

Untersuchung der Farbabstandsformeln
des CIELAB Farbraums
auf ihre Eignung, Farbrauschen
quantitativ und physiologisch richtig
zu beschreiben

Diplomarbeit
im Fachbereich Photoingenieurwesen
an der Fachhochschule Köln

vorgelegt von

Jan Fischer
Mat.-Nr.: 11000855

Michael Bantel
Mat.-Nr.: 11005277

Referent: Prof. Dr. M. Schrey
Koreferent: Dipl. Ing. D. Wüller

Köln, im Juli 2002



Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Erläuterung zur Thematik.....	1
1.2	Lösungsansatz.....	2
1.3	Grundlagen.....	3
1.3.1	Definitionen.....	3
1.3.2	Farbräume.....	4
	Das Normvalenzsystem (Normfarbtafel).....	4
	Das Farbdreieck.....	5
	CIELAB- und CIELCH- System.....	7
	CIELAB Farbdifferenzformeln.....	10
	CIE94.....	11
	CIEDE2000.....	12
1.3.3	Der Arbeitsplatz.....	15
	Leuchtdichteverhältnisse bei Monitoren mit positiver Polarität.....	17
1.3.4	Rauschen.....	18
	Grundlagen.....	18
	Farbrauschen.....	21
1.3.5	Einführung in Colormanagement.....	22
	Farbmanagement.....	22
	Aufbau von Farbprofilen.....	25
2	Material und Methoden.....	30
2.1	Das Programm.....	30
2.1.1	Grundlagen.....	30
2.1.2	Einführung.....	30
2.1.3	Farbumrechnung.....	32
2.1.4	Das Verrauschen.....	33
2.1.5	Die Auswertung.....	36
2.2	Der Arbeitsplatz.....	37
2.2.1	Verwendete Geräte.....	37
2.2.2	Die Testfarben.....	39
2.2.3	Arbeitsumgebung.....	42
	Betrachtungsabstand.....	43
2.2.4	Probanden und Versuchsablauf.....	45
3	Ergebnisse.....	46
4	Diskussion.....	55
5	Ausblick.....	58
6	Anhang.....	59
6.1	Literaturverzeichnis.....	59
6.2	Eidesstattliche Erklärung.....	61
6.3	Sperrvermerk.....	62
6.4	Weitergabeerklärung.....	62
6.5	Abdrucke elektronischer Quellen.....	63
	6.5.1 Introduction to S-CIELAB	63
6.6	Messwerte.....	65

6.6.1 Ergebnisse der einzelnen Probanden.....	65
6.6.2 Kontrollmessungen der Testfarben.....	70
6.7 Die Menues von Shiraz.....	71
6.7.1 Ablage.....	71
Neu.....	71
Öffnen.....	71
Beenden.....	71
6.7.2 Bearbeiten.....	72
Testfeld.....	72
Voreinstellungen.....	72
6.7.3 Auswertung.....	72
Auswertung starten.....	72
Set erstellen.....	73
Set öffnen.....	73
Set bearbeiten.....	73
Ergebnisse anzeigen.....	73
Ergebnisse exportieren.....	73
6.8 Die Fenster von Shiraz.....	74
6.8.1 Farbwahl.....	74
6.8.2 Sequenz.....	75
6.8.3 Voreinstellungen.....	76
6.8.4 Set bearbeiten.....	77
6.8.5 Auswertung.....	78
6.9 Der Quellcode von SHIRAZ.....	79
6.9.1 Klasse Farbmatrik.ICC_RGBmatrik.....	79
6.9.2 Klasse Farbmatrik.LABmatrik.....	89
6.9.3 Klasse Sequenz.Rauschen.....	94

Erläuterung zur Aufgabenaufteilung

Die Programmierung des Programmes „Shiraz“ erfolgte durch Jan Fischer.

Grundlagen und Arbeitsplatz von Michael Bantel.

Der Versuchsablauf und dessen Auswertung erfolgte durch beide Diplomanden.

Eine genauere Aufteilung der Aufgabengebiete dieser Arbeit lässt sich aufgrund des Themas nicht bewerkstelligen, da die Komplexität der Einzelgebiete häufige Diskussionen und gemeinsames Aneignen des Fachwissens erforderte.

1 Einleitung

1.1 Erläuterung zur Thematik

Das Farbrauschen – wie auch das Rauschen allgemein – ist meßbar. Es sollte jedoch nicht nur statistisch sondern auch physiologisch mit einer Meßgröße quantifizierbar sein. Zum Beispiel anhand des Ergebniswertes einer Farbabstandsformel, der den Abstand zwischen ursprünglicher und verrauschter Farbe beschreibt.

Der menschliche Sehsinn reagiert jedoch auf jede Farbe unterschiedlich, was eine Anpassung der Farbabstandsformel bzw. des Farbraums an die physiologischen Bedingungen nötig macht. Dafür wurden mehrere Farbraummodelle entwickelt, die versuchen, eine Annäherung an das menschliche Sehempfinden zu erreichen.

Ein sehr gängiger, von der Industrie bevorzugter Farbraum, ist der CIELAB-Farbraum. Dessen Farbabstandsformel wurde seit seiner Entstehung 1976 ständig angepasst und erweitert.

Die Frage ist nun, ob sich eine dieser Farbabstandsformeln eignet, Farbrauschen quantitativ zu beschreiben. Dafür müsste ein Farbabstand ΔE zu errechnen sein, dessen Wert unabhängig von der Farbe und der Richtung des Rauschens ist. Die Farbabstandsformeln wurden allerdings nicht entwickelt, um Farbrauschen zu beurteilen, sondern um Farbflächen zu vergleichen.

Es soll daher im folgenden gezeigt werden, daß die bisherigen Farbabstandsformeln des CIELAB Farbraums nicht dazu geeignet sind, Farbrauschen physiologisch richtig und unabhängig von Farbe und Helligkeit zu beschreiben.

1.2 Lösungsansatz

Um die Eignung einer Farbabstandsformel und damit des Farbraums beurteilen zu können, muß geprüft werden, ob der Farbabstand ΔE für die verschiedenen Arten der Verrauschung (Für den LCH-Farbraum zum Beispiel die Sättigung und die Helligkeit sowie der Bunttonwinkel) innerhalb einer Farbe die gleichen Werte ergibt. Außerdem müssen die Farbabstände auch bei unterschiedlichen Farben als gleich groß erkannt werden.

Dazu wird durch eine selbsterstellte Software eine monochrome Farbfläche künstlich nach der Helligkeit L, der Sättigung C und dem Farbwinkel h in mehreren Abstufungen mit einer Gaussverteilung verrauscht, deren verschiedene Rauschwerte der Stärke des Rauschens nach geordnet, einem Probanden nacheinander gezeigt werden. Die Stufe, die eine Versuchsperson als verrauscht erkennt, wird festgehalten.

Dies geschieht jeweils für die drei Komponenten L, C und h, die in die Farbabstandsformel mit einfließen, bei unterschiedlichen Farben. Unter anderem sind die additiven und subtraktiven Grundfarben Rot, Grün und Blau, Cyan, Magenta und Gelb sowie zwei Hauttöne vorgesehen.

Das ΔL , ΔC und Δh , mit dem die Stufe beschrieben wird, entspricht hierbei der Varianz σ^2 des Rauschens.

Die Ergebnisse mehrerer Testpersonen werden statistisch gemittelt, dann wird für die Komponenten der jeweiligen Farbe der Farbabstand ermittelt.

Da außer der ursprünglichen CIELAB-Farbabstandsformel von 1976 noch Erweiterungen entwickelt wurden, um die Ergebnisse physiometrisch besser dem Sehempfinden anzupassen, werden auch deren Ergebnisse ermittelt und auf Tauglichkeit überprüft.

1.3 Grundlagen

1.3.1 Definitionen

Normalerweise und umgangssprachlich wird in der Farbmeterik im Zusammenhang mit dem CIELAB- Farbraum von Helligkeit, Buntheit und von der Sättigung als x-Vektor gesprochen. Dies ist so eigentlich nicht richtig, da die Sättigung zwar für das CIELUV-System, nicht aber für den CIELAB-Farbraum definiert ist. Richtigerweise muß hier immer von Buntheit die Rede sein.

Die Sättigung ist die Kennzeichnung für den Grad der Farbigeit unabhängig von der Helligkeit. Die helligkeitsabhängige Kennzeichnung für den Grad der Farbigeit heißt Buntheit. Bei gleicher Buntheit nimmt die Sättigung mit abnehmender Helligkeit also zu [3, S.173].

Da im CIELAB- Farbraum immer – wie auch im Versuch dieser Arbeit – helligkeitsbezogen gearbeitet wird, muß hier von Buntheit gesprochen werden. Aus Gewohnheitsgründen und da diese Definition häufig nicht bekannt ist, wird jedoch im folgenden dem Begriff Sättigung weiterhin den Vorzug gegeben. Es ist aber die Buntheit gemeint!

Zur Verdeutlichung:

Sowohl Vektor A wie B entsprechen in einem imaginären kegelförmigen Farbraum einer Sättigung von 100%.

Die Buntheit (Die Länge der Vektoren) ist jedoch auf die Helligkeit bezogen unterschiedlich.

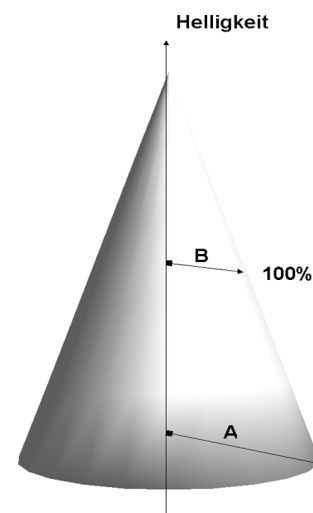


Abbildung 1 - Stilisierter Farbraum

1.3.2 Farbräume

Das Normvalenzsystem (Normfarbtafel)

Um eine präzise und international einheitliche Beschreibung von Farbe zu ermöglichen, ist ein System von anerkannten Bezugsfarben nützlich, die auch als Primärvalenzen bezeichnet werden.

Die Wahl von Primärvalenzen und ihrer zugehörigen Spektralwerte ist jedoch vollkommen willkürlich. In Abhängigkeit der empirisch ermittelten Eigenschaften des Normalbeobachters hat die CIE im Jahr 1931 die Spektralwertfunktionen der Primärvalenzen R, G und B festgelegt. Jeder Wellenlänge des sichtbaren Spektrums wird in dieser Systematik eine Kombination der Primärvalenzen zugeordnet.

Um einen Standard zu schaffen und zur besseren Handhabung dieser Werte, wurde von der CIE ein Normvalenzsystem empfohlen, dessen virtuelle Primärvalenzen XYZ als Normvalenzen bezeichnet werden und bei dem die Spektralwerte ausschließlich nicht negative Werte annehmen. Es handelt sich dabei um ein System mit sogenannten virtuellen Primärvalenzen, bei dem diesen keine realen Farbreize zugeordnet werden können. Dabei entspricht X einer imaginären (nicht real existierenden) roten, Y einer entsprechenden grünen und Z einer entsprechenden blauen Farbvalenz. Die auf diese Normvalenzen bezogenen Spektralwerte nennt man Normspektralwerte und die daraus errechneten Farbwerte Normfarbwerte.

Mittels linearer Transformationen können die Koordinaten des Normvalenzsystems beliebig in andere Primärvalenzsysteme, zum Beispiel in ein RGB-System umgerechnet werden.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Abhängig vom RGB-System müssen diese RGB-Werte jedoch noch mit Hilfe einer Gammakorrektur angepasst werden.

$$R_{\text{lin}} = R^{1/\gamma_{\text{rot}}}$$

$$G_{\text{lin}} = G^{1/\gamma_{\text{grün}}}$$

$$B_{\text{lin}} = B^{1/\gamma_{\text{blau}}}$$

Dabei hängen die Werte der Transformationsmatrix x_i , y_i und z_i sowie die Gamma-werte γ_{rot} , $\gamma_{\text{grün}}$ und γ_{blau} vom jeweiligen RGB-System ab.

Das Farbdreieck

Aus der Festlegung der Normspektralwerte durch die CIE folgen einige besondere Eigenschaften. So hat zum Beispiel ein ideales Weiß (bei energiegleichem Spektrum) die Normspektralwerte $X=Y=Z=100$ und die Helligkeit ist direkt durch den imaginären Normspektralwert Y gekennzeichnet.

Das CIEXYZ-Farbsystem stellt in der modernen Reproduktionstechnik einen wichtigen Referenzfarbenraum dar. Sowohl die Festlegungen der ICC als auch die Farbdefinitionen der Seitenbeschreibungssprache PostScript sehen CIEXYZ unter der Normlichtart D50 und einem Betrachtungswinkel von 2-Grad als Bezugsfarbenraum vor [10, S.21].

Stellt man das Verhältnis der Normfarbwerte zueinander dar, dann spricht man von den Normfarbwertanteilen x , y und z , die in der Summe stets 1 ergeben. Dementsprechend ist es auch nicht notwendig, alle drei Werte aufzuführen, da die

Angabe eines beliebigen Wertepaares in Differenz zu 1 als eindeutige Wertebeschreibung genügt. Durch diese Systematik ergibt sich nun eine neue Darstellungsvariante für die eindeutige Zuordnung von Farben auf der Basis der CIE-Primärvalenzen. Es werden nur die Normfarbwertanteile x und y angegeben, die zusammen ein Maß für Sättigung und Farbton bilden. Markiert man in diesem Visualisierungsschema die realen Farbvalenzen der Spektralfarben, so ergibt sich eine schuhsohlenförmige Darstellung, die auch als Spektralfarbenzug bezeichnet wird [10, S. 21].

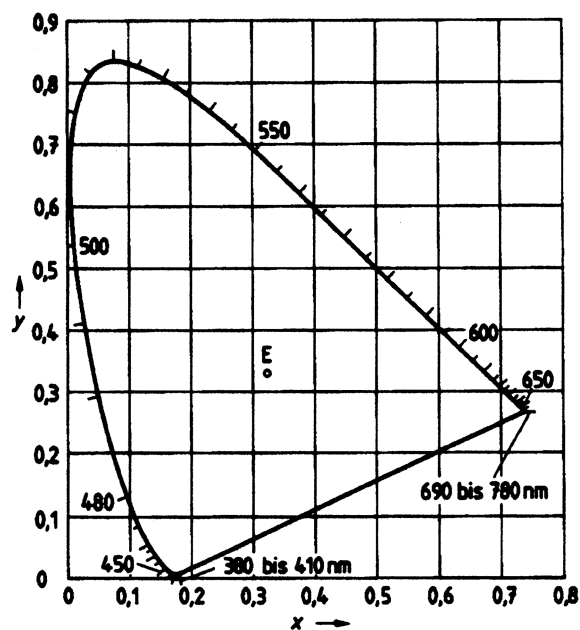


Abbildung 2 - Die CIE Normfarbtabelle [17]

Die sogenannte Mittelpunktsvalenz mit den Normwertsanteilen $x=y= 0,333$ entspricht im CIE Farbdreieck dem Unbuntpunkt. Bei Körperfarben gilt die Farbvalenz der jeweils verwendeten Lichtquelle als Mittelpunktsvalenz. Bei der Normlichtart D65 betragen die Normfarbwertsanteile beispielsweise $x= 0,3127$ und $y= 0,3290$; bei D50 gilt für $x= 0,3457$, für $y=0,3585$ [1].

Als Farbtemperatur ausgedrückt entsprechen diese Werte 6800K bzw. 5003K.

Der visuelle Unterschied zwischen zwei Farben läßt sich im CIE-Farbdreieck jedoch nicht ablesen. Auch wenn die Helligkeit in einer weiteren Darstellungsachse eingeführt wird, d.h. das CIE-Farbdreieck zum Farbkörper komplettiert wird, ist dies nicht möglich. Dieser beschreibt lediglich alle vom Auge des Normalbeobachters erfassbaren Farbvalenzen.

Trägt man die maximal erzielbare Helligkeit über die Sättigung und den Farbton auf, so ist festzustellen, daß sich im Bereich der gelben und grünen Farben bei hoher Sättigung eine deutlich größere Helligkeit erzielen läßt, als im blauen und roten Bereich. Der Farbkörper ist also deutlich asymmetrisch [10, S. 22].

CIELAB- und CIELCH- System

Das Normvalenzsystem ist empfindungsgemäß nicht gleichabständig. Somit können mit diesem Farbraum keine Farbunterschiede physiologisch richtig dargestellt und gemessen werden. Hierzu müssen die Farbwerte in einen visuell gleichabständigen Farbraum transformiert werden. 1976 wurden von der CIE aus diesem Gedanken heraus der LAB- und der LCH- Farbraum vorgeschlagen. Diese sind zumindest visuell annähernd gleichabständig und besitzen die zugehörigen Farbabstandsformeln, die allerdings immer wieder physiologisch besser angepaßt wurden. Eine ideale Anpassung wurde jedoch immer noch nicht erreicht.

Der CIELAB-Farbraum stellt – bedingt durch eine gewisse De-facto-Standardisierung – unter anderem durch die Festlegung der Commission Internationale de l'Eclairage (CIE) im Jahr 1976 das derzeit wohl wichtigste farbmtrische Ordnungssystem dar. Es wurde unabhängig von der Zuordnungssystematik des CIE-Farbdreiecks als Idee zur Schaffung gleichabständiger Farbordnungssysteme und Abstandsformeln durch die sogenannte Gegenfarbentheorie von Hering entwickelt [10, S. 25].

Er besitzt die senkrecht zueinander stehenden a^* und b^* Achsen, die sich im Unbuntpunkt (je nach Helligkeit von Schwarz bis Weiß) kreuzen. Als dritte Achse und

als Maß für die Helligkeit steht L^* im Unbuntpunkt senkrecht auf der durch a^* und b^* gebildeten Ebene. Farben gleichen Farbtons liegen auf Geraden vom Unbuntpunkt nach außen, wobei der Abstand zur Unbuntachse das Maß der Sättigung beschreibt.

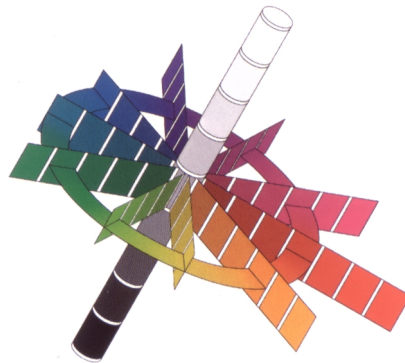


Abbildung 3 - CIELAB Farbraum
(schematisch) [10, S. 24]

Die Umrechnung von XYZ in CIELAB erfolgt durch Transformationsgleichungen. Für den CIELAB oder L^* , a^* , b^* Farbraum gilt nach DIN 6174 folgende Definition:

$$L^* = 116 Y^* - 16$$

$$a^* = 500 (X^* - Y^*)$$

$$b^* = 200 (Y^* - Z^*)$$

mit:

$$X^* = \begin{cases} \sqrt[3]{X/X_n} & \text{für } X/X_n > 0,008856 \\ 7,787 (X/X_n) & \text{für } X/X_n \leq 0,008856 \end{cases}$$

$$Y^* = \begin{cases} \sqrt[3]{Y/Y_n} & \text{für } Y/Y_n > 0,008856 \\ 7,787 (Y/Y_n) & \text{für } Y/Y_n \leq 0,008856 \end{cases}$$

$$Z^* = \begin{cases} \sqrt[3]{Z/Z_n} & \text{für } Z/Z_n > 0,008856 \\ 7,787 (Z/Z_n) & \text{für } Z/Z_n \leq 0,008856 \end{cases}$$

" X_n , Y_n , Z_n sind die Normfarbwerte der vollkommen mattweißen Fläche für die

Lichtart und den Normalbeobachter, auf die sich die Normfarbwerte X, Y, Z beziehen" [16].

Bei der Umwandlung in den LCH Farbraum wird der L*-Wert unverändert gelassen. Die anderen Werte werden folgendermaßen gebildet:

$$C_{ab}^* = \sqrt{a^{*2} + b^{*2}}$$

$$h_{ab}^* = \arctan\left(\frac{b^*}{a^*}\right)$$

"Der Bunttonwinkel h_{ab}^* nimmt von der positiven a^* -Achse ausgehend im entgegengesetzten Uhrzeigersinn zu" [16].

Der CIELAB-Farbraum hat ein a^*b^* -Diagramm, in dem gerade Linien aus der Normfarbtafel zu gekrümmten Linien werden, d.h. die additiven Mischfarben haben ihre Farbörter auf den gekrümmten Verbindungslinien der Farbörter der Komponenten.

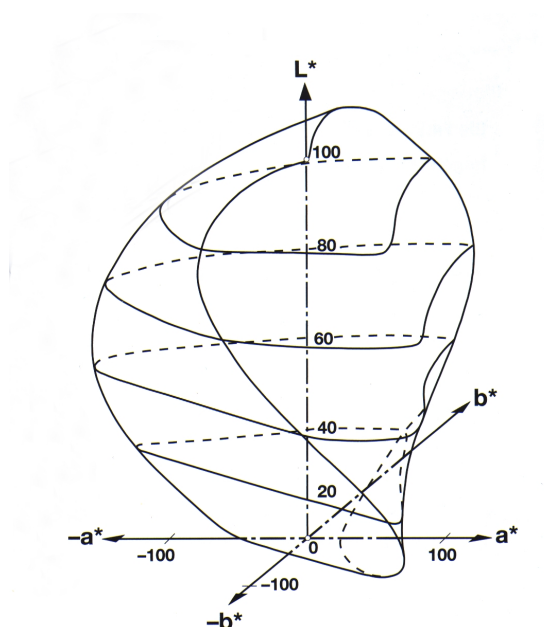


Abbildung 4 - CIELAB Farbraum [9, S. 79]

Der LCH Farbraum ist lediglich eine Variante des CIELAB-Farbraums. Alle Farben haben hier die gleiche Position wie im LAB-Modell. Anstatt die Farbe über ein rechtwinkliges Koordinatensystem zu definieren, wird hier die Farbart über einen Winkel und die Sättigung über den Abstand von der Grauachse beschrieben. Mit ihm lassen sich Farbnuancen sehr gut vergleichen [8, S. 44].

Mit dem LAB-System lassen sich Meßwerte jedoch einfacher notieren. Dies machte ihn unter anderem zur Verwendung in der Farbmeßtechnik gebräuchlicher. Es hat sich sogar als geräteunabhängiger Farbraum durchgesetzt. So können in ihm von jedem beliebigen digitalen Bilderfassungssystem gelieferte RGB Werte umgerechnet werden um sie dann an andere Geräte weitergeben zu können. Voraussetzung hierfür ist allerdings, daß der RGB-Farbraum, in dem die ursprünglichen RGB Farben entstanden sind, bekannt ist, oder durch entsprechende Geräteprofile Umrechnungstabellen erstellt werden.

Vor allem die Einbeziehung von Wahrnehmungseffekten (z.B. Simultankontrasten) in der systematischen Beschreibung von Farben stellt einen aktuellen farbwissenschaftlichen Forschungsstand dar. In sogenannten Color Appearance Modellen wird versucht, einen systematischen Zusammenhang zwischen Farbvalenz und Farbempfindung herzustellen [10, S. 25].

CIELAB Farbdifferenzformeln

Für die Bestimmung von Farbabständen im CIELAB Farbraum gibt es mehrere Formelsätze.

Nach der ursprünglichen Definition von 1976 wird der Abstand zweier Farben im CIELAB Farbraum nach folgender Methode berechnet [16]:

$$\Delta E_{ab}^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$$

Dabei gilt mit den Indices _P (Probe) und _B (Bezug):

$$\Delta L^* = L_P^* - L_B^*$$

$$\Delta a^* = a_P^* - a_B^*$$

$$\Delta b^* = b_P^* - b_B^*$$

Zuweilen kann es nützlich sein, die Farbe in den LCH Farbraum zu transformieren. Dann berechnet sich der Farbabstand wie folgt:

$$\Delta E_{ab}^* = \sqrt{(\Delta L^*)^2 + (\Delta C_{ab}^*)^2 + (\Delta H_{ab}^*)^2}$$

Mit:

$$\Delta L^* = L_P^* - L_B^*$$

$$\Delta C_{ab}^* = C_P^* - C_B^* = \sqrt{(a_P^*)^2 + (b_P^*)^2} - \sqrt{(a_B^*)^2 + (b_B^*)^2}$$

$$\Delta H_{ab}^* = \sqrt{(\Delta E_{ab}^*)^2 - (\Delta L^*)^2 - (\Delta C_{ab}^*)^2} = \sqrt{(\Delta a^*)^2 + (\Delta b^*)^2 - (\Delta C_{ab}^*)^2}$$

Der Buntton-Beitrag ΔH_{ab}^* darf nicht mit der Differenz zweier Bunttonwinkel Δh_{ab} verwechselt werden. Aus der Formel erhält man nur den absoluten Betrag für ΔH_{ab} , nicht jedoch sein Vorzeichen. Dafür gilt:

$$\Delta H_{ab}^* = \begin{cases} \text{positiv, wenn } a_P^* b_B^* - a_B^* b_P^* \leq 0 \\ \text{negativ, wenn } a_P^* b_B^* - a_B^* b_P^* > 0 \end{cases}$$

CIE94

Im Jahr 1995 verabschiedete die CIE eine Erweiterung der ursprünglichen Farbabstandsformel. Dieses System wird als CIE 1994 (ΔL^* ΔC_{ab}^* ΔH_{ab}^*) Farbdifferenz Modell bezeichnet, bzw. mit CIE94 abgekürzt.

Hier wurden zum einen die parametrische Faktoren k eingeführt, die das Anpas-

sen der Formel an bestimmte Betrachtungsbedingungen, wie z.B. die Oberflächenbeschaffenheit der Proben, ermöglichen. Zum anderen wurden die Skalierungsfaktoren S eingeführt, die die gesamte Performance verbessern sollen. Die CIE94 Formel ist eng an die CMC(l:c) Formel angelehnt, die ebenfalls mit dem CIELAB Farbraum arbeitet [19].

Die Farbabstandsformel lautet wie folgt:

$$\Delta E^*_{94} = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*_{ab}}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*_{ab}}{k_H S_H}\right)^2}$$

$$S_L = 1$$

$$S_C = 1 + 0,045 \overline{C^*_{ab}}$$

$$S_H = 1 + 0,015 \overline{C^*_{ab}}$$

Wobei $\overline{C^*_{ab}}$ das geometrische Mittel der Sättigungen von Probe und Bezug ist [19].

Die Parameter k_L , k_C und k_H dienen zur Anpassung der relativen Gewichtung des Helligkeits-, Buntheits- und Buntton-Beitrags, um die Formel an unterschiedliche Betrachtungsbedingungen und Anwendungen anzupassen, die von den Referenzbedingungen abweichen[12]. Diese können z.B. in [1, S. 94] nachgelesen werden. Für die Referenzbedingungen werden sie auf den Wert 1 gesetzt.

CIEDE2000

Eine weitere Erweiterung wird als CIEDE2000 bezeichnet. In dieser Farbdifferenzformel wurde zusätzlich zu den Gewichtungsfunktionen ein interaktiver Term für Sättigungs- und Farbwinkelunterschiede eingefügt, der vor allem im blauen Bereich bessere Ergebnisse bringt. Ein Skalierungsfaktor für die a^* -Achse unterstützt eine verbesserte Darstellung der Farbabstände bei Neutralfarben [13].

$$\Delta E^*_{00} = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'_{ab}}{k_C S_C}\right)^2 + \left(\frac{\Delta H'_{ab}}{k_H S_H}\right)^2 + R_T \left(\frac{\Delta C'_{ab}}{k_C S_C}\right) \left(\frac{\Delta H'_{ab}}{k_H S_H}\right)}$$

Die Berechnung der einzelnen Komponenten:

Schritt1: Berechnung der L^* , a^* , b^* und C^* Werte (siehe Kapitel CIELAB Farbdifferenzformeln).

Schritt2: Berechnung von a' , C' und h' :

$$L' = L^*$$

$$a' = (1 + G) a^*$$

$$b' = b^*$$

$$C'_{ab} = \sqrt{a'^2 + b'^2}$$

$$h'_{ab} = \arctan\left(\frac{b'}{a'}\right)$$

Mit:

$$G = \frac{1}{2} \left(1 - \sqrt{\frac{\overline{C^*_{ab}}^7}{\overline{C^*_{ab}}^7 + 25^7}} \right)$$

Wobei $\overline{C^*_{ab}}$ das arithmetische Mittel der Sättigungen von Probe und Bezug ist.

Schritt 3: Berechnung von $\Delta L'$, $\Delta C'$ und $\Delta H'$:

$$\Delta L' = L'_P - L'_B$$

$$\Delta C' = C'_P - C'_B$$

Die Indices $_P$ und $_B$ beziehen sich auf die Proben- und Bezugsfarbe.

$$\Delta H' = 2 \sqrt{C'_P C'_B} \sin\left(\frac{\Delta h'}{2}\right)$$

Mit

$$\Delta h' = h'_P - h'_B$$

Schritt 4: CIEDE2000 ΔE_{00} berechnen:

$$\Delta E^*_{00} = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'_{ab}}{k_C S_C}\right)^2 + \left(\frac{\Delta H'_{ab}}{k_H S_H}\right)^2 + R_T \left(\frac{\Delta C'_{ab}}{k_C S_C}\right) \left(\frac{\Delta H'_{ab}}{k_H S_H}\right)}$$

Mit:

$$S_L = 1 + \frac{0,015 (\bar{L}' - 50)^2}{\sqrt{20 + (\bar{L}' - 50)^2}}$$

$$S_C = 1 + 0,045 \bar{C}'$$

$$S_L = 1 + 0,015 \bar{C}' T$$

wobei:

$$T = 1 - 0,17 \cos(\bar{h}' - 30^\circ) + 0,24 \cos(2\bar{h}') + 0,32 \cos(3\bar{h}' + 6^\circ) - 0,20 \cos(4\bar{h}' - 63^\circ)$$

$$R_T = -\sin(2\Delta\theta) R_C$$

$$\Delta\theta = 30 \exp\left\{-\left[\frac{(\bar{h}' - 275^\circ)}{25}\right]^2\right\}$$

und

$$R_C = 2 \sqrt{\frac{\bar{C}'^7}{\bar{C}'^7 + 25^7}}$$

Dabei sind \bar{L}' , \bar{C}' und \bar{h}' die arithmetischen Mittel der L' , C' und h' Werte von Probe

und Bezugswert. Bei der Ermittlung von \bar{h}' muss bei Farben aufgepasst werden, die in unterschiedlichen Quadranten liegen. Wenn die Differenz zweier Farbwinkel kleiner als 180° ist, kann das arithmetische Mittel verwendet werden. Andernfalls muss man 360° vom größeren Winkel abziehen und dann erst das arithmetische Mittel berechnen [13].

Die Parameter k_L , k_C und k_H dienen auch hier zur Anpassung der relativen Gewichtung des Helligkeits-, Buntheits- und Buntton-Beitrags, um die Formel an unterschiedliche Betrachtungsbedingungen und Anwendungen anzupassen, die von den Referenzbedingungen abweichen. Für die Referenzbedingungen werden sie auf den Wert 1 gesetzt.

1.3.3 Der Arbeitsplatz

Um die Nachvollziehbarkeit und jederzeitige Wiederholbarkeit einer Messung zu garantieren, sollten die Bedingungen eines Versuchs festgelegt werden. Hierzu veröffentlichte 1978 die CIE Anhaltswerte, die sich speziell auf die Untersuchung und Forschung von Farbabständen bezog [6, S. 113 und 115].

Beleuchtung	D65
Beleuchtungsstärke	1000lx
Umgebung	gleichmäßiges Neutralgrau mit L= 50
Betrachtungsart	Objekt (kein Selbstleuchter)
Probengröße	größer 4°
Probenabstand	minimal bzw. in direktem Kontakt
Farbabstand d. Proben	0 – 5 Delta E
Oberflächenstruktur	homogen, ohne Ungleichmäßigkeiten

Diese Messbedingungen sind aber für eine Monitormessung nicht zu nutzen (ein Monitor ist zum Beispiel ein Selbstleuchter). Außerdem wird in graphischen Berei-

chen mehrheitlich die Normlichtart D50 genutzt und aus Adaptationsgründen auch der Bildschirm auf diese Farbtemperatur geeicht.

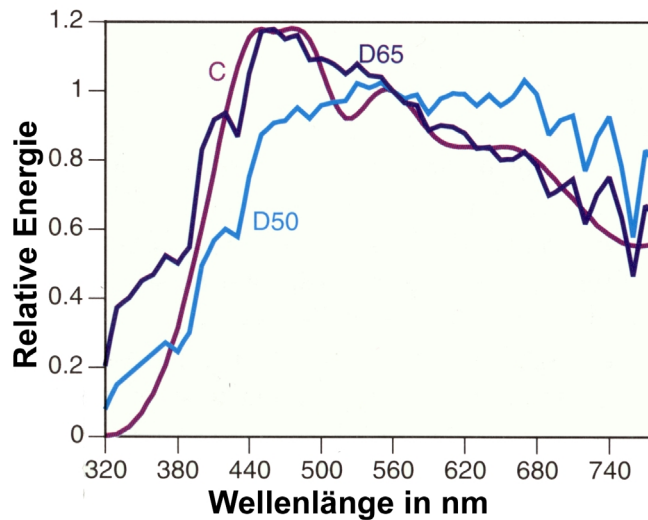


Abbildung 5 - Spektralverteilung der wichtigsten Normlichtquellen [6, S. 6]

Sinnvoll erscheint es jedoch, sich bei der Arbeit mit Testpersonen auf die von der DIN 5035-7 genormten Arbeitsplatzbedingungen für einen Bildschirmarbeitsplatz zu beziehen. So sind hier auch die Bildschirmumgebung sowie auch die Raumbeleuchtung festgelegt, um eine ausgewogene Leuchtdichteverteilung im Bereich des Gesichtsfeldes zu erreichen und somit eine visuelle Belastung zu begrenzen.

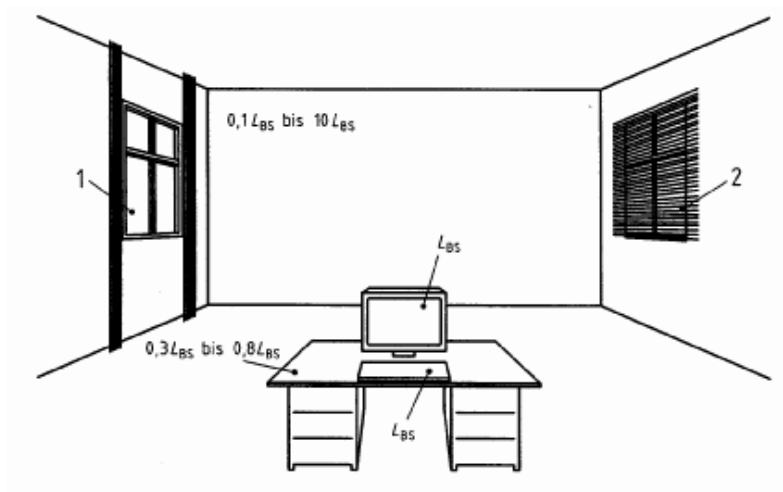


Abbildung 6 - Arbeitsplatzskizze aus DIN 5035-7 [15]

Leuchtdichteverhältnisse bei Monitoren mit positiver Polarität

(Dunkle Schriftzeichen auf hellem Hintergrund)

Die Erkennbarkeit der Bildschirminformation ist im hohen Maße abhängig von der Umgebung des Arbeitsplatzes: Helle Flächen, die sich aus der Sicht des Nutzers auf der Bildschirmoberfläche spiegeln, können zu einer Beeinträchtigung der Informationsaufnahme führen; helle Flächen, die um den Bildschirm angeordnet sind, können zu Adaptationsstörungen führen [15].

So wurden im Versuchsraum alle im Gesichtsfeld befindlichen Flächen in einem mittleren Grau gehalten. Arbeitsgegenstände, Papier und Ähnliches wurden vom Tisch entfernt.

Maßgebend bei der Bestimmung der Leuchtdichteverhältnisse im Gesichtsfeld sind die mittleren Leuchtdichten der Flächen. Diese sind von der Polarität des Bildschirmhintergrundes abhängig.

Ein heute typischer Wert der Leuchtdichte des Bildschirmhintergrundes von Bildschirmen mit positiver Polarität ist etwa 100 cd/m^2 . Für Bildschirme mit Leuchtdichten in dieser Größenordnung werden folgende Leuchtdichteverhältnisse empfohlen:

Schreibtisch- bzw. Tischoberfläche	$0,3 L_{BS} \leq L_T \leq 0,8 L_{BS}$
Größere Flächen im Raum	$0,1 L_{BS} \leq L_U \leq 10 L_{BS}$

Dabei ist L_{BS} die Leuchtdichte des Bildschirmhintergrundes, L_T die Leuchtdichte der Arbeitstischoberfläche und L_U die Leuchtdichte größerer Flächen in der Umgebung.

1.3.4 Rauschen

Grundlagen

Als Rauschen bezeichnet man ein Zufallssignal, das sich dem eigentlichen Bildsignal überlagert. Das Rauschen in Bildern stammt zum größten Teil von der bildgebenden Elektronik, insbesondere vom Bildsensor (z.B. CCD oder CMOS). Dabei gibt es die unterschiedlichsten Quellen für dieses Rauschen.

Grundsätzlich unterscheidet man zwischen inneren Rauschquellen, die ihre Ursache im Aufbau und Funktionsprinzip des Bildsensors haben, und äußeren Rauschquellen, die in Folge der Weiterverarbeitung des Signals entstehen.

Stellvertretend für die unterschiedlichen Bildsensoren soll hier das Entstehen von Rauschen anhand eines CCD-Bildsensors erläutert werden. Diese Art von Bildsensor ist sowohl in digitalen Kameras als auch in Scannern häufig vertreten. Auf seine Funktionsweise soll hier jedoch nicht eingegangen werden.

Zu den inneren Rauschquellen zählen z.B. das Dunkelstromrauschen (auch als thermisches Rauschen bezeichnet) und das Übertragungsrauschen.

Normalerweise entstehen freie Ladungsträger in einem CCD durch das Auftreffen von Licht auf die lichtempfindlichen Sensoren. Allerdings können auch durch thermische Generation Ladungsträger freigesetzt werden, die sich zum Nutzsignal hinzu addieren. Dies kann sowohl im lichtempfindlichen Bereich des CCDs geschehen, als auch im Verschieberegister. Dieses Rauschen wird als Dunkelstrom- oder thermisches Rauschen bezeichnet. Es ist vor allem von der Temperatur, aber auch der Größe eines CCD-Elements, der Anzahl der CCD-Zellen und der Taktfrequenz abhängig.

Beim Transport der Signalladungen durch das Verschiebe- bzw. Ausleseregister sind die Ladungen den unterschiedlichsten Einflüssen ausgesetzt, die das Nutzsi-

gnal verändern können. Die Einflüsse unterscheiden sich je nach verwendeter Technik der CCD. Man spricht dabei von Übertragungsrauschen.

Zu den äußeren Rauschquellen gehören u.a. auf der Eingangsseite, das elektrische Eingangsrauschen und das Photonenrauschen und auf der Ausgangsseite das Reset- oder Vorverstärkerrauschen und das Quantisierungsrauschen.

Das Photonenrauschen entsteht durch den nicht konstanten Photonenstrom. Dieser Effekt ist vor allem bei niedrigen Beleuchtungsstärken zu berücksichtigen. Bei normaler Beleuchtung kann er jedoch vernachlässigt werden.

Das elektrische Eingangsrauschen entsteht durch zufällige Schwankungen im Spannungspegel und der Impulse bei der Ansteuerung des CCD. Es ist sowohl von der Temperatur, als auch von der Eingangskapazität abhängig.

Auch beim Auslesen der Ladungen aus dem CCD kann Rauschen entstehen. Je nach verwendeter Variante spricht man hier z.B. von Resetrauschen und/oder Vorverstärkerrauschen. Diese Rauscharten haben die unterschiedlichsten Entstehungsparameter wie z.B. Temperatur, Größe der ausgelesenen Ladung, Taktfrequenz usw. [14].

Das Quantisierungsrauschen schließlich entsteht, wenn das analoge Signal des Bildgebers in ein digitales Signal gewandelt wird [11, S.161]. Liegen z.B. die analogen Signale zweier benachbarter Werte sehr dicht beieinander, aber auf der jeweils anderen Seite einer Quantisierungsschwelle, so kann dadurch ihr Abstand vergrößert werden. Das so entstandene Rauschen bezeichnet man als Quantisierungsrauschen.

All diese Rauschquellen haben unterschiedliche statistische Verteilungen und Erwartungswerte. Nach dem zentralen Grenzwertsatz kann man jedoch davon ausgehen, daß sie in ihrer Summe annähernd Normalverteilt $N(\mu; \sigma^2)$ sind mit dem Erwartungswert μ und der Varianz σ^2 [2, S. 62].

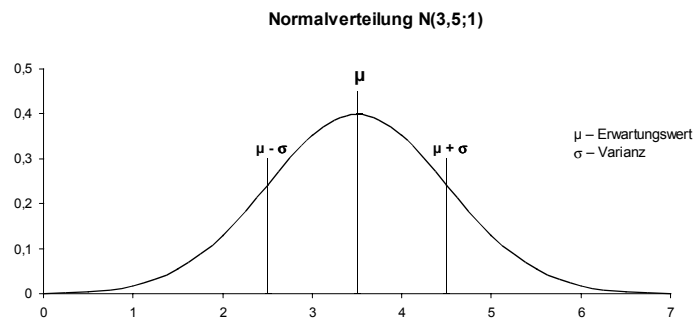


Abbildung 7 - Normalverteilung mit Erwartungswert $\mu = 3,5$ und Standardabweichung $\sigma = 1$

Des weiteren wird bei dieser Untersuchung davon ausgegangen, daß ihre statistische Verteilung über das gesamte Bildfeld konstant ist. Der Erwartungswert μ muß also an jeder Stelle im Bildfeld zum Nutzsignal addiert werden. Außerdem addiert sich die lokale Abweichung vom Mittelwert des Rauschens dazu. Letztere kann sowohl negativ, als auch positiv sein.

Damit hat das Bildsignal also den Wert:

Bildsignal = Nutzsignal + Mittelwert Rauschen + lokale Abweichung vom Mittelwert

Hier zeigt sich, daß das Rauschen vor allem bei niedrigem Nutzsignal ins Gewicht fällt. Der Rauschmittelwert μ erzeugt eine Änderung des Bildes, die jedoch gleichmäßig über das gesamte Bild verteilt ist. Dazu kommt dann noch die Änderung durch die lokale Abweichung vom Mittelwert. Eine Verbesserung des Bildes läßt sich also erreichen, wenn der Rauschmittelwert vom Bild subtrahiert wird. Damit ergibt sich für das Bildsignal dann:

Verbessertes Bildsignal = Nutzsignal + Abweichung vom Mittelwert

Die Abweichung vom Mittelwert des Rauschens ist eine $N(0;\sigma^2)$ -verteilte Zufalls-

funktion. Dabei kann die Varianz σ^2 als eine Maßzahl für die Stärke des Verrauschens betrachtet werden.

Daher wurde in dieser Versuchsreihe als Maß für die Verrauschung die Varianz der Farbabweichung ΔE im CIELAB-Farbraum gewählt.

Farbrauschen

Bei Graustufenbildern tritt Rauschen nur in einem Kanal auf, da die Bilder auch nur aus einem Kanal bestehen. Es äußert sich durch eine Änderung des Grauwertes und damit der Helligkeit.

Farbbilder bestehen jedoch aus mehreren Kanälen. Jeder dieser Kanäle kann mit Rauschen behaftet sein, wobei die Stärke von Kanal zu Kanal z.T. erheblich variieren kann. Je nach Zusammenspiel der Kanäle kann sich daher eine Farbe in Helligkeit, Sättigung und Farbton verändern. In der Praxis wird sich dabei eine Änderung nicht auf eine der drei Richtungen beschränken.

Die Art des Farbrauschens ist dabei auch stark abhängig von der Art der Farbkanäle. Bei den meist verwendeten RGB Sensoren wirkt sich ein Rauschen in einem der drei Kanäle auf Helligkeit, Farbsättigung und Farbton gleichzeitig aus.

Würde ein Sensor direkt mit einem LCh Sensor arbeiten, dessen drei Kanäle Helligkeit, Sättigung und Farbwinkel sind, so würde sich Rauschen in nur einem der drei Kanäle völlig anders auswirken. Es wäre ausschließlich die Helligkeit oder die Sättigung oder die Farbart betroffen.

Da der betrachtete CIELAB Farbraum in seinen Farbabstandsformeln mit den drei Richtungen Helligkeit, Farbton und Sättigung arbeitet, wurden die Testfarben, um eine Differenzierung der einzelnen Vektoren zu erreichen, nacheinander in diese Richtungen verrauscht. So läßt sich untersuchen, ob die unterschiedlichen Auswirkungen des Rauschens in den Kanälen L, C und h sich mit diesen Farbab-

standsformeln genügend genau beschreiben lassen.

In die Farbabstandformeln des CIELAB Farbraums gehen die Werte für ΔL und ΔC direkt ein, die Farbwinkeldifferenz Δh jedoch nicht. Diese wird durch den Bunttonbeitrag ΔH ersetzt, der nicht ausschließlich von Δh abhängt, sondern auch noch die Änderung der Sättigung ΔC berücksichtigt (siehe Kapitel 1.3.2). Der Bunttonbeitrag ΔH errechnet sich je nach verwendeter ΔE Formel außerdem unterschiedlich.

Die verwendeten Testbilder wurden daher nicht nach ΔH sondern nach Δh verauscht. Dadurch ergibt sich je nach Farbe ein unterschiedlich großes ΔH_{\max} bzw. ΔE_{\max} für diese Sequenzen.

1.3.5 Einführung in Colormanagement

Farbmanagement

Ein großes Problem bei der digitalen Bildverarbeitung stellt die fehlende Normung der verwendeten Primärfarben dar. Weder die Farborte der Monitorphosphore, noch Druckfarben oder spektrale Empfindlichkeiten von z.B. Scanner sind normt.

Damit sind die RGB Werte eines Bildes auf zwei unterschiedlichen Systemen nicht zwingend auch absolut z.B. im CIEXYZ Farbraum die selben Farben. Außerdem haben unterschiedliche Medien auch unterschiedlich große Farbräume, die sie darstellen können.

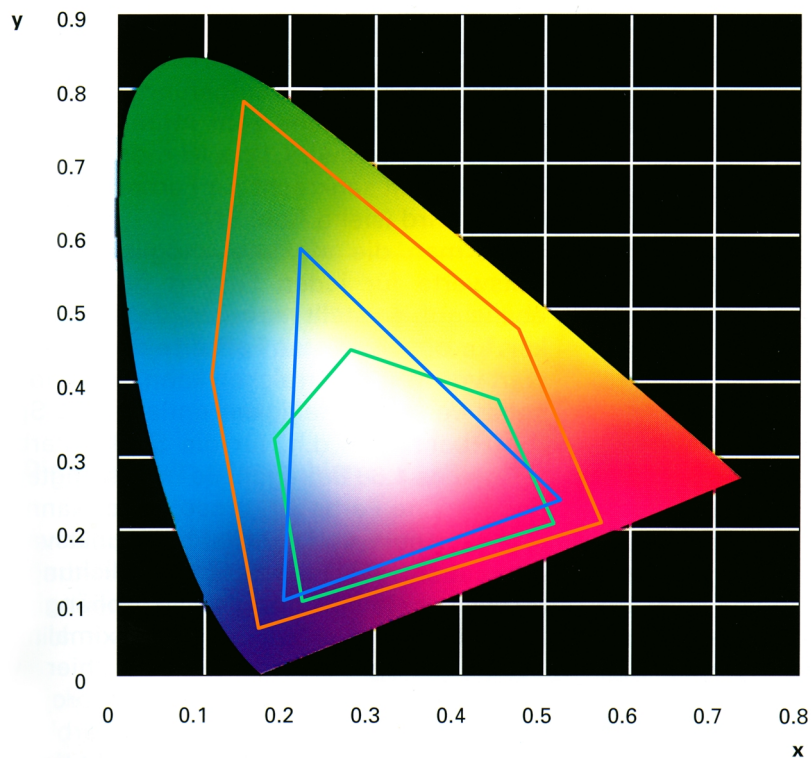


Abbildung 8 – Farbumfang eines Dias, Monitors und Druckers [5]

In Abbildung 8 ist der darstellbare Farbraum eines Dias (rot), eines Monitors (blau) und eines Druckers (grün) im CIExy-Diagramm abgebildet. Deutlich wird der unterschiedliche Farbumfang aufgezeigt.

Daher mußte ein System geschaffen werden, um die unterschiedlichen Farbräume zusammen zu bringen. Auf Initiative der FOGRA wurde 1993 das ICC (International Color Consortium) durch Hersteller von Soft- und Hardware gegründet. Das ICC schuf einen einheitlichen Standard, der ein Management der unterschiedlichen Gerätefarbräume ermöglicht. Somit war nun ein plattformübergreifendes Farbmanagement möglich.

Das Farbmanagement erfüllt zwei Aufgaben. Zum einen werden unterschiedliche Farbräume von verschiedenen Geräten ineinander überführt. Dadurch wird gewährleistet, daß Farben richtig wiedergegeben werden. Zum anderen wird ein Mechanismus bereitgestellt, nach dem mit Farben verfahren wird, die auf dem System nicht dargestellt werden können. Die dazu nötige Farbraumanpassung

(Gammutmapping) kann auf drei Arten erfolgen:

- Farbmétrisch richtig. Nicht darstellbare Farben werden dabei einfach abgeschnitten.
- Empfindungsgemäß richtig. Die Farben werden so angepaßt, daß der Gesamteindruck des Bildes bzw. das Verhältnis der Farben zueinander erhalten bleibt. Das genaue Verfahren ist herstellerabhängig.
- Mit richtiger Sättigung. Dabei wird die jeweilige Farbachse angepaßt, so daß die maximale Sättigung einer Farbe in einem Farbraum auch der maximalen Sättigung in einem anderen Farbraum entspricht. Das genaue Verfahren ist herstellerabhängig [4, S. 82f].

Um die Farbraumanpassung zu gewährleisten muß für jedes Gerät in einer Verarbeitungskette ein Farbprofil erstellt werden. In diesen Profilen sind die Informationen enthalten, die benötigt werden, um den geräteabhängigen in einen geräteunabhängigen Farbraum umzurechnen. Dieser unabhängige Farbraum wird auch als PCS (profile connection space = Profil verbindender Farbraum) bezeichnet. Dazu wird entweder der CIEXYZ oder der CIELAB Farbraum verwendet.

Mit der Hilfe dieser geräteunabhängigen Farbwerte kann das Farbmanagement Modul, das entweder im Betriebssystem (wie z.B. ColorSync oder ICM) oder direkt im Programm verankert ist, die Farbräume von unterschiedlichen Geräten ineinander überführen.

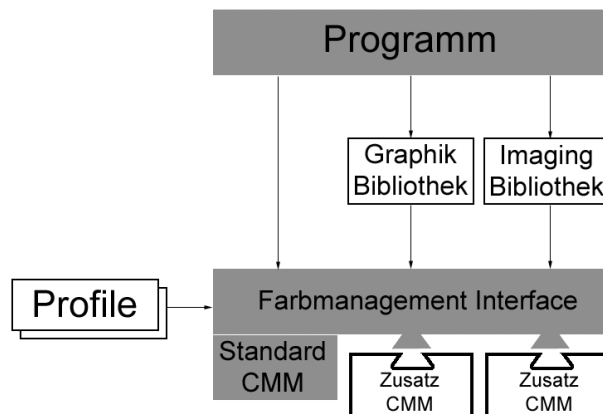


Abbildung 9 - Implementierung des CMM [4, S. 2]

Grundsätzlich gibt es drei verschiedene Typen für Geräteprofile: Eingabeprofile (z.B. für Scanner), Displayprofile (z.B. für Monitore) und Ausgabeprofile (z.B. Drucker). Jeder dieser Typen kann noch Unterklassen enthalten. Des Weiteren gibt es noch vier geräteunabhängige Profiltypen, die z.B. für abstrakte Farbraumtransformationen verwendet werden.

Für diese Arbeit sind jedoch nur die Monitorprofile von Bedeutung.

Aufbau von Farbprofilen

Das ICC (International Color Consortium) hat das Dateiformat für Profile standardisiert. Eine Profildatei besteht aus einem Dateikopf (den Header), der allgemeine Informationen wie die Größe des Profils, den PCS (CIE XYZ oder LAB), die Art des Profils (für Eingabegerät, Display oder Ausgabegerät) und andere Informationen enthält.

Die eigentlichen Daten werden in Blöcken innerhalb des Profils gespeichert. Auf diese Weise kann jede Anwendung bzw. das Farbmanagement Modul des Betriebssystems genau auf die Informationen zugreifen, die benötigt werden, ohne jedesmal das komplette Profil laden zu müssen.

Die einzelnen Datenblöcke werden durch sogenannte Tags identifiziert. Eine Liste der Tags, die ein Profil enthält, folgen dem Dateikopf. In dieser Liste stehen zum einen die Signaturen der enthaltenen Tags, sowie Länge und Speicherort der eigentlichen Daten innerhalb des Profils.

Jedes Tag hat eine eigene Signatur und ein festgelegtes Format für die eigentlichen Daten. Sowohl die Signatur als auch die Datenformate sind in den ICC-Spezifikationen festgelegt.

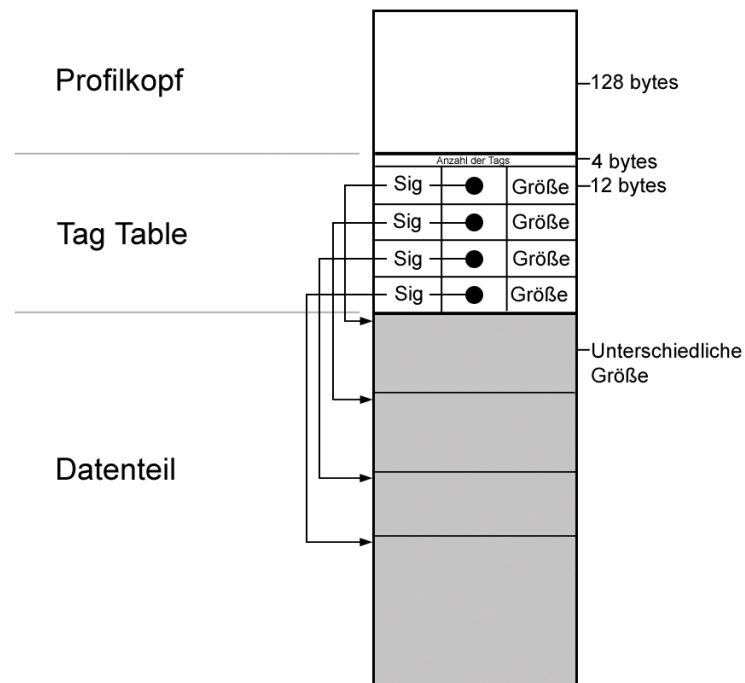


Abbildung 10 - Aufbau eines ICC Profils [4, S. 12]

Diese Art der Codierung ermöglicht ein offenes Verfahren, da Softwarehersteller für bestimmte Zwecke auch eigene Tags definieren können. Standardprogramme ignorieren diese sogenannten "privaten Tags" und holen sich aus dem Profil nur die Informationen, die sie auch verwerten können. Diese herstellereigenen Tags sollten beim ICC registriert werden. So wird verhindert, daß zwei verschiedene private Tags die selbe Signatur verwenden.

Für unterschiedliche Profiltypen bzw. Gerätetypen gibt es bestimmte Informationen, die in einem Profil enthalten sein müssen, und andere die enthalten sein können, um erweiterte Funktionen zu ermöglichen.

Bei Farbmonitoren gehören zu den erforderlichen Tags unter anderem die Matrix zur Umrechnung von RGB nach PCSXYZ, die Gammakurven, sowie der Weißpunkt des PCS. Die Umrechnung erfolgt dabei immer farbmétrisch korrekt [4, S. 20].

Zu den zusätzlichen Tags gehört z.B. eine multidimensionale LUT (look up table), die zur Umrechnung von RGB nach CIELAB verwendet wird. Dabei gibt es jeweils eine Tabelle zur farbmétrisch, empfindungsgemäß und sättigungsrichtigen Farb-anpassung [4, S. 20]. Nicht jede dieser Tabellen muß in einem Profil enthalten sein. Die ICC-Spezifikationen legen fest, wie verfahren werden soll, wenn nicht alle Tabellen enthalten sind.[4, S. 4].

Hierbei ist zu beachten, daß die im Profil enthaltene Matrix, die zur Umrechnung von RGB nach XYZ dient, keine absoluten CIEXYZ Werte liefert, sondern Werte, die relativ zum Weißpunkt des PCS liegen. Diese werden als PCSXYZ Werte bezeichnet. Der Weißpunkt des PCS liegt bei Farbmonitoren gemäß den ICC-Spezifikationen immer auf D50 (selbst wenn der Monitor tatsächlich z.B. auf eine Farbtemperatur von D65 oder D93 gestellt ist) [4, S. 38]. Die Umrechnung der relativen Koordinaten des PCS in absolute CIEXYZ Koordinaten kann nach drei Methoden vorgenommen werden.

Die einfachste Methode ist die lineare Reskalierung der CIEXYZ Werte, auch bekannt als "falsche" von Kries Transformation:

$$\begin{bmatrix} X_{absolut} \\ Y_{absolut} \\ Z_{absolut} \end{bmatrix} = \begin{bmatrix} \frac{X_W}{X_{D50}} & 0 & 0 \\ 0 & \frac{Y_W}{Y_{D50}} & 0 \\ 0 & 0 & \frac{Z_W}{Z_{D50}} \end{bmatrix} \begin{bmatrix} X_{relativ} \\ Y_{relativ} \\ Z_{relativ} \end{bmatrix}$$

bzw.

$$X_{\text{absolut}} = X_W / X_{D50} * X_{\text{relativ}}$$

$$Y_{\text{absolut}} = Y_W / Y_{D50} * Y_{\text{relativ}}$$

$$Z_{\text{absolut}} = Z_W / Z_{D50} * Z_{\text{relativ}}$$

mit:

X_{absolut} : absoluter CIE Wert

X_W : Wert des tatsächlichen Weißpunktes

X_{relativ} : relativer Wert des PCS

X_{D50} : Wert von D50

Ebenso für Y und Z.

Diese Art der Transformation wird als "falsche von Kries Transformation" bezeichnet, da nicht mit den Empfindlichkeiten der Zapfen des menschlichen Auges gearbeitet wird, sondern mit den Normspektralwertkurven des CIEXYZ Farbraumes.

Korrekt ist die von Kries Transformation oder die linearisierte Bradford / CIE-CAM97s Transformation. Dabei werden die CIEXYZ Werte zuerst linear unter Einbezug der spektralen Empfindlichkeit der Zapfen des menschlichen Auges in einen neuen Farbraum transformiert. Die Anpassung an die neuen Betrachtungsbedingungen erfolgt durch eine lineare Reskalierung dieser Farbwerte. Die so errechneten Werte werden anschließend linear in den CIEXYZ Farbraum transformiert. Der genaue Mechanismus und die Transformationsmatrix kann z.B. in [4, S 109f] nachgelesen werden.

Seit der Version 4 der ICC Profil Spezifikationen sollte in jedem Profil ein Tag für die Farbanpassung enthalten sein, wenn nicht D50 verwendet wird. In diesem Tag ist eine 3x3 Matrix enthalten, mit welcher die relativen PCSXYZ Werte nach einer der drei oben erwähnten Methoden in absolute CIEXYZ Werte umgerechnet wer-

den können [4, S. 35]. Allerdings wird diese Funktion derzeit von den meisten Systemen noch nicht unterstützt.

Da in dieser Arbeit nur mit dem CIELAB Farbraum gearbeitet wird - der ebenfalls auf der falschen von Kries Transformation aufbaut - werden die letzten beiden Möglichkeiten zur Farbanpassung hier nicht verwendet.

Für die Farbraum Transformationen von RGB nach CIELAB werden in dieser Arbeit aus dem Monitorprofil die Matrix zur Konvertierung von RGB nach PSCXYZ, die Gammakurven in Form einer LUT, sowie der Weißpunkt des PCS verwendet. Die genaue Berechnung wird in Kapitel 2.1.3 erklärt.

2 Material und Methoden

2.1 Das Programm

2.1.1 Grundlagen

Das Programm Shiraz wurde in Java programmiert. Als Entwicklungsumgebung wurde Code Warrior 7 academic eingesetzt. Die Quellcodes mitsamt der Projektdatei sind auf der beiliegenden CD-ROM zu finden.

Shiraz setzt als Laufumgebung Java 1.3.1 und Java 3D voraus. Die entsprechenden Installer sind ebenfalls auf der CD-ROM vorhanden.

Eine ausführliche Dokumentation der verwendeten Klassen und deren Funktionen findet sich als Javadoc-HTML ebenfalls auf der CD-ROM.

2.1.2 Einführung

Das Programm Shiraz ermöglicht die Auswahl beliebiger monochromer Farbflächen. Und zwar wahlweise aus dem RGB-Farbraum wie auch aus dem CIELAB-Farbraum. Dazu wird in der Benutzeroberfläche über einen Button die Farbraumwahl getroffen. Entsprechend der Entscheidung werden drei Slider aktiviert, mittels derer eine Bewegung auf den Achsen des jeweiligen Farbraummodells möglich wird. Zur Orientierung werden die Koordinatenwerte in Textfeldern unterhalb der Slider angezeigt. Visuell wird die momentan gewählte Farbe in einem Vorschaufeld in der Mitte der Benutzeroberfläche wiedergegeben.

Über den Button *OK* wird der Farbwert gespeichert und abgelegt. Ein Button *Abbrechen* ermöglicht den Ausstieg aus der Benutzeroberfläche.

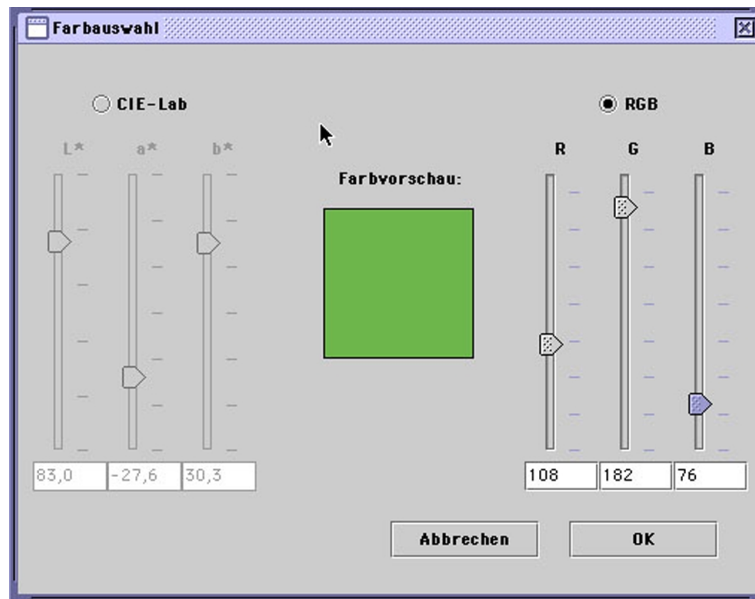


Abbildung 11 – Auswahl RGB

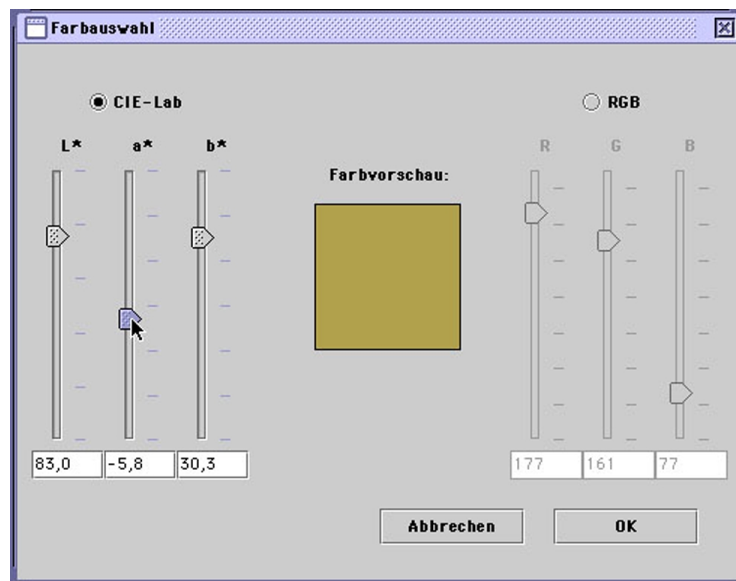


Abbildung 12 - Auswahl Lab

Für den Fall, daß die Farbwerte im CIELAB-Farbraum so gewählt werden, daß diese die Darstellungsmöglichkeiten des RGB-Farbraums überschreiten, wird im Vorschaufeld die Warnung „Illegale Farbe“ ausgegeben.

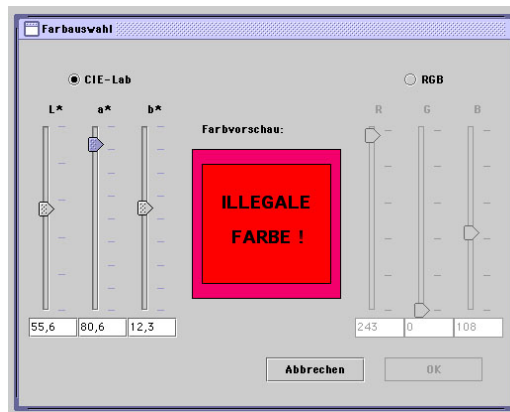


Abbildung 13 - Illegale Farbe

2.1.3 Farbumrechnung

Die Umrechnung der Farben erfolgt mit den Daten des gewählten Monitorprofils. Im Constructor der Klasse *ICC_RGBmetrik* wird das Profil geöffnet und die Matrix sowie die TRCs eingelesen. Da in einem Profil nur die Daten für die Rechnung von RGB nach PCSXYZ enthalten sind, müssen für die Rechnung von PCSXYZ nach RGB sowohl die Matrix als auch die TRCs invertiert werden.

Die Invertierung der Matrix übernimmt die Klasse *Matrix3f*, die Bestandteil des Java 3D Paketes ist. Die Invertierung der TRCs wird von *ICC_RGBmetrik* übernommen. Dazu werden die Werte der inversen TRCs schrittweise aus den originalen TRCs interpoliert. Die Interpolation erfolgt dabei auf der Voraussetzung, dass Zwischenwerte linear interpoliert werden können. Da dies eine Vorbedingung für TRCs nach den ICC-Spezifikationen ist, kann hier so verfahren werden.[4, S. 47]

Zur Umrechnung von CIEXYZ nach CIELAB wird der Weißpunkt verwendet, der sich im MediaWhitePoint Tag des Profils befindet. In unserer Versuchsreihe ist eine Adaption vom PCS zu den tatsächlichen CIEXYZ Werten nicht nötig. Die CIELAB Werte können direkt aus dem MediaWhitePoint und den PCSXYZ Werten errechnet werden.

Für die Umrechnung von RGB nach CIELAB werden die RGB Werte zuerst von den linearen in nichtlineare Werte mit Hilfe der TRCs konvertiert. Diese werden dann über die Matrix in den PCSXYZ Farbraum umgerechnet und von dort über den Weißpunkt des Profils nach CIELAB gerechnet.

Für die Umrechnung von CIELAB nach RGB wird einfach der umgekehrte Weg verwendet (mit der inversen Matrix und den invertierten TRCs).

Dabei ist zu beachten, daß auf Grund der Quantisierung der RGB Werte zwar jedem RGB Wert genau ein CIELAB Wert entspricht, jedoch ein paar CIELAB Werte - wenn sie eng genug beieinander liegen - den selben RGB Wert ergeben können! Daher wird bei der numerischen Eingabe von CIELAB Werten im entsprechenden Fenster dieser zunächst nach RGB und anschließend wieder nach CIELAB gerechnet. Dieser Wert erscheint dann im Eingabefeld. So kann es passieren, dass bei der Eingabe z.B. des Wertes 43.6 für L^* der Wert nach Drücken der *Enter* Taste auf 43.5 springt.

2.1.4 Das Verrauschen

In der Klasse *Sequenz.Rauschen* werden die Rauschwerte gebildet und zu den Farben hinzugerechnet. Zur Erzeugung der Rauschwerte wird die in Java implementierte Klasse *Random* verwendet. Diese erzeugt durch die Funktion *nextGaussian()* eine annähernd $N(0;1)$ verteilte Zufallsfunktion. Da bei gleichem Startwert (dem seed-Wert) von *Random* die Zufallsfunktion immer die selbe Abfolge von "Zufallszahlen" liefert, ist sichergestellt, dass jede erzeugte Farbe mit dem selben Muster verrauscht wird.

Als Maßzahl für den Verrauschungsgrad wird dabei die Varianz verwendet. Da die Zufallsfunktion keine tatsächlich $N(0;1)$ verteilte Zahlenfolge liefert, wurde die verwendete Zahlenfolge statistisch untersucht.

Die Abweichung wurde allerdings nicht als Korrekturfaktor in die Ergebnisse mit

eingerechnet. Das Ziel dieser Untersuchung ist nicht die Ermittlung einer absoluten Rauschschwelle. Daher kann diese Abweichung, die sich in allen Richtungen gleich auswirkt, vernachlässigt werden.

	Ohne Grenze	Grenzen bei $\pm 2,2$
Mittelwert	0,0009	0,0007
Varianz	0,9970	0,9468
Max. Wert	4,9065	2,20
Min. Wert	-4,1244	-2,20
Außerhalb	-	2,72%
Anzahl Werte	60000	60000
Startwert (Seed)	1	1

Da kein Gammutmapping erfolgt, müssen die Farben auf Werte begrenzt werden, die im RGB Farbraum des Monitors noch dargestellt werden können. Es muss also darauf geachtet werden, dass nicht nur die Grundfarbe, sondern auch der maximale Farbwert der Verrauschung noch darstellbar ist.

Damit der verwendbare Farbraum nicht zu sehr eingeschränkt wird, wurde daher den Wertebereich der Zufallszahlenfolge auf Werte zwischen -2,2 und +2,2 beschränkt. Die Werte, die außerhalb dieser Grenze liegen, werden auf den Grenzwert reduziert.

Damit ergibt sich bei dem gewählten Maximalwert von $\Delta E = 10$ ein Raum von $\Delta E = 22$ um jeden Farbwert, der noch darstellbar sein muss. Das Programm Shiraz kontrolliert schon beim Einstellen der Farbe, ob diese Bedingung erfüllt wird. Falls nicht, wird der OK Knopf deaktiviert und eine Farbwarnung erscheint im Farbfeld.

Nach dem Erzeugen der Rauschwerte in der Funktion *Rauschen.rauschzahlen()* werden die einzelnen Sequenzen in der Funktion *Rauschen.bilderErzeugen()* errechnet. Um die Bilder zu verrauschen, werden die CIELAB Werte zuerst in L^* , C^* und h^* umgerechnet. Diese Werte werden für die Berechnung des Farbabstandes

ΔE benutzt. Die Umrechnung ist eine einfache Umwandlung der kartesischen Koordinaten a^* und b^* in die Polarkoordinaten C und h . Die Helligkeit L^* bleibt unverändert bestehen.

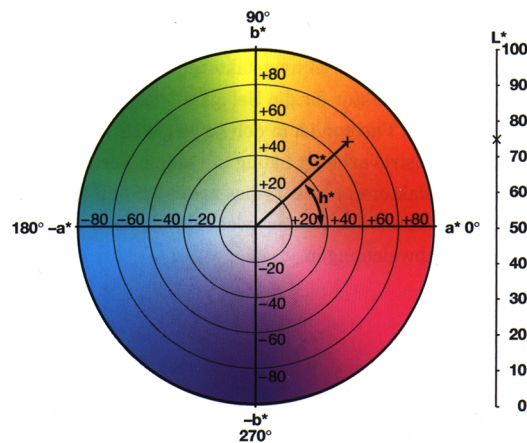


Abbildung 14 - Koordinaten im LCH Farbraum [10, S. 24]

Diese Werte werden anschließend nach folgendem Schema mit den Rauschwerten verrechnet:

$$L' = L^* + \text{Rauschwert} * \text{maximales Delta} * \text{aktueller Schritt} / \text{Anzahl der Schritte}$$

Für C und h ist das Verfahren identisch. Nach der Verrauschung erfolgt die Rückrechnung nach L^* , a^* und b^* und anschließend nach RGB. Die so ermittelten RGB Werte werden dann in ein Bild geschrieben.

Für jede der drei Richtungen L , C und h wurde eine Sequenz mit 31 Bildern von $\Delta E = 0$ bis $\Delta E = 10$ erstellt. Der Abstand der einzelnen Bilder der Sequenz beträgt also $\Delta E = 0,33$. Eine Ausnahme bildet dabei die Sequenz Δh . Hier sind die Schritte $\Delta h = 0,33$, was sich aber je nach Farbort der Grundfarbe in unterschiedlichen ΔH und damit auch ΔE auswirkt.

Die drei Sequenzen werden zum Schluß in eine Datei geschrieben.

Desweiteren wird eine Testsequenz erzeugt, deren erstes Bild die Grundfarbe und die maximale Farbabweichungen im Vergleich zeigt. Die weiteren Bilder zeigen die Abweichungen einzeln in der Reihenfolge $+\Delta L$, $-\Delta L$, $+\Delta C$, $-\Delta C$, $+\Delta h$ und $-\Delta h$ vollflächig an. Diese können somit farbmétrisch ausgemessen werden.

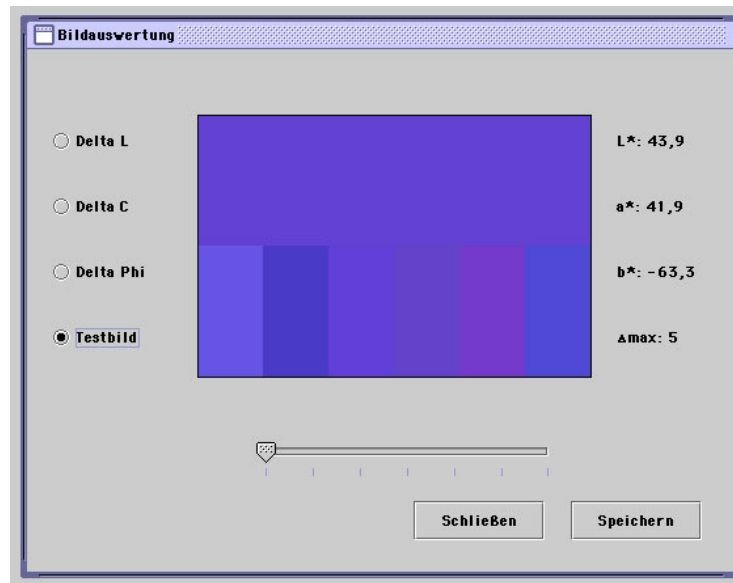


Abbildung 15 - Testbild

2.1.5 Die Auswertung

Bei der Auswertung soll der Proband bestimmen, ab welchem Grad er eine Farbe als verrauscht empfindet. Dazu können die einzelnen Bilder jeder Sequenz über einen Schieberegler angewählt werden. Die Reihenfolge ist dabei von der Grundfarbe (keine Verrauschung) bis maximal verrauscht.

Beim Erreichen der Rauschschwelle betätigt der Proband dann einen Schalter, der den aktuellen Wert des Schiebereglers in den dazugehörigen Delta-Wert umrechnet und abspeichert. Anschließend wird die nächste Sequenz geöffnet. Die Reihenfolge der Farben wurde zufällig gewählt, ist aber bei jedem Probanden die selbe. Innerhalb einer Farbe ist die Reihenfolge immer L^* , C^* und h^* .

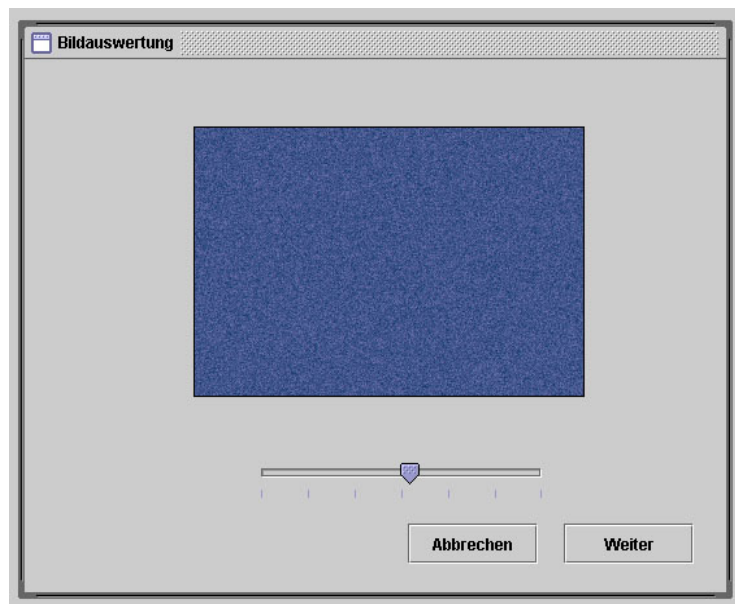


Abbildung 16 - Auswertung

Die so erhaltenen Messergebnisse lassen sich exportieren. Die Delta-Werte werden dabei auf zwei Nachkommastellen genau ausgegeben. Der Mittelwert und die Varianz werden ebenfalls direkt ermittelt.

Den Probanden wurde dabei vorgegeben, daß sie, falls kein Rauschen wahrgenommen werden kann, den Maximalwert einstellen müssen. Die Statistik wird vor der Auswertung um diese Werte bereinigt.

2.2 Der Arbeitsplatz

2.2.1 Verwendete Geräte

Der verwendete Computer läuft unter dem Betriebssystem Windows 2000®. Die im Rechner eingebaute Grafikkarte ist vom Typ Matrox Millennium mit 4 MB Grafikspeicher.

Der verwendete Monitor ist ein Step 21 Zoll Mod.Nr. NUB1107STTUW. Zur Kali-

brierung wurde die Software Profile Maker 3 und der Messkopf X-rite Monitor Optimizer DTP-92 verwendet.

Vor der Kalibrierung und der Profilerstellung sollte der Monitor mindestens eine Stunde laufen. Dadurch wird gewährleistet, daß sich der Monitor auf Arbeitstemperatur befindet. Farbschwankungen, wie sie in der Aufwärmphase eines Monitors oft zu beobachten sind, werden so stark reduziert.

Vor dem Erstellen des Profils wird der Monitor mit Hilfe von Profile Maker 3 kalibriert. Die Farbtemperatur wurde dabei auf 5500 K eingestellt. Die Leuchtdichte des Monitors beträgt laut Profile Maker 90 cd/m².

Nach der Kalibrierung wird der Monitor durch die Software ausgemessen und anschließend ein ICC-Profil erstellt. Dieses Profil *Diplommonitor15.05.icc* ist auf der CD-ROM zu finden. Es wurde in Windows als Standardprofil registriert.

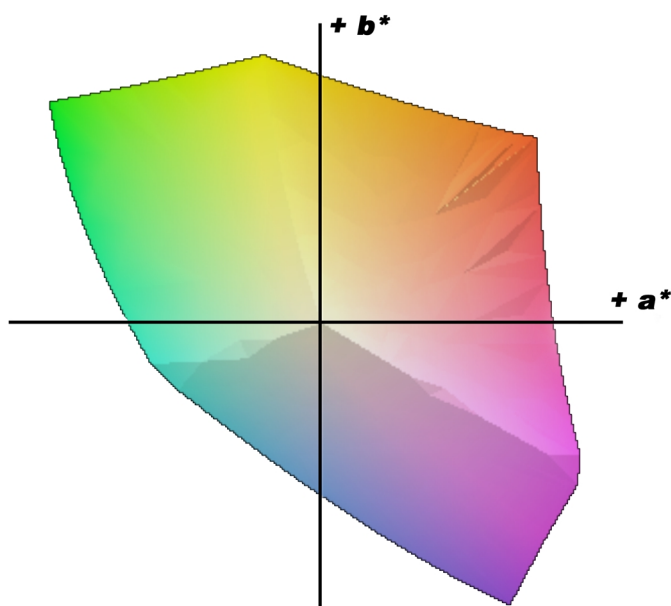


Abbildung 17 - Der Farbraum des Testmonitors in CIELAB

Das Profil wird ebenfalls an das Programm Shiraz übergeben. Es dient als Grundlage der Farbkonvertierung bei der Erstellung der Testfarben.

Es ist jedoch zu beachten, daß handelsübliche Monitore über ihre Bildfläche starken Farbschwankungen unterliegen können. Diese sind in ihrer Stärke farbabhängig. Beim verwendeten Diplommonitor wurden bei einer Beispielfarbe (Rot) Schwankungen über die Bilddiagonale von 7,5 ΔE zum Referenzpunkt in der Mitte gemessen. Messgeräte wie das DTP 92 müssen daher immer an derselben Stelle fixiert werden. Für den nachfolgenden Versuch kann so aber eine Wahrnehmungsungenauigkeit der Farbdifferenzen für den Probanden nicht ausgeschlossen werden.

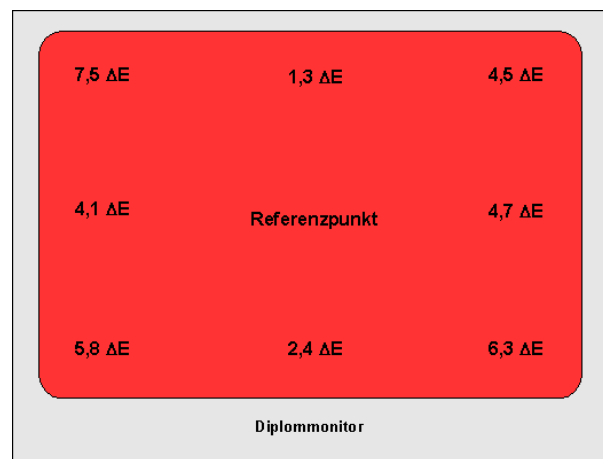


Abbildung 18 - Farbgenauigkeit des benutzten Monitors

2.2.2 Die Testfarben

Die Testfarben werden so ausgewählt, daß ein möglichst großer Bereich im zur Verfügung stehenden Farbraum abgedeckt wird. Es sollen die additiven und subtraktiven Grundfarben vertreten sein, sowie zwei Grautöne und zwei Hauttöne.



Die Farbpalette in Abb. 19 soll nur einen schnellen visuellen Überblick über die gewählten Testfarben geben.

Die Farbdarstellung kann durch das Druckprofil nicht den Originalfarben entsprechen. Die genauen CIELAB Werte können der unten stehenden Tabelle entnommen werden.

Anordnung der Farben:

Braun	Haut 1	Mittelgrün
Cyan	Haut 2	Orange
Dunkelgrau	Hellgrau	Pastell
Dunkelviolet	Magenta	Pseudorot
Gelb	Blau	Mittelblau

Abbildung 19 - Die verwendeten Testfarben

Durch den eingeschränkten Farbraum können die Grundfarben nicht als reine bzw. hoch gesättigte Farben verwendet werden. In Abbildung 20 ist der Farbraum des Monitors im CIELAB Farbraum dargestellt (rote Punkte). Die Werte wurden mit Hilfe des verwendeten Monitorprofils nach der Kalibrierung errechnet. Die innere Markierung stellt den reduzierten Farbraum dar (siehe Kapitel 2.1.4). Die blauen Punkte markieren die Farborte der von uns verwendeten Testfarben.

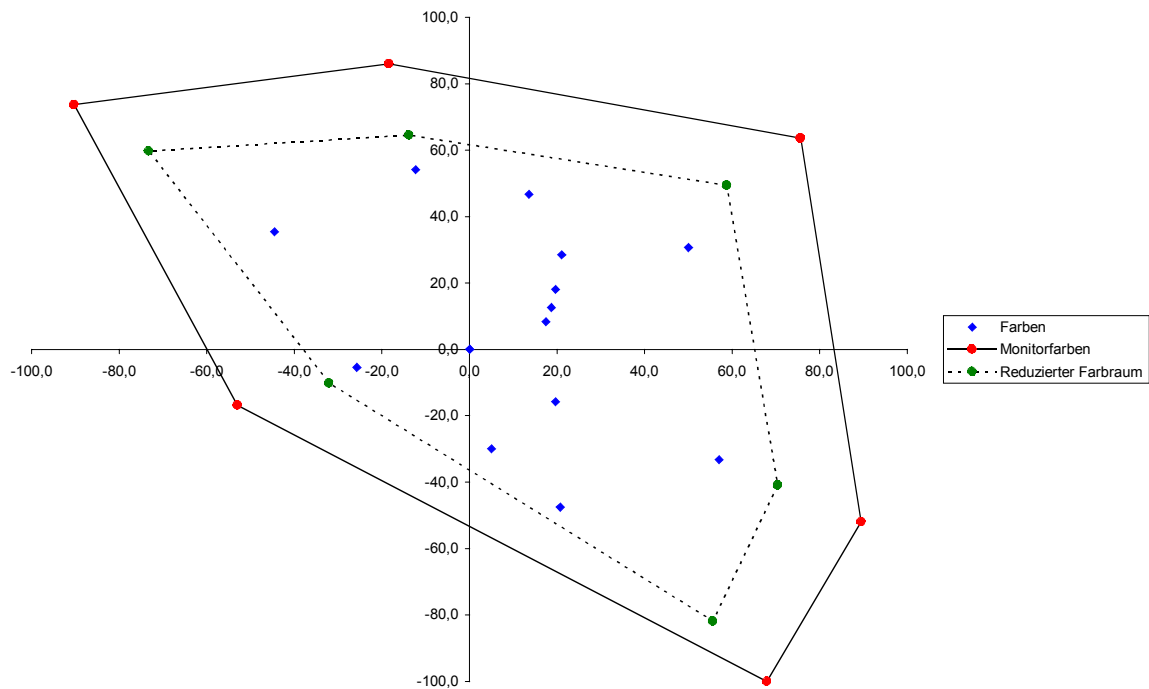


Abbildung 20 - Lage der Testfarben im CIELAB-Farbraum

Farbwerte der Testfarben

	L*	a*	b*	C	h
Blau	40,3	5,0	-30,0	30,4	279,5
Braun	36,2	21,1	28,5	35,5	53,5
Cyan	69,9	-25,7	-5,4	26,3	191,9
Dunkelgrau	30,0	0,1	0,1	0,1	45,0
Dunkelviolet	40,9	19,7	-15,8	25,3	321,3
Gelb	78,4	-12,3	54,2	55,6	102,8
Haut 1	64,0	18,7	12,6	22,5	34,0
Haut 2	56,0	19,7	18,1	26,8	42,6
Hellgrau	70,0	0,1	0,1	0,1	45,0
Magenta	53,7	57,0	-33,2	66,0	329,8
Mittelblau	42,2	20,7	-47,5	51,8	293,5
Mittelgrün	60,2	-44,5	35,4	56,9	141,5
Orange	66,0	13,6	46,7	48,6	73,8
Pastell	69,4	17,5	8,3	19,4	25,4
Pseudorot	50,7	50,0	30,7	58,7	31,5

Zur Kontrolle, ob die errechneten Farben auch korrekt am Monitor angezeigt werden, wurden stellvertretend einige Farben am Monitor mit dem *Monitor-Optimizer* und dem *Measuretool* von *Profilemaker 3* vermessen. Die Messwerte und deren Bewertung sind im Anhang 6.6.2 zu finden.

2.2.3 Arbeitsumgebung

Die Umgebung des Arbeitsplatzes wird so eingerichtet, daß die Beleuchtung nicht durch Tageslicht beeinflusst wurde, sondern nur durch zwei Leuchtstoffröhren vom Typ Osram Lumilux L18W/12 mit einer Farbtemperatur von 5460 K (gemessen mit Gossen Colormaster 3F) erfolgt.

Durch die Angleichung der Farbtemperatur von Monitor und Hintergrundbeleuchtung wird sichergestellt, daß die Farbadaptation der Probanden nicht zwischen Hintegrund und Monitor schwankt, sondern auf einen Wert festgelegt ist. Der Unterschied von ca. 40 K ist zu vernachlässigen.

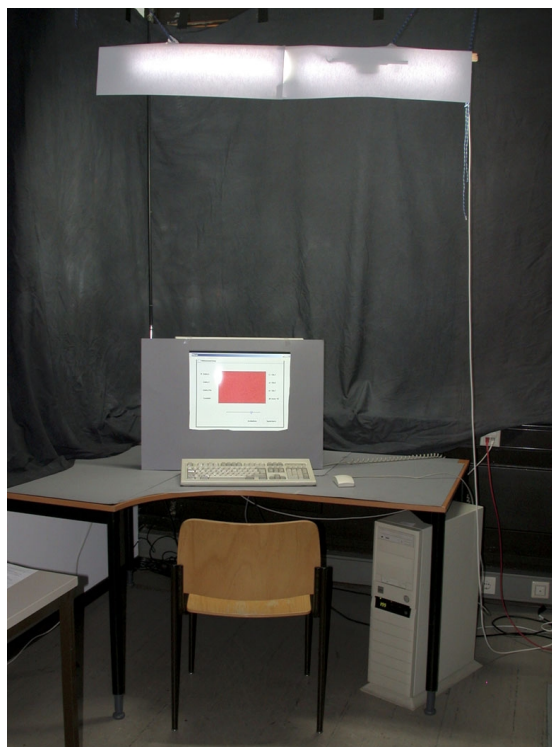


Abbildung 21 - Der verwendete Arbeitsplatz

Da jedem Probanden der Zweck der Untersuchung und die Funktion des Programms am Arbeitsplatz erläutert wurden, hatten dessen Augen genug Zeit, sich sowohl an die Helligkeit, als auch an die Farbtemperatur des Lichtes zu adaptieren.

Die Röhren werden so angebracht, daß dem Proband kein direktes Licht in die Augen fällt und keine Reflexionen auf dem Monitor entstehen.

Die Helligkeitsverteilung im Raum wird mit einem Gossen Colormaster 3F gemessen. Die Messung erfolgt dabei in einem Abstand von 20 cm von der zu messenden Fläche, wobei kein direktes Licht von den Röhren auf die Messzelle fällt. Das Messen in einem definierten Abstand läßt so die Möglichkeit zu, die erhaltenen Beleuchtungsstärken zur Beleuchtungsstärke des Monitors ins Verhältnis zu setzen. Da die Beleuchtungsstärke und die Leuchtdichte einander proportional sind, können so auch die ermittelten Beleuchtungsstärkeverhältnisse und die Leuchtdichteangaben der DIN 5035-7 unmittelbar miteinander verglichen werden. So kann der Arbeitsplatz eingemessen werden ohne die Leuchtdichte direkt ermitteln zu müssen.

Objekt	Messwerte [lx]	Verhältnis L/M	DIN 5035-7
Monitor	90	-	-
Tastatur	70	0,78	1
Schreibtisch	35	0,39	0,3 – 0,8
Hintergrund	15	0,17	0,1 – 10
Monitorumgebung	30	0,33	0,3 – 0,8

Um den Monitor visuell in den Vordergrund zu stellen wird der Arbeitsplatz an den unteren Grenzen der DIN 5035-7 gehalten.

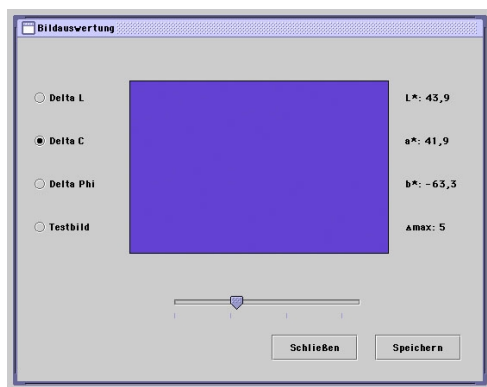
Betrachtungsabstand

Der ideale Betrachtungsabstand für den Probanden errechnet sich über den opti-

malen Sehwinkel (Das Gesichtsfeld, das ohne Kopfbewegung wahrgenommen werden kann) und die Monitorbreite. Um eine ständige Augenbewegung zu vermeiden, wurde der ideale Betrachtungsabstand nicht nur für das Testfeld alleine berechnet!

Der Sehwinkel wurde auf 30° festgelegt.

Die Breite des 21 Zoll – Monitors beträgt 38cm.



$$\begin{aligned}\text{Opt. Abstand} &= 38,0\text{cm} / \tan 30^\circ \\ &= \underline{65,8\text{cm}}\end{aligned}$$

Optimaler
Betrachtungsabstand

Die Maximalentfernung für eine Monitorbetrachtung wird jedoch über die Pixelgröße und den kleinsten auflösbaren Winkel (2') ermittelt:

$$\begin{aligned}\text{Pixelg.} &= \text{Monitorbreite} / \text{Auflösung} \\ &= 38,0\text{cm} / 640 \\ &= 0,59\text{mm}\end{aligned}$$

Somit:

$$\begin{aligned}\text{Max. Abstand} &= 0,59\text{mm} / \tan 2' \\ &= \underline{101,4\text{cm}}\end{aligned}$$

Somit werden die Testpersonen für die Bildschirmbetrachtung so gesetzt, daß die Augen einen Minimalabstand von ca. 65cm, keinesfalls aber mehr als einen Meter haben.

2.2.4 Probanden und Versuchsablauf

Als Probanden wurden 32 Personen aus dem Fachbereich Photoingenieurwesen zufällig ausgewählt. Alle sollten zur Farbbeurteilung geschult sein. Jeder der Probanden führte den Versuch nur einmal durch. Eine Zeitvorgabe bestand nicht.

Die Sequenz konnte beliebig oft durchgespielt werden. Dabei konnte zu einer sicheren Beurteilung die Rauschschwelle durch Vor- und Zurückbewegen des Sliders genau eingegrenzt werden. Eine einmal abgegebene Bewertung konnte jedoch nicht rückgängig gemacht werden.

Durchschnittlich benötigten die Testpersonen etwa 25 Minuten für den Versuch.

3 Ergebnisse

Um eine statistisch möglichst genaue Aussage machen zu können, wurden im Laufe des Versuchs 32 Probanden auf ihr Farbrauschempfinden getestet (Zum Vergleich: Für den Luo und Rigg Datensatz der ΔE_{94} und ΔE_{00} -Abstandsformeln standen 20 Personen zur Verfügung [21]). Alle wurden vorher an die Umgebungsbedingungen adaptiert. Dies wurde schon durch die benötigte Zeit für die Erklärung des Versuchs bei entsprechendem Licht erreicht.

Die Ergebnisse der Einzelpersonen lassen sich dabei der Tabelle im Anhang entnehmen. Die Mittelwerte $\bar{}$ und die Varianzen s^2 können den unten stehenden Tabellen entnommen werden.

	Braun	Cyan	Dunkelgrau	Dunkelviolett	Gelb	Haut 1	Haut 2	Hellgrau
L*	36,2	69,9	30,0	40,9	78,4	64,0	56,0	70,0
a*	21,1	-25,7	0,1	19,7	-12,3	18,7	19,7	0,1
b*	28,5	-5,4	0,1	-15,8	54,2	12,6	18,1	0,1
$\bar{\Delta L}$	0,7	0,8	0,7	0,6	0,7	0,7	0,6	0,7
$s^2(L)$	0,1	0,1	0,1	0,0	0,1	0,0	0,0	0,1
$\bar{\Delta C}$	0,6	0,8	0,7	0,6	0,7	0,6	0,6	0,7
$s^2(C)$	0,1	0,1	0,0	0,0	0,1	0,1	0,1	0,1
$\bar{\Delta h}$	4,7	6,0	5,0	7,2	8,3	4,6	4,3	4,6
$s^2(h)$	2,8	6,5	2,5	4,8	5,1	3,0	2,2	2,7

	Magenta	Mittelblau	Mittelgrün	Orange	Pastell	Pseudorot	Blau
L*	53,7	42,2	60,2	66,0	69,4	50,7	40,3
a*	57,0	20,7	-44,5	13,6	17,5	50,0	5,0
b*	-33,2	-47,5	35,4	46,7	8,3	30,7	-30,0
$\bar{\Delta L}$	0,7	0,8	0,7	0,6	0,6	0,7	0,8
$s^2(L)$	0,1	0,2	0,1	0,0	0,1	0,1	0,1
$\bar{\Delta C}$	0,7	0,7	0,7	0,6	0,6	0,6	0,7
$s^2(C)$	0,1	0,0	0,1	0,1	0,0	0,1	0,1
$\bar{\Delta h}$	4,8	5,4	7,0	6,2	4,1	3,7	5,4
$s^2(h)$	3,9	3,6	9,7	6,7	2,8	1,7	5,3

Bei der Betrachtung der Einzelergebnisse fällt auf, daß die Wahrnehmungsschwellen für die Helligkeit und für die Sättigung sehr niedrig sind, vor allem aber auch bei allen Testpersonen sehr eng beieinander liegen. Das bedeutet, daß sowohl Mittelwert als auch die Varianz sehr klein ausfällt. Anders beim Bunttonwinkel: Dort scheinen die Wahrnehmungsgrenzen von Farbe zu Farbe aber auch von Person zu Person sehr unterschiedlich zu liegen.

Für einen schnellen Überblick verdeutlichen dies auch die folgenden Graphiken für die beispielhaft gewählte Farbe Braun.

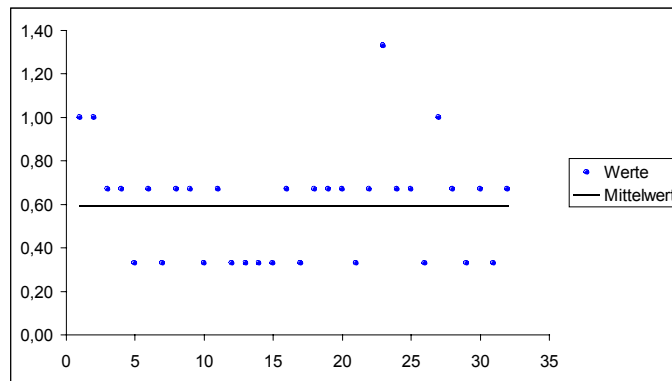


Abbildung 22 - ΔL Werte von Braun

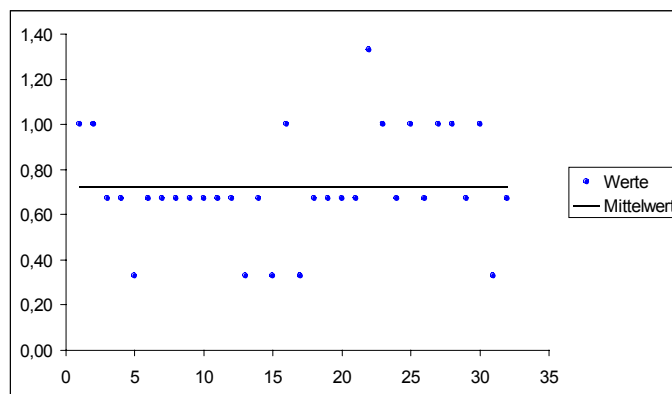


Abbildung 23 - ΔC Werte von Braun

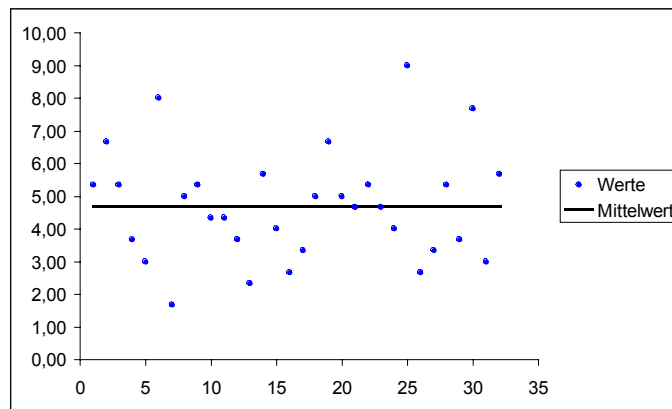


Abbildung 24 - Δh Werte von Braun

Zu beachten ist dabei, daß die Schrittweite bei $0,33 \Delta L$ bzw. ΔC bzw. Δh liegt. Bei den schon niedrigen L und C Werten scheinen die einzelnen Werte daher weiter auseinander zu liegen, als bei der Δh Sequenz. Tatsächlich wurde hier jedoch die Auflösungsgrenze der Sequenzen erreicht.

Deutlich wird aber auch, daß einzelne „Ausreißer“ bei allen Farben in der Δh Sequenz im oberen wie auch unteren Wertebereich die Statistik beeinflussen.

Die mit „10,0“ bezeichneten Höchstwerte werden aus der Tabelle und somit auch aus der weiteren Rechnung gestrichen, weil diese ein Nichterkennen von Farbrauschen der Einzelpersonen im Versuch kennzeichnen (siehe Kapitel 2.1.5).

Die Ursachen der weiteren Ausreißer sind im versehentlichen Mausclicken, durch Einbildung von Rauschen, fehlender Konzentration oder durch „Wettbewerbsdruck“ der Probanden zu suchen und werden ebenfalls eliminiert.

Um diese zu ermitteln wurden die Messwerte zusammen mit dem Mittelwert und den Grenzen $\text{Mittelwert} \pm \text{Varianz}/2$ in ein Diagramm eingetragen. Werte die sich weit außerhalb dieser Grenzen befinden, werden nicht in die Auswertung übernommen. Ziel dieser Art der Bereinigung ist es, Ausreißer auszuschließen, aber dennoch möglichst viele Werte in die Auswertung zu übernehmen.

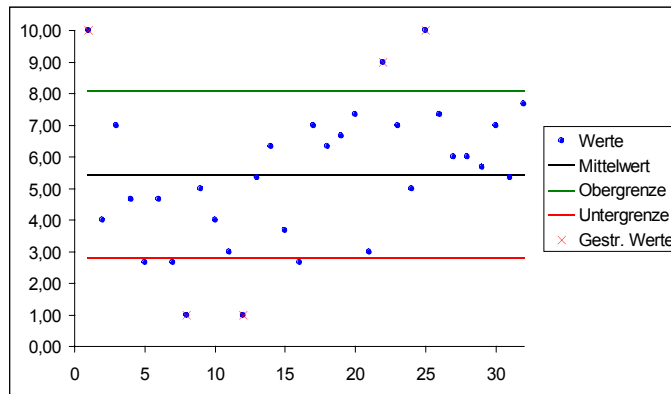


Abbildung 25 - Δh Werte von Blau

Abbildung 25 verdeutlicht die Auswahl der Ausreißer. Weitab von den Grenzen liegende Meßpunkte wurden ausgeschlossen, nahe bei der Grenze liegende Punkte wurden jedoch mit in die Statistik übernommen. Auf diese Weise fielen bei den 15 Farben 41 Werte per Definition als „Rauschen nicht erkannt“ und 30 Werte als zu weit abliegend aus der Statistik. Bei 480 Meßwerten sind das 14,8 Prozent.

	Braun	Cyan	Dunkelgrau	Dunkelviolett	Gelb	Haut 1	Haut 2	Hellgrau
L*	36,2	69,9	30,0	40,9	78,4	64,0	56,0	70,0
a*	21,1	-25,7	0,1	19,7	-12,3	18,7	19,7	0,1
b*	28,5	-5,4	0,1	-15,8	54,2	12,6	18,1	0,1
Ø ΔL	0,7	0,8	0,7	0,6	0,7	0,7	0,6	0,7
s² (L)	0,1	0,1	0,1	0,0	0,1	0,0	0,0	0,1
Ø ΔC	0,6	0,8	0,7	0,6	0,7	0,6	0,6	0,7
s² (C)	0,1	0,1	0,0	0,0	0,1	0,1	0,1	0,1
Ø Δh	4,4	5,6	5,0	6,3	7,8	4,2	4,1	4,6
s² (h)	1,4	4,3	2,5	2,8	3,4	1,2	1,2	2,7

	Magenta	Mittelblau	Mittelgrün	Orange	Pastell	Pseudorot	Blau
L*	53,7	42,2	60,2	66,0	69,4	50,7	40,3
a*	57,0	20,7	-44,5	13,6	17,5	50,0	5,0
b*	-33,2	-47,5	35,4	46,7	8,3	30,7	-30,0
∅ ΔL	0,7	0,8	0,7	0,6	0,6	0,7	0,8
s² (L)	0,1	0,2	0,1	0,0	0,1	0,1	0,1
∅ ΔC	0,7	0,7	0,7	0,6	0,6	0,6	0,7
s² (C)	0,1	0,0	0,1	0,1	0,0	0,1	0,1
∅ Δh	4,7	5,4	5,9	6,0	4,3	3,7	5,3
s² (h)	3,0	2,6	5,5	4,8	1,3	1,4	2,6

Durch die Bereinigung der Messwerte ändern sich, wie aus der obigen Tabelle zu entnehmen ist, die Mittelwerte kaum, wohl aber die Varianzen. Diese werden deutlich kleiner.

	Braun	Cyan	Dunkelgrau	Dunkelviolett	Gelb	Haut 1	Haut 2	Hellgrau
ΔE₇₆								
Sequenz L	0,72	0,83	0,67	0,64	0,72	0,65	0,59	0,67
Varianz	0,06	0,10	0,05	0,02	0,06	0,04	0,04	0,08
Sequenz C	0,59	0,80	0,74	0,61	0,72	0,62	0,61	0,68
Varianz	0,06	0,09	0,03	0,04	0,07	0,05	0,05	0,07
Sequenz h	4,49	2,58	-	2,76	7,54	1,64	1,91	-
Varianz	0,86	1,95	-	1,24	3,28	0,46	0,54	-
ΔE₉₄								
Sequenz L	0,72	0,83	0,67	0,64	0,72	0,65	0,59	0,67
Varianz	0,06	0,10	0,05	0,02	0,06	0,04	0,04	0,08
Sequenz C	0,23	0,36	0,72	0,28	0,20	0,31	0,28	0,67
Varianz	0,02	0,04	0,03	0,02	0,02	0,02	0,02	0,07
Sequenz h	2,93	1,85	-	2,00	4,11	1,23	1,36	-
Varianz	0,56	1,40	-	0,90	1,79	0,34	0,39	-
ΔE₀₀								
Sequenz L	0,60	0,64	0,52	0,57	0,50	0,54	0,55	0,52
Varianz	0,05	0,08	0,04	0,02	0,04	0,03	0,04	0,06
Sequenz C	0,23	0,38	0,92	0,29	0,20	0,33	0,28	0,84
Varianz	0,02	0,04	0,04	0,02	0,02	0,03	0,02	0,09
Sequenz h	2,07	1,83	-	1,91	4,47	1,34	1,55	-
Varianz	0,65	1,39	-	0,88	1,98	0,37	0,44	-

	Magenta	Mittelblau	Mittelgrün	Orange	Pastell	Pseudorot	Blau
ΔE_{76}							
Sequenz L	0,68	0,76	0,69	0,59	0,64	0,70	0,76
Varianz	0,05	0,16	0,08	0,04	0,06	0,05	0,07
Sequenz C	0,68	0,65	0,68	0,59	0,61	0,62	0,70
Varianz	0,05	0,04	0,06	0,05	0,04	0,06	0,06
Sequenz h	5,43	4,89	5,81	5,07	1,44	3,77	2,81
Varianz	3,48	2,35	5,46	4,11	0,45	1,40	1,40
ΔE_{94}							
Sequenz L	0,68	0,76	0,69	0,59	0,64	0,70	0,76
Varianz	0,05	0,16	0,08	0,04	0,06	0,05	0,07
Sequenz C	0,17	0,19	0,19	0,18	0,32	0,17	0,29
Varianz	0,01	0,01	0,02	0,02	0,02	0,02	0,03
Sequenz h	2,73	2,75	3,14	2,93	1,11	2,01	1,93
Varianz	1,75	1,32	2,95	2,38	0,35	0,74	0,96
ΔE_{00}							
Sequenz L	0,65	0,69	0,60	0,48	0,50	0,70	0,67
Varianz	0,05	0,15	0,07	0,03	0,05	0,05	0,06
Sequenz C	0,17	0,19	0,19	0,18	0,36	0,17	0,29
Varianz	0,01	0,01	0,02	0,02	0,02	0,02	0,03
Sequenz h	2,36	3,74	2,66	3,57	1,14	2,25	2,46
Varianz	1,53	1,81	2,50	2,90	0,35	0,83	1,22

Bei den Farben Dunkelgrau und Hellgrau erfolgte keine Anpassung der Messwerte, da bei diesen Farben der Winkel $h = 0$ ist. Daher macht eine Auswertung nach Δh hier keinen Sinn.

Die so gewonnenen Schwellenwerte für das Erkennen von Rauschen in einheitlichen Farbflächen wurden nun in die drei wichtigsten ΔE Formeln des CIELAB Farbraums eingesetzt. So kann ein unmittelbarer Vergleich der Farbabstände der einzelnen Farbdifferenzformeln und somit deren Eignung für eine Quantifizierung des Farbrauschens erfolgen.

Zur Verdeutlichung der ΔE -Differenzen werden diese in Abhängigkeit der Farben für jede der drei Farbabstandsformeln in ein Diagramm gesetzt. Ganz deutlich wird hier, daß der Farbabstand bei keiner der Abstandsformeln – im besonderen für den Bunttonwinkel h – einen gleichmäßigen Wertebereich aufweist.

In den Grafiken stellen die Verbindungslinien zwischen den Farben keine Zwischenwerte dar, sondern dienen nur zur Verdeutlichung der ΔE Unterschiede!

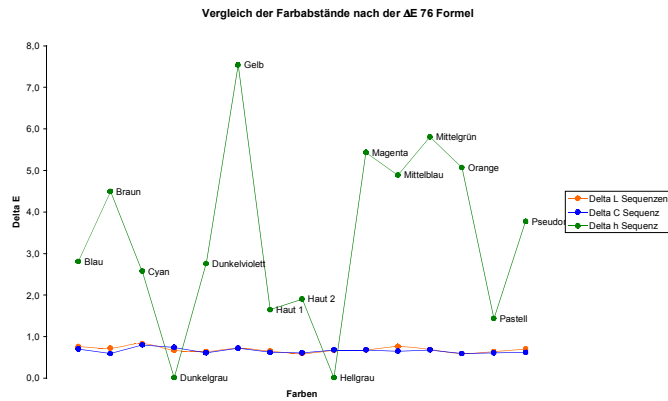


Abbildung 26 - Vergleich der Farbabstände nach der ΔE_{76} Formel

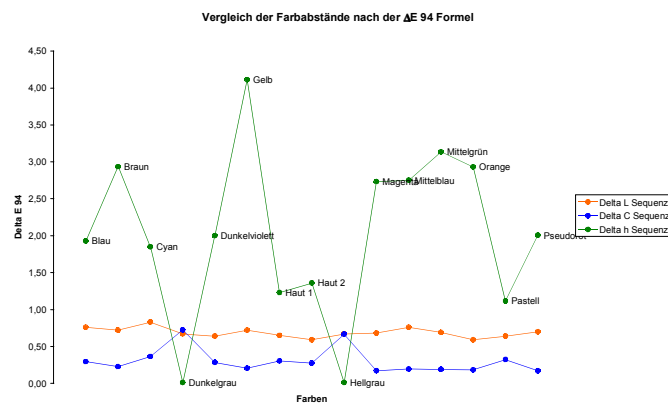


Abbildung 27 - Vergleich der Farbabstände nach der ΔE_{94} Formel

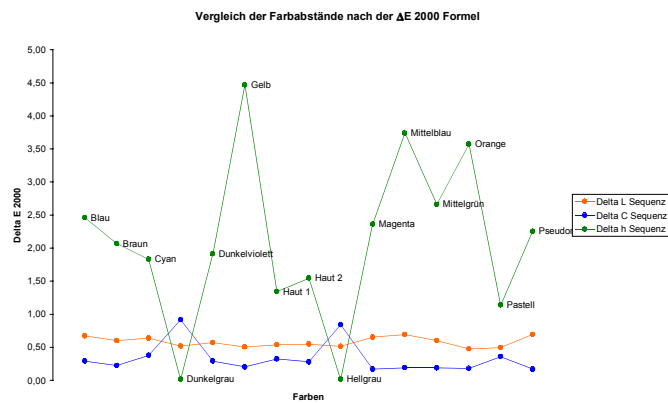


Abbildung 28 - Vergleich der Farbabstände nach der ΔE_{00} Formel

Nochmals verdeutlicht wird dies durch den Vergleich der Farbabstandsformeln bei der Δh Sequenz, bei welcher die größten Sprünge auftreten.

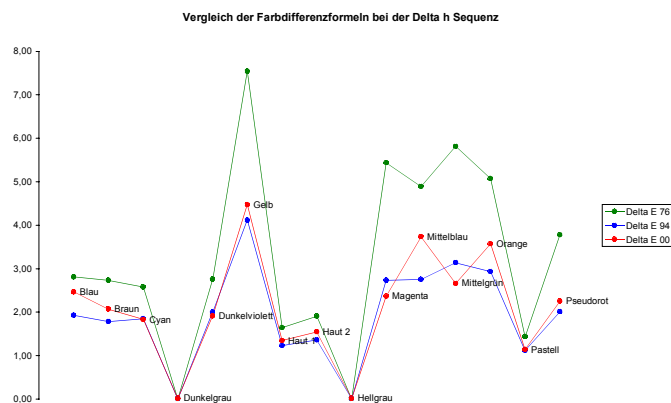


Abbildung 29 – Vergleich der Farbabstandsformeln bei der Δh Sequenz

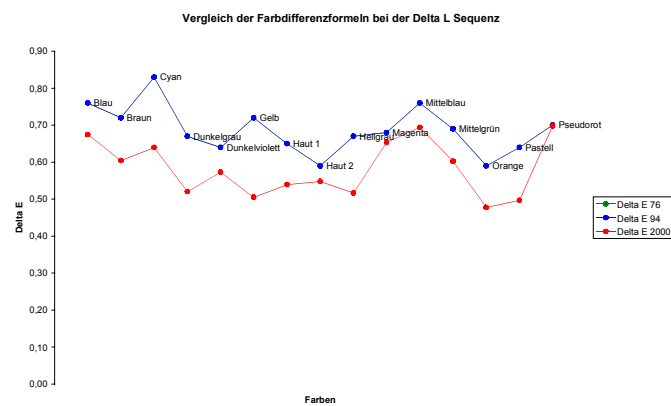


Abbildung 30 - Vergleich der Farbabstandsformeln bei der ΔL Sequenz

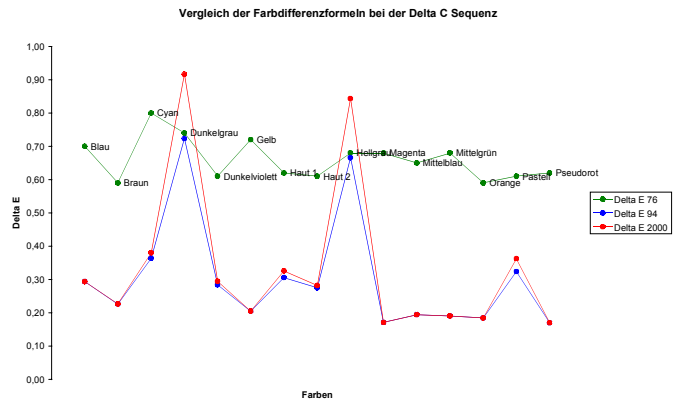


Abbildung 31 - Vergleich der Farbabstandsformeln bei der ΔL Sequenz

4 Diskussion

Grundsätzlich werden die zu erwartenden Ergebnisse bestätigt. Eine farbumabhängige Wahrnehmungsschwelle für Farbrauschen läßt sich nicht mit den bekannten Farbabstandsformeln des CIELAB-Farbraums beschreiben!

Die Messwerte zeigen vor allem beim Farbton deutlich unterschiedliche Schwellenwerte, während die Helligkeit und die Sättigung von Schwankungen relativ unberührt bleiben. Eine Quantifizierung von Farbrauschen ist daher im CIELAB-Farbraum nachweislich nicht möglich.

Es ist jedoch anzumerken, daß mit den zur Verfügung stehenden Möglichkeiten an Arbeitsmitteln eine Messung von absoluten Werten nicht machbar ist. Der verwendete Monitor ist zu ungenau in der Farbdarstellung, was sich in gemessenen Abweichungen von bis zu sieben ΔE über die Bildschirmdiagonale verdeutlicht. Ebenso zeigten die Meßgeräte und die zugehörige Meßsoftware nicht die gewünschte Präzision und Fehler in den Farbunterschieden ließen sich so nicht deutlich eingrenzen.

Um genauere Ergebnisse für den Schwellenwert bei der Helligkeit L und der Sättigung C zu bekommen, hätte eine kleinere Schrittweite als $\Delta E = 0,3$ gewählt werden müssen um so eine feinere Abstufung zu erzielen. Dies hätte eine höhere Anzahl von Bildern pro Sequenz zur Folge gehabt, was eine mit dem uns zur Verfügung gestellten Computer nicht mehr handlebare Datenmenge erzeugt hätte. Die gewählte Schrittweite von $\Delta E = 0,3$ ist insofern gerechtfertigt, als nach [3, S. 68] ein Farbunterschied von $\Delta E = 0,2$ praktisch nicht mehr wahrnehmbar ist.

Ebenso ist über eine Dunkel- oder Neutralphase zwischen den Einzelfarben und eines mehrmaligen Versuchsdurchgangs der Testperson nachzudenken. Diese ermöglicht eine bessere Anpassung der Augen des Probanden an die nachfolgende Farbe.

Doch ist mit der verwendeten Meßreihe eine deutliche Tendenz festzumachen, die die erwarteten Ergebnisse bestätigte. Dies rechtfertigt den verwendeten Versuchsablauf.

So zeigen die aus dem Versuch gewonnenen Diagramme deutlich, daß die Wahrnehmungsschwelle für Farbrauschen bei der Luminanz und der Chrominanz fast gleich ist, die Schwelle beim Bunttonwinkel in jeder Farbe jedoch deutlich höher und auch von Farbton zu Farbton sehr unterschiedlich ist. Die Abbildungen 29 bis 31 unterstreichen, daß ein Farbrauschen im Bunttonwinkel vom Probanden erst um mehrere ΔE -Werte später festgestellt wird.

Aufgefallen ist auch, daß im Buntton Farbrauschen in den Haut- und Pastelltönen am frühesten erkannt wird, daher das Auge dort wohl am sensibelsten ist. Ähnliches gilt für Rot und Cyan, während bei Gelb und Blau die Wahrnehmung einer Verrauschung erst sehr spät erfolgt. Gerade die letzten beiden Farbtöne sowie auch das Grün werden jedoch von den drei Farbabstandsformeln – besonders von der ΔE_{00} -Formel - sehr unterschiedlich bewertet, was ihre Möglichkeiten zur Berechnung eines absoluten Schwellenwertes für diese Anwendung ebenfalls in Frage stellt. Aufgrund der hohen Abweichungen bei allen Formeln scheint die Berechnung eines Korrekturfaktors kaum möglich.

Für die Sättigung und die Helligkeit gilt ähnliches. Auch hier schwanken die Werte bei den unterschiedlichen Farben; allerdings auf sehr niedrigem Niveau. Besonders fallen in der ΔC -Sequenz die Peaks der beiden Grautöne und des Pastelltones auf. Scheinbar haben Farben, die eine niedrige Schwelle in der ΔC -Sequenz haben eine hohe Schwelle in der Δh -Sequenz. Die Kurven von ΔE_{00} und ΔE_{94} aus der ΔC -Sequenz verhalten sich also nahezu komplementär zu den entsprechenden Kurven des Bunttonwinkels. Aufgrund der gewählten Schrittweite von $\Delta E = 0,3$ läßt sich jedoch nicht eindeutig sagen, ob die Schwankungen in der ΔL - und ΔC -Sequenz tatsächlich statistisch signifikant sind.

Auffällig ist, daß in der ΔL Sequenz für die Farbabstandsformeln ΔE_{76} und ΔE_{94}

gleiche Werte ermittelt werden. Dies begründet sich dadurch, daß es sich in diesem Fall um einen reinen Abstand im Bereich der Helligkeit handelt. Da die Korrekturfaktoren k_L und S_L den Wert 1 haben und die weiteren Komponenten ΔC und ΔH Null sind, ergeben sich für beide Varianten der Farbabstandsformel die gleichen Werte.

So erweist sich auch das grundsätzliche Problem, daß der CIELAB-Farbraum für Aufsichtsvorlagen und eine mindestens 4° Probengröße [6, S. 113 & 115] entwickelt wurde, als gegeben. Zwar zeigte sich in einer 2001 von M.R. Luo durchgeführten Untersuchung, daß Monitorfarben eines CRT-Monitors mit Aufsichtsfarben vergleichbar und die Farbabstandsformeln damit grundsätzlich anwendbar sind [18], doch entspricht die „Probengröße“ beim Farbrauschen (ein Pixel) bei weitem nicht den ursprünglichen Vorgaben.

5 Ausblick

Um Farbrauschen und dessen physiologische Auswirkungen umfassend zu beschreiben, lassen die Ergebnisse dieser Diplomarbeit sicherlich keine abschließenden Schlußfolgerungen zu.

Nicht untersucht wurde hier das ebenfalls wichtige Wahrnehmen von Farbrauschen in realen Bildern: So z. B. ob sich dieses ändert, wenn Bilder viel Information oder viele monochrome, große Flächen enthalten.

Ebenfalls kann darüber nachgedacht werden, die Versuche auf weitere Farbräume auszudehnen. Beispielsweise könnte sich der S-CIELAB Farbraum eignen, da er nicht nur die Farbwerte, sondern auch den Bildinhalt bewertet. Die Fähigkeit Farben zu unterscheiden, sinkt mit steigender Ortsfrequenz, vor allem auf der Blau-Gelb Achse [20]. Dies zeigt auch die vorliegende Untersuchung.

Für die Entwicklung einer Formel, die Farbrauschen physiologisch richtig und unabhängig von der Farbe beschreibt, ist die Ermittlung von absoluten Schwellenwerten nötig. Dazu sind jedoch gegenüber dem hier verwendeten Versuchsaufbau und -Ablauf (zusätzlich zu den in Kapitel 4 erwähnten) weitere Modifikationen zu empfehlen:

Jeder Proband sollte den Versuch über einen längeren Zeitraum mehrmals ausführen. Die Resultate sollten gemittelt werden. Dadurch können Schwankungen durch Müdigkeit, Zeitdruck oder ähnlichem reduziert und so die Ergebnisse verbessert werden.

Außerdem sollte gewährleistet sein, daß sich der Proband nur „von unten“ an die Wahrnehmungsschwelle herantasten kann und nicht durch ein vorheriges Betrachten des maximalen Rauschens ein Nichterkennen als Schwelle angeben kann.

6 Anhang

6.1 Literaturverzeichnis

- [1] Fairchild, M. D.: Color Appearance Models; Addison Wesley Publishing Company 1997
- [2] Storm, R.: Wahrscheinlichkeitsrechnung, Mathematische Statistik, Statistische Qualitätskontrolle; Fachbuchverlag Leipzig 1995
- [3] Schläpfer, K.: Farbmeterik in der Reproduktionstechnik und im Mehrfarbendruck. UGRA, St. Gallen 1993
- [4] International Color Consortium: Specification ICC.1:2001-12, File Format for Color Profiles (Version 4.0.0); ICC 2001 (<http://www.color.org>)
- [5] Linotype-Hell AG: Die kreative Welt der digitalen Daten – Color Management. Linotype-Hell AG, Eschborn
- [6] Burns, R.S.: Billmeyer and Saltzman's Principles of Color Technology. John Wiley & Sons, New York 2000
- [7] Wyszecki, G, Stiles, W.S.: Color Science. Wiley Classics Library, New York 2000
- [8] Homann, J.P.: Digitales Colormanagement. 2. erweiterte Auflage. Springer Verlag, Berlin 2000.
- [9] Heidelberger Druckmaschinen AG: Farbe & Qualität. Heidelberg 1999
- [10] Brües, S.: Postscriptum Colormanagement. GretagMacbeth, 2000
- [11] Webers, J.: Handbuch der Film- und Videotechnik. Franzis Verlag, Feldkirchen 1998. S. 161
- [12] Heggie, D., Wardman, R.H., Luo, M.R.: A comparison of the colour differences computed using the CIE94, CMC(l:c) and BFD(l:c) formulae. Journal of the Society of Dyers and Colourists. Volume 112 (October 1996), 264 – 269
- [13] Luo, M.R., Cui, G., Rigg, B.: The development of the CIE 2000 colour-difference formula – CIEDE2000. COLOR research and application, Volume 26, Number 5, October 2001, 340 – 350
- [14] Tschernyschov, M., Freyer, R.: Einige Möglichkeiten zur Reduzierung des Rauschens von CCD-Bildsensoren. Bild und Ton, Band 39 (1986) Heft 11, 325 – 330
- [15] DIN Norm 5035-7: Beleuchtungsbedingungen bei Bildschirmarbeitsplätzen

-
- [16] DIN 6174: Farbmetrische Bestimmung von Farbabständen bei Körperfarben nach der CIELAB-Formel.
- [17] DIN 5033-3: Farbmessung - Farbmaßzahlen
- [18] Cui, G., Luo, M.R., Rigg, B., Li, W.: Colour-Difference Evaluation Using CRT Colours. Part I: Data Gathering and Testing Colour Difference Formulae. COLOR research and application, Volume 26, Number 5, October 2001, 394 – 402
- [19] McDonald, R., Smith, K.J.: CIE94 – a new colour-difference formula. Journal of the Society of Dyers and Colourists. Volume 111 (December 1995), 376 – 379
- [20] Internetseite: Introduction to S-CIELAB. Adresse:
<http://white.stanford.edu/~brian/scielab/introduction.html>. Abdruck der Seite befindet sich im Anhang 6.5.1
- [21] Luo, M.R., Rigg, B.: Chromaticity-discrimination ellipses for surface colours. COLOR research and application, 1986;11: 25 - 42

6.2 Eidesstattliche Erklärung

Ich versichere hiermit, die vorgelegte Arbeit in dem gemeldeten Zeitraum ohne fremde Hilfe verfasst und mich keiner anderen als der angegebenen Hilfsmittel und Quellen bedient zu haben.

Köln, den 23.07.2002

Jan Fischer

Ich versichere hiermit, die vorgelegte Arbeit in dem gemeldeten Zeitraum ohne fremde Hilfe verfasst und mich keiner anderen als der angegebenen Hilfsmittel und Quellen bedient zu haben.

Köln, den 23.07.2002

Michael Bantel

6.3 Sperrvermerk

Die vorgelegte Arbeit unterliegt keinem Sperrvermerk.

6.4 Weitergabeerklärung

Ich erkläre hiermit mein Einverständnis, daß das vorliegende Exemplar meiner Diplomarbeit oder eine Kopie hiervon für wissenschaftliche Zwecke verwendet werden darf.

Köln, den 23.07.2002

Michael Bantel

Ich erkläre hiermit mein Einverständnis, daß das vorliegende Exemplar meiner Diplomarbeit oder eine Kopie hiervon für wissenschaftliche Zwecke verwendet werden darf.

Köln, den 23.07.2002

Jan Fischer

6.5 Abdrucke elektronischer Quellen

6.5.1 Introduction to S-CIELAB

1. Measuring perceptual color difference

Like the CIELAB Delta E metric, the S-CIELAB metric is a "perceptual color fidelity" metric. It measures how accurate the reproduction of a color is to the original when viewed by a human observer. The CIELAB metric is suitable for measuring color difference of large uniform color targets. The S-CIELAB metric extends the CIELAB Delta E metric to color images.

Color discrimination is determined by a lot of factors, such as the cone sensitivity of the eye, the ambient illumination, the spatial pattern of the targets, etc. The effects of some of these factors on color sensitivity are better understood than others. To learn the basics (and advances) about color science and vision in general, check out the book [Foundations of Vision](#), written by Brian Wandell.

2. CIE L*a*b* Delta E metric

One of the most widely used perceptual color fidelity metric is the Delta E metric, given as part of the [CIE L*a*b*](#) standard color space specification. To measure perceptual difference between two lights using this metric, the spectral power distribution of the two lights are first converted to XYZ representations, which reflect (within a linear transformation) the spectral power sensitivities of the three cones on the human retina. Then, the XYZ values are transformed into an L*a*b* space, in which equal distance is supposed to correspond to equal perceptual difference (a "perceptually uniform" space). Then, the perceptual difference between the two targets can be calculated by taking the Euclidean distance of the two in this L*a*b* space. The difference is expressed in "Delta E" units. One Delta E unit represents approximately the threshold detection level of the color difference.

The CIE L*a*b* Delta E metric is intended to be used on large uniform color targets (at least 2° visual angle in size).

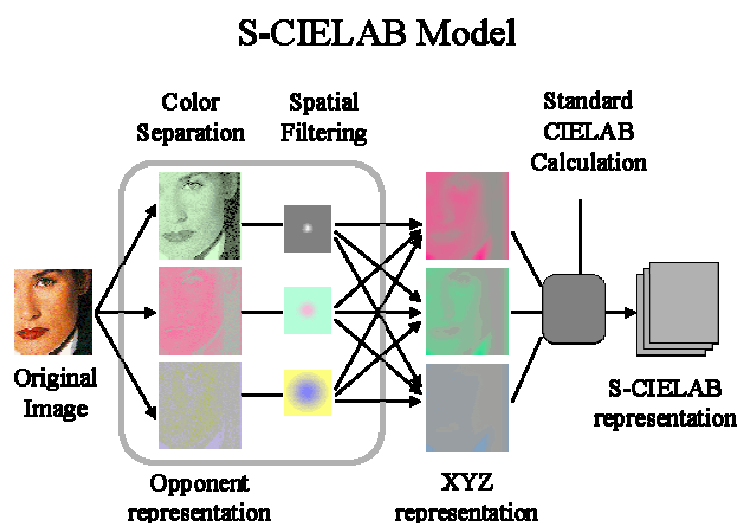
3. Color difference of patterned targets

Color discrimination and appearance is a function of spatial pattern. In general, as the spatial frequency of the target goes up (finer variations in space), color differences become harder to see, especially differences along the blue-yellow color direction. This result came from many studies, including Poirson & Wandell (1993, 1996), Kelly (1983), etc.

So, if we want to apply the CIE L*a*b* Delta E metric to color images, the spatial patterns of the image have to be taken into account. The goal of the S-CIELAB metric is to add a spatial pre-processing step to the standard CIELAB Delta E metric to account for the spatial-color sensitivity of the human eye.

4. Structure of S-CIELAB: A spatial extension to CIELAB

The S-CIELAB calculation is illustrated below. The key components are the color transformation and the spatial filtering steps before the standard CIELAB Delta E calculations. This is a "Pattern-Color Separable" architecture. For more information, please check out some related papers [here](#).



6.6 Messwerte

6.6.1 Ergebnisse der einzelnen Probanden

In den folgenden Tabellen sind die Ergebnisse der einzelnen Probanden aufgeführt. Die farbig unterlegten Werte wurden aus der Auswertung ausgeklammert.

Lfd. Nr.	Blau			Braun			Cyan		
	L*: 40,3	a*: 5,0	b*: -30,0	L*: 36,2	a*: 21,1	b*: 28,5	L*: 69,9	a*: -25,7	b*: -5,4
	ΔL	ΔC	Δh	ΔL	ΔC	Δh	ΔL	ΔC	Δh
1	0,67	1,00	10,00	1,00	1,00	5,33	1,00	1,33	10,00
2	0,67	0,67	4,00	1,00	1,00	6,67	0,67	1,00	6,33
3	0,67	0,67	7,00	0,67	0,67	5,33	1,33	1,00	9,33
4	1,00	0,67	4,67	0,67	0,67	3,67	0,67	0,67	3,67
5	0,33	0,33	2,67	0,33	0,33	3,00	0,67	0,33	4,67
6	0,67	1,00	4,67	0,67	0,67	8,00	1,00	1,33	9,67
7	0,67	0,67	2,67	0,67	0,33	1,67	0,67	1,00	3,67
8	0,67	0,67	1,00	0,67	0,67	5,00	0,67	0,67	4,33
9	0,67	0,67	5,00	0,67	0,67	5,33	1,00	0,67	2,67
10	0,33	0,33	4,00	0,67	0,33	4,33	0,33	0,33	4,00
11	0,33	0,33	3,00	0,67	0,67	4,33	0,67	0,67	4,67
12	0,67	0,67	1,00	0,67	0,33	3,67	0,67	0,67	5,00
13	0,67	0,67	5,33	0,33	0,33	2,33	0,67	0,33	4,67
14	1,33	0,67	6,33	0,67	0,33	5,67	1,00	0,67	5,33
15	0,67	0,33	3,67	0,33	0,33	4,00	0,67	0,33	7,67
16	1,00	0,67	2,67	1,00	0,67	2,67	1,33	1,33	3,33
17	0,67	0,67	7,00	0,33	0,33	3,33	0,33	0,67	5,33
18	0,67	0,67	6,33	0,67	0,67	5,00	1,00	0,67	9,67
19	0,67	0,67	6,67	0,67	0,67	6,67	1,00	1,00	10,00
20	1,00	1,00	7,33	0,67	0,67	5,00	0,67	1,00	5,00
21	0,67	0,67	3,00	0,67	0,33	4,67	0,67	0,67	3,00
22	1,33	1,33	9,00	1,33	0,67	5,33	1,67	1,00	8,67
23	1,00	1,00	7,00	1,00	1,33	4,67	1,33	1,33	5,00
24	0,67	0,67	5,00	0,67	0,67	4,00	1,00	1,00	10,00
25	1,00	1,00	10,00	1,00	0,67	9,00	1,33	1,00	10,00
26	1,00	0,33	7,33	0,67	0,33	2,67	0,33	0,33	5,33
27	1,33	1,00	6,00	1,00	1,00	3,33	1,00	1,00	4,67
28	1,00	1,00	6,00	1,00	0,67	5,33	0,67	0,67	6,33
29	0,67	0,67	5,67	0,67	0,33	3,67	0,67	0,67	1,33
30	0,67	0,67	7,00	1,00	0,67	7,67	0,67	1,00	7,33
31	0,33	0,33	5,33	0,33	0,33	3,00	0,67	0,67	4,33
32	0,67	0,67	7,67	0,67	0,67	5,67	0,67	0,67	8,33
Mittelwert	0,76	0,70	5,44	0,72	0,59	4,69	0,83	0,80	6,04
Varianz	0,07	0,06	5,26	0,06	0,06	2,81	0,10	0,09	6,46
Ber. Mittelwert	0,76	0,70	5,30	0,72	0,59	4,42	0,83	0,80	5,63
Ber. Varianz	0,07	0,06	2,65	0,06	0,06	1,39	0,10	0,09	4,26

Lfd. Nr.	Dunkelgrau			Dunkelviolett			Gelb		
	L*: 30,0	a*: 0,1	b*: 0,1	L*: 40,9	a*: 19,7	b*: -15,8	L*: 78,4	a*: -12,3	b*: 54,2
	ΔL	ΔC	Δh	ΔL	ΔC	Δh	ΔL	ΔC	Δh
1	0,67	1,00	8,33	0,67	0,67	10,00	0,67	1,33	9,67
2	1,00	1,00	5,33	0,67	1,00	7,00	1,33	1,33	9,00
3	0,00	0,67	8,00	0,67	0,67	10,00	0,67	0,67	10,00
4	0,67	0,67	4,33	0,67	0,67	4,33	0,67	0,67	4,67
5	0,67	0,67	4,00	0,67	0,67	6,67	1,00	0,67	8,67
6	0,67	1,00	4,33	0,67	0,67	7,00	0,67	0,67	10,00
7	0,67	0,67	2,67	0,67	0,67	3,67	0,67	1,00	3,00
8	0,67	0,67	3,67	0,67	0,33	6,67	0,67	0,67	8,67
9	0,67	0,67	4,67	0,67	0,67	7,33	0,67	0,67	7,00
10	0,33	0,33	5,00	0,33	0,33	10,00	0,33	0,33	9,67
11	0,67	0,67	3,33	0,67	0,67	5,67	0,67	0,67	8,33
12	0,67	0,67	3,67	0,67	0,67	10,00	0,67	0,67	10,00
13	0,67	0,67	4,00	0,33	0,33	5,33	0,67	0,33	6,00
14	0,67	1,00	5,33	0,67	0,67	7,33	0,67	0,33	5,67
15	0,33	0,67	1,67	0,33	0,33	8,00	0,33	0,33	6,33
16	0,67	1,00	4,33	0,67	1,33	4,00	0,67	0,67	5,00
17	0,33	0,33	4,00	0,67	0,33	4,67	0,33	0,67	2,67
18	0,67	0,67	6,33	0,67	0,33	5,67	0,67	0,67	10,00
19	0,67	0,67	6,33	0,67	0,67	10,00	1,00	1,00	10,00
20	0,67	0,67	6,33	0,67	0,67	10,00	0,67	0,67	9,33
21	0,67	0,67	3,00	0,67	0,33	4,33	0,67	0,67	5,00
22	1,00	1,00	6,00	1,00	0,67	10,00	1,33	1,00	10,00
23	1,00	1,00	6,33	0,67	0,67	3,67	1,00	0,67	10,00
24	1,00	1,00	7,00	1,00	0,67	8,33	1,00	1,00	8,67
25	1,00	1,00	8,00	0,67	0,67	9,00	0,67	0,67	9,33
26	0,33	0,67	5,00	0,33	0,33	10,00	0,33	0,33	10,00
27	1,00	0,67	5,33	0,67	0,67	4,67	0,67	0,67	9,67
28	0,67	0,67	5,33	0,67	0,67	8,67	0,67	1,00	10,00
29	0,67	0,67	4,00	0,67	0,67	6,00	0,67	0,67	9,33
30	0,67	0,67	4,67	0,67	0,67	6,67	1,00	1,00	10,00
31	0,67	0,67	3,67	0,33	0,67	6,67	0,67	0,67	10,00
32	0,67	0,67	5,67	0,67	0,67	9,00	0,67	0,67	10,00
Mittelwert	0,67	0,74	4,99	0,64	0,62	7,20	0,72	0,72	8,30
Varianz	0,05	0,03	2,46	0,02	0,04	4,79	0,06	0,07	5,06
Ber. Mittelwert	0,67	0,74	4,99	0,64	0,62	6,26	0,72	0,72	7,78
Ber. Varianz	0,05	0,03	2,46	0,02	0,04	2,82	0,06	0,07	3,39

Lfd. Nr.	Haut 1			Haut 2			Hellgrau		
	L*: 64,0	a*: 18,7	b*: 12,6	L*: 56,0	a*: 19,7	b*: 18,1	L*: 70,0	a*: 0,1	b*: 0,1
	ΔL	ΔC	Δh	ΔL	ΔC	Δh	ΔL	ΔC	Δh
1	1,00	1,33	9,67	0,67	1,33	6,33	1,33	1,00	6,00
2	0,67	1,00	5,00	1,00	0,67	4,00	1,00	0,67	4,33
3	0,67	0,67	5,67	0,67	0,67	5,33	1,00	0,67	7,33
4	0,67	0,67	4,00	0,67	0,67	4,67	0,67	0,67	4,67
5	0,67	0,67	2,67	0,67	0,67	3,67	0,67	0,67	3,67
6	0,67	0,67	5,67	0,33	0,33	4,67	0,67	0,67	4,67
7	0,67	0,67	5,33	0,67	0,67	3,67	0,67	0,67	4,33
8	0,67	0,67	4,67	0,67	0,67	3,33	0,67	0,67	4,00
9	0,33	0,67	4,00	0,67	0,67	3,33	0,67	1,00	5,67
10	0,33	0,33	2,00	0,33	0,33	4,00	0,33	0,33	2,33
11	0,67	0,67	3,33	0,67	0,67	3,33	0,67	0,67	3,67
12	0,67	0,67	3,67	0,67	0,67	4,00	0,67	0,67	6,00
13	0,33	0,67	4,33	0,33	0,67	4,33	0,67	0,67	3,33
14	0,67	0,33	7,00	0,33	0,67	2,67	0,33	0,33	5,00
15	0,33	0,33	1,33	0,33	0,33	2,67	0,00	0,33	3,00
16	0,67	0,67	5,00	1,00	1,00	6,33	0,67	1,33	5,67
17	0,33	0,33	3,00	0,33	0,33	2,67	0,33	0,33	1,67
18	0,67	0,67	5,67	0,33	0,67	4,00	0,67	0,67	5,00
19	0,67	0,67	5,67	0,67	0,67	5,33	0,67	0,67	6,67
20	0,33	0,67	4,00	0,33	0,33	4,67	0,33	0,33	3,00
21	0,67	0,67	3,00	0,67	0,33	3,00	0,67	0,67	4,33
22	1,00	0,67	7,00	1,00	1,00	5,33	1,33	1,33	5,67
23	1,00	1,00	3,00	0,67	0,67	4,00	1,00	1,00	3,67
24	1,00	0,67	5,00	0,67	0,67	5,67	0,67	1,00	8,00
25	1,00	0,67	7,67	0,67	0,67	10,00	1,00	0,67	8,00
26	0,33	0,33	4,33	0,33	0,67	3,00	0,33	0,33	4,00
27	0,67	0,67	2,33	0,67	0,67	3,00	0,67	0,33	1,33
28	0,67	0,67	4,33	0,67	0,67	4,33	0,67	0,67	5,00
29	0,67	0,33	4,33	0,33	0,33	3,00	0,67	0,67	2,67
30	0,67	0,67	4,33	0,67	0,67	3,33	0,67	0,67	4,67
31	0,67	0,33	3,33	0,67	0,33	3,33	0,33	0,67	6,33
32	0,67	0,33	5,33	0,67	0,33	5,67	0,67	0,67	4,67
Mittelwert	0,65	0,63	4,55	0,59	0,62	4,27	0,67	0,68	4,64
Varianz	0,04	0,05	2,98	0,04	0,05	2,22	0,08	0,07	2,71
Ber. Mittelwert	0,65	0,63	4,18	0,59	0,62	4,09	0,67	0,68	4,64
Ber. Varianz	0,04	0,05	1,17	0,04	0,05	1,16	0,08	0,07	2,71

Lfd. Nr.	Magenta			Mittelblau			Mittelgrün		
	ΔL	ΔC	Δh	ΔL	ΔC	Δh	ΔL	ΔC	Δh
	L*: 53,7 a*: 57,0 b*: -33,2			L*: 42,2 a*: 20,7 b*: -47,5			L*: 60,2 a*: -44,5 b*: 35,4		
1	0,67	1,00	8,00	1,33	1,00	5,00	1,00	0,67	7,67
2	1,00	1,00	4,33	0,67	0,67	4,67	0,67	1,00	8,00
3	0,67	0,67	7,33	0,67	0,67	7,67	1,00	0,67	8,33
4	0,67	0,67	5,33	0,67	0,67	4,67	0,67	0,67	6,00
5	0,67	0,67	1,67	1,00	0,67	2,67	0,67	0,67	4,33
6	0,67	0,67	6,00	0,67	0,67	8,33	1,00	0,67	10,00
7	0,67	0,67	2,00	0,67	0,67	3,33	0,67	0,67	7,67
8	0,67	0,67	5,33	0,67	0,67	5,00	0,33	0,67	2,67
9	0,67	0,67	4,67	0,67	0,67	5,67	0,33	0,67	10,00
10	0,33	0,33	4,67	0,33	0,33	4,00	0,33	0,33	8,33
11	0,67	0,67	2,33	0,67	0,67	3,33	1,00	0,67	1,33
12	0,67	0,67	5,67	0,67	0,67	6,67	0,67	0,67	10,00
13	0,67	0,67	4,67	0,67	0,67	5,00	0,67	0,67	7,33
14	0,67	0,67	5,00	0,67	0,67	6,00	0,33	0,67	7,00
15	0,33	0,33	4,00	0,67	0,33	0,67	0,33	0,33	3,67
16	1,00	0,67	5,33	1,00	1,00	5,67	1,33	1,33	5,67
17	0,67	0,67	3,00	0,33	0,33	3,00	0,33	0,33	3,33
18	0,67	0,67	5,00	0,67	0,67	6,67	0,67	0,67	9,33
19	1,00	1,00	6,33	1,00	1,00	7,33	1,00	1,00	10,00
20	0,33	0,67	2,33	0,67	0,33	4,00	0,67	0,33	2,00
21	0,67	0,33	3,00	0,67	0,67	3,00	1,00	0,67	4,33
22	1,33	1,33	7,33	2,67	1,00	7,67	1,33	1,33	10,00
23	1,00	1,00	4,67	0,67	0,67	4,67	0,67	0,67	5,00
24	1,00	0,67	7,00	1,00	0,67	6,00	0,67	0,67	10,00
25	0,67	1,00	7,33	0,67	0,67	9,00	0,67	0,67	10,00
26	0,33	0,33	7,33	0,33	0,33	8,00	0,33	0,33	10,00
27	0,67	0,33	2,00	0,67	0,67	5,00	0,67	1,00	2,33
28	0,67	0,67	5,67	0,67	0,67	7,33	0,67	0,67	10,00
29	0,33	0,67	1,00	0,67	0,67	4,00	0,33	0,33	0,33
30	0,67	0,33	3,00	0,67	0,33	5,00	0,67	0,67	10,00
31	0,33	0,67	4,67	0,67	0,67	6,33	0,67	0,67	10,00
32	0,67	0,67	8,00	0,67	0,67	6,67	0,67	0,67	8,33
Mittelwert	0,68	0,68	4,81	0,76	0,65	5,38	0,69	0,68	6,97
Varianz	0,05	0,05	3,87	0,16	0,04	3,57	0,08	0,06	9,66
Ber. Mittelwert	0,68	0,68	4,72	0,76	0,65	5,41	0,69	0,68	5,86
Ber. Varianz	0,05	0,05	3,03	0,16	0,04	2,60	0,08	0,06	5,51

Lfd. Nr.	Orange			Pastell			Pseudorot		
	L*: 66,0	a*: 13,6	b*: 46,7	L*: 69,4	a*: 17,5	b*: 8,3	L*: 50,7	a*: 50,0	b*: 30,7
	ΔL	ΔC	Δh	ΔL	ΔC	Δh	ΔL	ΔC	Δh
1	1,00	0,67	7,00	0,67	0,67	5,33	1,00	0,67	4,67
2	0,67	1,00	8,67	1,00	0,67	5,33	1,00	1,00	4,33
3	0,67	0,67	8,00	0,67	0,67	6,00	0,67	0,67	2,67
4	0,67	0,33	5,00	0,67	0,67	4,33	0,67	0,67	3,67
5	1,00	0,67	3,67	0,67	0,67	2,67	0,67	0,67	1,33
6	0,67	0,67	10,00	0,67	0,33	5,00	0,67	0,67	6,33
7	0,67	0,67	3,00	0,33	0,67	3,67	0,67	0,67	3,67
8	0,33	0,33	5,33	0,67	0,33	3,67	0,67	0,33	2,67
9	0,67	0,67	2,67	0,67	0,67	1,00	0,67	0,67	4,67
10	0,33	0,33	6,00	0,33	0,33	3,00	0,67	0,33	4,00
11	0,67	0,67	5,67	0,67	0,67	3,00	0,67	0,67	3,33
12	0,67	0,67	7,33	0,67	0,67	3,67	0,67	0,67	5,67
13	0,67	0,33	6,00	0,33	0,67	4,33	0,67	0,67	2,33
14	0,33	0,67	6,33	0,33	0,67	4,00	0,33	0,33	3,33
15	0,33	0,33	2,67	0,33	0,33	5,00	0,33	0,33	4,67
16	0,67	0,67	5,67	1,00	1,00	4,67	1,00	1,00	4,33
17	0,33	0,33	3,33	0,67	0,67	2,67	0,67	0,33	2,00
18	0,67	0,67	10,00	0,67	0,67	5,67	0,67	0,67	4,00
19	0,67	1,00	10,00	1,00	1,00	6,33	1,00	1,00	5,00
20	0,33	0,33	2,00	0,33	0,33	2,67	0,67	0,33	2,67
21	0,67	0,33	3,33	0,67	0,33	3,67	1,00	0,67	3,33
22	1,00	1,00	9,67	1,33	1,00	6,33	1,00	1,33	5,33
23	0,67	0,67	4,67	0,67	0,67	2,67	0,67	0,67	4,33
24	0,67	0,67	6,00	0,67	0,67	5,33	0,67	0,67	5,33
25	0,67	1,00	9,00	1,00	1,00	9,00	1,00	1,00	5,33
26	0,33	0,33	7,67	0,33	0,33	1,33	0,33	0,33	2,00
27	0,67	0,67	5,67	0,67	0,67	3,33	1,00	0,67	2,33
28	0,67	0,67	7,67	0,67	0,67	4,33	0,67	0,67	4,00
29	0,33	0,33	0,67	0,33	0,33	1,00	0,33	0,33	1,67
30	0,67	0,67	9,00	0,67	0,67	5,33	0,67	0,33	2,33
31	0,33	0,33	8,33	0,67	0,33	3,33	0,33	0,33	2,33
32	0,33	0,67	8,00	0,33	0,67	4,00	0,67	0,67	4,67
Mittelwert	0,59	0,59	6,19	0,64	0,62	4,11	0,70	0,63	3,70
Varianz	0,04	0,05	6,65	0,06	0,04	2,83	0,05	0,06	1,69
Ber. Mittelwert	0,59	0,59	5,98	0,64	0,62	4,26	0,70	0,63	3,69
Ber. Varianz	0,04	0,05	4,85	0,06	0,04	1,33	0,05	0,06	1,37

6.6.2 Kontrollmessungen der Testfarben

Testweise wurden zwei Testfarben am Monitor mit dem x-Rite DTP 92 und dem *Measure Tool* von *Profilemaker 3* ausgemessen.

Farbe	Ist			Soll			ΔE
	L*	a*	b*	L*	a*	b*	
Gelb	77,6	-15,2	57,0	78,4	-12,3	54,2	4,1
+ ΔL	87,7	-14,0	56,3	88,4	-12,3	54,2	2,8
- ΔL	67,2	-16,2	55,8	68,4	-12,3	54,2	4,4
+ ΔC	77,3	-17,4	65,7	78,4	-14,5	64,0	3,5
- ΔC	77,5	-13,5	46,6	78,4	-10,1	44,4	4,1
+ Δh	77,7	-25,4	53,8	78,4	-21,5	51,2	4,7
- Δh	77,5	-5,4	58,1	78,4	-2,7	55,5	3,8
Magenta	52,4	59,3	-34,9	53,7	57,0	-33,2	3,1
+ ΔL	62,7	54,3	-32,5	63,7	57,0	-33,2	3,0
- ΔL	42,4	58,4	-34,5	43,7	57,0	-33,2	2,3
+ ΔC	52,4	68,2	-39,6	53,7	65,6	-38,2	3,2
- ΔC	52,4	50,8	-29,9	53,7	48,4	-28,2	3,3
+ Δh	52,6	59,6	-23,8	53,7	61,9	-22,8	2,7
- Δh	52,4	50,4	-42,3	53,7	50,4	-42,6	1,3

Aufgrund der geringen Farbdifferenzen wurden im weiteren die errechneten Farbwerte verwendet.

6.7 Die Menues von Shiraz

Nicht alle Menüpunkte sind aktive Befehle. Einige wurden nur der Vollständigkeit halber mit in die einzelnen Menues übernommen, haben aber keine Funktion und lassen sich auch nicht anwählen.

6.7.1 Ablage

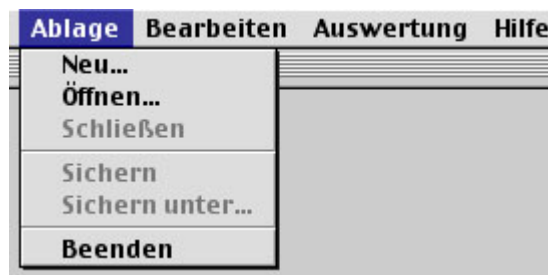


Abbildung 32 - Menue Ablage

Neu...

Hiermit wird das Fenster zu Auswahl einer Farbe geöffnet.

Öffnen...

Bereits erstellte und gespeicherte Sequenzen können mit diesem Menüpunkt nochmals geöffnet werden.

Beenden

Beendet das Programm. Nicht gesicherte Werte gehen verloren.

6.7.2 Bearbeiten



Abbildung 33 - Menue Bearbeiten

Testfeld...

Öffnet ein Fenster mit einem weißen Feld. Dient zum Ausmessen des Weißpunktes.

Voreinstellungen...

Öffnet das Fenster für die Voreinstellungen. Hier können die Bildschirmgröße, das zu verwendende ICC-Profil und der Ordner mit den gespeicherten Bildern gewählt werden.

6.7.3 Auswertung

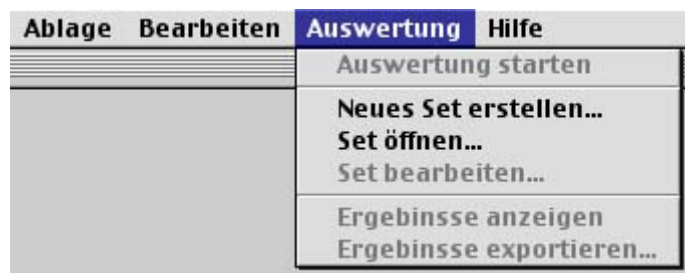


Abbildung 34 - Menue Auswertung

Auswertung starten...

Startet eine neue Auswertung mit dem ausgewählten Set. Wird erst aktiviert, wenn

ein Set geöffnet oder neu erstellt wurde.

Set erstellen...

Erstellt ein neues Set für die Auswertung. Zuerst wird nach dem Namen und dem Speicherort gefragt.

Als Standardordner wird der Ordner Sets innerhalb des Arbeitsordners geöffnet. Die Dateiendung `.set` wird automatisch erzeugt.

Anschließend können die ersten Bilder, die in das Set sollen, ausgewählt werden. Das Programm unterstützt den sogenannten Multiselect-Modus: mit Hilfe der Steuerungstaste (strg oder ctrl) können mehrere Dateien gleichzeitig ausgewählt werden.

Es erfolgt keine Kontrolle, ob die ausgewählten Dateien tatsächlich vom richtigen Typ sind!

Set öffnen...

Öffnet ein bereits erstelltes Set.

Set bearbeiten...

Ermöglicht Bilder zum Set hinzuzufügen oder herauszunehmen. Werden Bilder entfernt, so gehen auch die bisherigen Messergebnisse verloren.

Ergebnisse anzeigen

Zeigt die Messergebnisse des aktuellen Sets an.

Ergebnisse exportieren...

Exportiert die Messergebnisse des aktuellen Sets im `.txt`-Format, z.B. zur Nachbearbeitung in einem anderen Programm. Die einzelnen Datenfelder werden dabei mit Tabulatoren getrennt.

6.8 Die Fenster von Shiraz

Im folgenden werden die Funktionen und Möglichkeiten in den einzelnen Fenstern des Programms erläutert.

6.8.1 Farbwahl

Die Farbwahl kann entweder im RGB oder im CIELAB Modus erfolgen. Die Wahl des Farbraums erfolgt über die Radiobuttons. Die Wahl der einzelnen Farbwerte kann entweder mit Hilfe der Schieberegler oder numerisch geschehen. Bei der Eingabe von Werten im CIELAB Raum ist darauf zu achten, daß Dezimalwerte mit einem Punkt, nicht mit einem Komma eingegeben werden. Jede numerische Eingabe muß mit der Return-Taste bestätigt werden.

Die Farbwerte werden unter Einbeziehung des gewählten ICC-Profiles in den jeweils anderen Farbraum umgerechnet. Dabei erfolgt kein Gammutmapping. Werden also Werte im CIELAB Farbraum gewählt, die nicht direkt in den RGB Farbraum umgerechnet werden können, so erfolgt eine Warnung. Das Erstellen einer Sequenz mit solchen Werten ist nicht möglich. Des weiteren wird bei der Einstellung der Farbe überprüft, ob auch die 2,2-fache Maximalabweichung der gewünschten Farbe noch als RGB dargestellt werden kann (siehe Kapitel 2.1.3).

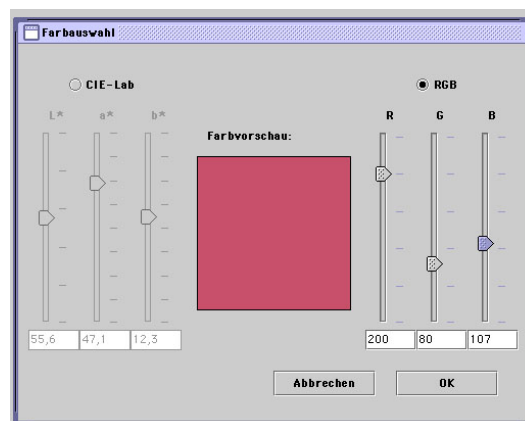


Abbildung 35 - Fenster Farbauswahl

Durch klicken auf den *OK* wird die Sequenz erstellt.

Durch klicken auf *Abbrechen* werden die Einstellungen verworfen.

6.8.2 Sequenz

Eine fertige Sequenz kann in diesem Fenster betrachtet werden. Es öffnet sich nachdem eine Sequenz erzeugt wurde oder wenn eine vorhandene Sequenz mit *Öffnen...* geöffnet wurde.

Mit den ersten drei Radio-Buttons kann die Richtung der Verrauschung gewählt werden und mit dem Schieberegler kann die Sequenz von der Verrauschung $\Delta E=0$ bis zur maximalen Stufe betrachtet werden. Die maximalen Stufe kann zusammen mit den Farbwerten der Grundfarbe am rechten Rand des Fensters abgelesen werden.

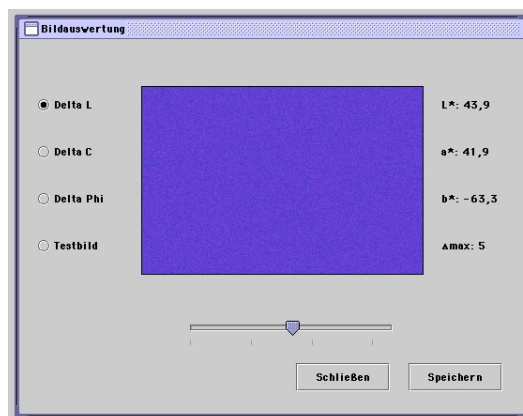


Abbildung 36 - Fenster Bildsequenz

Der Radiobutton *Testbild* zeigt eine Sequenz, bei der das erste Bild in der oberen Hälfte die Grundfarbe und in der Unteren die maximalen Werte der Verrauschung in folgender Reihenfolge von links nach rechts: $+\Delta L$, $-\Delta L$, $+\Delta C$, $-\Delta C$, $+\Delta h$, $-\Delta h$. Die weiteren Bilder enthalten die maximalen Farbabweichungen nochmals als Vollbild, was die Messung mit dem X-rite Monitor Optimizer aufgrund der Größe des Farbfeldes erleichtert. Mit Hilfe dieses Testbildes können die Farbwerte ausgemessen und überprüft werden.

Der Schalter *Speichern* gibt die Möglichkeit die Sequenz zu speichern. Als Standardordner wird das in den Voreinstellungen angegebene Arbeitsverzeichnis geöffnet. Die Dateiendung *.mjs* wird vom Programm immer automatisch angefügt - auch wenn man sie manuell an den Dateinamen anfügt wird sie nochmals angefügt.

6.8.3 Voreinstellungen

Das Fenster für die Voreinstellungen besitzt drei Tabulatoren. Einen für die Bildschirmgröße, einen für das ICC-Profil und einen für den Arbeitspfad.

Für die Bildschirmgröße stehen fünf gebräuchliche Werte zur Auswahl. Die Bildschirmgröße steuert einerseits die Größe des Hauptfensters, wird aber auch zur Positionierung der anderen Fenster benötigt. Standardwert ist 832 x 624 Pixel.



Abbildung 37 - Einstellung Bildgröße

Bei der Wahl des ICC-Profiles wird überprüft, ob es sich bei der gewählten Datei tatsächlich um ein ICC-konformes RGB Profil handelt. Falls nicht, so wird keine Änderung vorgenommen. Standard ist Javas sRGB Profil (wird automatisch mit Java installiert).



Abbildung 38 - Einstellung ICC Profil

Das Arbeitsverzeichnis bezeichnet den Ordner, in dem die Bilddateien zu finden

sind. Durch die Wahl des entsprechenden Ordners wird gleichzeitig ein Unterordner mit dem Namen *Sets* erstellt, in dem die einzelnen Datensets für die Auswertungen gespeichert werden können. Des Weiteren wird ein Ordner mit Namen *Ergebnisse* erstellt, in den Ergebnisse exportiert werden können.

Mit dem Button *Anzeigen* wird der vollständige Pfad angezeigt. Mit *Ändern* kann ein anderer Ordner gewählt werden.



Abbildung 39 - Einstellung Bildordner

Nachdem ein Pfad geändert wurde, wird er noch nicht korrekt angezeigt. Erst wenn das Voreinstellungsfenster mit *OK* geschlossen und anschließend wieder geöffnet wird, kann der Pfad richtig angezeigt werden!

Ist noch keine Datei angelegt, so wird eine Voreinstellungsdatei mit den Standardwerten und dem Namen *Shiraz Prefs.dat* angelegt. Sie liegt im selben Verzeichnis wie die Programmdatei selbst. Wird sie verschoben oder gelöscht, so wird beim nächsten Start wieder eine Neue angelegt.

6.8.4 Set bearbeiten

In diesem Fenster werden alle Bildsequenzdateien angezeigt, die in dem aktuellen Set enthalten sind.

Mit *Hinzufügen* können weitere Dateien zu dem Set hinzugefügt werden. Das Programm unterstützt den sogenannten Multiselect-Modus: mit Hilfe der Steuerungstaste (strg oder ctrl) können mehrere Dateien gleichzeitig ausgewählt werden. Es erfolgt keine Kontrolle, ob die ausgewählten Dateien tatsächlich vom rich-

tigen Typ sind!



Abbildung 40 - Set bearbeiten

Mit den Checkboxes können einzelne Dateien ausgewählt werden. *Entfernen* löscht die ausgewählten Dateien aus dem Set. Durch das Entfernen von Dateien aus einem Set gehen auch die bisherigen Messergebnisse dieser Dateien verloren! Es erfolgt keine weitere Warnung oder Abfrage. Dieser Schritt ist auch nicht wieder rückgängig zu machen!

Mit *Fertig* wird das Fenster wieder geschlossen. Es gibt keine Möglichkeit die bisherigen Änderungen zu verwerfen.

6.8.5 Auswertung

In diesem Fenster findet die eigentliche Auswertung statt. Die Sequenzen werden in der Reihenfolge in der sie im Set gespeichert sind aufgerufen. Innerhalb einer Sequenz gilt die Reihenfolge $\Delta L - \Delta C - \Delta H$.

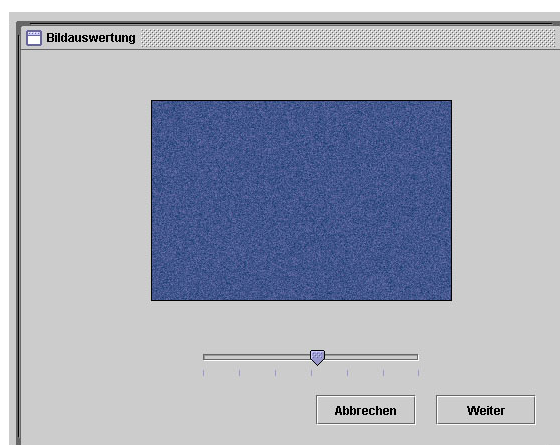


Abbildung 41 - Bildauswertung

Mit dem Schieberegler wird vom Probanden der Schwellenwert eingestellt, ab dem er das Farbfeld als verrauscht empfindet. Durch klicken auf *Weiter* wird dieser Wert innerhalb des Sets gespeichert und das nächste Bild wird für die Auswertung geöffnet. Nach der letzten Sequenz erfolgt eine kurze Meldung, daß die Auswertung beendet ist.

Wird die Auswertung durch *Abbrechen* vorzeitig beendet, so werden die bisherigen Werte gespeichert.

6.9 Der Quellcode von SHIRAZ

Der komplette Quellcode ist zusammen mit den entsprechenden CodeWarrior Projektdateien auf der beiliegenden CD zu finden. Da die meisten Klassen nur für die Steuerung des Programms und den Aufbau der Fenster zuständig sind, werden hier nur die Klassen aufgeführt, die direkt mit der Farbtransformation, der Verrauschung der Farben und der Erzeugung der Bilder zu tun haben.

Daher werden im folgenden Quellcode zum Teil Funktionen und Klassen aufgerufen, die hier nicht abgedruckt sind. Sie befinden sich aber auf der CD-ROM.

6.9.1 Klasse Farbmatrik.ICC_RGBmetrik

```
/*
  ICC_RGBmetrik.java
  Title:      Shiraz
  Author:    Jan Fischer
  Description: Programm zur definierten Verrauschung von CIE-Lab Farben
              im RGB Farbraum unter Einbeziehung eines CMM.
*/

package Farbmatrik;

import java.awt.color.*;
import javax.swing.*;
import java.io.*;
import javax.vecmath.*;
import com.jan.mathe.Zahlen;
import com.jan.color.*;
```

```

        mat.getRow( 1, invMat[ 1 ] ); mat.getRow( 2, invMat[ 2 ] );
    }
    catch ( SingularMatrixException sme ) { JOptionPane.showMessageDialog( null,
        "Die Matrix des Profils lößt sich nicht invertieren!",
        "FEHLER !!!", JOptionPane.ERROR_MESSAGE );
    }
// Gamma-Korrektur entweder über TRCs und inverse TRCs
    try {
        rotTRC = dasProfil.getRotTRC();
        gruenTRC = dasProfil.getGruenTRC();
        blauTRC = dasProfil.getBlauTRC();
// Inverse TRCs erstellen
        rotInversTRC = new short[ rotTRC.length ];
        gruenInversTRC = new short[ gruenTRC.length ];
        blauInversTRC = new short[ blauTRC.length ];
// Falls die Arrays gleich lang sind (was eigentlich der Fall sein sollte)
        if( rotTRC.length == gruenTRC.length && rotTRC.length == blauTRC.length ) {
            int r = 1;
            int g = 1;
            int b = 1;
            float rotOben = 0, rotUnten = 0, gruenOben = 0, gruenUnten = 0,
                blauOben = 0, blauUnten = 0;
            float faktor = ( rotTRC.length - 1 ) / 65535f;
            for( int i = 0; i < rotTRC.length; i++ ) {
                boolean rotOK = false, gruenOK = false, blauOK = false;
                while( !( rotOK && gruenOK && blauOK ) ) {
                    if( i <= Zahlen.sigToUnsig( rotTRC[ r ] ) * faktor ) {
                        rotOben = Zahlen.sigToUnsig( rotTRC[ r ] ) * faktor;
                        rotUnten = Zahlen.sigToUnsig( rotTRC[ r - 1 ] ) * faktor;
                        rotOK = true;
                    }
                    else
                        r++;

                    if( i <= Zahlen.sigToUnsig( gruenTRC[ g ] ) * faktor ) {
                        gruenOben = Zahlen.sigToUnsig( gruenTRC[ g ] ) * faktor;
                        gruenUnten = Zahlen.sigToUnsig( gruenTRC[ g - 1 ] ) * faktor;
                        gruenOK = true;
                    }
                    else
                        g++;

                    if( i <= Zahlen.sigToUnsig( blauTRC[ b ] ) * faktor ) {
                        blauOben = Zahlen.sigToUnsig( blauTRC[ b ] ) * faktor;
                        blauUnten = Zahlen.sigToUnsig( blauTRC[ b - 1 ] ) * faktor;
                        blauOK = true;
                    }
                    else
                        b++;
                }
            }
// Zwischenwerte werden interpoliert (Zweipunkt Geradenform)
            rotInversTRC[ i ] = ( short ) ( ( ( i - rotUnten ) / ( rotOben - rotUnten )
                + r - 1 ) / faktor );
            gruenInversTRC[ i ] = ( short ) ( ( ( i - gruenUnten ) / ( gruenOben - gruenUnten )
                + g - 1 ) / faktor );
            blauInversTRC[ i ] = ( short ) ( ( ( i - blauUnten ) / ( blauOben - blauUnten )
                + b - 1 ) / faktor );
            r -= 1;

```

```

        g -= 1;
        b -= 1;
    }
    hatTRC = true;
}
// Falls nicht, muss jede inverse TRC gesondert erstellt werden
else {
    int j = 1;
    float rotOben = 0, rotUnten = 0,
        gruenOben = 0, gruenUnten = 0, blauOben = 0, blauUnten = 0;
    float faktor = ( rotTRC.length - 1 ) / 65535f;
    // Rote inverse TRC
    for( int i = 0; i < rotTRC.length; i++ ) {
        boolean rotOK = false;
        while( !rotOK ) {
            if( i <= (float) ( Zahlen.sigToUnsig( rotTRC[ j ] ) * faktor ) ) {
                rotOben = Zahlen.sigToUnsig( rotTRC[ j ] ) * faktor;
                rotUnten = Zahlen.sigToUnsig( rotTRC[ j - 1 ] ) * faktor;
                rotOK = true;
            }
            j++;
        }
        rotInversTRC[ i ] = ( short ) ( ( ( i - rotUnten ) / ( rotOben - rotUnten )
                                        + j - 1 ) / faktor );

        j -= 1;
    }
    // Grüne inverse TRC
    j = 1;
    faktor = ( gruenTRC.length - 1 ) / 65535f;
    for( int i = 0; i < gruenTRC.length; i++ ) {
        boolean gruenOK = false;
        while( !gruenOK ) {
            if( i <= (float) ( Zahlen.sigToUnsig( gruenTRC[ j ] ) * faktor ) ) {
                gruenOben = Zahlen.sigToUnsig( gruenTRC[ j ] ) * faktor;
                gruenUnten = Zahlen.sigToUnsig( gruenTRC[ j - 1 ] ) * faktor;
                gruenOK = true;
            }
            j++;
        }
        gruenInversTRC[ i ] = ( short ) ( ( ( i - gruenUnten ) / ( gruenOben - gruenUnten )
                                        + j - 1 ) / faktor );

        j -= 1;
    }
    // Blaue inverse TRC
    j = 1;
    faktor = ( blauTRC.length - 1 ) / 65535f;
    for( int i = 0; i < blauTRC.length; i++ ) { boolean blauOK = false;
        while( !blauOK ) {
            if( i <= (float) ( Zahlen.sigToUnsig( blauTRC[ j ] ) * faktor ) ) {
                blauOben = Zahlen.sigToUnsig( blauTRC[ j ] ) * faktor;
                blauUnten = Zahlen.sigToUnsig( blauTRC[ j - 1 ] ) * faktor;
                blauOK = true;
            }
            j++;
        }
        blauInversTRC[ i ] = ( short ) ( ( ( i - blauUnten ) / ( blauOben - blauUnten )
                                        + j - 1 ) / faktor );

        j -= 1;
    }
}

```

```

        }
        hatTRC = true;
    }
}
// Wenn keine TRC im Profil ist, Gamma-Wert abfragen
catch( NoSuchTagException nste ) {
    try {
        rotGamma = dasProfil.getRotGamma();
        gruenGamma = dasProfil.getGruenGamma();
        blauGamma = dasProfil.getBlauGamma(); hatTRC = false;
    }
    catch( NoSuchTagException exc ) {
    }
}
}
}
catch( NoProfileException npe ) {
    JOptionPane.showMessageDialog( null, "Die Datei ist kein ICC-RGB Profil!",
        "FEHLER !!!", JOptionPane.ERROR_MESSAGE );
    npe.printStackTrace();
}
catch( NoSuchTagException exc ) {
    JOptionPane.showMessageDialog( null, exc.getMessage(),
        "FEHLER !!!", JOptionPane.ERROR_MESSAGE ); exc.printStackTrace();
}
catch( IOException ioe ) {
    JOptionPane.showMessageDialog( null, "Fehler beim ...ffnen des Profils",
        "FEHLER !!!", JOptionPane.ERROR_MESSAGE ); ioe.printStackTrace();
}
}
}

/**
 * Nimmt die RGB-Werte zur Weiterverarbeitung an.
 * @param r Rot
 * @param g Grün
 * @param b Blau
 */

public void setRGB( int r, int g, int b ) {
    rot = (short) r;
    gruen = (short) g;
    blau = (short) b;
}

/**
 * Nimmt die RGB-Werte zur Weiterverarbeitung an. Dabei werden die übergebenen
 * Werte getestet, ob sie innerhalb der mit <code>setMinMax()</code> gesetzten Grenzen
 * liegen. Wurden keine expliziten Grenzen gesetzt, dann gilt der Wertebereich
 * von 0 bis 255.
 * @param rgb int-Array mit den RGB-Werten in Reihenfolge R - G - B
 * @return <code>true</code>, wenn die Werte angenommen wurden
 */

public void setRGB( int[] rgb ) {
    if( rgb.length < 3 ) {
        JOptionPane.showMessageDialog( null, "Ungültige RGB-Werte! Array zu kurz.",
            "FEHLER !!!", JOptionPane.ERROR_MESSAGE );
    }
}

```

```

        return;
    }
    setRGB( rgb[ 0 ], rgb[ 1 ], rgb[ 2 ] );
}

/**
 * Zur Abfrage der aktuellen RGB-Werte.
 * @return rgb <code>short</code>-Array aus 3 Werten mit R - G - B
 * @see #getLegal
 */

public short[] getRGB() {
    short[] rgb = new short[ 3 ];
    rgb[ 0 ] = rot;
    rgb[ 1 ] = gruen;
    rgb[ 2 ] = blau;
    return rgb;
}

/**
 * Rechnet von CIE-XYZ nach RGB mit den übergebenen Werten. Die neuen RGB-Werte
 * können dann mit <code>getRGB()</code> abgerufen werden.
 * <p>
 * @param xyz <code>double</code>Array mit den X, Y und Z Anteilen der Farbe
 * @return rgb <code>short</code>Array mit den RGB Werten.
 * Reihenfolge R - G - B
 */

public short[] xyzNachRGB( double[] xyz ) {
    return xyzNachRGB( xyz[ 0 ], xyz[ 1 ], xyz[ 2 ] );
}

/**
 * Rechnet von CIE-XYZ nach RGB mit den übergebenen Werten. Die neuen RGB-Werte
 * können dann mit <code>getRGB()</code> abgerufen werden.
 * @param x X-Anteil der Farbe
 * @param y Y-Anteil der Farbe
 * @param z Z-Anteil der Farbe
 * @return rgb <code>short</code>-Array mit den RGB Werten.
 * Reihenfolge R - G - B
 */

public short[] xyzNachRGB( double x, double y, double z ) {
    setXYZ( x, y, z );
    xyzNachRGB();
    return getRGB();
}

/**
 * Rechnet von CIE-XYZ nach RGB mit den internen Werten. Die neuen RGB-Werte
 * können dann mit <code>getRGB()</code> abgerufen werden.
 * <p>
 * Ungültige RGB-Werte werden <b>nicht</b> angepasst! Eine Kontrolle der Werte
 * muss also noch erfolgen. Ob Werte außerhalb des Wertebereiches liegen kann

```

```

* mit <code>getLegal()</code> abgefragt werden.
*/

public void xyzNachRGB() {
    double[] xyzGet = getXYZ();
    float x, y, z, linRot, linGruen, linBlau;
    // XYZ normalisieren
    x = (float) xyzGet[ 0 ] / 100f;
    y = (float) xyzGet[ 1 ] / 100f;
    z = (float) xyzGet[ 2 ] / 100f;
    // Via Matrix die XYZ Werte errechnen
    linRot = ( invMat[0][0] * x + invMat[0][1] * y + invMat[0][2] * z );
    linGruen = ( invMat[1][0] * x + invMat[1][1] * y + invMat[1][2] * z );
    linBlau = ( invMat[2][0] * x + invMat[2][1] * y + invMat[2][2] * z );
    // Test auf gültige RGB-Werte
    legal = true;    // Annahme, die Farbe ist ok!
    if( linRot > ( 1f ) ) {
        linRot = ( 1 );
        legal = false;
    }
    if( linRot < ( 0 ) ) {
        linRot = ( 0 );
        legal = false;
    }
    if( linGruen > ( 1 ) ) {
        linGruen = ( 1 );
        legal = false;
    }
    if( linGruen < ( 0 ) ) {
        linGruen = ( 0 );
        legal = false;
    }
    if( linBlau > ( 1 ) ) {
        linBlau = ( 1 );
        legal = false;
    }
    if( linBlau < ( 0 ) ) {
        linBlau = ( 0 );
        legal = false;
    }
    // Inverse Gamma-Korrektur
    if( hatTRC ) {
        float faktor = ( float ) ( ( 255f + 0.5 ) / 65535f );
        int rotTemp, gruenTemp, blauTemp;
        rotTemp = ( int ) ( linRot * ( rotTRC.length - 1 ) );
        gruenTemp = ( int ) ( linGruen * ( gruenTRC.length - 1 ) );
        blauTemp = ( int ) ( linBlau * ( blauTRC.length - 1 ) );
        rot = ( short ) ( Zahlen.sigToUnsig( rotInversTRC[ rotTemp ] ) * faktor );
        gruen = ( short ) ( Zahlen.sigToUnsig( gruenInversTRC[ gruenTemp ] ) * faktor );
        blau = ( short ) ( Zahlen.sigToUnsig( blauInversTRC[ blauTemp ] ) * faktor );
    }
    else {
        rot = ( short ) ( Math.pow( linRot, 1/rotGamma ) * 255f + 0.5 );
        gruen = ( short ) ( Math.pow( linGruen, 1/gruenGamma ) * 255f + 0.5 );
        blau = ( short ) ( Math.pow( linBlau, 1/blauGamma ) * 255f + 0.5 );
    }
}

```

```

/**
 * Zum Testen, ob auch die verrauschten Farbwerte innerhalb des RGB-Farbraums sind.
 * Die Funktion wird nur aufgerufen, wenn die aktuellen RGB-Werte ansonsten darstellbar
 * sind. Andrenfalls ist eine Kontrolle, ob auch die Delta L, C und Phi-Werte innerhalb
 * von RGB liegen, unnötig.
 */

private void testLegal() {
    short[] rgbTemp;
    double[] labTemp, labNeu; float testwert = 2.2f;
// Aktuelle Werte sichern
    rgbTemp = getRGB();
    labTemp = getLAB();
    labNeu = getLAB();
//// Häufig benötigte Werte ausrechnen ////
    double c, c1;
    double phi, phi1, phiGrad;
    int rgbDeltaLPlus, rgbDeltaLMinus, rgbDeltaCPlus,
        rgbDeltaCMinus, rgbDeltaPhiPlus, rgbDeltaPhiMinus;
// C* ausrechnen
    c = Math.sqrt( labTemp[ 1 ] * labTemp[ 1 ] + labTemp[ 2 ] * labTemp[ 2 ] );
// Phi ausrechnen (Phi in Rad!)
    phi = Math.atan( labTemp[ 2 ] / labTemp[ 1 ] );
// Phi dem richtigen Quadranten zuweisen!
    if( labTemp[ 1 ] < 0 ) {
        phi += Math.PI;
    }
    if( labTemp[ 2 ] < 0 && labTemp[ 1 ] > 0 ) {
        phi += 2.0 * Math.PI;
    }
}
// Phi in Grad umwandeln!
    phiGrad = phi * 180.0 / Math.PI;
// Sinus und Cosinus von phi
    double sinus, cosinus; sinus = Math.sin( phi ); cosinus = Math.cos( phi );
// Maximalwerte von Delta L für das Testbild
    labNeu[ 0 ] = labTemp[ 0 ] + testwert * Sequenz.Bilder.MAX_DELTA;
    rgbDeltaLPlus = labNachRGB( labNeu );
    if( legal ) {
        labNeu[ 0 ] = labTemp[ 0 ] - testwert * Sequenz.Bilder.MAX_DELTA;
        rgbDeltaLMinus = labNachRGB( labNeu );
    }
    labNeu[ 0 ] = labTemp[ 0 ];
// Maximalwerte von Delta C für das Testbild
    if( legal ) {
        c1 = c + testwert * Sequenz.Bilder.MAX_DELTA;
        labNeu[ 1 ] = c1 * cosinus;
        labNeu[ 2 ] = c1 * sinus; rgbDeltaCPlus = labNachRGB( labNeu );
    }
    if( legal ) {
        c1 = c - testwert * Sequenz.Bilder.MAX_DELTA; labNeu[ 1 ] = c1 * cosinus;
        labNeu[ 2 ] = c1 * sinus;
        rgbDeltaCMinus = labNachRGB( labNeu );
    }
}
// Maximalwerte von Delta Phi für das Testbild
    if( legal ) {
        phi1 = phiGrad + testwert * Sequenz.Bilder.MAX_DELTA;
        phi1 = phi1 * Math.PI / 180.0;

```

```

        labNeu[ 1 ] = c * Math.cos( phi1 );
        labNeu[ 2 ] = c * Math.sin( phi1 );
        rgbDeltaPhiPlus = labNachRGB( labNeu );
    }
    if( legal ) {
        phi1 = phiGrad - testwert * Sequenz.Bilder.MAX_DELTA; phi1 = phi1 * Math.PI / 180.0;
        labNeu[ 1 ] = c * Math.cos( phi1 );
        labNeu[ 2 ] = c * Math.sin( phi1 );
        rgbDeltaPhiMinus = labNachRGB( labNeu );
    }
// Alte LAB und RGB-Werte wieder herstellen
    setRGB( rgbTemp[ 0 ], rgbTemp[ 1 ], rgbTemp[ 2 ] );
    setLAB( labTemp );
}

/**
 * Zur Abfrage, ob die aktuellen RGB-Werte innerhalb des Wertebereichs liegen.
 * @return <code>true</code> wenn die RGB-Werte gültig sind
 */

public boolean getLegal() {
    if( legal )
        testLegal();
    return legal;
}

/**
 * Rechnet von RGB nach XYZ mit den übergebenen Werten und gibt die XYZ-Werte zurück.
 * @param r R-Anteil der Farbe
 * @param g G-Anteil der Farbe
 * @param b B-Anteil der Farbe
 * @return <code>double</code>-Array mit den XYZ-Werten. Reihenfolge: X - Y - Z
 */

public double[] rgbNachXYZ( int[] rgb ) {
    setRGB( rgb );
    rgbNachXYZ();
    return getXYZ();
}

/**
 * Rechnet von RGB nach XYZ mit den übergebenen Werten und gibt die XYZ-Werte zurück.
 * @param r R-Anteil der Farbe
 * @param g G-Anteil der Farbe
 * @param b B-Anteil der Farbe
 * @return <code>double</code>-Array mit den XYZ-Werten. Reihenfolge: X - Y - Z
 */

public double[] rgbNachXYZ( int r, int g, int b ) {
    setRGB( r, g, b );
    rgbNachXYZ();
    return getXYZ();
}

```

```

/**
 * Rechnet von RGB nach XYZ mit den internen Werten. Die neuen
 * XYZ-Werte können dann mit <code>getXYZ()</code> abgerufen werden.
 */

public void rgbNachXYZ() {
    float linRot, linGruen, linBlau;
    float xTemp, yTemp, zTemp;
    legal = true;
// Die Gamma-Korrektur in die RGB-Werte rechnen
    if( hatTRC ) {
        float nenner = ( Short.MAX_VALUE - Short.MIN_VALUE );
        int rotTemp, gruenTemp, blauTemp;
        rotTemp = ( int ) ( rot / 255.0 * ( rotTRC.length - 1 ) );
        gruenTemp = ( int ) ( gruen / 255.0 * ( gruenTRC.length - 1 ) );
        blauTemp = ( int ) ( blau / 255.0 * ( blauTRC.length - 1 ) );
        linRot = Zahlen.sigToUnsig( rotTRC[ rotTemp ] ) / nenner;
        linGruen = Zahlen.sigToUnsig( gruenTRC[ gruenTemp ] ) / nenner;
        linBlau = Zahlen.sigToUnsig( blauTRC[ blauTemp ] ) / nenner;
    }
    else {
        linRot = ( float ) Math.pow( rot / 255d, rotGamma );
        linGruen = ( float ) Math.pow( gruen / 255d, gruenGamma );
        linBlau = ( float ) Math.pow( blau / 255d, blauGamma );
    }
// Via Matrix die XYZ Werte errechnen
    xTemp = ( matrix[0][0] * linRot + matrix[0][1] * linGruen + matrix[0][2] * linBlau ) * 100;
    yTemp = ( matrix[1][0] * linRot + matrix[1][1] * linGruen + matrix[1][2] * linBlau ) * 100;
    zTemp = ( matrix[2][0] * linRot + matrix[2][1] * linGruen + matrix[2][2] * linBlau ) * 100;
    setXYZ( xTemp, yTemp, zTemp );
}

/**
 * Rechnet von CIE-Lab nach RGB mit den übergebenen Werten. Die neuen
 * RGB-Werte werden als 32-bit-<code>int</code>-Wert zurückgegeben.
 * <p>
 * Dabei gilt Standardmäßig: Blau: bit 0-7; Gruen: bit 8-15; Rot: bit 16-23;
 * Alpha (wird hier nicht benutzt): bit 24-31.
 * <p>
 * Die RGB-Werte werden auf den mit <code>setMinMax()</code> gesetzten
 * Wertebereich begrenzt.
 * @param lab Die CIE-Lab-Werte als <code>double</code>-Array
 * @return Eine 32-bit Zahl mit den RGB-Werten
 */

public int labNachRGB( double[] lab ) {
// RGB-Werte aus den übergebenen CIE-Lab Werten errechnen
// Die neuen RGB-Werte liegen in rot, gruen und blau
    labNachXYZ( lab );
    xyzNachRGB();
// RGB begrenzen
    if( rot < 0 ) { rot = 0; }
    if( rot > 255 ) { rot = 255; }
    if( gruen < 0 ) { gruen = 0; }
    if( gruen > 255 ) { gruen = 255; }
    if( blau < 0 ) { blau = 0; }
    if( blau > 255 ) { blau = 255; }
}

```

```

// Alpha auf opaque setzten
int rgb = 0xff000000;
// RGB-Werte in die int-Variable zuweisen
rgb = ( rgb & 0xff000000 ) | ( rot << 16 ) | ( gruen << 8 ) | ( blau << 0 );
return rgb;
}
}

```

6.9.2 Klasse Farbmeterik.LABmetrik

```

/*
LABmetrik.java
Title:      Shiraz
Author:     Jan Fischer
Description: Programm zur definierten Verrauschung von CIE-Lab Farben
            im RGB Farbraum unter Einbeziehung eines CMM.
*/

package Farbmeterik;

import javax.swing.*;
import java.text.DecimalFormat;

/**
 * Ein <code>LABmetrik</code> Objekt rechnet von CIE-XYZ nach CIE-Lab
 * und zurück unter Einbeziehung eines Standard- oder eines eigenen
 * Weißpunktes. Die CIE-XYZ und CIE-Lab Werte sind dabei nicht normalisiert!
 * @version 1.1
 * @author Jan Fischer
 * @date 14.05.2002
 * @see Farbmeterik.RGBmetrik
 * @see Farbmeterik.ICC_RGBmetrik
 */

public class LABmetrik {
    private double lWert, aWert, bWert;           // CIE-Lab Werte
    private double xWert, yWert, zWert;         // CIE-XYZ Werte
    private double xWeiss, yWeiss, zWeiss;      // CIE-XYZ Werte des Weisspunktes
    private final double drittel = 1.0 / 3.0;   // wird für die XYZ -> Lab Rechnung benötigt
    private final double grenzwert = 0.008856; // Grenzwert in der XYZ -> Lab Berechnung

    /**
     * Standard-Constructor für das Farbmeterikmodul.
     * <p>
     * Initialisiert den Weißpunkt auf D65.
     */

    public LABmetrik() {
        setWeisspunkt( 95.04, 100.00, 108.89 );
    }
}

```

```

/**
 * Constructor für das Farbmotrikmodul.
 * <p>
 * Initialisiert den Weißpunkt mit den angegebenen X, Y und Z Werten.
 * @param x X-Wert des Weißpunktes im CIE-XYZ Farbraum
 * @param y Y-Wert des Weißpunktes im CIE-XYZ Farbraum
 * @param z Z-Wert des Weißpunktes im CIE-XYZ Farbraum
 */

public LABmetrik( double x, double y, double z ) {
    setWeisspunkt( x, y, z );
}

/**
 * Constructor für das Farbmotrikmodul.
 * <p>
 * Initialisiert den Weißpunkt mit den angegebenen X, Y und Z Werten.
 * @param array Array aus drei Feldern mit den Weisswerten.
 * Reihenfolge: X - Y - Z
 */

public LABmetrik( double[] array ) {
    setWeisspunkt( array );
}

/**
 * Initialisiert den Weißpunkt mit den angegebenen X, Y und Z Werten.
 * <p>
 * @param x X-Anteil des Weißpunktes
 * @param y Y-Anteil des Weißpunktes
 * @param z Z-Anteil des Weißpunktes
 */

public void setWeisspunkt( double x, double y, double z ) {
    xWeiss = x;
    yWeiss = y;
    zWeiss = z;
}

/**
 * Initialisiert den Weißpunkt mit den angegebenen X, Y und Z Werten.
 * <p>
 * Ist der Weißpunkt nicht gültig (ein Wert kleiner Null oder größer
 * 150 bzw. 100 bei Y oder das Array hat weniger als drei Werte),
 * so wird er auf D65 gesetzt.
 * @param array Array aus den drei Weißwerten. Reihenfolge: X - Y - Z
 */

public void setWeisspunkt( double[] array ) {
    // Länge des Arrays checken
    if( array.length < 3 ) {
        JOptionPane.showMessageDialog( null, "Ungültiger Weißpunkt. Array zu kurz!",
            "FEHLER !!!", JOptionPane.ERROR_MESSAGE );
        return;
    }
    setWeisspunkt( array[ 0 ], array[ 1 ], array[ 2 ] );
}

```

```

}

/**
 * Gibt die X,Y und Z Werte des Weißpunktes als <code>double</code>-Array aus.
 * @return Array aus drei double Werten mit [0] = X, [1] = Y und [3] = Z
 */

public double[] getWeisspunkt() {
    double[] werte = new double[ 3 ];
    werte[ 0 ] = xWeiss;
    werte[ 1 ] = yWeiss;
    werte[ 2 ] = zWeiss;
    return werte;
}

/**
 * Setzt die X, Y, und Z Werte. Die Werte werden kontrolliert, ob sie größer-gleich
 * 0 sind und kleiner-gleich dem Weißwert.
 * <p>
 * Werden im Array mehr als drei Werte übergeben, werden die Überzähligen
 * ignoriert. Bei weniger als drei Werten wird <code>false</code> zurückgegeben
 * und keine neuen Werte gesetzt! Der Wertebereich wird nur auf positive Werte
 * überprüft, nach oben wird <B>keine</B> Kontrolle gemacht!
 * @param xyz Array aus (mind.) 3 Werten in der Reihenfolge X - Y - Z
 * @return <code>true</code>, wenn die Aktion erfolgreich war.
 */

public void setXYZ( double[] xyz ) {
    if( xyz.length < 3 ) {
        JOptionPane.showMessageDialog( null, "Ungültiger Weißpunkt. Array zu kurz!",
            "FEHLER !!!", JOptionPane.ERROR_MESSAGE );
        return;
    }
    setXYZ( xyz[ 0 ], xyz[ 1 ], xyz[ 2 ] );
}

/**
 * Setzt die X, Y, und Z Werte.
 * @param x X-Anteil der Farbe
 * @param y Y-Anteil der Farbe
 * @param z Z-Anteil der Farbe
 */
public void setXYZ( double x, double y, double z ) {
    xWert = x;
    yWert = y;
    zWert = z;
}

/**
 * Gibt die X,Y und Z Werte als <code>double</code>-Array aus.
 * @return Array aus drei double Werten mit [0] = X, [1] = Y und [3] = Z
 */

public double[] getXYZ() {
    double[] werte = new double[ 3 ];

```

```

    werte[ 0 ] = xWert;
    werte[ 1 ] = yWert;
    werte[ 2 ] = zWert;
    return werte;
}

/**
 * Setzt die CIE-Lab Werte.
 * <p>
 * Es erfolgt keine Kontrolle, ob die übergebenen Werte korrekt sein können
 * und ob dass Array wirklich aus drei Elementen besteht!
 * @param lab Array aus (mind.) 3 Werten in der Reihenfolge L* - a* - b*
 */

public void setLAB( double[] lab ) {
    lWert = lab[ 0 ];
    aWert = lab[ 1 ];
    bWert = lab[ 2 ];
}

/**
 * Gibt die CIE-Lab Werte als Array aus.
 * @return Array aus drei <code>double</code>-Werten mit
 * [0] = L*, [1] = a* und [2] = b*
 */

public double[] getLAB() {
    double[] werte = new double[ 3 ];
    werte[ 0 ] = lWert;
    werte[ 1 ] = aWert;
    werte[ 2 ] = bWert;
    return werte;
}

/**
 * Rechnet die übergeben Werte von CIE-Lab nach XYZ unter Verwendung des
 * internen Weißpunktes.
 * @param lab <code>double</code>-Array aus den drei Werten für L*, a*, und b*
 * @return <code>double</code>-Array mit den drei Werten für X, Y, und Z
 */

public double[] labNachXYZ( double[] lab ) {
    setLAB( lab );
    labNachXYZ();
    return getXYZ();
}

/**
 * Rechnet mit den internen Werten von CIE-Lab nach XYZ
 * unter Verwendung des vorher definierten Weißpunktes.
 * Standardweißpunkt ist D65 ( falls kein anderer definiert wurde).
 */

public void labNachXYZ() {
    double yTemp;

```

```

// CIE-L* testen und Y berechnen
double test = Math.pow( ( IWert + 16 ) / 116.0, 3.0 );
if( test > grenzwert ) {
    yWert = test * yWeiss;
    yTemp = Math.pow( yWert / yWeiss, drittel );
}
else {
    yWert = IWert / 903.292 * yWeiss;
    yTemp = ( 7.787 * yWert / yWeiss ) + 16.0/116.0;
}
// CIE-a* testen und X berechnen
test = Math.pow( ( aWert / 500.0 ) + yTemp, 3.0 );
if( test > grenzwert ) {
    xWert = test * xWeiss;
}
else {
    xWert = ( ( aWert / 500.0 ) + yTemp - 16.0/116.0 ) / 7.787 * xWeiss;
}
// CIE-b* testen und Z berechnen
test = Math.pow( yTemp - ( bWert / 200.0 ), 3.0 );
if( test > grenzwert ) {
    zWert = test * zWeiss;
}
else {
    zWert = ( yTemp - ( bWert / 200.0 ) - 16/116 ) / 7.787 * zWeiss;
}
}

/**
 * Rechnet die übergeben Werte von XYZ nach CIE-Lab unter Verwendung des
 * internen Weißpunktes. Gibt das Ergebnis in einer 3er Array aus <code>double</code>
 * -Werten zurück.
 * @param x X-Wert der Farbe
 * @param y Y-Wert der Farbe
 * @param z Z-Wert der Farbe
 * @return <code>double</code>-Array mit den drei Werten für L*, a* und b*
 */

public double[] xyzNachLAB( double x, double y, double z ) {
    double[] werte = new double[ 3 ];
    if( !setXYZ( x, y, z ) ) { // Sollte das übergebene Array nicht OK sein,
        return werte; // wird ein leeres Array zurückgegeben.
    }
    xyzNachLAB();
    return getLAB();
}

/**
 * Rechnet die übergeben Werte von XYZ nach CIE-Lab unter Verwendung des
 * internen Weißpunktes. Gibt das Ergebnis in einer 3er Array aus <code>double</code>
 * -Werten zurück.
 * @param xyz <code>double</code>-Array aus den drei Werten für X, Y, und Z
 * @return xyz <code>double</code>-Array mit den drei Werten für L*, a*, und b*
 */

public double[] xyzNachLAB( double[] xyz ) {
    setXYZ( xyz );
}

```

```

        xyzNachLAB();
        return getLAB();
    }

    /**
     * Rechnet mit den internen Werten von XYZ nach CIE-Lab
     * unter Verwendung des vorher definierten Weißpunktes.
     * Standardweißpunkt ist D65 ( falls kein anderer definiert wurde).
     */

    public void xyzNachLAB() {
        double yFaktor, xFaktor, zFaktor, faktor;
        faktor = 16.0 / 116.0;
        xFaktor = ( xWert / xWeiss > grenzwert ?
            Math.pow( xWert / xWeiss, drittel ) : 7.787 * ( xWert / xWeiss ) + faktor );
        yFaktor = ( yWert / yWeiss > grenzwert ?
            Math.pow( yWert / yWeiss, drittel ) : 7.787 * ( yWert / yWeiss ) + faktor );
        zFaktor = ( zWert / zWeiss > grenzwert ?
            Math.pow( zWert / zWeiss, drittel ) : 7.787 * ( zWert / zWeiss ) + faktor );
        lWert = 116 * yFaktor - 16;
        aWert = 500 * ( xFaktor - yFaktor );
        bWert = 200 * ( yFaktor - zFaktor );
    }
}

```

6.9.3 Klasse Sequenz.Rauschen

```

/*
  Rauschen.java
  Title:      Shiraz
  Author:     Jan Fischer
  Description: Programm zur definierten Verrauschung von CIE-Lab Farben
              im RGB Farbraum unter Einbeziehung eines CMM.
*/

package Sequenz;

import java.awt.*;
import javax.swing.*;
import java.util.*;
import Farbmeterik.*;
import java.lang.*;

/**
 * <code>Rauschen</code> enthält die Algorithmen, um aus den vom Steuerungsfeld
 * übernommenen CIE-Lab-Werten eine Sequenz von verrauschten Bildern zu erzeugen.
 * <p>
 * Dabei werden jedoch nicht die Bilder direkt erzeugt, sondern nur <code>int</code>
 * Arrays mit den RGB-Werten. Diese werden dann in der <code>Bilder</code>-Klasse
 * zu Bildern gemacht.
 * <p>
 * @version 1.0
 * @author Jan Fischer
 * @date 27.02.2002
 */

```

```

* @see Sequenz.Bildfenster
* @see Sequenz.Bilder
* @see Sequenz.Bilddaten
*/

```

```

public class Rauschen {
    private double[] rausch;
    private int[][] deltaL;
    private int[][] deltaC;
    private int[][] deltaPhi;
    private int[] testbild;

    /**
     * Standard-Constructor. Legt Größe und Lage des <code>Bild</code>-Panels fest.
     * @param werte Die CIE-Lab-Werte als <code>double</code>-Array
     * @param iccProfil Das <code>File</code>-Objekt des verwendeten ICC-Profiles
     */

    public Rauschen( double[] werte, java.io.File iccProfil ) {
        System.gc();
        int pixel = Bilder.BREITE * Bilder.HOEHE;
        rausch = new double[ pixel ];
        rauschzahlen( pixel );
        try{
            deltaL = new int[ Bilder.ANZAHL ][ pixel ];
            deltaC = new int[ Bilder.ANZAHL ][ pixel ];
            deltaPhi = new int[ Bilder.ANZAHL ][ pixel ];
            testbild = new int[ pixel ];
        }
        catch( OutOfMemoryError oome ) {
            // Alles auf Anfang
            if( deltaL != null )
                deltaL = null;
            if( deltaC != null )
                deltaC = null;
            if( deltaPhi != null )
                deltaPhi = null;
            if( testbild != null )
                testbild = null;
            System.gc();
            // Neuer Versuch
            try{
                deltaL = new int[ Bilder.ANZAHL ][ pixel ];
                deltaC = new int[ Bilder.ANZAHL ][ pixel ];
                deltaPhi = new int[ Bilder.ANZAHL ][ pixel ];
                testbild = new int[ pixel ];
            }
            catch( OutOfMemoryError er ) {
                JOptionPane.showMessageDialog( null, "Nicht genug freier Speicher vorhanden!!!",
                    "MIST !!!", JOptionPane.ERROR_MESSAGE );
            }
        }
        bilderErzeugen( werte, iccProfil );
    }

    /**

```

```

* Errechnet die drei Bild-Sequenzen.
* @param labAlt <code>double</code>-Array mit den originalen CIE-Lab Werten
* @param iccProfil Das <code>File</code>-Objekt des verwendeten ICC-Profiles
*/

```

```

private void bilderErzeugen( double[] labAlt, java.io.File iccProfil ) {
    ICC_RGBmetrik farbmodul = new ICC_RGBmetrik( iccProfil );
    int rgbOriginal = farbmodul.labNachRGB( labAlt );
    int rgb, rgbDeltaLPlus, rgbDeltaLMinus,
        rgbDeltaCPlus, rgbDeltaCMinus, rgbDeltaPhiPlus, rgbDeltaPhiMinus;
    int pixel = Bilder.BREITE * Bilder.HOEHE;
    double bruch = (double) Bilder.MAX_DELTA / (double) Bilder.ANZAHL ;
    int i, j;
    int[][] tempTestbild = new int[ Bilder.HOEHE ][ Bilder.BREITE ];
    //// Grundbild erzeugen (Delta = 0 ) ////
    for( i = 0; i < ( pixel ); i++ ) {
        deltaL[ 0 ][ i ] = rgbOriginal;
        deltaC[ 0 ][ i ] = rgbOriginal;
        deltaPhi[ 0 ][ i ] = rgbOriginal;
    }
    double[] labNeu = new double[ 3 ];
    labNeu[ 0 ] = labAlt[ 0 ];
    labNeu[ 1 ] = labAlt[ 1 ];
    labNeu[ 2 ] = labAlt[ 2 ];
    double c, c1;
    double phi, phi1, phiGrad;
    //// Häufig benötigte Werte ausrechnen ////
    // C* ausrechnen
    c = Math.sqrt( labAlt[ 1 ] * labAlt[ 1 ] + labAlt[ 2 ] * labAlt[ 2 ] );
    // Phi ausrechnen (Phi in Rad!)
    phi = Math.atan( labAlt[ 2 ] / labAlt[ 1 ] );
    // Phi dem richtigen Quadranten zuweisen!
    if( labAlt[ 1 ] < 0 ) {
        phi += Math.PI;
    }
    if( labAlt[ 2 ] < 0 && labAlt[ 1 ] > 0 ) {
        phi += 2.0 * Math.PI;
    }
    // Phi in Grad umwandeln!
    phiGrad = phi * 180.0 / Math.PI;
    // Sinus und Cosinus von phi
    double sinus, cosinus;
    sinus = Math.sin( phi );
    cosinus = Math.cos( phi );
    // Rauschbilder berechnen
    for( j = 1; j < Bilder.ANZAHL; j++ ) {
        for( i = 0; i < ( pixel ); i++ ) {
            double delta = j * bruch * rausch[ i ];

            //// Neuer L* Wert ( = L* + Delta L* )
            labNeu[ 0 ] = labAlt[ 0 ] + delta;
            // RGB Koordinaten
            rgb = farbmodul.labNachRGB( labNeu );
            deltaL[ j ][ i ] = rgb;

            labNeu[ 0 ] = labAlt[ 0 ];          // L* ändert sich ab hier nicht mehr

            //// Neuer C*-Wert ( = C* + Delta C* )
            c1 = c + delta;

```

```

// a*-Wert des neuen Wertes
    labNeu[ 1 ] = c1 * cosinus;
// b*-Wert des neuen Wertes
    labNeu[ 2 ] = c1 * sinus;
// Neue RGB-Koordinaten
    deltaC[ j ][ i ] = farbmodul.labNachRGB( labNeu );

//// Neuer Phi-Wert (= Phi + Delta Phi) in Grad!
    phi1 = phiGrad + delta;
// Neuen Phi-Wert in Rad umwandeln
    phi1 = phi1 * Math.PI / 180.0;
// a*-Wert des neuen Wertes
    labNeu[ 1 ] = c * Math.cos( phi1 );
// b*-Wert des neuen Wertes
    labNeu[ 2 ] = c * Math.sin( phi1 );
// Neue RGB-Koordinaten
    deltaPhi[ j ][ i ] = farbmodul.labNachRGB( labNeu );
}
}

// Maximalwerte von Delta L für das Testbild
labNeu[ 0 ] = labAlt[ 0 ] + Bilder.MAX_DELTA;
rgbDeltaLPlus = farbmodul.labNachRGB( labNeu );
labNeu[ 0 ] = labAlt[ 0 ] - Bilder.MAX_DELTA;
rgbDeltaLMinus = farbmodul.labNachRGB( labNeu );

labNeu[ 0 ] = labAlt[ 0 ];

// Maximalwerte von Delta C für das Testbild
c1 = c + Bilder.MAX_DELTA;
labNeu[ 1 ] = c1 * cosinus; labNeu[ 2 ] = c1 * sinus;
rgbDeltaCPlus = farbmodul.labNachRGB( labNeu );
c1 = c - Bilder.MAX_DELTA;
labNeu[ 1 ] = c1 * cosinus;
labNeu[ 2 ] = c1 * sinus;
rgbDeltaCMinus = farbmodul.labNachRGB( labNeu );

// Maximalwerte von Delta Phi für das Testbild
phi1 = phiGrad + Bilder.MAX_DELTA;
phi1 = phi1 * Math.PI / 180.0;
labNeu[ 1 ] = c * Math.cos( phi1 );
labNeu[ 2 ] = c * Math.sin( phi1 );
rgbDeltaPhiPlus = farbmodul.labNachRGB( labNeu );
phi1 = phiGrad - Bilder.MAX_DELTA;
phi1 = phi1 * Math.PI / 180.0;
labNeu[ 1 ] = c * Math.cos( phi1 );
labNeu[ 2 ] = c * Math.sin( phi1 );
rgbDeltaPhiMinus = farbmodul.labNachRGB( labNeu );

//// Testbild erzeugen

// Original-Fläche
for( i = 0; i < 100; i++ ) {
    for( j = 0; j < 300; j++ ) {
        tempTestbild[ i ][ j ] = rgbOriginal;
    }
}

// Delta-Flächen

```

```

    for( i = 100; i < 200; i++ ) { // Zeile
        for( j = 0; j < 50; j++ ) { // Spalte
            tempTestbild[ i ][ j ] = rgbDeltaLPlus;
            tempTestbild[ i ][ j + 50 ] = rgbDeltaLMinus;
            tempTestbild[ i ][ j + 100 ] = rgbDeltaCPlus;
            tempTestbild[ i ][ j + 150 ] = rgbDeltaCMinus;
            tempTestbild[ i ][ j + 200 ] = rgbDeltaPhiPlus;
            tempTestbild[ i ][ j + 250 ] = rgbDeltaPhiMinus;
        }
    }

// Das Testbild in ein eindimensionales Array packen
int z = 0;
for( i = 0; i < Bilder.HOEHE; i++ ) {
    for( j = 0; j < Bilder.BREITE; j++ ) {
        testbild[ z ] = tempTestbild[ i ][ j ];
        z++;
    }
}

/**
 * Erzeugt die Sequenz von Rauschwerten. Der feste Startwert garantiert, dass
 * bei jedem Aufruf die selbe Sequenz von normalverteilten "Zufallszahlen"
 * erzeugt wird.
 * @param anzahl Die Anzahl der zu erzeugenden Rauschwerte
 */

private void rauschzahlen( int anzahl ) {
    Random zufall = new Random( 1 );
    for( int i = 0; i < anzahl; i++ ) {
        rausch[ i ] = zufall.nextGaussian();
        if( rausch[ i ] > 2.2 ) rausch[ i ] = 2.2;
        if( rausch[ i ] < -2.2 ) rausch[ i ] = -2.2;
    }
}

/**
 * Zum Abrufen des fertigen deltaL-Arrays.
 * @return Das <code>int</code>-Array mit den in RGB umgerechneten
 * Werten für die verschiedenen Rauschbilder
 */

public int[][] getDeltaL() {
    return deltaL;
}

/**
 * Zum Abrufen des fertigen deltaC-Arrays.
 * @return Das <code>int</code>-Array mit den in RGB umgerechneten
 * Werten für die verschiedenen Rauschbilder
 */

public int[][] getDeltaC() {
    return deltaC;
}

```

```
/**
 * Zum Abrufen des fertigen deltaPhi-Arrays.
 * @return Das <code>int</code>-Array mit den in RGB umgerechneten
 * Werten für die verschiedenen Rauschbilder
 */

public int[][] getDeltaPhi() {
    return deltaPhi;
}

/**
 * Zum Abrufen des fertigen testbild-Arrays.
 * @return Das <code>int</code>-Array mit den in RGB umgerechneten
 * Werten für das Testbild
 */

public int[] getTestbild() {
    return testbild;
}
}
```