertierung vom RGB-Farbraum zum YCbCr-Farbraum erfolgt durch Matrizenmultiplikation und ist vom Umfang des YCbCr-Farbraums abhängig, weshalb man zwischen SDTV, HDTV und full-range unterscheidet. SDTV steht für „Standard Definition Television“, HDTV für „High Definition Television“. Dabei umfasst der YCbCr-Farbraum Y-Werte von 16 bis 235 und Cb/Cr-Werte im Bereich von 16 bis 240; bei der full-range Konvertierung sind die Farben des gesamten YCbCr-Farbraums, also mit Werten von 0 bis 255, möglich. Für jede Berechnung sind eine Matrix und ein sogenannter „offset“-Vektor vorgegeben, mit denen man zwischen RGB und YCbCr konvertieren kann. Im Folgenden wird nur die Formel für die full-range Konvertierung betrachtet.

Die Werte des YCbCr-Vektors ergeben sich, indem man auf den vorgegebenen RGB-Vektor von links die Konvertierungsmatrix hinzu multipliziert und den offset-Vektor dazu addiert:

$$\begin{pmatrix} y \\ cb \\ cr \end{pmatrix} = \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \cdot \begin{pmatrix} r \\ g \\ b \end{pmatrix}$$

Da die RGB-Werte am Computer zwischen 0 und 1 liegen, müssen die Komponenten des offset-Vektors durch 256 geteilt werden. Nach der Konvertierung entsteht folgendes Bild:

Abb. 8: Originalbild

Abb. 9: Bild nach der Konvertierung

Zur Übersicht sind in den Abbildungen 14 bis 16 die einzelnen Farbkomponenten getrennt dargestellt (s. Abb. 10, Abb. 11, Abb. 12).

Abb. 10: nur Y

Abb. 11: nur Cb

Abb. 12: nur Cr

3.2.5 Full-range YCbCr nach RGB

Bei der Rück-Konvertierung berechnet man die RGB-Werte ein bisschen anders: Die Konvertierungsmatrix wird zwar von links auf den übergebenen YCbCr-Vektor hinzu multipliziert, allerdings werden vorher von der Cb- und Cr-Komponente des Vektors 128.0 (am Computer wieder 0.5) abgezogen und es gibt keinen offset-Vektor.

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} 1.000 & 1.000 & 1.000 \\ 0.000 & -0.343 & -0.419 \\ 1.400 & -0.711 & 0.000 \end{pmatrix} \cdot \begin{pmatrix} y \\ cb - 128 \\ cr - 128 \end{pmatrix}$$

Nach der Konvertierung erhält man wieder das Originalbild:

Abb. 13: Bild im YCbCr-Farbraum

Abb. 14: Bild nach der Konvertierung

3.3 Limitierungen

Das Farbraum-Konvertierverfahren ist bezüglich der Qualität der Bilder sehr eingeschränkt. Bei den obigen Beispielen hat die Konvertierung hervorragend funktioniert, da das Bild in Originalauflösung eingespeist wurde. Wird das Bild allerdings komprimiert, wodurch viele Farbinformationen verloren gehen, und dann konvertiert, treten Fehler auf: Beim konvertierten Bild kann man einzelne Pixelblöcke erkennen und an manchen Stellen schlägt die Konvertierung fehl, wodurch schwarze Flecken auf dem Bild zu sehen sind (s. Abb. 15). Dadurch gibt es auch Probleme bei der Rück-Konvertierung (s. Abb. 16). Die hier gezeigten Bilder wurden zuerst komprimiert und dann konvertiert.

Abb. 15: RGB nach HSV
Quelle: w.hsv

Abb. 16: HSV nach RGB mit vergrößertem Ausschnitt

4 Fazit und Ausblick

Die digitale Abbildung der Realität ist durch die vorhandenen Farbmodellen schon sehr fortgeschritten, doch leider sind die technischen Einschränkungen im Moment noch sehr hoch: Da fast alle Computer nur Farben im RGB-Farbmodell darstellen, können sich andere Farbmodelle, die der menschlichen Wahrnehmung mehr entsprechen, beispielsweise das HSV-Modell, nicht durchsetzen. Außerdem treten, wie oben beschrieben, bei komprimierten Bildern noch gravierende Fehler auf. Ein weiteres Problem ist, dass die Auflösungen der Bildschirme kontinuierlich verbessert werden; zurzeit sogar schon von 4K (4096 x 2160 Pixel) auf 8K (7680 x 4320 Pixel). Die dadurch erhöhte Pixelanzahl und deren Darstellung fordert von den Geräten enorme Leistungen, die bisher nur von wenigen gewährleistet werden können. Für genauere Informationen siehe [?].

5 Referenzen

Bildquellen

- Abb. 1 <http://www.itwissen.info/definition/lexikon/Standard-RGB-sRGB-standard-RGB.html> (31.01.2017)
- Abb. 2 <https://www.saxoprint.de/blog/der-farbraum-rgb-und-cmyk-im-vergleich/> (31.01.2017)
- Abb. 3 <http://de.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html?requestedDomain=www.mathworks.com> (07.02.2017)
- Abb. 4 Ulrike Kulzer
- Abb. 15 Originalbild: https://www.tutorialspoint.com/java_dip/color_space_conversion.htm (08.02.2017)