

Eine Farbe ist eine Farbe ist eine Farbe oder

Warum sehen meine Bilder denn so dunkel aus ?

von Thomas Engel

Der zufriedene Joe User braust durch das WorldWideWeb und erfreut sich der vielen bunten Bilder, die da langsam über die verstopfte Leitung gekrochen kommen. Doch brr...was ist das ? GNUstep ist ja schon und gut...aber die Jungs sollten erstmal lernen wie man Bilder malt die nicht aussehen als ob im Raum alle Lampen kaputt gegangen sind. Das ist ja alles viel zu dunkel !

Als hilfsbereiter Mensch schickt Herr User gleich eine Nachricht an die Webmasterin und teilt ihr freundlich mit, daß ihre Farben doch sehr dunkel geraten sind. Sie muß sich doch darum kümmern. Danke.

Doch wie sagt sie es den Kindern...millionenköpfiger Schar?

Die bunte Welt der Farben

Nun ist ja die Sache mit den Farben mehr oder minder trivial. Farbe ist was die Banane gelb, das Stop-Schild rot und die Tinte auf den Fingern der Kinder blau macht.

Jedoch ist Farbe etwas sehr subjektives, da das menschliche Gehirn auch hier eine ganze Ecke mitzureden hat.

Menschen nehmen „gleiche“ Farben unterschiedlich wahr und in extremen Fällen können sie manche Farben gar nicht wahrneh-

men oder unterscheiden (rot-grün Blindheit).

Ändern sich die Lichtverhältnisse, so biegt unser Gehirn die physikalisch vorhandenen Farbtöne so um, daß sie nach Möglichkeit dem entsprechen was wir sehen wollen. Setzt man sich z.B. eine gelbe Brille auf, so wird der Himmel nach kurzer Zeit doch wieder blau und der Schnee wieder weiß. Erstaunliches Ding ... dieses Gehirn.

Doch was hat das alles mit Webmasterins Problem zu tun? Joe User trägt sicherlich keine gelbe Brille und wird auch nicht nur auf ihrer Webseite plötzlich farbenblind.

Die bunte Welt der Technik

So verwirrend wie es in der „echten“ Welt ist, ist es auch in der Welt der bits & bytes. Unterschiedliche Ausgabegeräte haben unterschiedliche „Vorstellungen“ von Farbe.

Auf einem Bildschirm werden Farben nach dem RGB-Modell in *additiver* Weise erzeugt (aktives Leuchten), wobei Drucker im Normalfall im CMYK Modus arbeiten und dabei die Arbeit der *substraktiven* Farbmischung überlassen (passives Filtern).

Doch selbst bei gleichem physikalischem Modell ist eine Farbe nicht eine Farbe. Das verwendete Papier kann bei einem Drucker

sehr stark die Farben beeinflussen, je nachdem wie gut es Licht reflektiert oder wie stark es z.B. die Farben aufsaugt.

In den Bildschirmen werden dagegen unterschiedliche Leuchtstoffe verwendet und die Elektronenkanonen der verschiedenen Geräte verhalten sich auch nicht identisch. Selbst innerhalb der gleichen Baureihe kann es Unterschiede geben.

Der Verwirrung nicht genug, hat man nun jedoch noch eine ganze Reihe anderer Probleme, die besonders die Welt der Druckereien auf Trab halten.

Die Farbtöne, die sich mit den verschiedensten Mitteln erzeugen lassen (Photo, RGB, CMYK), sind nicht identisch. So kann ein Bildschirm nicht alle Farben wiedergeben, die ein Diapositiv enthält und der Drucker kann schon gleich gar nicht alles ausdrucken, was man am Bildschirm sieht. Jedoch kann man auch mit dem Drucker Farben erzeugen, die kein Photo oder Bildschirm auf die Reihe bringt (am Ende des Größtenbereiches).

Und was überhaupt nicht geht, ist *alle* physikalisch möglichen Farbtöne mit einem der technischen Geräte darzustellen.

Bei all den Problemen kann man sich doch nur freuen, daß man überhaupt farbige Bilder am Computer zu sehen bekommt, die doch eigentlich recht gut dem entsprechen, was man erwartet.

„Ja...ja“ würde Joe ihr sicherlich antworten. „Das ist ja alles schön und gut, aber ich habe nur einen Bildschirm, die Probleme aus der Welt der Druckereien interessieren mich nicht und zudem sind ja die meisten Bilder im WWW viel besser als Eure.“

Frau Webmasterin konnte ihn mit diesem Ausflug in die Physik sicherlich nicht überzeugen.

Die bunte Welt der Computer

Um mit den von Joe User beschriebenen Problem fertig zu werden, gibt es eine „einfache“ Lösung: *Gamma Korrektur*.

Unter *Gamma Korrektur* versteht man einen Vorgang, bei dem die gespeicherten Farbwerte einer Bilddatei so verdreht werden, daß die bei der Ausgabe unvermeidlichen Farbfehler wieder kompensiert werden. Im Idealfall entstehen so genau die vom Betrachter gewünschten Farbeindrücke – trotz all der am Anfang beschriebenen menschlichen und technischen Besonderheiten.

Nehmen wir ein einfaches Beispiel und betrachten ein Graustufenbild. Bei diesem wird die Farbe Schwarz als 0 codiert und Weiß als 1 codiert (pingelige Zeitgenossen werden hier anmerken wollen, daß Schwarz und Weiß keine Farben im physikalischen Sinne sind – doch das sind die schon oft gespaltenen Haare).

Alle Grautöne dazwischen sollen linear in ihrer Helligkeit den Zahlenwerten entsprechen, d.h. „mittelgrau“ würde einem Wert von 0.5 entsprechen, was wir im Folgenden als Grau bezeichnen. In der Welt der Farbeimer entspricht das anschaulich einer 1:1 Mischung aus Schwarz und Weiß.

Betrachten wir nun den Weg eines Bildpunktes der Farbe Grau

(0.5) auf seinem idealisierten Weg zum Bildschirm:

Der Bildschirmtreiber liest den Wert 0.5 aus dem Bildspeicher aus und schickt ihn zur Grafikkarte. Diese legt entsprechend eine 50%ige Stromspannung an die Leitung zum Bildschirm, wo eine Elektronenkanone mit 50%iger Intensität auf die armen Phosphorpunkte ballert. Der betroffene Punkt leuchtet mit 50%iger Intensität und erzeugt genau den vom Betrachter erwarteten Farbeindruck.

Das ging ja richtig einfach. Zu einfach!

In der analogen Realität ist die gewünschte 50%igkeit an fast jeder der einzelnen Stellen recht unwahrscheinlich und für das gesamte System grenzt es an ein halbes Wunder. Es würde eine absolute Linearität aller beteiligten Komponenten bedeuten.

Wo welcher Fehler auftritt soll uns nicht weiter interessieren und wir fassen einfach alle Einzelfehler zu einem großen Systemfehler zusammen.

In unserem Beispiel nehmen wir an, daß die wahrgenommene Farbe eher einem helleren Grau mit dem Wert 0.4 entspricht und somit das System unser 0.5er Grau um 0.1 Einheiten zu dunkel anzeigt.

Also: 0.5 (Bildpunkt) - 0.1 (Fehler) = 0.4 (Sichtbar)

Würden wir auf dem gleichen System eine hellere 0.6er Grau anzeigen würde der 0.1er Fehler es als 0.5er Grau darstellen.

Also: 0.6 (Bildpunkt) - 0.1 (Fehler) = 0.5 (Sichtbar)

Jedoch, in beiden Fällen sieht die Betrachterin nicht das was sie sehen sollte, denn die Farbe (der Grauton) ist immer zu dunkel.

Gamma Korrektur

Genau an dieser Stelle kommt die bereits angekündigte Gamma Korrektur ins Spiel. Wenn jemand ein 0.5er Grau sehen will, muß man dem Bildschirm ja einfach nur ein 0.6er Grau vorgaukeln.

Dazu setzen wir eine Gamma Korrektur von 0.1 Einheiten ein.

Also: 0.5 (Bildpunkt) + 0.1 (Gamma Kor.) -> 0.6 (korrigierter Bildpunkt) - 0.1 (Fehler) = 0.5 (Sichtbar)

Bingo.

Auch wenn in der Realität die Gamma Korrektur nie durch einen festen Wert sondern durch eine mathematische Funktion beschrieben wird (meist eine einfache Multiplikation) ändert sich nichts an der oben beschriebenen Idee. Die Addition ist jedoch in unserem Fall anschaulicher.

Nun ist zwar klar, wozu Gamma Korrektur gut ist, doch schuldet unsere Webmasterin dem lieben Joe immer noch eine zufriedenstellende Antwort auf sein Problem – dunkle Bilder im WWW.

Der einfache Weg: Korrigierter Bild-Farbraum

Wann und wo soll sie also passieren...unsere Gamma Korrektur.

Ein sehr einfacher Weg ist es die vorliegenden Bilddaten (z.B. einer TIFF oder GIF Datei) *einmal zu korrigieren* bis sie dem Farbbeurteil des Künstlers zusagen und das Bild dann *abzuspeichern*.

Wir würden damit alle unsere 0.5er Graupixel in 0.6er Pixel wandeln und die Welt für den Betrachter wäre in Ordnung.

Von vorher wissen wir aber, daß der Systemfehler von Computer zu Computer schwankt und so

mit eigentlich auch eine andere Gamma Korrektur verlangen würde. Oder?

Das ein derart korrigiertes Bild auf einem anderen Computer dennoch recht gut aussieht, liegt daran, daß die Art der Systemfehler in einem gewissen Rahmen immer identisch ist, solange „vergleichbare“ Technik Verwendung findet (also Kathodenstrahl-Bildschirme). Alle Systeme würden die Farben eher zu dunkel als zu hell anzeigen und damit gehen die Korrekturen grundsätzlich schon in die richtige Richtung. Unabhängig vom tatsächlichen Systemfehler.

Tatsächlich liegen weiterhin kleine Farbschwankungen vor, die für den normalen Betrachter jedoch kaum von Bedeutung sind. Erst im professionellen Publishing Bereich werden einem derartige Schwankungen möglicherweise belustigen.

Bei dieser Methode geht also das Betriebssystem von Bilddaten aus, die für einen Bildschirm korrigiert wurden und erspart sich damit jegliche Arbeit.

Wir werden diese im folgenden als „Microsoft-Methode“ bezeichnen, da DOS und Windows nach diesem Verfahren arbeiten (bzw. ja eigentlich nix Arbeiten, sondern nur faul weiterreichen).

Der saubere Weg: Linearer Bild-Farbraum

Ein etwas anderer Ansatz geht von Bildern mit linearem Farbraum aus. Diese Farben werden hierbei vom Betriebssystem für die Anzeige am Bildschirm automatisch korrigiert, ohne dabei die Daten im gespeicherten Bild zu ändern.

Damit dies effizient durchgeführt werden kann, benötigt man die Unterstützung der Grafikkarte,

wo eine brauchbare Gamma Korrektur von der Hardware durchgeführt werden kann ohne das System zu belasten.

Zwar ist diese Unterstützung nicht zwingend, jedoch bedeutet eine Softwarelösung entsprechende Prozessorleistung, die man anders besser „verbraten“ könnte.

Wichtige Voraussetzung ist, daß der Hersteller eines derartigen Systems die richtige Korrektur vorinstellt oder es der Künstlerin erlaubt, ihr System – wie es so schön heißt – zu kalibrieren.

Adobes Display PostScript System, wie es NeXT in OPENSTEP verwendet, geht diesen Weg. NeXT liefert für die eigenen alten NeXT Computer die entsprechenden Voreinstellungen und erlaubt PC-Besitzern die Gamma Korrektur auf ihr System abzustimmen (Interessierte sollten sich die DPS Funktionen *getframebuffertransfer*, *setframebuffertransfer* und die NetInfo Einträge in */localconfig/screens* anschauen)

Nennen wir diesen Weg die „NeXT-Methode“ da die zugrundeliegenden Framebuffer Funktionen nicht Bestandteil des Adobe PostScript Level 2 Standards sind.

Vergleichsangebot...was ist besser

Nun ist es immer verdrängend, wenn in einer NeXT-orientierten Publikation die „NeXT-Methode“ besser abschneidet als die „Microsoft-Methode“. Zwar kann hier nicht auf alle technischen Details eingegangen werden, aber dennoch sollen einige Punkte angesprochen werden.

Der einzige Vorteil der „Microsoft-Methode“ ist, daß das Betriebssystem sich jegliche Arbeit erspart und damit ganz einfach jeglicher Verantwortung entzieht.

Aus den geliebten Kompatibilitäts-Gründen ist es für Microsoft nicht möglich, Farbkorrektur systemweit durchzuführen, da dadurch bereits korrigierte Bilder wieder verflucht würden.

Bereits korrigierte Bilder sind oftmals nicht zu erkennen und gemachte Korrekturen können nur schwer automatisch rückgängig gemacht werden. Sehr wenige Bildformate erlauben die Speicherung eines Vermerks über die gemachten Korrekturen (z.B. TIFF) aber noch weniger Programme berücksichtigen diese Information dann auch noch.

Alle Vorteile der „NeXT-Methode“ kann man als Nachteile der „Microsoft-Methode“ werten. Im Groben sind das:

- sehr gute Übereinstimmung der Farben bei unterschiedlicher Wiedergabe, sofern die einzelnen Ausgabegeräte richtig kalibriert wurden (also auch auf Druckern).

- das Verlagern transparenter Farbwerte entspricht im linearen Farbraum eher den erwünschten Ergebnissen.

- ein linearer Farbraum erlaubt „besseres“ Dithering auf Systemen mit wenigen Farben (z.B. bei 8Bit Grafikmodi).

- „mehr Farbinformation“ bei gleicher Bittiefe der gespeicherten Daten (dieser Punkt ist etwas schwammig formuliert – aber der zweifelnde Leser kann ja nochmal nachbohren)

Die „Apple-Methode“ liegt etwas zwischen der „Microsoft“ und „NeXT-Methode“ wobei die „X-Window“ sich wiederum nicht festlegen will, aber eher zur „Microsoft-Methode“ neigt.

Im X-Window Bereich gibt es jedoch „Ausrutscher“ wie die „SGI-Methode“, wo automatische Gamma Korrektur auch zum gut-

en Ton gehört und somit eher der „NeXT-Methode“ entspricht.

Doch wie sag ich's meinem Kinde ?

All dies wird der Webmasterin nicht viel nützen. Was soll sie nun Joe User antworten ?

Wenn Joe ein X-Window/UNIX Anwender könnte sie sein Problem mit der für die X-Welt üblichen Empfehlung beantworten: „Kaufen sie sich doch einfach eine neue 32Bit Grafikkarte – upgrade your hardware“.

Aber kann sie diesem Joe – ach was – kann sie den zig Millionen DOS/Windows Usern einfach sagen, daß Ihr System und Ihr Web-Browser keine Ahnung von Farben haben. Egal ob 98% aller Websites „richtig“ dargestellt werden ?

Wieso sollte Joe ihr glauben, daß gerade ihre WWW Seiten „gute“ Farben enthalten und der ganze Rest „problematische“ Farben verwendet ?

Sollte sie sich wirklich die Mühe machen und versuchen es auch ihm zu erklären ?...so wie sie es bei den ganzen anderen versucht hat, die sich vor ihm schon beschwert hatten.

Sollte die diese lästigen Anfragen in Zukunft einfach besser ignorieren oder doch lieber ihre Bilder „verfälschen“, damit die Windows-Leser zufrieden sind ?

Sie entschied sich das zu tun, was die meisten Menschen tun. Also erst einmal nichts tun, dann darüber eine Nacht schlafen um sich am nächsten Tag zu wundern, daß sie Joes Mail scheinbar gestern versehentlich gelöscht hatte. Schade – Joe hätte sicherlich sein Betriebssystem dafür schuldig gesprochen. Obwohl ?

thomas engel
tsengel@cip.informatik.uni-erlangen.de