
Masterarbeit

Herr/Frau
Christian Greim

**Entwicklung eines
gleichabständigen Farbraumes
mittels neuronaler Netze,
basierend auf klassischen
Messergebnissen**

2020

Masterarbeit

Entwicklung eines gleichabständigen Farbraumes mittels neuronaler Netze, basierend auf klassischen Messergebnissen

Autor:
Herr Christian Greim

Studiengang:
Information and Communication Science

Seminargruppe:
IC14w1-M

Erstprüfer:
Prof. Dr.-Ing. Rainer Zschockelt

Zweitprüfer:
Prof. Dr.-Ing. Frank Zimmer

Einreichung:
Mittweida, 12.2.2020

MASTER THESIS

**Development of an equidistant
colorspace by neural networks,
based on conventionally used test
results**

author:
Mr. Christian Greim

course of studies:
Information and Communication Science

seminar group:
IC14w1-M

first examiner:
Prof. Dr.-Ing. Rainer Zschockelt

second examiner:
Prof. Dr.-Ing. Frank Zimmer

submission:
Mittweida, 12.2.2020

Bibliografische Angaben:

Greim, Christian:

Entwicklung eines gleichabständigen Farbraumes mittels neuronaler Netze basierend auf klassischen Messergebnissen

Development of an equidistant colorspace by neural networks,
based on conventionally used test results

2020 - 99 Seiten

Mittweida, Hochschule Mittweida (FH), University of Applied Sciences,
Fakultät Medien, Masterarbeit, 2020

Abstract

Genormte Farbordnungssysteme wie CIE-Yxy, CIE-Lab bis hin zu CIECAM02 haben das Problem, dass Farbabstände, die von Menschen als gleich wahrgenommen werden, in ihren Zahlenwerten um Faktoren größer 4 voneinander abweichen. Das wäre so, als wollte man Längendifferenzen mit Linealen bestimmen, deren Abstände auf der Skala zwischen 1 und 4 cm oder stärker willkürlich variieren. Farbdifferenzen lassen sich also nach momentanem Stand der Normen nur sehr grob zahlenmäßig beschreiben.

Um hier einen substantiellen Fortschritt zu erzielen, werden die grundlegenden Arbeiten von Wright aus dem Jahr 1931 und von MacAdam aus dem Jahr 1942, im Licht neuerer Erkenntnisse detailliert kritisch gewürdigt.

Trotz der aufgezeigten Kritik wird unter der Annahme, dass die Untersuchungen von MacAdam ideal seien, versucht, ein gleichabständiges Farbsystem analog zu CIE-Lab zu finden. Hierzu wird das Farbsehen als neuronales Netz aufgefasst und beschrieben. Danach lassen sich Verfahren des Deep Learning, insbesondere die Backpropagation anwenden.

Die neuronal gewonnenen sinnvollen Ergebnisse erinnern stark an diejenigen, die auf herkömmlichem Weg bereits vor über 60 Jahren gewonnen wurden, was ein starker Hinweis auf die Sinnhaftigkeit des Verfahrens ist. Damit wird ein Weg skizziert, wie mittels großer, sorgfältig gemessener Datensätze von Farbdifferenzen sich Farbsysteme finden lassen sollten, mit denen sich Farbdifferenzen besser beschreiben lassen, als mit den Formeln in den aktuellen Normen.

Inhaltsverzeichnis

Abstract.....	V
1 Einleitung.....	13
1.1 Problemstellung.....	13
1.2 Form und Quellen.....	14
1.3 Methodik.....	15
2 Beschreibung der Versuche von Wright.....	16
3 Die Wright-Versuche und die Physiologie.....	21
4 Kritik an der Linearisierung des Farbreizes.....	24
4.1 Bereits verwendete nicht-lineare Farbreize.....	24
4.2 Vorschlag zur Vereinfachung bisheriger Formeln.....	25
4.3 Linear oder nicht-linear?.....	26
5 Vereinfachte Anwendung linearisierter Reize.....	27
6 MacAdam-Versuche.....	29
6.1 Grundsätzliche Bedeutung.....	29
6.2 Spektrale Betrachtungen.....	29
6.3 Helligkeiten im MacAdam-Versuch.....	32
6.4 MacAdam nicht nur Ellipsen.....	32
6.5 MacAdam als Gütekriterium.....	38
7 Zusammenfassende Kritik am Stand der Technik.....	40
7.1 Grundsätzliches Problem.....	40
7.2 Kritik im Detail.....	40
8 Die Sigmoidfunktion.....	42
9 Farbmetrische Formeln neuronal formuliert.....	44
10 Neuronale Optimierung.....	49
11 Architekturen Neuronaler Netze.....	52
11.1 Zwei- oder Dreidimensionales Ergebnis?.....	52

11.2	Unterschiedliche Architekturen und ihre Ergebnisse.....	57
12	Fazit.....	61
13	Ausblick.....	61
	Literaturverzeichnis.....	X
	Anlagen.....	XIV
	Eigenständigkeitserklärung.....	XV

Abkürzungen

CIE: *Die Internationale Beleuchtungskommission beziehungsweise frz. Commission Internationale de l'Éclairage (CIE) oder engl. International Commission on Illumination ist eine unabhängige Non-Profit-Organisation mit dem Ziel der internationalen Kooperation und des Informationsaustausches in allen Belangen der Wissenschaft und der Kunst der Beleuchtung. [...] Die Organisation wird von der ISO als internationale Standardisierungskörperschaft anerkannt; im Rahmen von Kooperationen arbeitet die CIE zusammen mit der ISO, der International Electrotechnical Commission sowie dem European Committee for Standardization.*¹

CIECAM02 *ist ein Farberscheinungsmodell (engl. Color Appearance Model) von der internationalen Beleuchtungskommission CIE (commission internationale de l'éclairage) und der Nachfolger von CIECAM97.*²

*Da ein Rechner nicht die Fähigkeit des menschlichen Auges besitzt, Farben abhängig von ihrer Umgebung zu verändern, [...] müssen [...] Berechnungen diese Funktion übernehmen. Diese Berechnungen sind in dem Modell von CIECAM02 integriert, das am Ende die angeglichenen Farbwerte zurückgibt.*³

CIE-Lab: Annähernd gleichabständiges Farbmodell von 1976⁴

CIE-XYZ: Farbsystem der CIE von 1931⁵

CIE-Yxy: CIE-Normfarbsystem, wegen seiner Form auch oft Schuhsohle genannt.⁶

CMF: Color Matching Function, schwer zu übersetzen, am besten mit Farbensensitivitätsfunktionen, wird auch wiedergegeben mit Norm-Spektralwertkurven. Gemeint sind die tatsächlichen oder nur angenommenen im Bereich zwischen 400 und 800 nm Lichtwellenlänge spektral differenzierten Empfindlichkeitskurven der Farbrezeptoren im menschlichen Auge bzw. der menschlichen Wahrnehmung.⁷

ISO: Internationale Organisation für Normung⁸

UCS: uniform color space⁹

ΔE: Farbabstand vorzugsweise in CIE-Lab¹⁰

1 [Wikipedia] „Internationale Beleuchtungskommission“, 26. Januar 2020

2 [Wikipedia] „CIECAM02“, 26. Januar 2020

3 [Wasilewski Kostova 2009], Seite 5

4 [Wikipedia] „Lab-Farbraum“, 24. Januar 2020

5 [Wikipedia] „DIN99-Farbraum“, 24. Januar 2020

6 [Wikipedia] „CIE-Normvalenzsystem“, 24. Januar 2020

7 [Wikipedia en] „CIE 1931 color space“, 24. Januar 2020

8 [Wikipedia] „ISO“, 24. Januar 2020

9 [Safdar 2017], Seite 1

10 [ECI 1999], Seite 8

Abbildungsverzeichnis

Abbildung 1: schmalbandiges Spektrum.....	16
Abbildung 2: idealisierte Darstellung aller beteiligten Spektren in einer Messung im Sinne von Guild und Wright.....	16
Abbildung 3: breitbandiges Spektrum.....	16
Abbildung 4: Spektrum mit zwei Maxima. Es können aber auch viele Maxima sein.....	17
Abbildung 5: Prinzipieller Versuchsaufbau bei Guild und Wright, nach [Kamm 1987].....	17
Abbildung 6: Versuchsergebnis von Guild und Wright, eigener Graf mit Zahlenwerten aus [Wyszecki 1982].....	17
Abbildung 7: Prinzipieller Versuchsaufbau bei Guild und Wright, nach [Kamm 1987] für „negative“ Farben oder besser virtuelle Farben.....	18
Abbildung 8: CIE1924 $V(\lambda)$ -Kurve, eigener Graf mit Zahlenwerten aus [Wyszecki 1982].....	18
Abbildung 9: Ergebnisse von Guild und Wright nur verrechnet mit CIE1924 $V(\lambda)$ -Kurve, eigener Graf mit Zahlenwerten aus [Wyszecki 1982].....	18
Abbildung 10: Vorab-Normierung $N(\lambda)$ vor Faltung von $r(\lambda)$, $g(\lambda)$ und $b(\lambda)$ mit CIE1924 $V(\lambda)$ -Kurve, eigener Graf mit Zahlenwerten aus [Wyszecki 1982] im Vergleich mit Linsendichte gemäß Stockman [cvrl], „Lens density spectra“	19
Abbildung 11: Faltung von $r(\lambda)$, $g(\lambda)$ und $b(\lambda)$ mit CIE1924 $V(\lambda)$ -Kurve mit Normierung $N(\lambda)$, eigener Graf mit Zahlenwerten aus [Wyszecki 1982].....	19
Abbildung 12: Aus der Abbildung11 mit der Formel oben errechnete Werte für CIE-XYZ.....	20
Abbildung 13: individual variations in photopic sensitivity results for 52 individuals, based on heterochromatic step by step brightness matching; from Gibson & Tyndall (1923) aus [MacEvoy 2015].....	21
Abbildung 14: Aus [Asano 2015], Seite 43.....	21
Abbildung 15: The normalized spectral sensitivity of human cone cells of short-, middle- and long-wavelength types. Wikipedia englisch „CIE 1931 color space“ basierend auf Daten aus [Stockman 1993], Seite 2516ff.....	22
Abbildung 16: Spektrale Absorptionskurven der verschiedenen Sehpigmente der Fotorezeptoren. Zur besseren Vergleichbarkeit wurden die Maxima auf 100 Prozent gesetzt. Wikipedia „Zapfen (Auge) vom 6.7.2019“ nach [Dartnall 1980], Seite 505.....	22
Abbildung 17: [Xu 1997], Seite 955.....	22
Abbildung 18: Vorab-Normierung $N(\lambda)$ vor Faltung von $r(\lambda)$, $g(\lambda)$ und $b(\lambda)$ mit CIE1924 $V(\lambda)$ -Kurve, eigener Graf mit Zahlenwerten aus [Wyszecki 1982] im Vergleich mit Linsendichte gemäß Stockman [cvrl], „Lens density spectra“	23

Abbildung 19: Unterschiedliche Kurven, die Lichtmenge zu Sinneseindruck oder Reiz zu Reizreaktion beschreiben.....	24
Abbildung 20: MacAdam und andere Ellipsen im CIE-Lab-Farbraum dargestellt aus [Luo 2000], Seite 341.....	25
Abbildung 21: Messwertkurven logarithmisch umgerechnet unter Abzug der Referenzfarbe, die bei den Experimenten in den „negativen“, üblicherweise virtuellen Farben mit vorhanden sind.....	27
Abbildung 22: XYZ-Kurven errechnet analog Wright, allerdings auf Grundlage logarithmischer Messwertkurven, wie in Abbildung 21 gezeigt.....	28
Abbildung 23: Calcite (CaCO ₃), or Iceland Spar, is a useful crystalline substrate most commonly used in polarization optics. https://www.thorlabs.com/newgrouppage9.cfm?object-group_id=6973&tabname=Calcite	30
Abbildung 24: Spektrale Transmissionskurven aus [Kodak 1928] gemäß [MacAdam1942] dargestellt.....	30
Abbildung 25: Spektrale Transmissionskurven aus [MacAdam1942].....	30
Abbildung 26: Spektrale Transmissionskurven aus [MacAdam1942].....	31
Abbildung 27: Spektrale Transmissionskurven aus [Kodak 1928]gemäß [MacAdam1942] dargestellt.....	31
Abbildung 28: Aus [Richter 1996], Seite 97 übliche Darstellung von MacAdamEllipsen.....	32
Abbildung 29: Darstellung der Originalmesswerte für die MacAdam-Ellipsen, eingetragen in ein CIE-Yxy-Diagramm streng nach den Messwerten in der MacAdams Arbeit von 1942 [MacAdam 1942], Seite 260f. Die Striche erscheinen sehr kurz, weil die Ellipsen eben nicht, wie sonst üblich vergrößert wurden.....	34
Abbildung 30: Allgemeine Messwerte von MacAdam eingetragen in ein CIE-Yxy-Diagramm streng nach den Messwerten in der MacAdams Arbeit von 1942 [MacAdam 1942], Seite 258f. Sehr hilfreich zur Überprüfung entstehender Grafen sind gerade diejenigen Mischgeraden, die möglichst nahe an den äußersten Rändern des CIE-Yxy-Diagrammes liegen.....	35
Abbildung 31: Allgemeine Messwerte von MacAdam eingetragen in ein CIE-Lab-Diagramm streng nach den Messwerten in der MacAdams Arbeit von 1942 [MacAdam 1942], Seite 258f. Da für CIE-Lab ein Helligkeitswert angenommen werden muss, hat sich die Grafik angeboten, eine möglichst gute Einbeschreibung in den Spektralfarbenzug zu suchen. In diesem Fall bei einem L-Wert von 21,6.....	36
Abbildung 32: Allgemeine Messwerte von MacAdam eingetragen in ein DIN-Lab99-Diagramm streng nach den Messwerten in der MacAdams Arbeit von 1942 [MacAdam 1942], Seite 258f. Da für DIN-Lab99 ein Helligkeitswert angenommen werden muss, hat sich die Grafik angeboten, eine möglichst gute Einbeschreibung in den Spektralfarbenzug zu suchen. In diesem Fall bei einem L99-Wert von 31.....	37
Abbildung 33: Allgemeine Messwerte von MacAdam eingetragen in ein CIECAM02-UCS-Diagramm streng nach den Messwerten in der MacAdams Arbeit von 1942 [MacAdam 1942],	

Seite 258f. Da CIECAM02-UCS eine Lichtart verlangt, wurde Normlicht C verwendet, wie dies MacAdam angibt. Dieses hat die CIE-XYZ-Koordinaten: 0,309 0,319 0,372. $L_A = 24$, Weiß der Probe in Relation zum Weiß der Umgebung $Y_b = 10$, VIEWING_CONDITIONS = 'Dark', angestrebte Helligkeit ist wieder die Lab-Helligkeit von von 21,6.....	38
Abbildung 34: Wie sind im DIN99-Diagramm die allgemeinen Messungen von MacAdam verteilt? Bei einem Abstand zwischen 20 und 30 gibt es vier Werte, die extrem nach oben abweichen und erst bei einem Abstand von über vierzig gibt es viele extrem kleine Werte. Diese extrem gesättigten Farben spielen im Alltag bei Aufsichtsfarben kaum eine Rolle.....	39
Abbildung 35: Die Simoidfunktion, wie sie häufig in Neuronalen Netzen bzw. beim Deep Learning zum Einsatz kommt.....	42
Abbildung 36: Reiz-Reaktionskurve für CIECAM02.....	42
Abbildung 37: Die Simoidfunktion, um 0,5 subtrahiert und anschließend mit 2 multipliziert, im Vergleich zu der Übertragungsfunktion in Abbildung 19.....	43
Abbildung 38: Die für CIECAM02-UCS geänderten Normvalenzen, wobei bemerkenswert ist, dass die geringen negativen Anteile anscheinend nicht ins Gewicht fallen.....	44
Abbildung 39: Die bekannten Normvalenzen.....	44
Abbildung 40: Architektur eines Neuronalen Netzes analog CIE-Lab allerdings mit den modifizierten XYZ-Werten von CIECAM02-UCS und Sigmoidfunktionen statt Kubikwurzeln.....	46
Abbildung 41: Werte für die MacAdam-Ellipsen eingetragen in ein Diagramm, das aus Matrix-Sigmoid-Berechnungen analog zu CIE-Lab gewonnen wurde.....	47
Abbildung 42: Werte für die allgemeinen Messwerte von MacAdam, eingetragen in ein Diagramm, das aus MatrixSigmoid-Berechnungen analog zu CIE-Lab gewonnen wurde.....	48
Abbildung 43: Skizze zur Errechnung des idealen Farbortes der zweiten Randfarbe im Gegensatz zum realen Farbort.....	50
Abbildung 44: Architektur eines Neuronalen 3-5-3-Netzes.....	51
Abbildung 45: a-b-Ebene Allgemeine MacAdam-Messwerte für Neuronales 3-5-3 Netzwerk...52	52
Abbildung 46: Blau-Gelb-Differenz b^* für die Farbschwellen als Funktion der Rot-Grün-Buntheit a^* und der Blau-Gelb -Buntheit b^* , aus [Kittelman 2010], Seite 52.....	53
Abbildung 47: L-a-Ebene Allgemeine MacAdam-Messwerte für Neuronales 3-5-3 Netzwerk...54	54
Abbildung 48: L-b-Ebene Allgemeine MacAdam-Messwerte für Neuronales 3-5-3 Netzwerk...55	55
Abbildung 49: Gekrümmte Fläche einheitlicher Helligkeit nach MacAdam. Aus https://www.colors-system.com/?page_id=907&lang=en gelesen am 2.10.2019 analog zur Abbildung in [Wyszecki 1982], Seite 668.....	56
Abbildung 50: Gekrümmte Fläche einheitlicher Helligkeit nach Stiles. Aus https://www.colors-system.com/?page_id=915 gelesen am 2.10.2019 analog zur Abbildung in [Wyszecki 1982], Seite 669.....	56

Abbildung 51: Werte für die allgemeinen Messwerte von MacAdam, eingetragen in ein Diagramm, nach Berechnungen eines Neuronalen 3-5-2-Netzes.....	57
Abbildung 52: Architektur eines Neuronalen 3-5-2-Netzes.....	58
Abbildung 53: Architektur eines Neuronalen 3-2-2-Netzes.....	58
Abbildung 54: Werte für die allgemeinen Messwerte von MacAdam, eingetragen in ein Diagramm, nach Berechnungen eines Neuronalen 3-2-2-Netzes.....	59
Abbildung 55: Architektur eines Neuronalen 3-3-4-2-Netzes.....	59
Abbildung 56: Werte für die allgemeinen Messwerte von MacAdam, eingetragen in ein Diagramm, nach Berechnungen eines Neuronalen 3-3-4-2-Netzes.....	60

Tabellenverzeichnis

CIE-Farbsysteme.....	39
Allgemeine_Messwerte.....	45
Ellipsen-Messwerte.....	45

1 Einleitung

1.1 Problemstellung

In der Drucktechnik bildet die richtige Farbwiedergabe den vielleicht wichtigsten Problemkreis. Stimmt die meistens vom Bildschirm, manchmal auch von Farbmustern vorgegebene Farbe mit dem Druckergebnis überein? Welche Toleranzen sollen dabei zugelassen werden? Die Ergebnisse hängen dabei auch vom Umgebungslicht ab, weil abgesehen von Fluoreszenz maximal nur reflektiert werden kann, was vorher auf den Bedruckstoff gefallen ist. Zudem adaptiert das Auge auf das Umgebungslicht. Bei Farben auf Bildschirmen ist das etwas anders, weil hier das Auge weitgehend auf den Bildschirm adaptiert, der gleichzeitig die Lichtquelle ist. In beiden Fällen ist aber letztlich das Licht entscheidend, das im Auge den Reiz auslöst. Die farbliche Zusammensetzung dieses einfallenden Lichtes lässt sich mit einem Spektrum, wie in den Abbildungen 1 bis 4 gezeigt, gut beschreiben.

Schon 1931 verabschiedete die CIE den sogenannten CIE-XYZ-Farbraum als Norm.¹¹ Dieser ermöglicht es von einer Spektralen Messung im Bereich der Wellenlängen zwischen 400 und 800 nm Lichtwellenlänge eindeutig auch einen Punkt in jenem CIE-XYZ-Farbraum zu schließen. Theoretisch gibt es unendlich viele Spektren, die auf den selben Punkt (X, Y, Z) fallen. Das ist genau die Absicht: aus der riesigen Menge an Information eines gemessenen Spektrums auf die relativ kleine Menge an Information zu schließen, die unser Sehsinn dem entnimmt. In modernen Farbsystemen ist das eine Information über die Helligkeit, eine über die Lage zwischen Gelb und Blau und schließlich eine über die Lage zwischen Rot und Grün. Letztlich arbeitet unser Sehsinn nur in diesen drei Dimensionen: Helligkeit, Blau-Gelb-Färbung und Rot-Grün-Färbung. Demgegenüber werden in einem Spektrum üblicherweise Messungen in 10 nm-Schritten gemacht, also eine bei 400 nm, die nächste bei 410 nm und so weiter. Jede dieser 40 Messungen liefert einen Wert zwischen 0 und 1. Eine Messung beinhaltet 40 Werte, die voneinander unabhängig beliebig zwischen 0

und 1 liegen können. Mathematisch betrachtet haben wir also einen 40-dimensionalen endlichen Raum, den wir auf den 3-dimensionalen Raum menschlicher Farbwahrnehmung abbilden.

Der CIE-XYZ-Farbraum erfüllt diese mathematischen Anforderungen, aber mit einem entscheidenden Haken. Die Empfindlichkeit menschlichen Sehens ist an unterschiedlichen Orten des CIE-XYZ-Farbraums sehr unterschiedlich. Das hört sich erst einmal nach einer akademischen Finesse an, hat aber schwierige Probleme in der Praxis zur Folge. Es ist etwa so, als ob zwei Personen zur Messung von Abständen zwei gleiche Lineale verwenden würden, deren Einteilungen nicht gleichmäßig sind. Wenn die eine Person eine Länge in Zahlen übermitteln will, dann kann der Empfänger der Zahl genau die gleiche Länge ermitteln. Bei Abweichungen kann aber der Empfänger nur sehr umständlich angeben, wie viel die Abweichung beträgt. Toleranzen lassen sich auf diese Weise nur höchst umständlich definieren. Gerade bei der Darstellung von Farben in der grafischen Industrie geht es ständig um Farbabweichungen und deren Zulässigkeit. Zudem kann ein Druckverfahren in Abhängigkeit vom Bedruckstoff nur einen bestimmten Farbraum abbilden. Der Farbraum am Monitor eines Auftraggebers muss also in den Farbraum der Druckmaschine umgerechnet werden. Dies funktioniert zwar trotz der untauglichen „Lineale“ für Farben, aber es könnte einfacher sein.

Bisher wurde davon ausgegangen, dass der CIE-XYZ-Farbraum die menschliche Wahrnehmung prinzipiell gut wiedergibt und man mit mathematischen Methoden diesen Farbraum so verzerren kann, dass er wahrgenommene Farbabweichungen in allen Bereichen gleichmäßig wiedergeben kann. Der erste genormte Versuch war der CIE-Lab-Farbraum von 1976, der nach wie vor seinen Dienst in vielen Normen wie den Normen ISO 12647 für die grafische Industrie tut. Aber auch dieser gibt wahrgenommene Farbabweichungen nur ungleichmäßig wieder, so dass die aktuelle Norm ISO 12647 auch alternative Farbabstandsformeln zulässt wie CIE-DE2000. In neueren Farbraummodellen wie CIE-CAM02 werden auch das Umgebungslicht und die Beleuchtungsstärke berücksichtigt¹², was bei reinen Untersuchungen an Bildschirmen kaum nötig ist.

11 [Wikipedia] „Lab-Farbraum“, 25. Juni.2019

12 [Wasilewski Kostova 2009], Seite 10f

Nach nun langjährigen und zahlreichen Versuchen, die immer nur kompliziertere Formeln hervorgebracht haben, ist es an der Zeit, die Frage deutlich zu stellen, ob nicht die Ungleichmäßigkeiten in allen CIE-Farbräumen schon in den Experimenten von Wright, auf denen letztlich alles fußt, und in den zugehörigen mathematischen Überlegungen begründet liegen. Da man auch aus Fehlern lernt, ist in dem derzeitigen unbefriedigenden Zustand auch das Wissen über Farbwahrnehmung gewachsen. Nun soll also kritisiert werden, wo das Vorgehen von Wright die Gleichabständigkeit der Farbmessung schon in den Grundlagen verhindert. Gleichzeitig soll versucht werden, mit dem aktuellen Wissen über Farbwahrnehmung Alternativen aufzuzeigen.

Ähnlich grundlegend wie die Versuche von Wright ist die Arbeit von MacAdam aus dem Jahr 1942¹³. Die sprichwörtlichen MacAdam-Ellipsen werden nach wie vor als Gütekriterium für Farbabstände heran gezogen. Auch sie sollen in dieser Arbeit kritisch gewürdigt werden. Im Gegensatz zu seinen Nachfolgern hat aber MacAdam sehr penibel auch mehr oder weniger die Rohdaten veröffentlicht, so dass diese kritische Würdigung überhaupt möglich ist und seine Arbeit zu recht auch heute noch als Meilenstein gilt. Um die Rohdaten von MacAdam mit modernen Programmen und Programmierkonzepten verarbeiten zu können, ist es notwendig sich mit allen Details der Arbeit von MacAdam vertraut zu machen. Durch die intensive Beschäftigung mit und Nutzung der Messdaten und Formeln von MacAdam, ergibt sich ein Bild von Leistungsfähigkeit und Grenzen der MacAdam-Ellipsen. Heute bilden die MacAdam-Ellipsen ein übliches Gütekriterium für Farbordnungssysteme.¹⁴ Dieses Kriterium wurde von MacAdam aus seinen Messwerten abgeleitet. Da inzwischen auch kleine Schreibtischcomputer mit großen Datenmengen umgehen können, ist es möglich, die gesamten Messwerte von MacAdam als Gütekriterium heran zu ziehen und nicht nur die Ellipsen. Dieses erweiterte Gütekriterium lässt sich auf ältere und aktuelle Farbsysteme anwenden und ermöglicht über Neuronale Netze und Deep Learning sogar die Entwicklung neuer Farbsysteme, wenn auch nur in gewissen Grenzen. Die Vorschläge für neue Systeme am Ende der Arbeit können

leider nur die Richtung für erfolgversprechende Methoden weisen, wie mit besseren Daten im Sinne von MacAdam tatsächlich ein einheitlich gleichabständiges Farbsystem gewonnen werden könnte.

1.2 Form und Quellen

Etwas ungewöhnlicherweise nehmen einen großen Teil dieser Arbeit Programmcode und Diagramme ein. Diagramme können die komplizierten Sachverhalte in der Farbmeterik oft einfacher darstellen, als lange Beschreibungen. Gerade Unterschiede in ein oder mehreren Diagrammen verdeutlichen gut Entwicklungsschritte oder mögliche Alternativen. In einem guten Diagramm steckt oft viel Arbeit, so dass dann letztlich nicht ganz so viel Textmenge gebraucht wird. Ein großer Teil der Diagramme wurde nicht in Excel oder genauer in LibreOffice Calc gemacht, sondern wurde wegen der großen Datenmengen und der Komplexität mittels Python in der Programmierumgebung Anaconda erzeugt. Die meisten Diagramme benötigen also keine Quellenangabe, da sie selbst erzeugt sind. Bei den Übrigen steht die Quelle im Text oder in der Bildunterschrift.

Die Listings finden sich im umfangreichen Anhang der Arbeit. Programmierung zwingt dazu, alle Sachverhalte, Konstanten, Grunddaten usw. wirklich sehr genau zu recherchieren und zu bedenken. Man kann sich anders als bei Alltagssprache nicht mit sprachlichen Kniffen über Ungenauigkeiten hinwegformulieren. Der Computer ist in seinen Ergebnissen oder Nicht-Ergebnissen sehr unbarmherzig. Dies zwingt zu einem sehr genauen Durchdringen der Materie, ist aber auch sehr zeitaufwändig. Ich musste mich nach längerer Zeit der Programmierabstinenz erst wieder an dieses sehr strukturierte Denken gewöhnen, das noch dadurch erschwert wurde, dass ich bisher noch nichts in Python programmiert hatte. Auch die Auswahl einer geeigneten Programmierumgebung hat eine Zeit in Anspruch genommen. Ausdrücklich sehr dankbar bin ich für das Buch „Tariq Rashid, Frank Langenau (Übersetzung): Neuronale Netze selbst programmieren, Ein verständlicher Einstieg mit Python“ [Rashid 2017], das einen guten praxisnahen Einstieg auf genau dem richtigen Niveau geboten hat. „Üblicherweise rechnet man mit einer insgesamten Produktivität – also inklusive aller Projektstätigkeiten – von

13 [MacAdam 1942]

14 [Luo 2000], Seite 347

10 bis 50 Lines of Code je Mitarbeiter und Tag¹⁵, was eher für geübte Programmierer gelten dürfte als für Wiedereinsteiger, womit klar wird, welcher zeitliche Aufwand im jeweiligen Programmcode steckt.

Auch wenn es nicht gerne gesehen wird, wird in dieser Arbeit häufig aus Wikipedia zitiert. Formal steht immer das Lesedatum dabei, da Wikipedia-Artikel stetig verbessert werden. Im wesentlichen wird dieses vereinfachte Zitieren nur für Physikalische Größen und den Stand der Technik verwendet, welcher unter Fachleuten der Farbmotrik derzeit als unumstritten gelten kann. Umstrittene oder neue Aussagen werden meistens aus Fachartikeln oder autorisierten Webseiten von Forschungsinstituten belegt. Bücher sind auf diesem Gebiet schwierig zu finden, weil sie entweder zu anwendungsbezogen oder bereits veraltet sind. Das weltweit anerkannte Standardwerk auf diesem Gebiet ist: Wyszecki, Günther, Stiles, W. S.: Color Science, Concepts and Methods, Quantitative Data Formulae. Dem Autor stand die Ausgabe von 2000 unbegrenzt zur Verfügung. Diese wurde hauptsächlich verwendet, da die Kapitel über die Versuche von Wright und MacAdam in den neueren Ausgaben nicht verändert wurden.

Die Fußnoten enthalten meistens die Quelle für die zentrale Aussage des vorangehenden Absatzes oder Satzes. Wörtliche Zitate sind durch Anführungszeichen oder *kursiven* Satz gekennzeichnet.

Wenn es viel Leerraum in der gedruckten Arbeit gibt, so ist dies den Eigenheiten von LibreOffice Writer geschuldet. Es ist insbesondere schwierig, Abbildungen, die zwei Spalten breit sind, in den fortlaufenden Text zu integrieren.

1.3 Methodik

Der erste beschreibende Teil unterzieht bisherige grundlegende Forschungsergebnisse einer kritischen Würdigung. Spätere Forschungserkenntnisse stellen manche expliziten oder stillschweigenden Annahmen der grundlegenden Experimente und Ergebnisse in Frage. In den jeweiligen Kapiteln wird der damalige Stand des Wissens und die Einwände durch modernere Forschung beschrieben. Dann wird abgewogen, ob der Einwand berechtigt ist und wie weit die ursprünglichen Ergebnisse in Zweifel

gezogen werden müssen. Diese Abwägungen können unterschiedlich ausfallen und bewegen sich zwischen folgenden Extremen: Ein prinzipiell schwerwiegender Einwand kann sich in so kleinen Größenordnungen auswirken, dass die Gesamtergebnisse kaum zu beanstanden sind. Oder ein anderer Einwand ist zwar schon seit den damaligen Untersuchungen bekannt, wird aber seit dem nicht ausführlich diskutiert, um die Ergebnisse nicht prinzipiell in Frage stellen zu müssen.

In Kapitel 6.4 werden, soweit dem Autor bekannt, erstmalig die Messergebnisse und nicht die fertigen Ellipsen in anderen als dem CIE-Yxy-Farbraum dargestellt. Allein diese Darstellungen dürften für ein Fachpublikum durchaus von Interesse sein und für eine realistischere Einschätzung der Ergebnisse von MacAdam sorgen.

In Kapitel 7 werden die Problemstellungen aller bis dort hin getroffenen Kritikpunkte noch einmal als Zwischenstand kurz zusammengefasst.

Der zweite eher forschende Teil beschreibt zunächst das Farbsehen in der Sprache und Form neuronaler Netze. Die Messergebnisse von MacAdam werden im weiteren trotz aller Einwände als Ideal angenommen, wider besseres Wissen. Diese Messergebnisse werden als grundlegendes Gütekriterium für Farbabstandswahrnehmung definiert, um damit unterschiedliche Architekturen von Neuronalen Netzwerken zu trainieren. Durch das Training werden Koeffizienten ermittelt, die zusammen mit der Netzwerkarchitektur Rechenvorschriften für neue Farbräume bilden, die dann die ursprünglich sehr stark variierenden Farbabstände, die gerade nicht mehr wahrnehmbar sind, in wesentlich homogenere Zahlen umwandeln. Es wird nicht einfach ein fertiges System für neuronale Netze verwendet, sondern eine bestehende Software für Schriftenerkennung wird für den gewünschten Zweck umgeschrieben und angepasst, was den Vorteil hat, im Detail nachvollziehen zu können, was genau gerechnet wird.

Schließlich werden die neu ermittelten Farbräume anhand des Gütekriteriums, MacAdam-Messwerte, mit Farbräumen verglichen, wie sie aktuelle Normen und Standards zur Verfügung stellen.

2 Beschreibung der Versuche von Wright

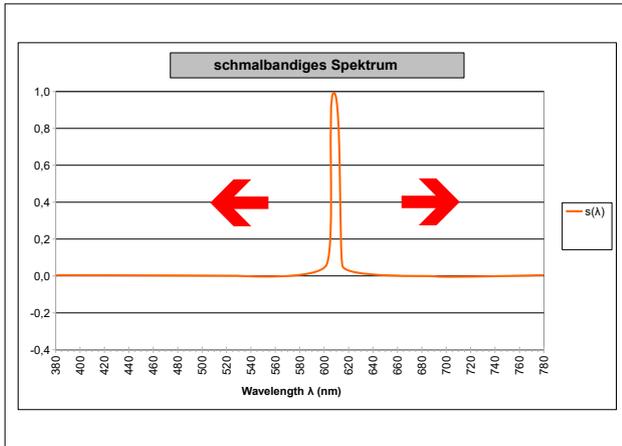


Abbildung 1: schmalbandiges Spektrum

Die Versuche von Wright (1928) bzw. Guild (1931) dienen erst einmal nicht dazu spektrale Farbempfindlichkeitsfunktionen zu ermitteln. Das sichtbare Licht wird dabei in einzelne Wellenlängenbereiche von jeweils 10 nm Breite eingeteilt. Es ergeben sich also im Bereich von 400 bis 800 nm vierzig Wellenbereiche. Wenn man einen dieser Bereiche ansprechen und messtechnisch auswerten will, ergeben sich Messkurven wie in Abbildung 1 bis 4 gezeigt. Aufgabe der Versuche ist es nun, einen solchen Wellenbereich durch eine Mischung aus drei anderen Bereichen zu beschreiben. Dabei wird jeder mögliche Wellenbereich 400 bis 410 nm, 410 nm bis 420 nm und sofort bis 790 bis 800 nm, getrennt untersucht. Unsere Abbildung 1 zeigt in etwa den Bereich 600 bis 610 nm.

Wright hat versucht, die unterschiedlichen schmalbandigen Spektren $s(\lambda)$ aus folgenden drei schmalbandigen Spektren nach zu mischen: für die kurzen Wellenlängen ein schmalbandiges Spektrum mit einem Maximum bei 436 nm, für die mittleren Wellenlängen eines mit einem Maximum bei 546 nm und schließlich für den langwelligen Bereich mit einem Maximum bei 700 nm. Wegen ihrer besonderen Eigenschaften wurden bei den Experimenten nur schmalbandige Spektren verwendet. Diese sind im normalen Leben aber eher eine Ausnahmeer-

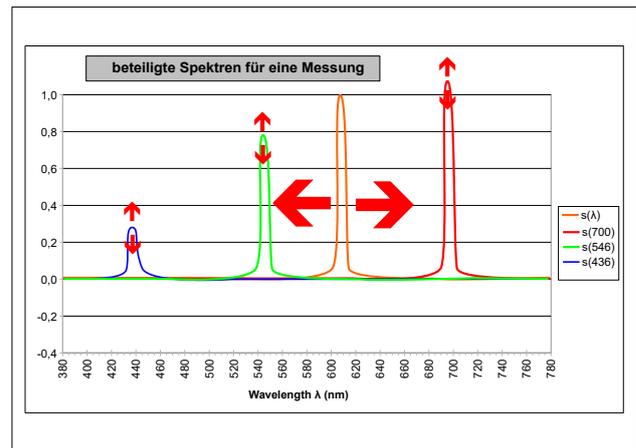


Abbildung 2: idealisierte Darstellung aller beteiligten Spektren in einer Messung im Sinne von Guild und Wright

scheinung, der wir allenfalls in Gestalt eines Schmetterlings oder Regenbogens begegnen. Im Normalfall sehen die Spektren von Farben eher aus, wie in Abbildung 3 und Abbildung 4 gezeigt. Diese sind modellhaft so gewählt, dass sie vielleicht sogar als gleich wahrgenommen würden.

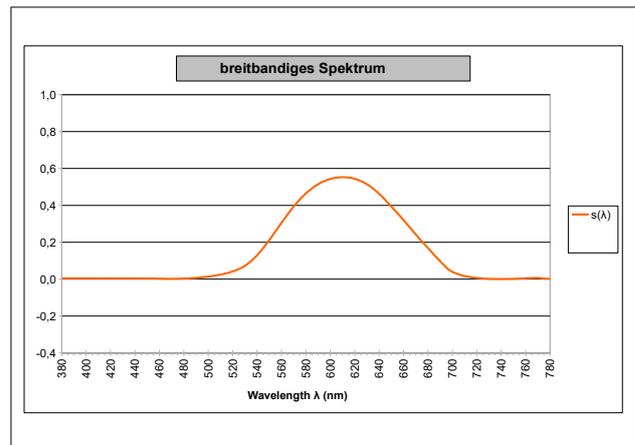


Abbildung 3: breitbandiges Spektrum

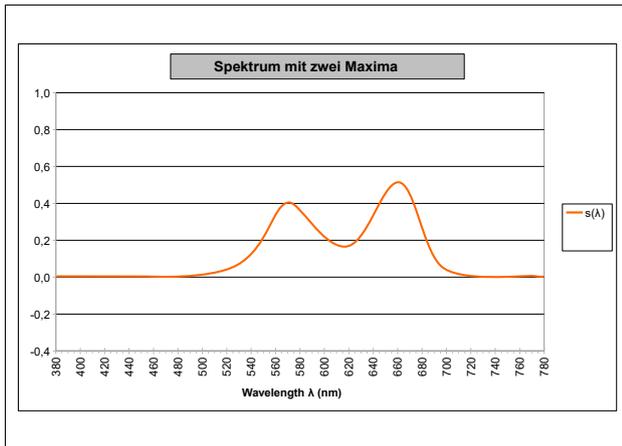


Abbildung 4: Spektrum mit zwei Maxima. Es können aber auch viele Maxima sein.

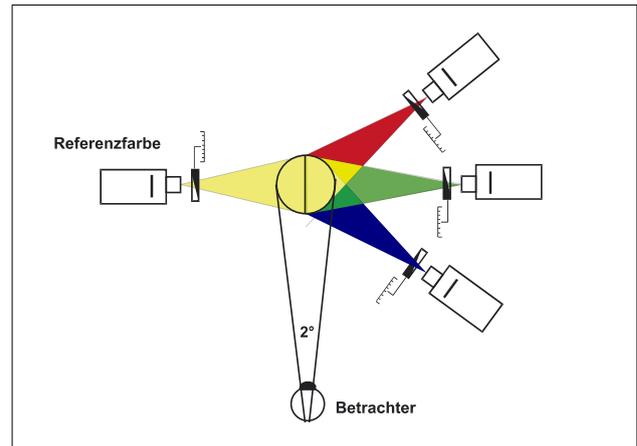


Abbildung 5: Prinzipieller Versuchsaufbau bei Guild und Wright, nach [Kamm 1987]

Die Grundüberlegung Wrights für die Messungen war, dass letztlich jeder Farbreiz wie modellhaft in Abbildung 3 und Abbildung 4 gezeigt aus unterschiedlichen Anteilen solcher Wellenbereiche wie in Abbildung 1 gezeigt zusammengesetzt ist.

Man sollte sich im Klaren sein, dass für jede Messung eines schmalbandigen Referenzspektrums $s(\lambda)$ drei weitere schmalbandige Spektren in unterschiedlicher Stärke beteiligt sind, wie in Abbildung 2 gezeigt. Der Versuchsaufbau wird in Abbildung 5 gezeigt. Ein Beobachter blickt auf ein zerteiltes rundes Betrachtungsfeld. In unserer Darstellung Abbildung 5 wird links die Referenzfarbe $s(\lambda)$ projiziert, deren Maximum bei jedem Versuch variiert wird. Mittels eines Helligkeitsreglers kann dafür gesorgt werden, dass die unterschiedlichen Referenzfarben $s(\lambda)$ für den Beobachter immer gleich hell erscheinen. Rechts werden immer die gleichen drei Grundfarben, $s(436)$, $s(546)$ und $s(700)$, mit Maxima bei 436, 546 und 700nm, projiziert. Mit den 2° wird letztlich die Größe des Betrachtungsfeldes angegeben. Je nach Größe des Betrachtungsfeldes muss der Beobachtungsabstand so gewählt werden, dass der Raumwinkel 2° beträgt.

Die Beobachter sollen nun die Intensität der Grundfarben, $s(436)$, $s(546)$ und $s(700)$, mittels dreier Helligkeitsregler jeweils so verändert werden, dass die linke und rechte Bildhälfte nicht zu unterscheiden sind. Auf diese Weise wird für jedes $s(\lambda)$ ein Mischungsverhältnis aus $s(436)$, $s(546)$ und $s(700)$ ermittelt. Für jeden Versuch würde ein Satz Formeln folgender Form entstehen: $s(400) = r_1 \cdot s(700) + g_1 \cdot s(546) + b_1 \cdot s(436)$; $s(410) = r_2 \cdot s(700) + g_2 \cdot s(546) + b_3 \cdot s(436)$; bis $s(800) = r_{40} \cdot s(700) + g_{40} \cdot s(546) +$

$b_{40} \cdot s(436)$. Die so gewonnenen Werte r_1 bis r_{40} , g_1 bis g_{40} , b_1 bis b_{40} lassen sich dann ihrerseits wieder als Kurven in Abhängigkeit von der zugehörigen Wellenlänge von $s(\lambda)$ darstellen. Diese werden dann mit $r(\lambda)$, $g(\lambda)$ und $b(\lambda)$ bezeichnet. Das Ergebnis wird in Abbildung 6 dargestellt.

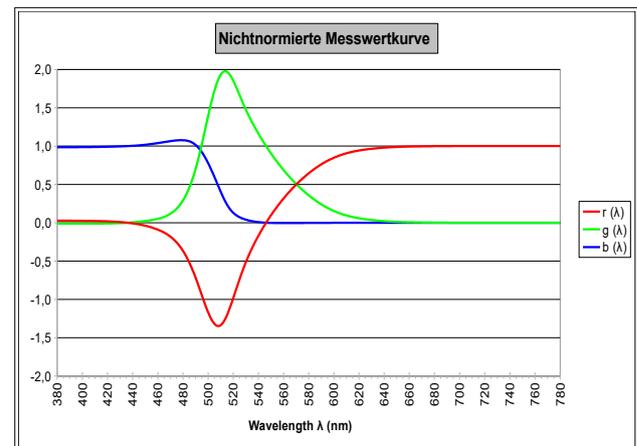


Abbildung 6: Versuchsergebnis von Guild und Wright, eigener Graf mit Zahlenwerten aus [Wyszecki 1982]

Bei dem Grafen springen sofort die negativen Werte der $r(\lambda)$ -Kurve ins Auge. Und damit die Frage: Was soll eine negative Farbe sein? Wir untersuchen hier aber noch keine Farben, sondern Mischungsverhältnisse. Besser sollten wir fragen: Wie kommen diese negativen Werte zu Stande? Bei den Experimenten wurde schnell klar, dass sich viele Farben $s(\lambda)$ nicht aus den drei anderen $s(436)$, $s(546)$ und $s(700)$ mischen lassen. Um nun trotzdem ein Mischungsverhältnis angeben zu können wurde eine der Grundfarben $s(436)$, $s(546)$ oder $s(700)$ zu-

sätzlich auf die Seite der Referenzfarbe $s(\lambda)$ projiziert, wie in Abbildung 7 dargestellt.

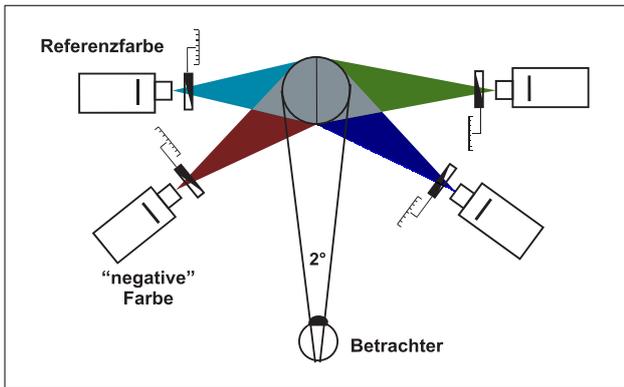


Abbildung 7: Prinzipieller Versuchsaufbau bei Guild und Wright, nach [Kamm 1987] für „negative“ Farben oder besser virtuelle Farben.

In der Abbildung 7 wird $s(700)$ mit auf die Referenzfarbe $s(\lambda)$ projiziert. Auch hier kann die Intensität mittels Helligkeitsregler verändert werden. Nur wird diese bei der Formel mit einem negativen Vorzeichen berücksichtigt, also beispielsweise: $s(500) = b_{10} \cdot s(436) - r_{10} \cdot s(700) + g_{10} \cdot s(546)$. Dieses Vorgehen mutet zu Recht etwas befremdlich an, aber wie wir später sehen werden fußt darauf die ganze Farbmetrik nach CIE. Es scheint also zu funktionieren, verursacht aber wahrscheinlich Probleme, die später noch zu erörtern sein werden.

Abbildung 6 liefert offensichtlich noch keine gute Beschreibung für das menschliche Farbsehen. An den Enden des Spektrums bei 380 nm und bei 780 nm würde so ein strahlendes Blau oder strahlendes Rot leuchten. Das widerspricht Messungen zur Helligkeitswahrnehmung von reinen Spektralfarben, die schon aus dem Jahr 1924 stammen¹⁶, wonach die Helligkeitswahrnehmung zu den Enden des Spektrums hin immer geringer, mithin die Farben immer dunkler werden. Diese CIE1924 $V(\lambda)$ -Kurve, welche die spektrale Hellempfindlichkeit des menschlichen Auges beschreibt, spielt für die weiteren Überlegungen noch eine große Rolle, weshalb sie in Abbildung 8 wiedergegeben ist.

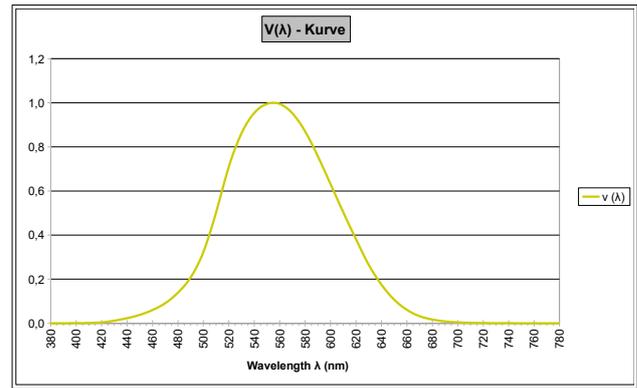


Abbildung 8: CIE1924 $V(\lambda)$ -Kurve, eigener Graf mit Zahlenwerten aus [Wyszecki 1982]

Die Messungen von Guild und Wright enden in etwa bei Abbildung 6. Um nun die Messungen mit der CIE1924 $V(\lambda)$ -Kurve in Einklang zu bringen, wurde diese mit den jeweiligen $r(\lambda)$, $g(\lambda)$ und $b(\lambda)$ im mathematischen Sinne gefaltet. Das geschieht so, dass bei einem bestimmten λ die jeweiligen Werte miteinander multipliziert werden, also $R(\lambda) = r(\lambda) \cdot V(\lambda)$, oder konkreter beispielsweise: $R(550) = r(550) \cdot V(550)$. Wobei in Abbildung 9 auffällt, dass die Blaukurve in Relation zu den beiden anderen sehr geringe Ausschläge zeigt.

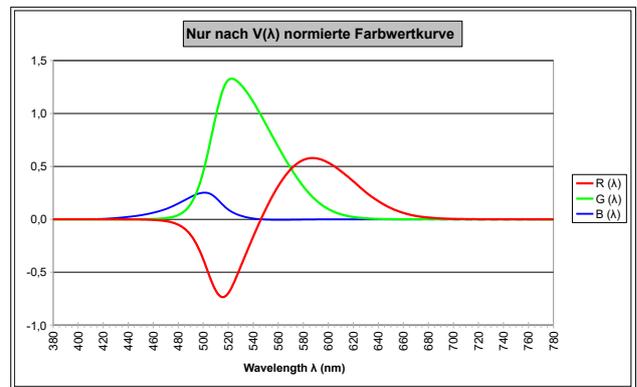


Abbildung 9: Ergebnisse von Guild und Wright nur verrechnet mit CIE1924 $V(\lambda)$ -Kurve, eigener Graf mit Zahlenwerten aus [Wyszecki 1982]

Um trotzdem ähnlichere Größenordnungen zu erhalten, werden alle Werte mittels einer etwas abstrakten Formel normiert. Diese lautet abweichend zur leicht nachvollziehbaren Formel von oben: $R(\lambda) = r(\lambda) \cdot V(\lambda) : (r(\lambda) \cdot 1,00002 + g(\lambda) \cdot 4,59069 +$

$b(\lambda) \cdot 0,06007$). Es ist durchaus erhellend, diese eigentlich sehr abstrakten Normierungsfaktoren einmal darzustellen, wie das in Abbildung 10 mit dem grünen Grafen gezeigt wird. Der orangefarbene Graf stellt die Filterfunktion der Linse und des Glaskörpers im Auge dar.¹⁷ Man sieht, die beiden Grafen haben nur sehr entfernt miteinander zu tun. Ob die Ähnlichkeit beabsichtigt oder nur zufällig ist, kann nach so vielen Jahren leider nicht entschieden werden. Näheres dazu siehe Ende Kapitel 3.

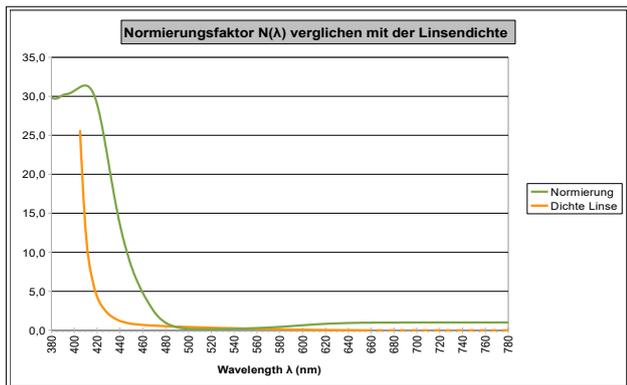


Abbildung 10: Vorab-Normierung $N(\lambda)$ vor Faltung von $r(\lambda)$, $g(\lambda)$ und $b(\lambda)$ mit CIE1924 $V(\lambda)$ -Kurve, eigener Graf mit Zahlenwerten aus [Wyszecki 1982] im Vergleich mit Linsendichte gemäß Stockman [cvrl], „Lens density spectra“

Im weiteren soll die grüne Kurve in Abbildung 10 mit $N(\lambda)$ bezeichnet werden. Es gilt also: $N(\lambda) = 1 : (r(\lambda) \cdot 1,00002 + g(\lambda) \cdot 4,59069 + b(\lambda) \cdot 0,06007)$. Mit dieser Normierung werden, wie klar ersichtlich die in Abbildung 9 kaum mehr vorhandenen $b(\lambda)$ -Werte wieder deutlich angehoben. Zudem erhält man, was Ziel dieser abstrakten Normierung war, folgenden Zusammenhang: $V(\lambda) = R(\lambda) \cdot 1,00002 + G(\lambda) \cdot 4,59069 + B(\lambda) \cdot 0,06007$. Wenn man also die gewonnenen Farbempfindlichkeitsfunktionen $R(\lambda)$, $G(\lambda)$ und $B(\lambda)$ unter Berücksichtigung bestimmter Faktoren verknüpft, erhält man die genormte Helligkeitskurve $V(\lambda)$. Man sollte sich dabei aber im Klaren sein, dass die Normierung und die konstanten Faktoren 1,00002, 4,59069 und 0,06007 mehr oder weniger willkürlich gewählt sind. Die Faktoren bewirken, dass die Flächen unter den drei Grafen $R(\lambda)$, $G(\lambda)$ und $B(\lambda)$, man könnte etwas unwissenschaftlicher sagen, die bestimmten Integrale, gleich sind. Die

endgültigen Farbempfindlichkeitsfunktionen $R(\lambda)$, $G(\lambda)$ und $B(\lambda)$ erhalten wir also mit der modifizierten Formel: $R(\lambda) = r(\lambda) \cdot V(\lambda) \cdot N(\lambda)$; $G(\lambda) = g(\lambda) \cdot V(\lambda) \cdot N(\lambda)$; $B(\lambda) = b(\lambda) \cdot V(\lambda) \cdot N(\lambda)$; zu sehen in Abbildung 11.

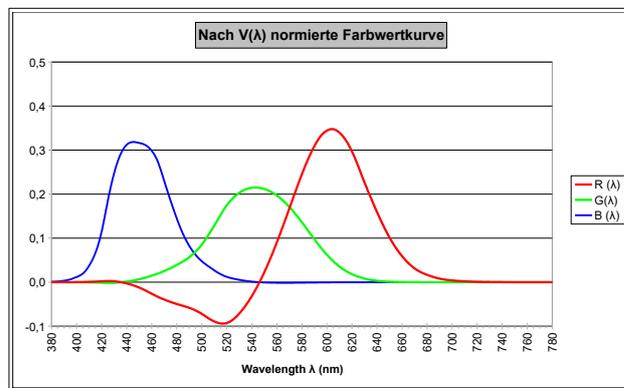


Abbildung 11: Faltung von $r(\lambda)$, $g(\lambda)$ und $b(\lambda)$ mit CIE1924 $V(\lambda)$ -Kurve mit Normierung $N(\lambda)$, eigener Graf mit Zahlenwerten aus [Wyszecki 1982]

Um nun noch die negativen Farbwerte zu beseitigen wurden die kurven $R(\lambda)$, $G(\lambda)$ und $B(\lambda)$ miteinander linear kombiniert und zwar nach der Formel:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.768831 & 1.751709 & 1.130135 \\ 1.000019 & 4.590609 & 0.060067 \\ 0.000000 & 0.056507 & 5.594169 \end{bmatrix} \begin{bmatrix} R(\lambda) \\ G(\lambda) \\ B(\lambda) \end{bmatrix}$$

Die Zahlen in dieser Matrix sind notwendig zur Errechnung des 2°-Beobachters. Diese Zahlen sind mehr oder weniger willkürlich gewählt und sorgen nicht nur für das Verschwinden negativer Farbwerte, sondern auch dafür, dass die aufsummierten Werte in 10 nm-Schritten immer die gleiche Summe ergeben. Man könnte auch sagen, dass das Integral unter den Grafen jeweils die gleiche Summe ergibt. Als Grafik erhält man die grundlegenden Normvalenten für den CIE-XYZ-Farbraum (2°-Normalbeobachter) Abbildung 12:

17 [cvrl] dort unter dem Begriff „Lens density spectra“ zu finden

18 [Fairman 1997], Seite 18, wird laut Wikipedia englisch „CIE 1931 color space“, 11. Januar 2020 auch vom CIE-Standard so definiert

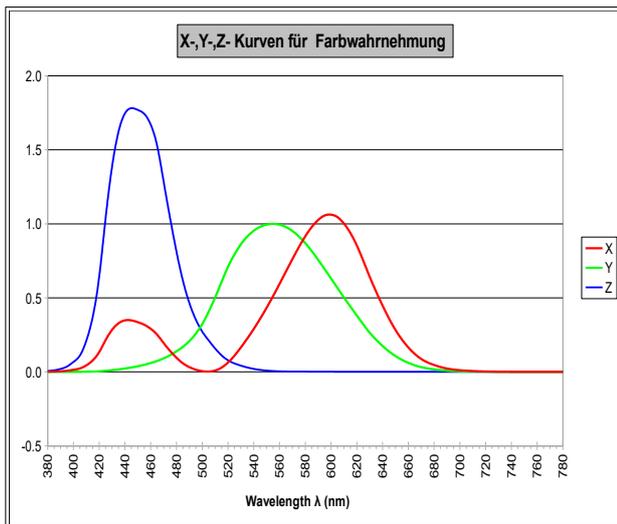


Abbildung 12: Aus der Abbildung 11 mit der Formel oben errechnete Werte für CIE-XYZ

The CIE 1931 2-deg CMFs (CIE, 1932), which form the basis for most practical colorimetry, are based on the chromaticity coordinates obtained by Guild (1931) and by Wright (1928). Chromaticity coordinates, however, provide only a relative measure of the ratios of the three primaries needed to match each spectrum color, whereas CMFs specify absolute energy values. In order to reconstruct the CMFs from the Wright and Guild data, it was assumed that the CIE1924 $V(\lambda)$ photopic luminosity function (CIE, 1926) is a linear combination of the three CMFs (see Wyszecki & Stiles, 1982), for a description of the reconstruction and for the tabulated values. It has long been clear that the CIE1924 $V(\lambda)$ that was used to construct the CIE 1931 2-deg CMFs seriously underestimates sensitivity at wavelengths below 460 nm, so that these CMFs are seriously in error at short wavelengths.¹⁹

Übersetzt: Die Daten der CIE von 1931 für die spektralen Farbempfindlichkeiten für den 2-Grad-Beobachter, bilden die übliche Grundlage für die angewandte Farbmetrik. Sie basieren auf den Farbkoordinaten, die von Guild (1931) und Wright (1928) gewonnen wurden, die aber nur eine relative Messung des Verhältnisses der notwendigen drei Grundfarben beschreiben, um jede reine Spektralfarbe darstellen zu können. Wohingegen Farbempfindlichkeitsfunktionen (CMF) absolute Werte für die Reizenergie liefern müssten. Um nun von den Daten von Wright und Guild auf die Farbempfind-

lichkeitsfunktionen schließen zu können, wurde angenommen, dass die spektrale Helligkeitsfunktion $V(\lambda)$, der CIE von 1924 eine Linearkombination der drei Farbempfindlichkeitsfunktionen (CMF) ist. Für eine genaue Beschreibung des Verfahrens und die tabellarischen Daten siehe das Buch von Wyszecki und Stiles von 1982 [Wyszecki 1982]. Es ist schon seit langem klar, dass die CIE1924- $V(\lambda)$ -Funktion, mit Hilfe derer die CIE 1931 2-Grad-Beobachter-Farbempfindlichkeitsfunktionen gewonnen wurden, die niedrigen Wellenlängen unter 460 nm stark unterschätzen, was zu Fehlern bei den kurzen Wellenlängen führt.

¹⁹ [CVRL], dort im Menü „Older CIE Standards“, dort „Colour matching functions“, dort der zugehörige Informationstext

3 Die Wright-Versuche und die Physiologie

Die Ergebnisse von Wright scheinen in Stein gemeißelt zu sein. Vom einfachsten Lehrbuch für Auszubildende bis zu modernsten Forschungen, werden die dort gewonnenen Normvalenzen allenfalls ein wenig modifiziert, aber niemals grundsätzlich überprüft. Das mag daran liegen, dass die Herleitung von den experimentell gewonnenen Ergebnissen hin zu den fertigen Normvalenzen mathematisch unbestreitbar ist, aber es werden einige Annahmen stillschweigend gemacht und letztlich nie wirklich begründet. Diese ergeben zwar schöne Symmetrien in den Zahlenwerten, haben aber keine physiologische Begründung. Eine herausragende Schlüsselrolle spielt die 1923 gewonnene $V(\lambda)$ -Kurve²⁰. Dass diese keineswegs unstrittig ist, zeigt Abbildung 13.

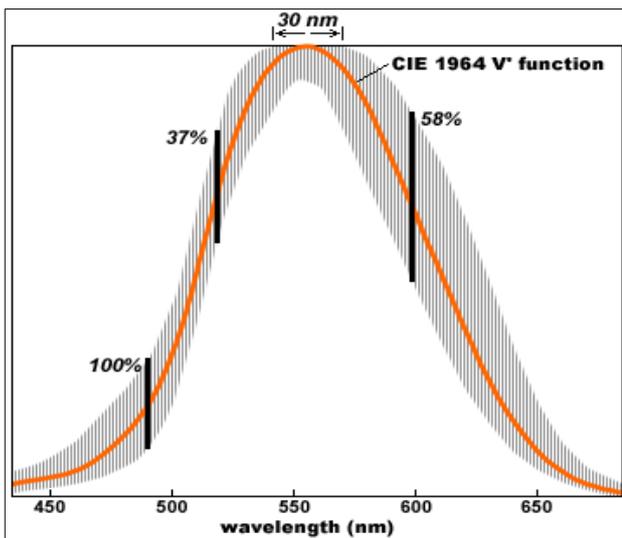


Abbildung 13: individual variations in photopic sensitivity results for 52 individuals, based on heterochromatic step by step brightness matching; from Gibson & Tyndall (1923) aus [MacEvoy 2015]

Diese wird aber von Wright als unumstößlich bewertet und in den entsprechenden Veröffentlichungen sogar bis auf die achte Nachkommastelle angegeben, was die Datenlage doch um einiges überstrapaziert. Zugegebenermaßen waren aber beispielsweise die tatsächlichen spektralen Empfindlichkeiten

ten der Zapfen im Auge, wie in Abbildung 16 gezeigt, seinerzeit noch nicht bekannt. Sie resultieren tatsächlich aus Messungen an Versuchspersonen. Die Datenlage wird auf der Website des Colour & Vision Research laboratory wie folgt beschrieben: Microspectrophotometric (MSP) results from the eyes of seven persons. Four kinds of spectra were obtained with absorbance peaks at the following wavelengths: rods, 496.3 \pm 2.3 nm (n=39); red cones, 558.4 \pm 5.2 nm (n=58); green cones, 530.8 \pm 3.5 nm (n=45); and blue cones 419.0 \pm 3.6 nm (n=5).²¹ Die Maxima der Absorptionskurven weichen doch erheblich von denen bei Wright bzw. den späteren Überlegungen der CIE ab. Ein Ähnliches Bild ergibt sich, wenn wir die Abbildung 11 betrachten im Vergleich zur Abbildung 14. Diese Abbildung entstammt einer neueren Untersuchung aus dem Jahr 2016²², die rechnerisch die Abweichungen des Farbsehens bei unterschiedlichen Beobachtern nachzuvollziehen versucht. Die rechnerischen Ergebnisse sind dargestellt und stimmen recht gut überein mit den gemessenen Streubreiten wie man sie bei Wyszecki & Stiles nachlesen kann²³.

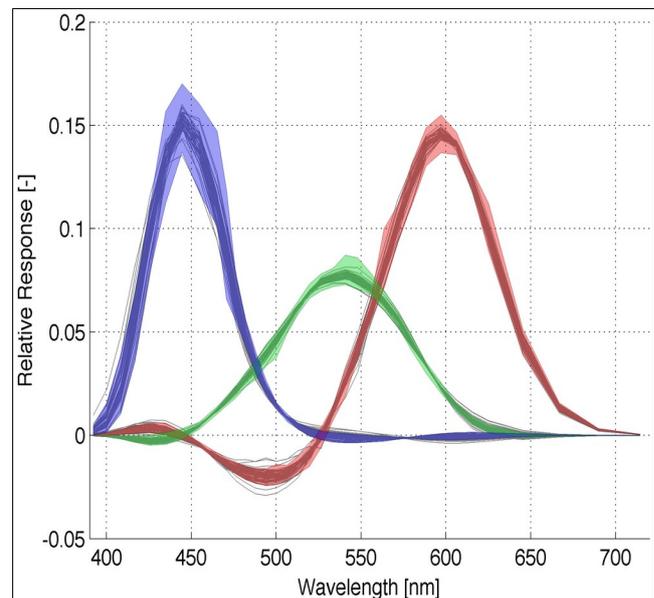


Abbildung 14: Aus [Asano 2015], Seite 43

20 Im Deutschen gesprochen: Vau-Lambda

21 Data from Table 2 of Dartnall, H. J. A., Bowmaker, J. K., & Mollon, J. D. (1983). Human visual pigments: microspectrophotometric results from the eyes of seven persons. Proceedings of the Royal Society of London, B 220, 115-130., Gefunden auf [cvrl]

22 [Fairchild 2016]

23 [Wyszecki 1982], Seiten 343 und 345

Auch die neueren Überlegungen der CIE von 2006, die Maxima der Absorptionskurven zu berücksichtigen, fußen letztlich wieder auf den Normvalenzen von Wright, aus denen sie per Lineartransformation abgeleitet werden. Sie sind in Abbildung 15 dargestellt.

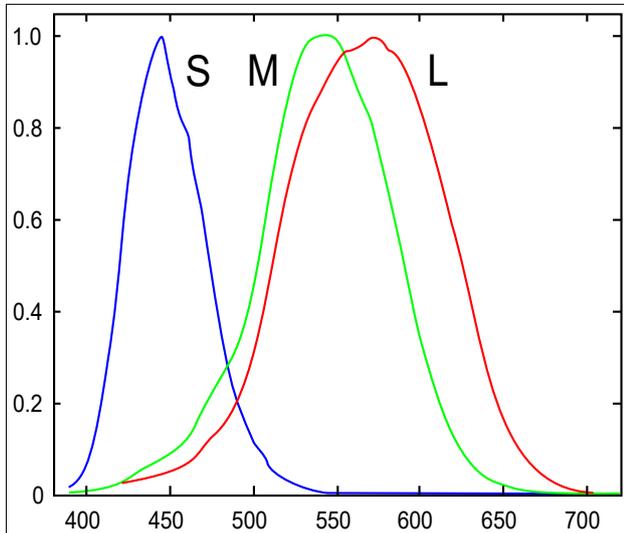


Abbildung 15: The normalized spectral sensitivity of human cone cells of short-, middle- and long-wavelength types. Wikipedia englisch „CIE 1931 color space“ basierend auf Daten aus [Stockman 1993], Seite 2516ff

Diese normierten spektralen Empfindlichkeiten der menschlichen Zapfen werden aber letztlich auch wieder aus den CIE-X, -Y und -Z abgeleitet. In der Form

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad 24$$

Die Parameter a1 bis c3 werden mehr oder weniger gut empirisch ermittelt. Beim Vergleich dieser speziellen LMS-Spektralkurven mit den Absorptionsspektren der Fotorezeptoren fällt auf, dass die Maxima der Absorptionskurven für die Rot- oder L-Zapfen und der Grün- oder M-Zapfen recht gut übereinstimmen, während die Maxima für die Blau- oder S-Zapfen um etwa 25 nm auseinander liegen. Man erkennt das beim Vergleich von Abbildung 15 mit Abbildung 16. Durch Linearkombinationen lässt sich anscheinend keine gute Übereinstimmung mit Empfindlichkeitskurven der Farbzeptoren erreichen.

24 [Wikipedia en], „CIE 1931 color space“, 24. Januar 2020

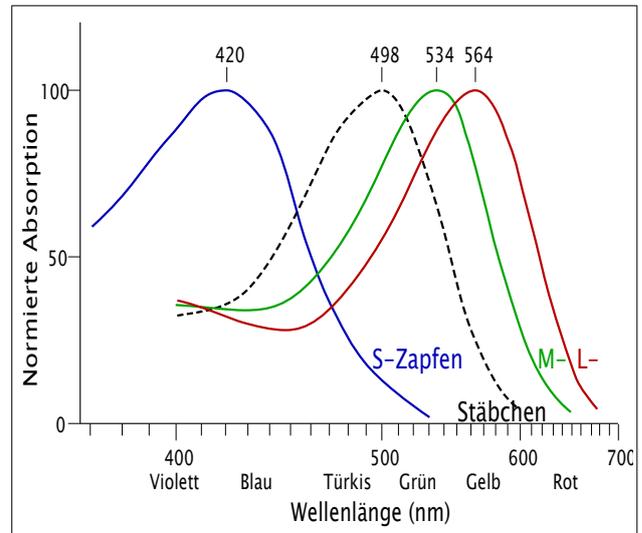


Abbildung 16: Spektrale Absorptionskurven der verschiedenen Sehpigmente der Fotorezeptoren. Zur besseren Vergleichbarkeit wurden die Maxima auf 100 Prozent gesetzt. Wikipedia „Zapfen (Auge) vom 6.7.2019“ nach [Dartnall 1980], Seite 505

Die Normierungskurve $N(\lambda)$ von oben, die eigentlich nur rechnerisch zu begründen ist, damit die Blau-Rezeptoren $b(\lambda)$ nicht unterbewertet werden, wie sie in Abbildung 10 dargestellt ist, hat erstaunliche Ähnlichkeit mit Kurven, welche die Optische Dichte der Linsen im menschlichen Auge beschreiben. Ergebnisse einer entsprechenden physiologischen Untersuchung sind in Abbildung 17 gezeigt²⁵.

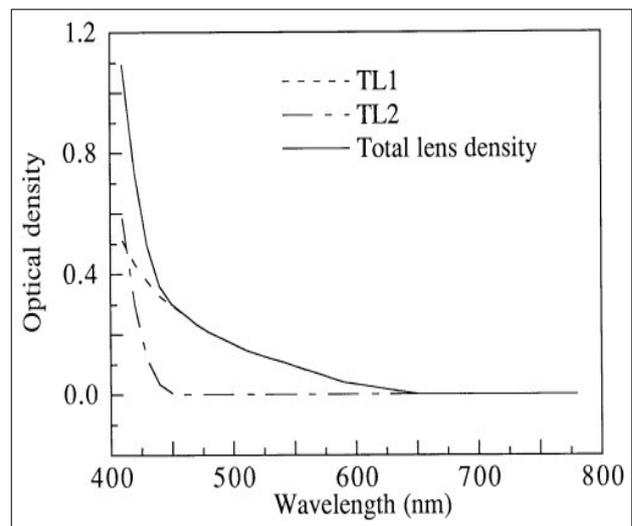


Abbildung 17: [Xu 1997], Seite 955

25 [cvrl] dort unter dem Begriff „Lens density spectra“ zu finden

Die mehr rechnerisch ermittelten Tabellenwerte des Colour & Vision Research laboratory sind ebenfalls nur logarithmisch angegeben und mussten noch entsprechend umgeformt werden, um sie mit $N(\lambda)$ vergleichen zu können. Das Ergebnis ist nochmals in Abbildung 18 gezeigt.

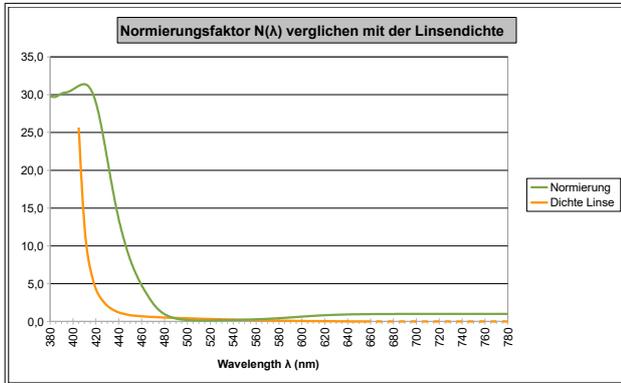


Abbildung 18: Vorab-Normierung $N(\lambda)$ vor Faltung von $r(\lambda)$, $g(\lambda)$ und $b(\lambda)$ mit CIE1924 $V(\lambda)$ -Kurve, eigener Graf mit Zahlenwerten aus [Wyszecki 1982] im Vergleich mit Linsendichte gemäß Stockman [cvrl], „Lens density spectra“

4 Kritik an der Linearisierung des Farbreizes

4.1 Bereits verwendete nicht-lineare Farbreize

Die Übertragungskennlinie ergibt sich gemäß der Definition von CIE-Lab nach der Formel²⁶:

$$y = 1,16 \cdot \sqrt[3]{\frac{x}{100}} - 0,16$$

Die Abbildung 20 zeigt Ellipsen innerhalb derer Farben als nicht unterschiedlich wahrgenommen werden. Diese experimentell und rechnerisch er-

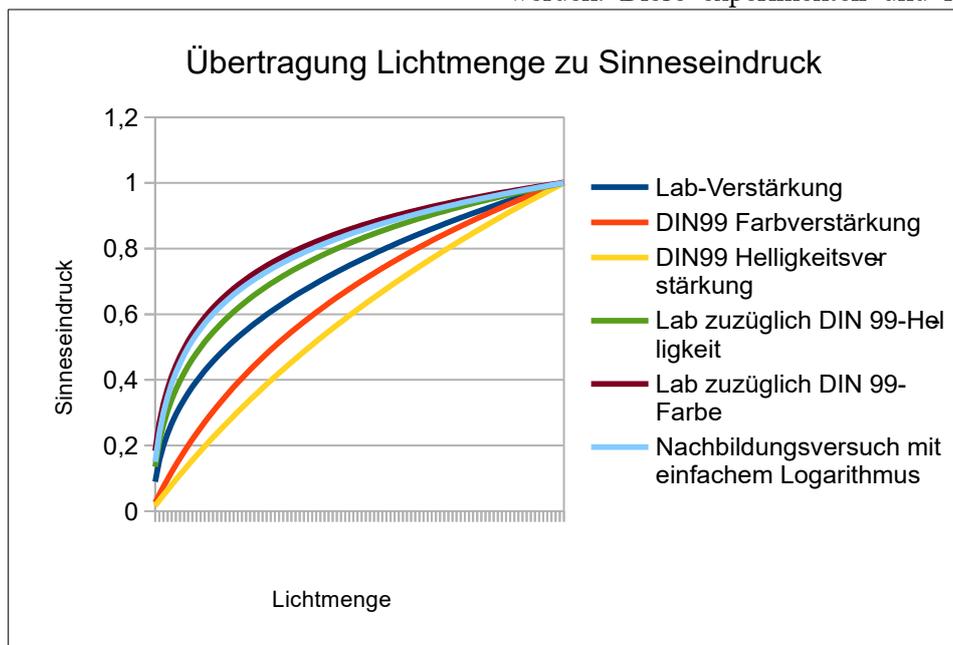


Abbildung 19: Unterschiedliche Kurven, die Lichtmenge zu Sinneseindruck oder Reiz zu Reizreaktion beschreiben.

Schon 1976 wurde mit der Einführung des CIE-Lab-Farbraums ein erster Schritt in Richtung gleichabständige Wahrnehmung gemacht. Dabei wurde anders als im CIE-XYZ-Farbraum davon ausgegangen, dass die Lichtenergie eines Reizes und die Stärke des resultierenden Sinneseindrucks nicht in einem linearen Zusammenhang stehen, sondern kleine Lichtmengen vergleichsweise starke Sinneseindrücke erzeugen, während eine weitere Erhöhung der Lichtmenge den Sinneseindruck nur noch vergleichsweise wenig ändern kann. In der Abbildung 19 ist die Übertragungskennlinie für CIE-Lab in der Mitte Dunkelblau dargestellt. Dabei ist die Lichtmenge in Prozentzahlen von 1 bis 100 und die Stärke des Sinneseindrucks von 0 bis 1 angegeben.

mittelten Ellipsen sind um so kleiner je näher sie der Grauchse in der Mitte des Koordinatensystems von Lab liegen. Dort ist also das Farbsehen empfindlicher als in den Bereichen mit gesättigteren Farben.

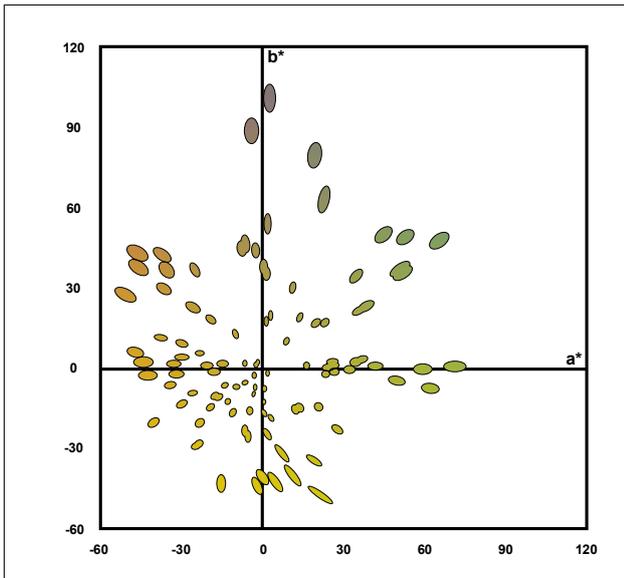


Abbildung 20: MacAdam und andere Ellipsen im CIE-Lab-Farbraum dargestellt aus [Luo 2000], Seite 341

Daraus ergibt sich, dass die Farben in der Nähe der Grauchse noch stärker gewichtet werden müssen, als es der CIE-Lab-Farbraum ohnehin schon tut. Um diese Schwäche auszugleichen, gibt es mehrere Vorschläge. Gegenüber den meisten anderen hat die sogenannte DIN99-Farbabstandsformel den entscheidenden Vorzug, durch mathematische Verzerrung einen neuen Farbraum zu erzeugen. Dort werden die Farbkoordinaten des CIE-Lab-Farbraumes nochmals umgerechnet und zwar nach der Formel unter der Annahme, dass nur Werte zwischen 0 und 1 entstehen dürfen²⁷:

$$y = \frac{1}{1,7} \cdot \ln(1 + 0,045 \cdot x)$$

Die Entsprechende Übertragungskennlinie findet sich in Abbildung 19 orange, die zweite Linie von unten. Die Helligkeit wird für DIN99 etwas anders errechnet, nämlich nach der Formel²⁸:

$$y = 105,51 \cdot \ln(1 + 0,0158 \cdot x)$$

Man sieht, dass die Helligkeiten nicht ganz so stark angehoben werden müssen wie die Farben. Nun haben wir also zwei Übertragungskennlinien. Zum ersten diejenige durch den CIE-Lab-Farbraum zum zweiten diejenige durch DIN99. Wenn man entspre-

chend kombiniert erhält man in der Abbildung 19 die oberste Kurve in Braun und die dritte von oben in grün.

4.2 Vorschlag zur Vereinfachung bisheriger Formeln

Diese relativ komplizierten Formeln lassen sich zu einem einfachen logarithmischen Zusammenhang verrechnen, wie er in der zweitobersten, hellblauen Übertragungskennlinie in Abbildung 19 abgebildet ist:

$$y = \frac{1}{4,615} \cdot \ln(1 + x)$$

Die Formel ist deshalb so einfach, weil als Eingabe Prozentzahlen von 0 bis 100 fungieren. Die Werte müssten also eigentlich durch 100 geteilt werden. Diese Formel liegt sehr nahe am Weber-Fechner'schen Gesetz, in dem schon 1860 der Zusammenhang zwischen Reizstärke R und Stärke der Empfindung E beschrieben wurde²⁹:

$$E = c \cdot \ln\left(\frac{R}{R_0}\right)$$

c ist eine Konstante

R ist der Reiz

R₀ ist die Reizschwelle

Dabei ist R₀ eine Reizschwelle unterhalb der kein Sinneseindruck erzeugt wird. In der Formel, die in Abbildung 19 verwendet wurde, müsste x durch 100 geteilt werden. Um dies auszugleichen müsste diese 100 mit 0,01, also einem Prozent, multipliziert werden. Der Zusammenhang sei hier kurz dargestellt:

$$y = \frac{1}{4,615} \cdot \ln\left(1 + \frac{x}{100 \cdot 0,01}\right)$$

für x angegeben in %

Wir nehmen also nach dem Weber-Fechner'schen Gesetz eine Reizschwelle von einem Prozent an. Wenn man in einem Druckprodukt eine schwarze Fläche druckt und daneben eine, die nur 99% Schwarz ist, wird man keinen Unterschied bemerken. Wenn man den Lichtfang mit einbezieht, ist eine 97%-Schwärzung immer noch nicht erkennbar.

27 [Wikipedia] „DIN99-Farbraum“, 24. Januar 2020

28 [Wikipedia] „DIN99-Farbraum“, 24. Januar 2020

29 [Wikipedia] „Weber-Fechner-Gesetz“, 24. Januar 2020

Hier handelt es sich um ein Kalkül, das man auch anders annehmen kann. Für die grundsätzlichen Überlegungen ist diese Annahme erst einmal sinnvoll, weil sie die Formel etwas vereinfacht, ohne ihren grundsätzlichen Charakter zu verändern. In den oben verwendeten Formeln wird innerhalb des Logarithmus jeweils noch ein „+1“ hinzu gefügt. Dies widerspricht oberflächlich betrachtet zunächst dem Weber-Fechner'schen Gesetz. Dieses geht in der oben dargestellten Form sinnvollerweise davon aus, dass der Reiz über der Reizschwelle liegen muss, der Bruch im Logarithmus somit stets größer 1 wird. Sollte er kleiner 1 sein, entstünden nämlich negative Werte für den Sinneseindruck. Was soll aber ein negativer Sinneseindruck sein? Man kann einen stärkeren oder einen schwächeren Sinneseindruck generieren, aber keinen negativen. Auch eine Nervenzelle kann als Reizantwort ein schwaches oder stärkeres Signal geben, oder auch gar keines, aber kein negatives. Nach neueren Forschungen kann aber tatsächlich das Fehlen von Licht bei bestimmten Sehzellen einen Impuls auslösen.³⁰ Dies wird in den bisherigen Farbsystemen noch nicht berücksichtigt. Da in der Farbmeterik für den Reiz üblicherweise mit Werten zwischen 0 und 1 gearbeitet wird, ist das Hinzufügen des Terms „+1“ notwendig, um sinnvolle Werte zu erhalten. Für einen Farbraum, der im Ergebnis gleiche Abstände für gleiche Wahrnehmungsunterschiede liefern soll, müsste eine entsprechende Übertragungskennlinie schon bei der Ermittlung der Grundvalenzen mit einbezogen werden. Dies ist bei der Ermittlung der Normalvalenzen für CIE-XYZ nicht geschehen.

4.3 Linear oder nicht-linear?

Oben wurde beschrieben, wie nicht-lineare Zusammenhänge bei der Farbmeterik zumindest im CIE-Lab-Farbraum verwendet werden. Nun geht aber die Farbmeterik schon bei ihren Experimenten bis hin zum CIE-XYZ-Farbraum von ausschließlich linearen Zusammenhängen von Reiz und Sinneseindruck aus. Bei dieser Annahme entspricht dem Reiz ein Sinneseindruck multipliziert mit einem festen Faktor. Auch in Fachdiskussionen entsteht hier ein anscheinend unüberbrückbarer Widerspruch zwischen linear und nicht-linear. Die entscheidenden Versuche von Wright wurden mit sehr schmalbandi-

gen Spektren durchgeführt. Die Versuche von Mac-Adam wurden hauptsächlich mit Filtern starker Sättigung und begrenzt auf kleine Spektralbereiche durchgeführt. Insbesondere bei ganz entscheidenden Versuchen von Wright wurden also nur sehr kleine Bereiche der jeweiligen Rezeptoren angesprochen. Wenn wir davon ausgehen, dass die schmalbandigen Spektren tatsächlich nur etwa 10 nm breit waren, dann werden selbst in günstigsten Kombinationen die Rezeptoren zu maximal 25% angesprochen, in den weitaus meisten Fällen viel weniger. Es ist deshalb durchaus möglich, dass in diesen Bereichen die Reizantwort annähernd linear ausfällt. Wenn wir die Abbildungen 19 aus Kapitel 4.1 und 37 aus Kapitel 8 betrachten, sehen wir deutlich, dass die meisten Übertragungskurven bei kleineren Werten sich annähernd linear verhalten. Die üblichen Reiz-Reaktions-Kurven künstlicher neuronaler Netze nutzen oft bewusst Kurven, die sich in den Bereichen mit kleinen Eingangswerten annähernd linear verhalten, dann langsamer ansteigen und bei sehr starken Reizen einer Sättigung zustreben.³¹ Diese sogenannte Sigmoidfunktion soll in Kapitel 8 noch genauer beschrieben werden.

30 [DfwG-Report], P.Bodrogi/T.Q. Khanh, 2017/3, Seite 23

31 [Lippe 2005], Seite 50

5 Vereinfachte Anwendung linearisierter Reize

Wright hat für die Berechnung des CIE-XYZ-Normalbeobachters konsequent nur lineare Transformationen durchgeführt. Das Wright'sche Grundexperiment wurde bereits in Kapitel 2 ausführlich erklärt. Im einfachsten Fall könnte man vereinfachend die in Kapitel 4 gewonnene Übertragungskennlinie auf alle Reize anwenden, egal von welchem Lichtwellenbereich sie stammen. In Abbildung 21 wird nicht wie bei Wright linear transformiert, sondern auf die Kurven in Abbildung 11 wird die erste Formel aus Kapitel 4.2 angewendet.

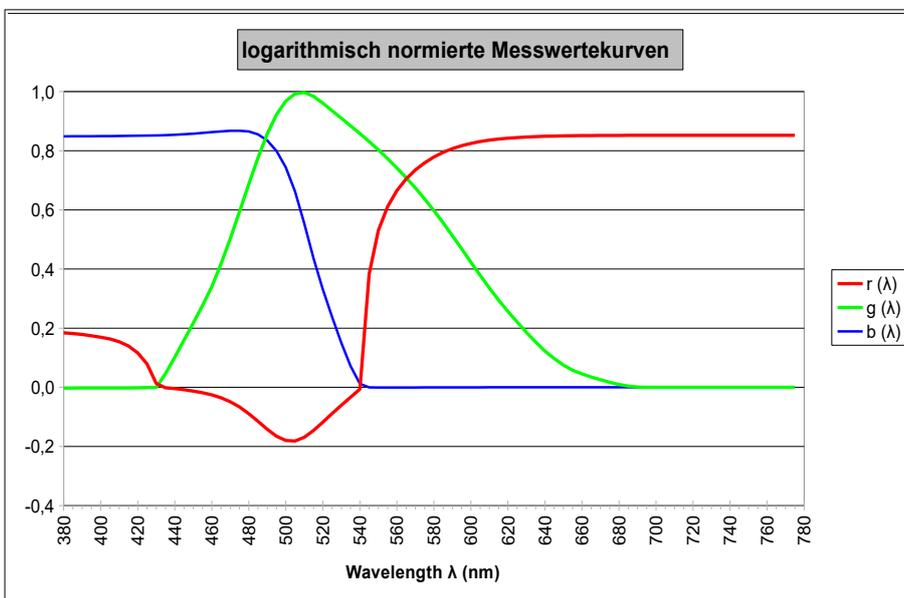


Abbildung 21: Messwertkurven logarithmisch umgerechnet unter Abzug der Referenzfarbe, die bei den Experimenten in den „negativen“, üblicherweise virtuellen Farben mit vorhanden sind.

Problem sind die „negativen“ Farben. Sie müssen so behandelt werden, dass bereits der Reiz der Referenzfarbe ausgeübt wird, und die „negative“ Farbe nur noch hinzu kommt. Es ist also nur die Veränderung eines Sinneseindrucks zu berücksichtigen, die wesentlich kleiner ist, als wenn der Reiz der „negativen“ Farbe alleine ausgeübt würde. An den Graphen ist dann aber deutlich zu erkennen, dass Knickstellen entstehen, die sich leider durch alle Lineartransformationen fortpflanzen, wie in Abbildung 22 gezeigt.

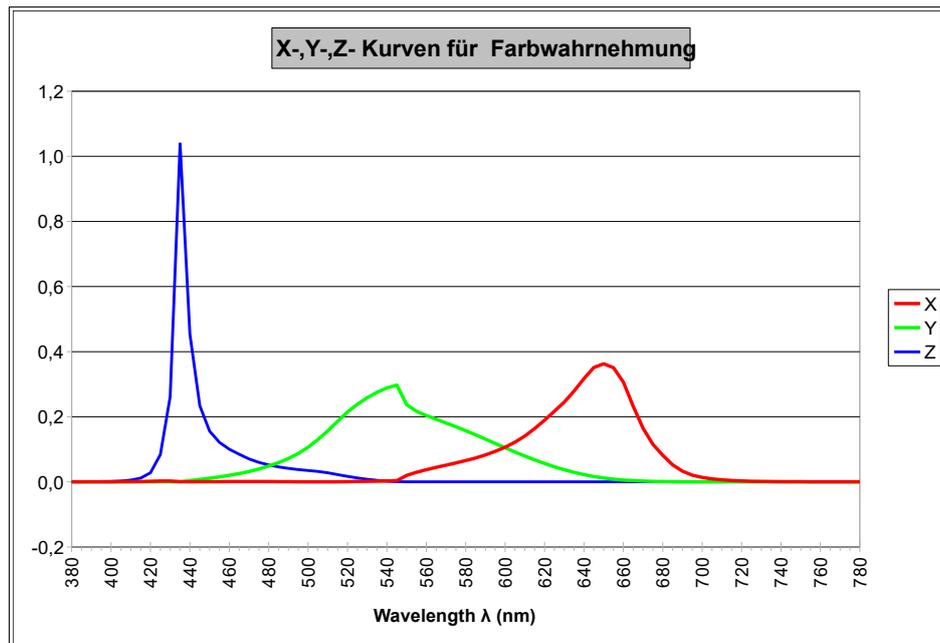


Abbildung 22: XYZ-Kurven errechnet analog Wright, allerdings auf Grundlage logarithmischer Messwertkurven, wie in Abbildung 21 gezeigt.

Es muss dann natürlich eine andere Lineartransformation als von Wright angewendet werden, wenn die Flächen unter den Grafen untereinander gleich und keine negativen Zahlwerte vorkommen sollen. In diesem Fall ist die entsprechende Matrix folgende:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2,76888 & 1,75174 & 1,13016 \\ 1,00002 & 4,59069 & 0,06007 \\ 0,00000 & 0,05651 & 5,59427 \end{bmatrix} \begin{bmatrix} R(\lambda) \\ G(\lambda) \\ B(\lambda) \end{bmatrix}$$

Diese Matrix ist sehr bewusst an diejenige von Wright angenähert. Das Ergebnis in Abbildung 22 mit seinen beiden Knickstellen sieht nicht sehr überzeugend aus. Auch die Herleitung der CIE-XYZ-Normvalenzen ist, wie in den voran gegangenen Kapiteln 2 und 3 gezeigt, dermaßen unsicher, so dass die Einführung der Nicht-Linearität zwar eventuell eine Verbesserung brächte, aber die Unsicherheit nur noch steigern würde. Am Ende von Kapitel 4.2 wurde bereits festgestellt, dass in den Versuchen von Wright eher schwache Reize verwendet wurden, was zur Folge hat, dass wahrscheinlich noch in Bereichen gearbeitet wurde, wo noch annähernd ein linearer Zusammenhang zwischen Reiz und Reaktion gelten dürfte. Auch die ab Kapitel 4.3 gewonnenen Einsichten lassen vermuten, dass in diesem Fall vereinfachend ein linearer Zusammenhang verwendet werden kann. Deshalb wurden hier keine weiteren Berechnungen unternommen. Mittels neuronaler Netze sollte es zwar

auch hier die Möglichkeit geben, neue Kurven zu ermitteln, dies wäre aber ein eigener Forschungsansatz, der in dieser Arbeit nicht weiter ausgeführt werden soll.

6 MacAdam-Versuche

6.1 Grundsätzliche Bedeutung

David L. MacAdam von den Research Laboratories, Eastman Kodak Company stellte 1942 seine Messergebnisse zu Farbähnlichkeiten im Norm-Diagramm von 1933 vor. Das Ergebnis waren Ellipsen in diesem Diagramm, innerhalb derer normalsichtige Menschen keine Farbunterschiede wahrnehmen, siehe Abbildung 28. Genauer gesagt, wenn der Mittelpunkt der jeweiligen Ellipse als Referenzfarbe dient, werden alle Farben, deren Koordinaten innerhalb der zugehörigen Ellipsen liegen, nicht als verschieden von der Referenzfarbe wahrgenommen. Diese sogenannten MacAdam-Ellipsen stellen die fundamentale Kritik am Normbeobachter von 1931 dar, auch wenn MacAdam selbst ausdrücklich am Normbeobachter nichts geändert wissen wollte: No modification of this standard system of specifying colors will be recommended. Any modification of the chromaticity diagram adequately representing these data on noticeabilities of color differences would be so complicated as to be useless in practice.³² MacAdam ist offensichtlich schon froh, dass er Farbkoordinaten nach Norm-Beobachter angeben kann und nicht mehr die bis dahin sehr ungenaue übliche Angabe von dominanter Wellenlänge und Reinheit angeben muss.³³

Die Untersuchungen von MacAdam werden in neuerer Zeit nicht nur in dieser Arbeit, sondern auch an anderer Stelle immer mehr der Kritik unterzogen. So kritisiert zum Beispiel Kittelmann in seiner Arbeit von 2010: *Die in dieser Untersuchung ermittelten Ellipsen in der Normfarbtafel zeigen teilweise deutlich andere Eigenschaften als die MacAdam-Ellipsen. Die MacAdam-Ellipsen sind am kleinsten im blauen Bereich und am größten im grünen. Hingegen liegt bei dieser Untersuchung das Maximum im gelben Bereich, und gerade im grünen Bereich ist die Ellipse sogar kleiner als in anderen. Auch im blauen Bereich zeigt sich eine stärkere Ausdehnung der Ellipsen. Die Werte um das Mittelgrau haben die geringsten Ellipsengrößen. Auch der maximale Größenunterschied zwischen den Ellipsen ist geringer als bei de-*

*nen nach MacAdam. Ein Unterschied zwischen den MacAdam-Ellipsen und den in dieser Untersuchung bestimmten Ellipsen dürfte dadurch entstehen, dass in dieser Untersuchung die Schwellen an Körperfarben gemessen werden, während MacAdam seine Untersuchungen 1942 an Lichtfarben durchgeführt hat.*³⁴

6.2 Spektrale Betrachtungen

Es ist schwierig, die genaue Emmissionskurve der Lichtquelle herauszufinden, die in der Arbeit von MacAdam mit „Mazda C-8“³⁵ beschrieben wird. Der Kodak Wratten Filter Nummer 78AA soll aber in Verbindung mit Mazda C Lampen, die offensichtlich Tungsten, also Glühlampen sind, Tageslicht erzeugen³⁶. In der Publikation³⁷ wird als Tageslicht-Ziel D50, also Licht mit 5000 K³⁸, angegeben. Dort wird eine Lichtquelle mit 2900 °K verwendet, während MacAdam 2848 K angibt, was aber keine all zu große Abweichung definiert. Die Mazda C Lampen emittieren laut MacAdam annähernd Normlicht A³⁹. Transmissionskurven für die verwendeten Wollaston- und Rochon-Prismen zur Polarisierung und Strahlteilung und -mischung⁴⁰ sind in der Arbeit leider nicht angegeben. Vermutlich wurden um diese Zeit Calcite verwendet⁴¹. Die Größe der Calcite ist in der Beschreibung von MacAdam leider nicht angegeben. Eine Transmissionskurve für Calcite ist in Abbildung 23 dargestellt. Der fast lineare Anstieg der Transmission zwischen 400 und 700 nm dürfte kaum ein Problem darstellen, sehr wohl aber die starke Filterwirkung von 400 bis 500 nm.

32 [MacAdam 1942], Seite 247

33 [MacAdam 1942], Seite 248

34 [Kittelmann 2010], Seite 38

35 [MacAdam 1942], Seite 250

36 [Kodak 1928], Seite 19

37 [Davis 1931], Seite 9f

38 [Davis 1931], Seite 14ff

39 [MacAdam 1942], Seite 251

40 [MacAdam 1942], Seite 250

41 [Wikipedia] „Wollaston-Prisma“, 24. Januar 2020

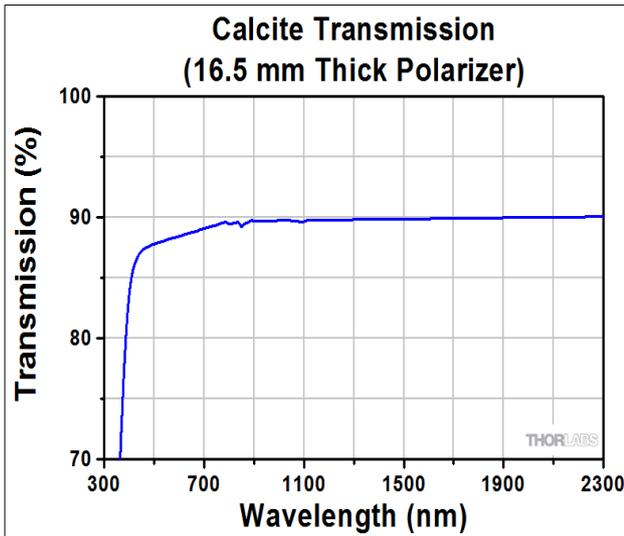


Abbildung 23: Calcite (CaCO_3), or Iceland Spar, is a useful crystalline substrate most commonly used in polarization optics. https://www.thorlabs.com/newgrouppage9.cfm?object-group_id=6973&tabname=Calcite

Die Filter, die MacAdam verwendet, sind einige Spezialfilter und die sogenannten WRATTEN-LIGHT FILTERS, deren optische Eigenschaften in [Kodak 1928] verzeichnet sind. Die Genauigkeit der Angaben ist leider nicht mehr überprüfbar. Für einige Spezialfilter, in der Arbeit von 1942 mit „X“ bezeichnet, liegen keine Transmissionskurven vor.⁴² Es ist prinzipiell nicht klar, wie und ob die Polarisationsfilter anzusetzen sind. Nach einem Schriftwechsel mit Sherman L Guth über researchgate.net, der nach eigenen Angaben ein Student von Wright war, gibt es dazu keine detaillierteren Veröffentlichungen als die Filtertransmissionskurven von Kodak. Versuche, die Ergebnisse auf Spektralniveau nach zu rechnen, waren leider nicht von Erfolg gekrönt. Durch Probieren könnte mit einigem Aufwand der Einsatz der Polfilter und die tatsächliche optische Wirksamkeit der Filter ermittelt werden, was aber immer unter dem Vorbehalt steht, ob die angegebenen XYZ-Werte genau genug sind. Die Probleme sollen an wenigen Grafen verdeutlicht werden. Auf Seite 254 und 255 seiner Arbeit⁴³ hat MacAdam einige wenige spektrale Transmissionskurven abgebildet. Diese sollen denen der Wrattenfilter-Transmissionskurven gegenüber gestellt werden, siehe Abbildungen 24 bis 27.

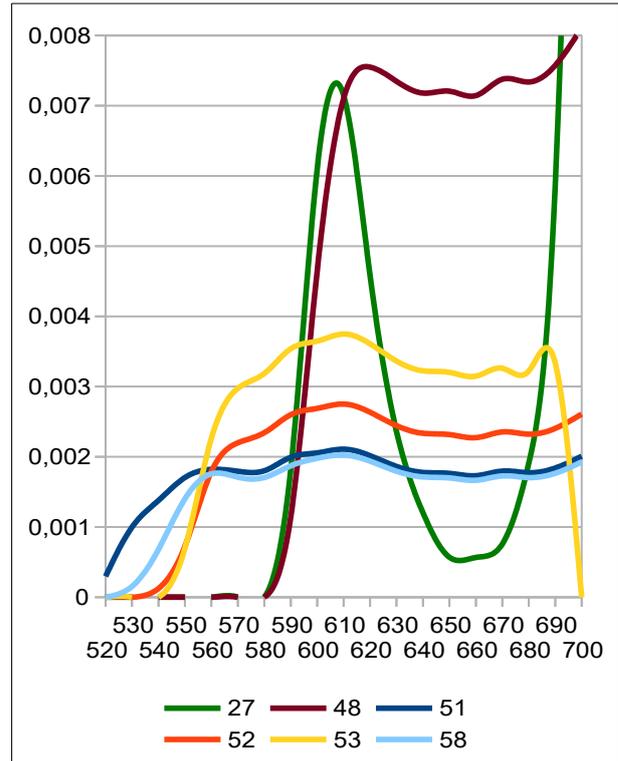


Abbildung 24: Spektrale Transmissionskurven aus [Kodak 1928] gemäß [MacAdam1942] dargestellt

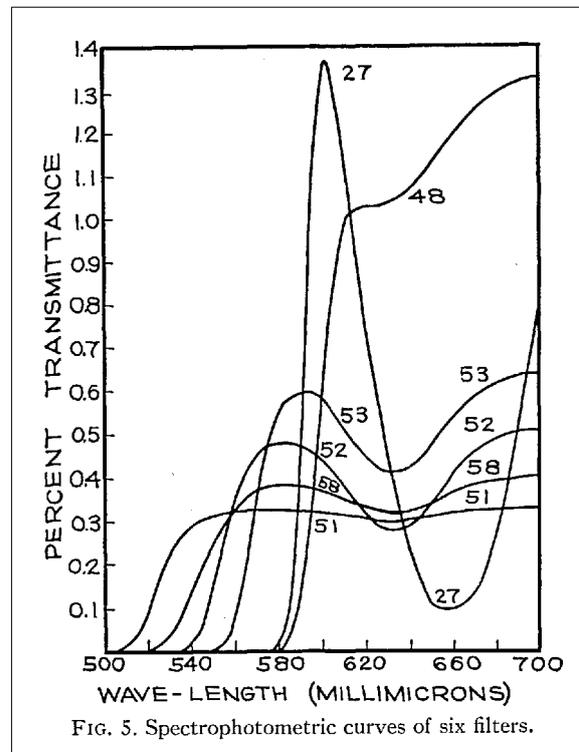


Abbildung 25: Spektrale Transmissionskurven aus [MacAdam1942]

42 [MacAdam 1942], Seite 253 unten.

43 [MacAdam 1942], Seite 254f

Es sind deutliche Ähnlichkeiten sichtbar, aber für fundamentale Messergebnisse sicher unbefriedigend.

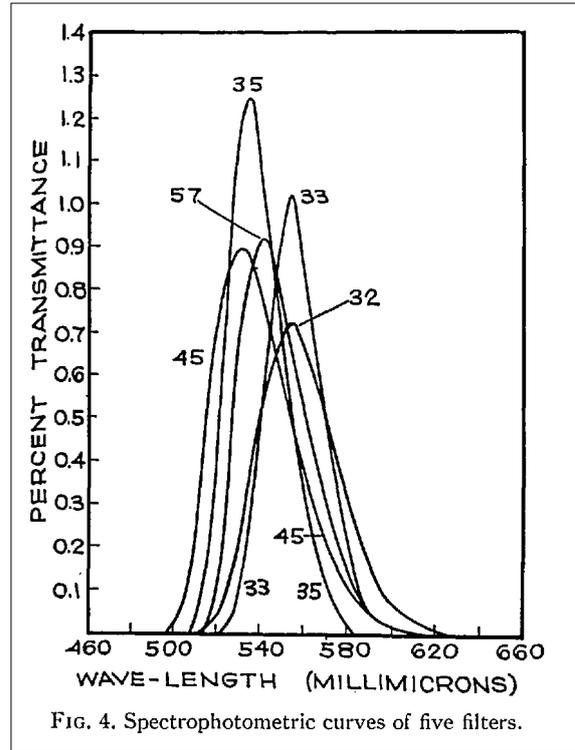


Abbildung 26: Spektrale Transmissionskurven aus [MacAdam1942]

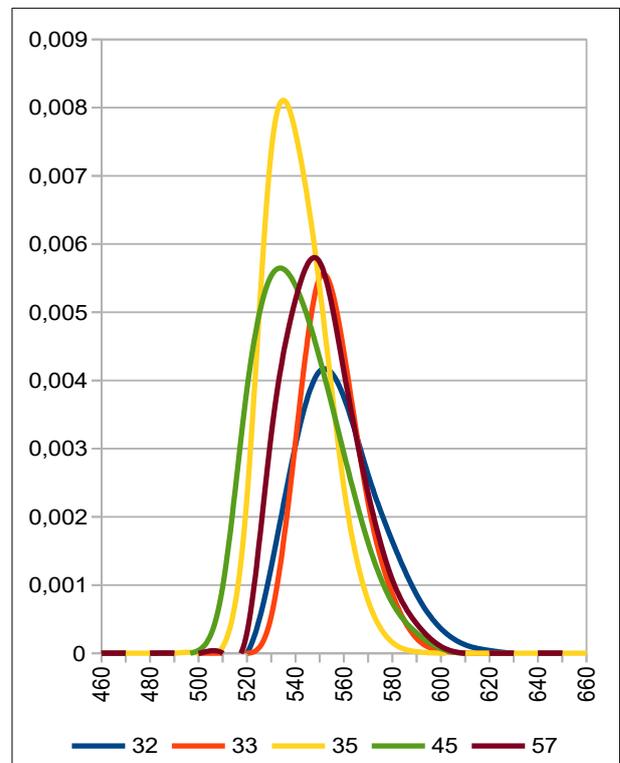


Abbildung 27: Spektrale Transmissionskurven aus [Kodak 1928] gemäß [MacAdam1942] dargestellt

6.3 Helligkeiten im MacAdam-Versuch

In seinem Experiment wird beim Beobachter deutlich zwischen dem eigentlichen Testfeld mit einem Blickwinkel von 2° und der Umgebung von je 21° unterschieden. Das Testfeld wird mit einer Leuchtdichte von 15 millilambert beleuchtet, während die Umgebung mit 7.5 millilambert mit der Normlichtart C beleuchtet wird.⁴⁴ Es wird nicht ausdrücklich erwähnt, aber die Aussage legt nahe, dass das Testfeld mit der doppelten Helligkeit der Umgebung auch mit dem Faktor 2 für alle Filter bewertet werden sollte. Auch dies könnte dazu beitragen, dass die von MacAdam angegebenen CIE-XYZ-Werte⁴⁵ der einzelnen Filter meistens größere Werte haben, als die aus den Wratten-Filtern berechneten XYZ-Werte, die das ganze optische System mit allen Linsen und Polarisationsfiltern unberücksichtigt lassen. Die Y-Werte der verwendeten Filter im CIE-XYZ-Farbraum liegen alle zwischen 0,0025 und 0,00315⁴⁶. Dies einfach unbedacht in CIE-L umgerechnet, ergäbe Werte zwischen 2.2582 und 2.8454. Wenn man die Verdoppelung der Werte, wegen des dunkleren Umgebungslichtes berücksichtigt, bekommt man Werte zwischen 4.5165 und 5.6908. Nach dem derzeitigen Standard für Offsetdruck sollte das Schwarz einen CIE-L-Wert von 16 oder darunter haben. Die unreflektiert aus den Angaben von MacAdam errechneten Werte sind also so dunkel, dass sie völlig sinnlos sind. Wir werden später sehen, dass man aus der Geometrie der Messwerte, in der Nähe des Spektralkurvenzuges und der Purpurgeraden, in anderen Farbsystemen wie CIE-Lab und CIECAM02-UCS auf die vom Beobachter empfundene Helligkeit rückschließen kann. MacAdam verwendet zur Angabe der Leuchtdichte häufig die Einheit Lambert. Lambert ist eine nicht SI-Einheit. 1 lambert (L) = $1/\pi$ candela pro Quadratcentimeter ($0.3183 \text{ cd}/\text{cm}^2$) oder $(10^4/\pi) * \text{cd m}^{-2}$, Quelle:⁴⁷ 15 Millilambert entsprechen $48 \text{ cd}/\text{m}^2$. Erst bei Leuchtdichten oberhalb 3 bis $30 \text{ cd}/\text{m}^2$ tritt photopisches Sehen auf⁴⁸. Insofern wurden die Experimente an

44 [MacAdam 1942], Seite 247

45 [MacAdam 1942], Seite 253

46 [MacAdam 1942], Seite 253

47 [Wikipedia en] „Lambert_(unit)“, 24. Januar 2020

48 [Wikipedia] „Photopisches_und_skotopisches_Sehen“, 24. Januar 2020

der Schwelle zum mesopischen Sehen ausgeführt. Heutige Experimentatoren versuchen peinlich, dies zu vermeiden, obwohl es eigentlich eher der Realität in einem normal beleuchteten Büro entspricht.

6.4 MacAdam nicht nur Ellipsen

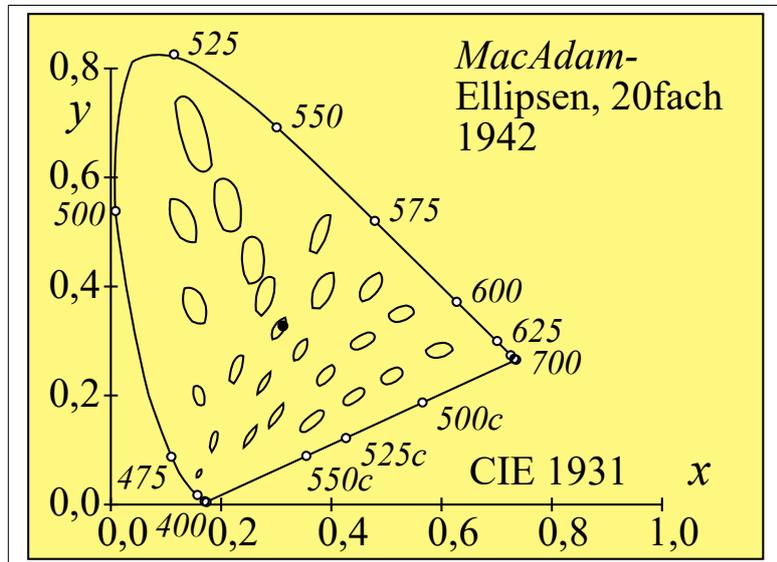


Abbildung 28: Aus [Richter 1996], Seite 97 übliche Darstellung von MacAdamEllipsen

Bei der üblichen Darstellung der MacAdam-Ellipsen sind die einzelnen Ellipsen meistens um einen Faktor vergrößert, damit die Darstellung klarer wird. Leider werden auch in grundlegenden Lehrbüchern diese Ellipsen einfach als solche angegeben⁴⁹. In Abbildung 28 wird korrekt darauf hingewiesen, dass die einzelnen Ellipsen um das 20fache vergrößert dargestellt sind, allerdings sind die Ellipsen bedingt durch die Vektorisierung etwas verzerrt. Diese Ellipsen werden in dessen maßgeblicher Arbeit von 1942 eigentlich aus Farbabständen errechnet, die durch Ausmischung zweier unterschiedlicher Farbfilter ermittelt werden. Wenn man dies in ein Farbraum-Diagramm wie beispielsweise CIE-Yxy oder CIE-Lab einträgt, ergeben sich viele kleine Striche deren Länge den Farbabstand definiert, den ein Mensch an dieser Stelle gerade nicht mehr wahrnehmen kann. Es fällt auf, dass sich die Striche für die Ellipsen keineswegs alle in einem Punkt schnei-

49 [Wyszecki 1982], Seite 793ff

den, sondern lediglich dicht beieinander liegen. Die Striche ein wenig so zu verschieben, dass sich die zu einer Ellipse gehörigen in einem Punkt schneiden, wertet MacAdam offenbar als zulässig, in der Annahme, dass sich das Farbsehen an bestimmten Orten des Farbraumes nicht schlagartig, sondern nur allmählich ändern sollte. Wie weit diese stillschweigende Annahme zulässig ist, bedürfte der Klärung. Jedenfalls lassen sich die Ellipsen nur unter dieser Annahme in üblicher Weise darstellen. MacAdam hat wesentlich mehr solcher Farbabstände gemessen und in seiner maßgeblichen Arbeit von 1942⁵⁰ veröffentlicht, als für die eigentlichen Ellipsen notwendig sind. Diese Werte tauchen in der Fachliteratur kaum auf, sind aber trotzdem sehr interessant, wie in Abbildung 30 gezeigt.

Der Helligkeitswert spielt bei Einträgen in ein CIE-XYZ-Diagramm zunächst keine Rolle, da er direkt über das CIE-Y definiert ist. Auch bei den CIE-Yxy-Werten ist er deshalb nicht wichtig. „These ellipses cannot be transformed into equal-sized circles by any projective transformation of the standard chromaticity diagram.“⁵¹ So schrieb MacAdam ursprünglich über seine Ellipsen. Trotzdem wurden diese immer wieder in anderen Farbsystemen dargestellt und auch MacAdam ist seinem hier zitierten Grundsatz nicht so ganz treu geblieben, siehe Abbildung 49. Bei den CIE-Lab und den DIN-Lab99-Werten ist aber der zugehörige Spektralfarbenzug stark von der angenommenen Helligkeit abhängig. Die erweiterten MacAdam-Striche lassen sehr gut auf die Ränder des Spektralfarbenzugs schließen. Wenn man nun die Helligkeit des angenommenen Spektralfarbenzuges verändert, kann man suchen, bei welcher Helligkeit sich die MacAdam-Striche am besten in den Spektralfarbenzug mit zugehöriger Purpurachse einbeschreiben lassen, wie in Abbildung 31 gezeigt.

Dies ist für CIE-Lab ein Helligkeitswert von CIE-L 21,6 und für DIN-L99 von 31, wie in Abbildung 32 gezeigt. Diese Helligkeitswerte sind hier nur empirisch aus der Geometrie der Messwerte und des zugehörigen Spektralkurvenzugs bzw. der Purpurgeraden abgeleitet. Auch das Diagramm des derzeit üblichen, modernen Farbraums CIECAM02-UCS ist durchaus aufschlussreich, wie Abbildung 33 zeigt.

50 [MacAdam 1942], Seite 258f

51 [MacAdam 1942], Seite 247

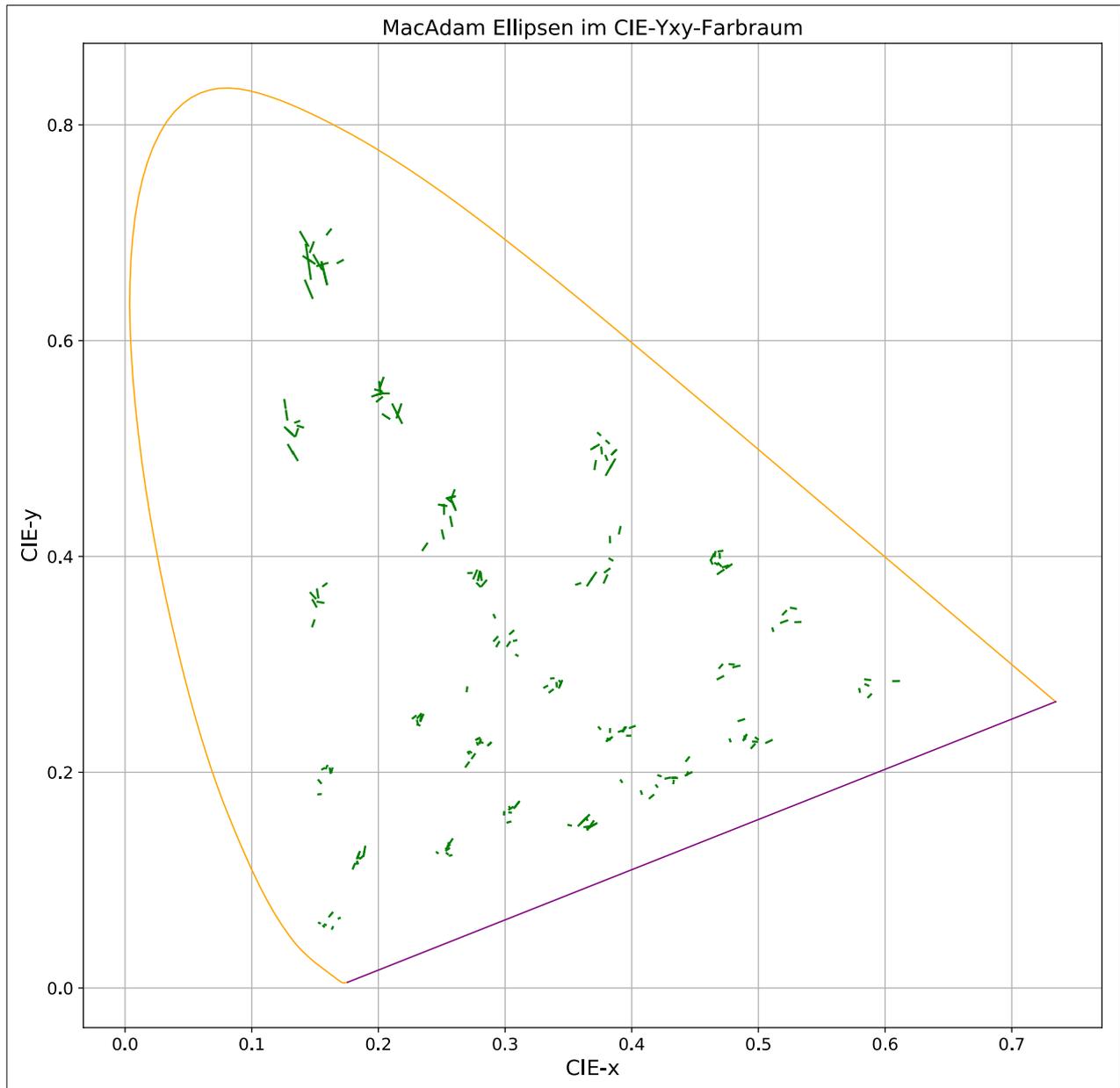


Abbildung 29: Darstellung der Originalmesswerte für die MacAdam-Ellipsen, eingetragen in ein CIE-Yxy-Diagramm streng nach den Messwerten in der MacAdams Arbeit von 1942 [MacAdam 1942], Seite 260f. Die Striche erscheinen sehr kurz, weil die Ellipsen eben nicht, wie sonst üblich vergrößert wurden.

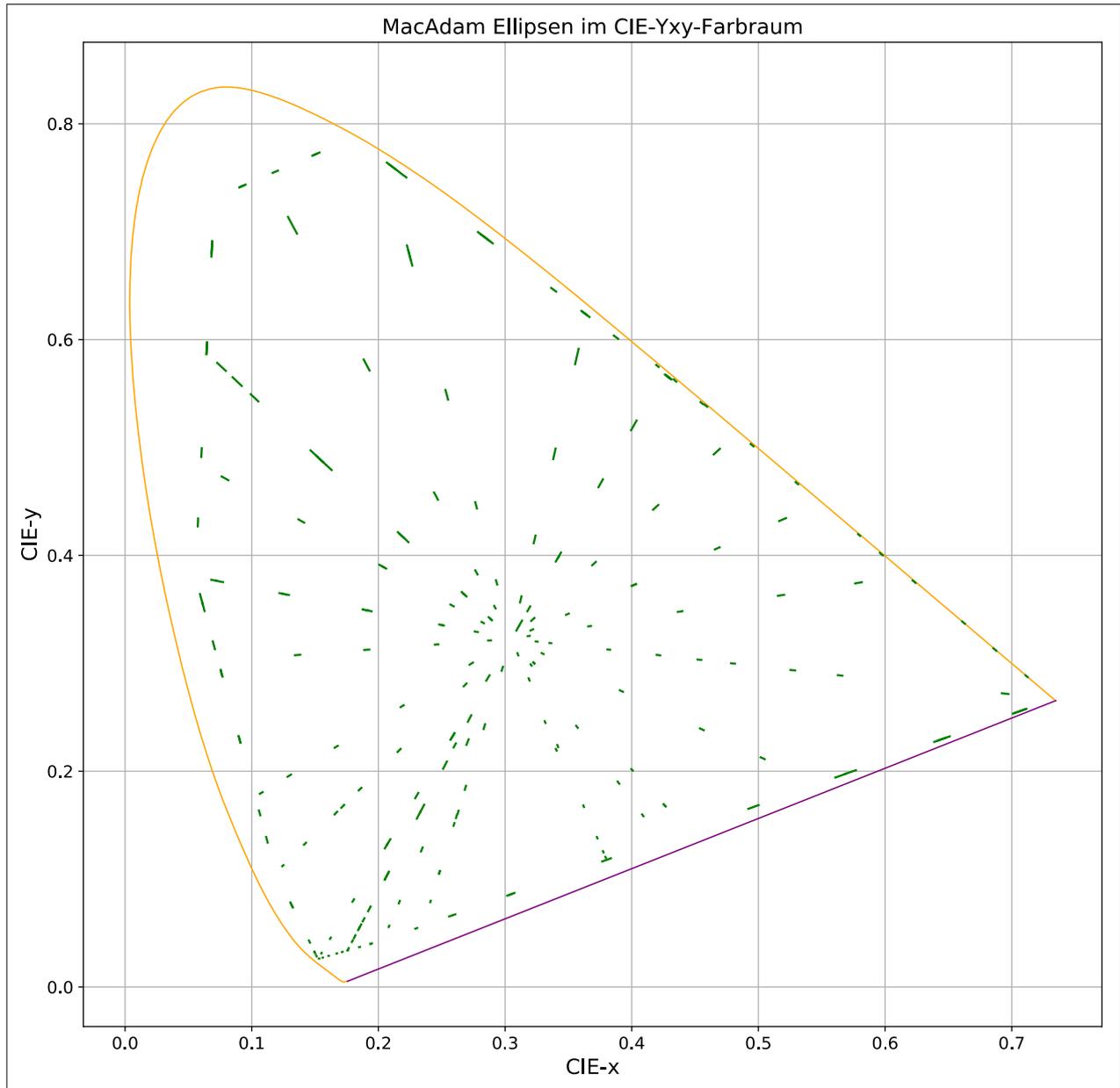


Abbildung 30: Allgemeine Messwerte von MacAdam eingetragen in ein CIE-Yxy-Diagramm streng nach den Messwerten in der MacAdams Arbeit von 1942 [MacAdam 1942], Seite 258f. Sehr hilfreich zur Überprüfung entstehender Grafen sind gerade diejenigen Mischgeraden, die möglichst nahe an den äußersten Rändern des CIE-Yxy-Diagrammes liegen.

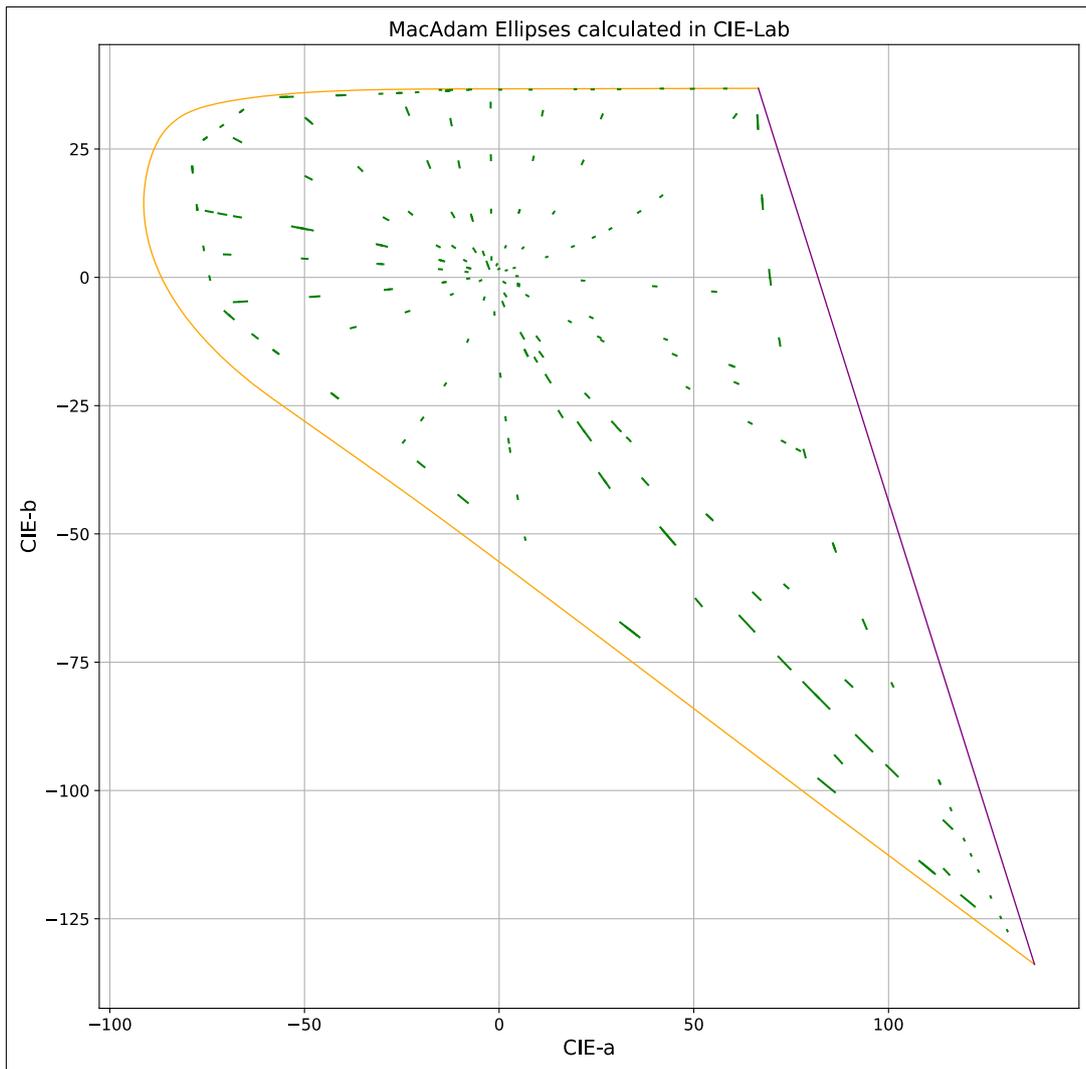


Abbildung 31: Allgemeine Messwerte von MacAdam eingetragen in ein CIE-Lab-Diagramm streng nach den Messwerten in der MacAdams Arbeit von 1942 [MacAdam 1942], Seite 258f. Da für CIE-Lab ein Helligkeitswert angenommen werden muss, hat sich die Grafik angeboten, eine möglichst gute Einbeschreibung in den Spektralfarbenzug zu suchen. In diesem Fall bei einem L-Wert von 21,6.

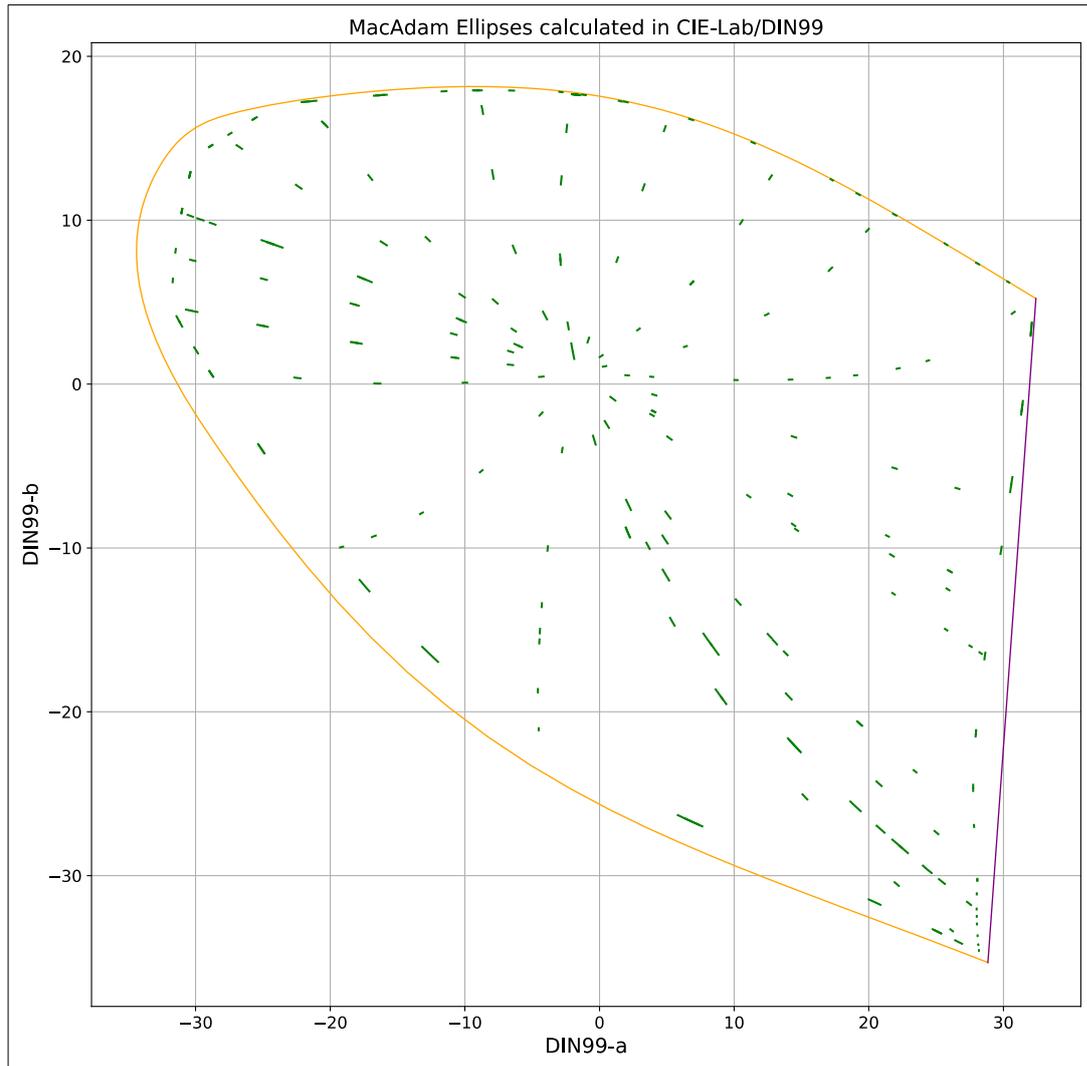


Abbildung 32: Allgemeine Messwerte von MacAdam eingetragen in ein DIN-Lab99-Diagramm streng nach den Messwerten in der MacAdams Arbeit von 1942 [MacAdam 1942], Seite 258f. Da für DIN-Lab99 ein Helligkeitswert angenommen werden muss, hat sich die Grafik angeboten, eine möglichst gute Einbeschreibung in den Spektralfarbentzug zu suchen. In diesem Fall bei einem L99-Wert von 31.

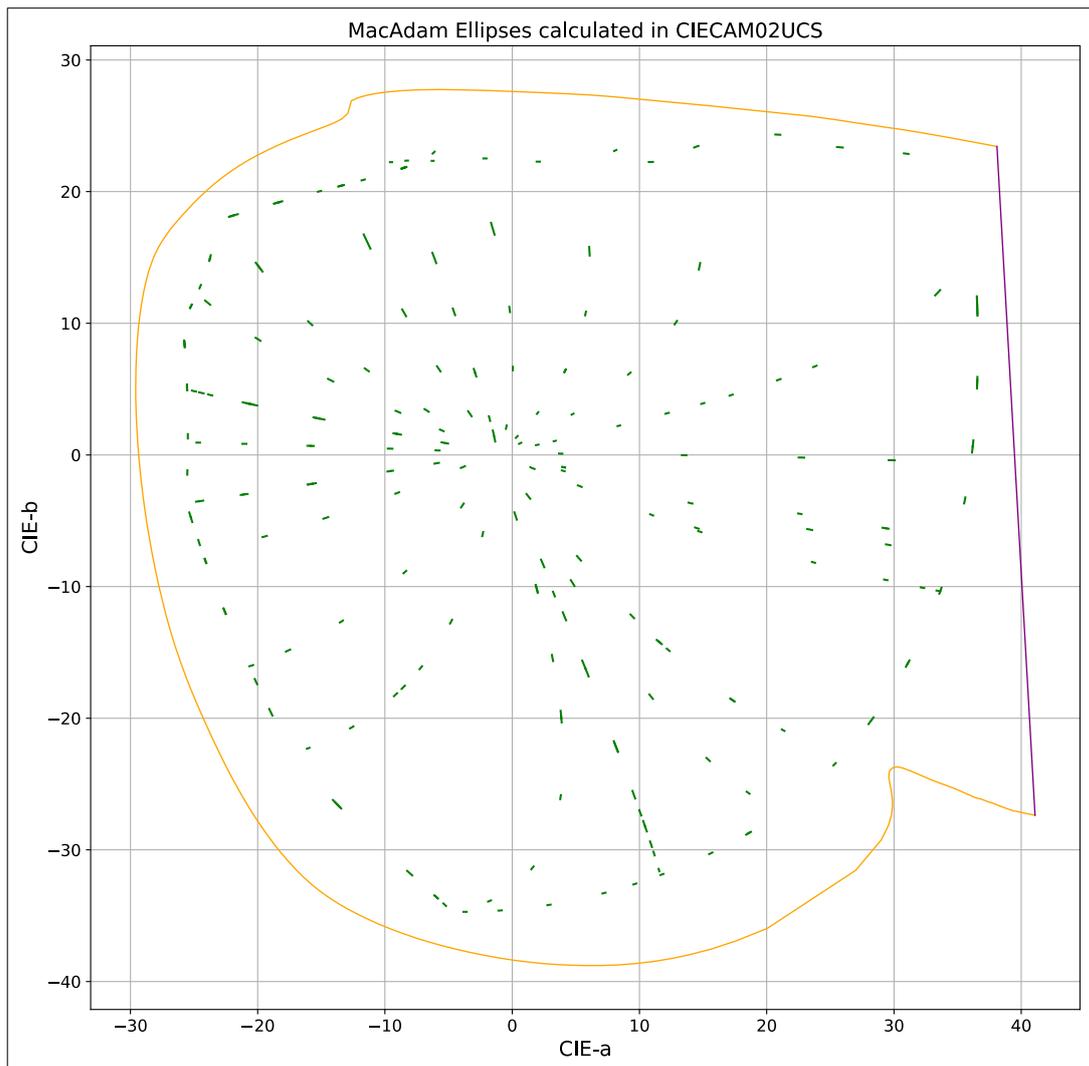


Abbildung 33: Allgemeine Messwerte von MacAdam eingetragen in ein CIECAM02-UCS-Diagramm streng nach den Messwerten in der MacAdams Arbeit von 1942 [MacAdam 1942], Seite 258f. Da CIECAM02-UCS eine Lichtart verlangt, wurde Normlicht C verwendet, wie dies MacAdam angibt. Dieses hat die CIE-XYZ-Koordinaten: 0,309 0,319 0,372. $L_A = 24$, Weiß der Probe in Relation zum Weiß der Umgebung $Y_b = 10$, VIEWING_CONDITIONS = 'Dark', angestrebte Helligkeit ist wieder die Lab-Helligkeit von von 21,6.

6.5 MacAdam als Gütekriterium

Die vielen Abbildungen sind nun kein Selbstzweck. Was der Industrie oder auch der Farbforschung fehlt, ist ein Farbsystem, das Farbunterschied so einordnet, dass tatsächlich empfundene Farbunterschiede eines Beobachters zuverlässig und einheitlich abgelesen werden können. Nehmen wir nun an, dass die MacAdam-Experimente hinreichend genau sind und nehmen wir ferner an, dass die CIE-Normalenzen nach Wright bzw. für CIECAM02-UCS, die

daraus neueren Normvalenzen ebenfalls hinreichend genau sind, dann lassen sich mittels der MacAdam-Experimente die einzelnen Farbsysteme hinsichtlich ihrer Güte bezüglich Farbabständen einordnen. In einem idealen Modell müssten alle Striche der MacAdam-Experimente gleich lang sein. Je größer die Unterschiede, desto schlechter erfüllt das untersuchte Farbsystem die Anforderung der Abstandsgleichheit. Nachfolgend eine Tabelle, die nur die Verhältnisse von maximaler zu minimaler Strichlänge in unterschiedlichen Farbsystemen angibt, unterschieden in die oben häufig gezeigten allgemeinen Ergebnisse von MacAdam und nur diejenigen Daten, die für die Ellipsen verwendet werden. Letztere sind wichtig, weil sie nicht in den extrem

gesättigten Farben liegen, die im Alltag, insbesondere bei gedruckten Farben praktisch keine Rolle spielen:

	„Ellipsen“ Maximum/Minimum	Allgemeine Messungen von MacAdam Maximum zu Minimum
CIECAM02-UCS	3.69	8.20
DIN99	9.72	54.8
CIE-Lab	9.82	20.2
CIE-Yxy	29.0	64.2

Bemerkenswert ist das erstaunlich schlechte Abschneiden von DIN99, obwohl es in der Praxis, also mit druckbaren Farben, recht zuverlässige Vorhersagen über Farbabweichungen erlaubt, die ähnliche Güte haben wie Einordnung nach CIEDE 2000, wie Philipp Kittelmann das in seiner Untersuchung aus dem Jahr 2010 feststellt.⁵² Vielleicht hilft hier eine Betrachtung der Verteilung der einzelnen Längen weiter, wie in Abbildung 34 gezeigt:

Bei einem Abstand zwischen 20 und 30 gibt es vier Werte, die extrem nach oben abweichen und erst ab einem Abstand von über vierzig, gibt es viele extrem kleine Werte. Wenn wir diese relativ wenigen extremen Werte außer Acht lassen, bewegen wir uns in einem Bereich zwischen 0,13 und 1,1, also einem Verhältnis von etwa 8,5, das mit den Werten in CIECAM02-UCS durchaus mithalten kann. Diese extrem gesättigten Farben spielen im Alltag bei Aufsichtsfarben kaum eine Rolle, weshalb das Ergebnis für die Messwerte der eigentlichen Ellipsen auch relativ gut ist.

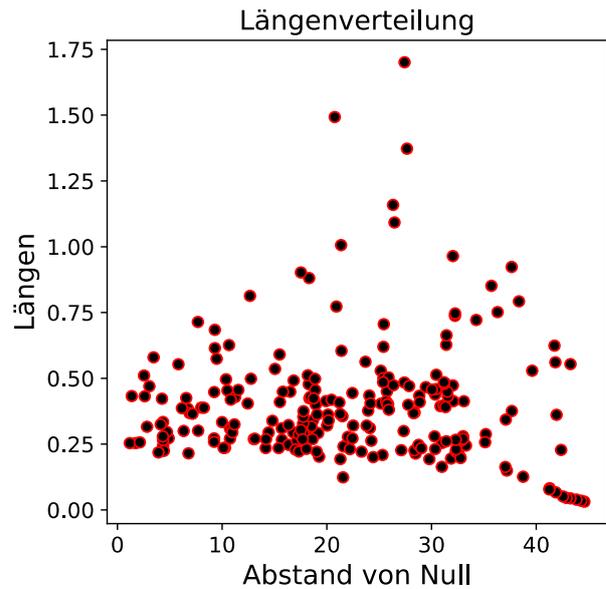


Abbildung 34: Wie sind im DIN99-Diagramm die allgemeinen Messungen von MacAdam verteilt? Bei einem Abstand zwischen 20 und 30 gibt es vier Werte, die extrem nach oben abweichen und erst bei einem Abstand von über vierzig gibt es viele extrem kleine Werte. Diese extrem gesättigten Farben spielen im Alltag bei Aufsichtsfarben kaum eine Rolle.

52 [Kittelmann 2010], Seite 60

7 Zusammenfassende Kritik am Stand der Technik

7.1 Grundsätzliches Problem

Es ist momentan nicht möglich eine Referenzfarbe zu definieren, die für eine Probe gelten soll und dafür einen einheitlichen Wert für die zulässige Farbabweichung zu fordern. Wenn man trotzdem einen einheitlichen Zahlenwert für die Farbabweichung festlegt, werden Beobachter in der Realität sehr stark unterschiedliche Farbabweichungen feststellen. An Farborten insbesondere in der Nähe der Grauchse, wo rechnerisch kaum Differenzen sichtbar sein sollten wird man durchaus Farbunterschiede wahrnehmen, an Farborten mit gesättigten Farben wird man rechnerisch große Abstände erhalten, aber eventuell gar keine Unterschiede wahrnehmen. Nehmen wir zur Veranschaulichung wie in Kapitel 1.1 an, die Einteilungen auf Linealen wäre nicht gleichmäßig, sondern willkürlich. Die 1 und die 2 wären näher beieinander als beispielsweise die 5 und die 6. Alle Lineale wären aber einheitlich produziert. Man könnte per Telefon und über Texte durchaus Längen mitteilen und jeder, der ein Lineal hätte, könnte nachmessen, ob Längen übereinstimmen. Wenn man nun aber fordern wollte, Stäbe mit unterschiedlichen Längen x_1 , x_2 , x_3 ... sollen eine einheitliche Toleranz von y haben, dann würde es schwierig. Wäre mit y ein Vielfaches des Abstandes zwischen 1 und 2 gemeint oder ein Vielfaches des Abstandes zwischen 5 und 6 oder ganz etwas anderes. In diesem Fall ließe sich dann relativ einfach eine Konvention aushandeln. Bei der Farbmotrik gibt es aber bisher keine Beschreibung, wie man aus gemessenen Spektren von Referenzfarbe und Probe Abstände errechnen könnte, die im ganzen Farbraum einheitlich wären. Angenommen man wählt die MacAdam-Ellipsen als Gütekriterium, was durchaus getan wird, wie beispielsweise in [Asano 2015], dann kann man dagegen Einwände erheben, wie das in dieser Arbeit in Kapitel 6 getan wurde. Nehmen wir trotzdem an, dass die MacAdam-Messwerte sinnvolle Aussagen zulassen. Dann werden die errechneten einheitlichen Farbabstände in den CIE-Diagrammen, egal ob CIE-Yxy, CIE-Lab oder CIECAM02-UCS für Beobachter mindestens

um den Faktor vier schwanken, bei den meisten Farbraumdiagrammen noch wesentlich mehr. Genauere Werte finden sich in der Tabelle in Kapitel 6.5. Für die praktische Anwendung bekommt man mit den bisherigen Farbdiagrammen leider nicht mehr als einen groben Anhaltspunkt für Farbdifferenzen. In der Druckindustrie, wo es um kleine Farbabweichungen auf Druckprodukten geht, ist gegenüber den Kunden kaum vermittelbar, warum eine Farbdifferenz bei Hauttönen im CIE-Lab-Farbraum von ΔE gleich 3 wahrscheinlich deutlich wahrnehmbar sein wird, während eine Abweichung in einem gesättigten Rot von ΔE gleich 10 eventuell gar nicht sichtbar sein wird. Mit CIECAM02-UCS wird das Problem zwar verringert, aber nicht gelöst, zudem ist der Berechnungsaufwand wesentlich höher als bei CIE-Lab. Es müssen nämlich zusätzlich zu den gemessenen Remissionsspektren von Referenz und Probe noch die Beleuchtungsfarbe und ein Faktor für die Gesamthelligkeit berücksichtigt werden.⁵³ Kurz gesagt, liegt das Problem darin, dass es bisher keinen gleichabständigen Farbraum gibt, um das Bild mit den Linealen aufzugreifen: Wir haben keine Lineale für Farbe, mit einer einheitlichen Einteilung.

7.2 Kritik im Detail

Kritik an den Versuchen von Wright:

- Die Wellenlängen unter 460 nm, werden zu schwach berücksichtigt siehe Ende Kapitel 2.
- Es wird stillschweigend ein lineares Verhältnis von Energiemenge des Reizes zu Reizantwort angenommen, was zwar prinzipiell bedenklich ist, aber wohl keine großen Auswirkungen hat, wie in den Kapitel 5 und 4.3 ausgeführt.
- Die Maxima der bisherigen Farbempfindlichkeitsfunktionen entsprechen nicht den Maxima der Absorptionskurven der Farbrezeptoren im Auge, siehe Vergleich von Abbildung 15 und 16.
- Relativ willkürliche Definition der Linearumrechnung zur Vermeidung negativer Farbwerte, siehe Kapitel 2 nach Abbildung 11.
- Die Messwerte von Wright lassen eine größere Streubreite erwarten, als sie Wright tatsächlich gemessen hat, siehe Abbildung 14.

53 [Wasilewski Kostova 2009], Seite 12 f

- Die $V(\lambda)$ -Kurve, auf der die ganzen Berechnungen von Wright fußen, hat selbst schon eine recht große Streubreite, siehe Abbildung 13

Kritik an den Versuchen von MacAdam:

- MacAdam hat alle seine Messungen bei geringer Beleuchtungsstärke gemessen, die man heute mit einem L-Wert zwischen 21 und 31 ausdrücken würde, siehe dazu das Ende von Kapitel 6.4.
- MacAdam hat erklärtermaßen nur Farborte auf einer einzigen Helligkeitsebene vermessen.⁵⁴
- MacAdam hat seine Untersuchungen zum ganz überwiegenden Teil an nur einem einzigen Probanden durchgeführt.⁵⁵ Die Ergebnisse wurden für spezielle Messfelder lediglich mit acht weiteren Probanden abgeglichen.⁵⁶ Repräsentativ ist das eher nicht.
- Welche Filterfunktion MacAdam für die verwendeten Polarisationsfilter annimmt, wird in seiner Arbeit [MacAdam 1942] nicht beschrieben, so dass wir heute auf Vermutungen angewiesen sind, siehe Kapitel 6.2 um Abbildung 23.
- Für einige verwendete Filter und Filterkombinationen sind die Transmissionskurven nicht mehr nachvollziehbar, da es sich nicht um die sonst verwendeten WRATTENLIGHT FILTERS von Kodak handelt. Siehe Kapitel 6.2.

54 [MacAdam 1942], Seite 247, gleich in der Einleitung: „...without variation of luminance.“

55 [MacAdam 1942], Seite 247, gleich in der Einleitung: „...Over twenty-five thousand trials at color matching have been recorded for a single observer...“

56 [MacAdam 1942], Seite 273, Tabellenkopf

8 Die Sigmoidfunktion

Bei neuronalen Netzen hat es sich bewährt, die Reaktion eines „Neurons“ auf ein Eingangssignal mit

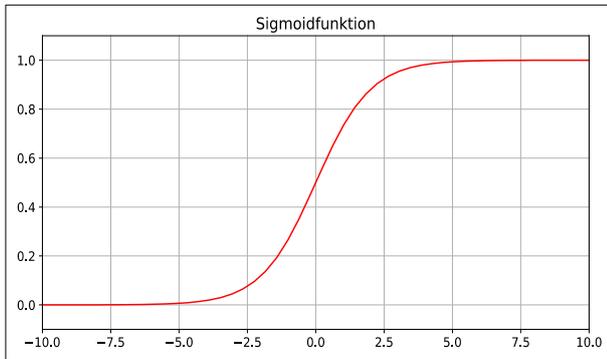


Abbildung 35: Die Sigmoidfunktion, wie sie häufig in neuronalen Netzen bzw. beim Deep Learning zum Einsatz kommt.

einer sogenannten Sigmoidfunktion zu realisieren.

Es gibt deren mehrere, aber üblich ist $y = \frac{1}{1 + e^{-x}}$

Diese Sigmoidfunktionen sind erfreulich leicht abzuleiten, nämlich mit $\text{sig}'(t) = \text{sig}(t) (1 - \text{sig}(t))$ ⁵⁸. Man sieht am Grafen, dass Eingabewerte für Sigmoidfunktion kleiner als 8 und größer als +8 nur noch einen Wert gegen 0 oder 1 liefern. Sie erreichen dort bereits einen Wert von 0.0003 und 0.9997. Oft wird auch gar keine Reaktion des „Neurons“ gewünscht, weshalb größere Werte manchmal trotzdem verwendet werden. Einen stetigen Zusammenhang von Eingangs- und Ausgangsgröße gibt es im Wesentlichen nur innerhalb des Bereiches von -8 bis +8. Im Vergleich zu den in der Farbmeterik verwendeten Logarithmen oder Wurzeln, hat diese Funktion mehrere große Vorteile. Sie läuft erstens nie gegen Unendlich. Zweitens ist diese Reiz-Reaktions-Kurve sehr ähnlich der Kurve, wie sie für modernere Farbsysteme wie CIECAM02 verwendet wird und in Abbildung 36 gezeigt ist⁵⁹.

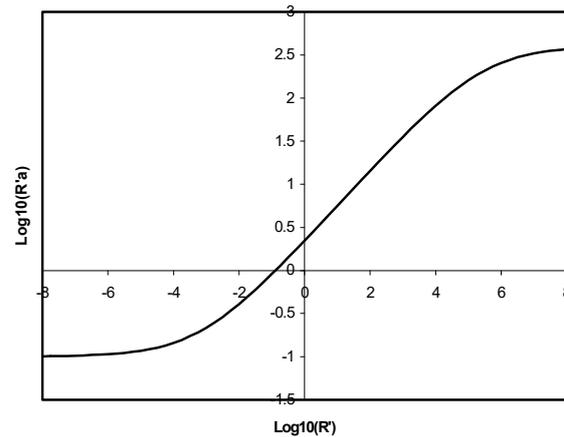


Abbildung 36: Reiz-Reaktionskurve für CIECAM02

Die Formel für die Übertragungskennlinie in CIECAM 02 ist allerdings ganz anders:

$$R'_a = \frac{400 F_L R' / 100^{0.42}}{27.13 + F_L R' / 100^{0.42}} + 0.1$$

wobei F alleine bereits für einen komplexen logarithmischen Zusammenhang steht.

Durch einfache Subtraktion von der Sigmoidfunktion mit 0,5 erhält man bei $x = 0$ wieder 0 und durch eine anschließende Multiplikation wieder Werte von 0 bis 1. Die Abbildung 37 zeigt, dass eine Annäherung an bisher verwendete Übertragungsfunktionen im Farbsehen wie in Abbildung 19 gezeigt, schwierig werden könnte. In den späteren Berechnungen stellt sich bemerkenswerter Weise heraus, dass das anscheinend kein all zu großes Problem ist.

57 [Rashid 2017], Seite 34

58 https://wikimedia.org/api/rest_v1/media/math/render/svg/0959aabb72c4cb49e258501af9be7a76b7e5d2fe gelesen am 4. Oktober 2018

59 [Moroney 2002], Seite 26

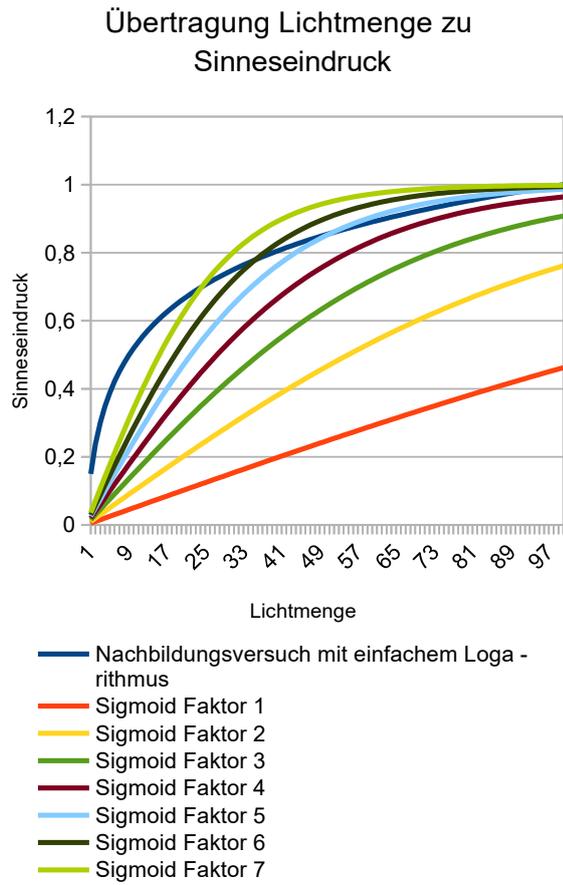


Abbildung 37: Die Sigmoidfunktion, um 0,5 subtrahiert und anschließend mit 2 multipliziert, im Vergleich zu der Übertragungsfunktion in Abbildung 19

9 Farbmétrische Formeln neuronal formuliert

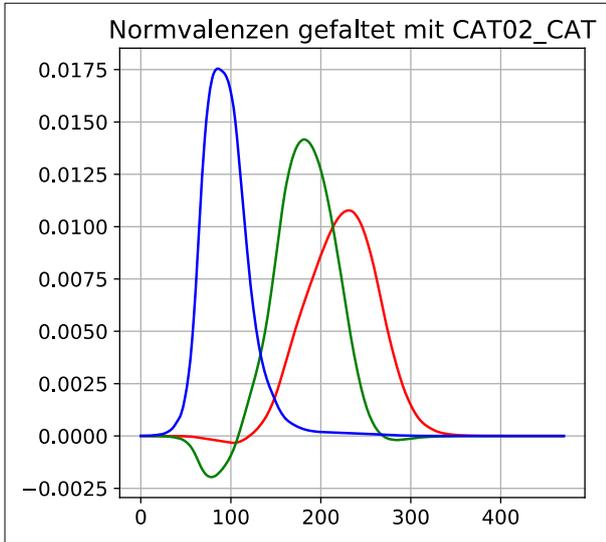


Abbildung 38: Die für CIECAM02-UCS geänderten Normvalenzen, wobei bemerkenswert ist, dass die geringen negativen Anteile anscheinend nicht ins Gewicht fallen.

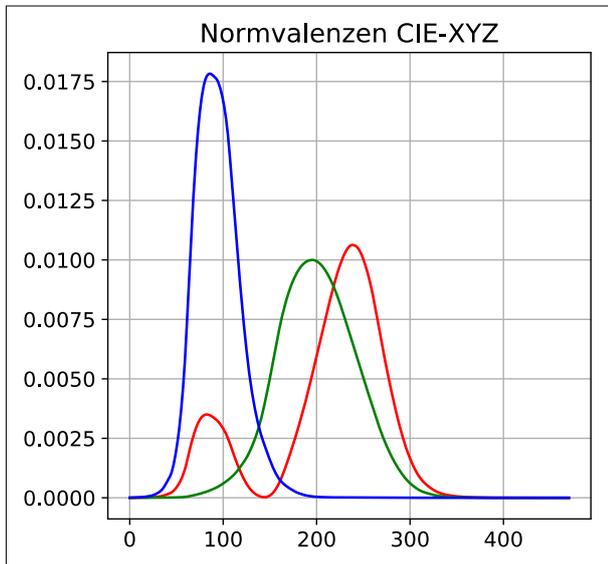


Abbildung 39: Die bekannten Normvalenzen

Im Prinzip lassen sich die bisher entdeckten oder entwickelten farbmétrischen Formeln zur Farbortbestimmung auch im Sinne neuronaler Netze oder Deep-Learnings beschreiben. Die Neuronen reagieren mit einer Sigmoidfunktion und die einzelnen

Neuronen werden gewichtet miteinander verbunden. Mit jedem Test werden die Gewichte ein wenig verändert. Die „Intelligenz“ steckt sozusagen in den Gewichten. Die Gewichte werden üblicherweise als zweidimensionale Matrize beschrieben. Auch in den Formeln für CIECAM02-UCS werden solche Matrizenrechnungen verwendet. So werden aus den Normvalenzen, wie sie für CIE-XYZ verwendet werden, abgeleitete Matrizen für CIECAM02-UCS, siehe Abbildungen 39 und 38. Für ein Ergebnis, das sich an CIECAM02 anlehnt, ergibt sich ein Modell mit den Gewichtungsfaktoren wie sie für CIECAM02 verwendet werden, um von den CIE-XYZ-Werten in verbesserte Werte umzurechnen.

$$9 \times \begin{bmatrix} 0.7328 & 0.4296 & -0.1624 \\ -0.7036 & 1.6975 & 0.0061 \\ 0.0030 & 0.0136 & 0.9834 \end{bmatrix}^{60}, \text{ also}$$

$$\begin{bmatrix} 6.5952 & 3.8664 & -1.4616 \\ -6.3324 & 15.2775 & 0.0549 \\ 0.027 & 0.1224 & 8.8506 \end{bmatrix} \text{ die 9 ist not-}$$

wendig, damit man mit der Sigmoidfunktion in einen Wertebereich kommt, der eine stetige Veränderung zulässt. Man kann sich also die Eingangswerte in CIE-XYZ-Koordinaten als Eingangsschicht vorstellen, die dann mittels der angegebenen Matrize umgerechnet und an die erste verborgene neuronale Schicht übergeben wird. Von dort wird dann eine zweite Matrize verwendet, um die Daten mit der Ausgangsneuronenschicht zu verrechnen. Das Schichtenmodell ist in Abbildung 40 dargestellt. In Anlehnung an CIE-Lab, hat sich folgende Matrize

als sinnvoll erwiesen: $\begin{bmatrix} 0 & 6 & 0 \\ 21 & -7 & 0 \\ 6 & 6 & -15 \end{bmatrix}$.

Oben wird die Helligkeit nur aus Y gebildet, mit dem Faktor 6, der dafür sorgt, dass mittels Sigmoidfunktion eine nicht-lineare Übertragung vollzogen wird. In der Mitte wird analog zu Lab-a die X-Koordinate gegen die Y-Koordinate verrechnet und eine Art Grün-Rot-Achse gebildet. Und unten wird analog zu Lab-b die Summe aus X und Y gegen Z verrechnet, wobei das Z mehr als doppelt gewichtet wird. Dadurch wird eine Art Gelb-Blau-Achse gebildet. Die glatten Zahlen wurden heuristisch ermittelt. Abbildung 40 zeigt, wie dieses Neuronale Netz üblicherweise grafisch dargestellt würde. Da-

60 [Wasilewski Kostova 2009], Seite 12

mit leichter von der Matrizenschreibweise auf die grafische Darstellung geschlossen werden kann, sind hier die einzelnen Gewichtungsfaktoren mit eingezeichnet. Die erzielbaren Ergebnisse sind erstaunlich gut, wie man in den nachfolgenden Diagrammen Abbildung 41 und 42 sehen kann. Die Längen der MacAdam-Abstände, die nicht für die Ellipsen verwendet werden, führen zu Werten, wie sie in der folgenden Tabelle dargestellt werden. Diese sind erfreulicherweise genau so gut wie CIE-Lab, aber wie zu erwarten einiges schlechter als CIE-CAM02, wie man in den folgenden Tabellen sehen kann. CIECAM02 liefert fast immer bessere Werte als CIE-Lab. Die erste Tabelle zeigt Auswertungen für die ungeordneten Messwerte von MacAdam. Die zweite Tabelle zeigt Auswertungen nur für diejenigen Messwerte von MacAdam, die für die eigentlichen Ellipsen verwendet werden.

Berechnungen für allgemeine Messwerte von MacAdam

MacAdam-Abstände	Durchschnitt	Median	Minimum	Maximum	Verhältnis Maximum/Minimum
CIE-Lab	1.21	0.883	0.260	5.24	20.2
CIE-CAM02	0.371	0.326	0.166	1.36	8.20
Matrix-Sigmoid	0.570	0.469	0.0985	2.00	20.4

Berechnungen für Ellipsen-Messwerte von MacAdam

MacAdam-Ellipsen	Durchschnitt	Median	Minimum	Maximum	Verhältnis Maximum/Minimum
CIE-Lab	0.931	0.787	0.244	2.40	9.82
CIE-CAM02	0.361	0.330	0.190	1.36	3.69
Matrix-Sigmoid	0.563	0.511	0.164	1.44	8.77

Dargestellt ist zunächst der Durchschnittswert aller grünen Striche im Diagramm, dann der Median aller Strichlängen, der kürzeste und der längste grüne Strich. Schließlich das Verhältnis von kürzestem zu längstem grünen Strich. Letzteres wird im Folgenden als das wesentliche Gütekriterium für ein Farbsystem verwendet. An Abbildung 42 fällt vielleicht auf, dass der Unbuntpunkt keinesfalls auf 0,0 liegt. Dies ließe sich mit einer einfachen Subtraktion beheben. Die Berechnung der Farbabstände würde dann trotzdem keine anderen Werte liefern.

Wenn man nur die Daten für die MacAdam-Ellipsen heranzieht, dann ist die MatrixSigmoid-Berechnung ähnlich gut wie CIE-Lab. Entscheidend als Gütekriterium ist das Verhältnis, also die letzte Spalte der Tabelle. Es lässt sich festhalten, dass man mittels Matrizen und Sigmoidfunktionen mit relativ wenig Aufwand ein Farbsystem nachbilden lässt, dass vergleichbar gute Ergebnisse liefert, wie CIE-Lab. Daraus lässt sich schließen, dass eine neuronale Modellierung analog zu neuronalen Netzen oder Deep-Learning anscheinend sinnvolle Ergebnisse liefert. Dies alles vorerst heuristisch, ohne die Näherungstechniken neuronaler Netze selbst zu verwenden.

Kalkuliertes neuronales Netzwerk analog CIE-Lab,
jedoch mit den XYZ-Werten von CIECAM02UCS und
Sigmoidfunktionen als Übertragungsfunktionen

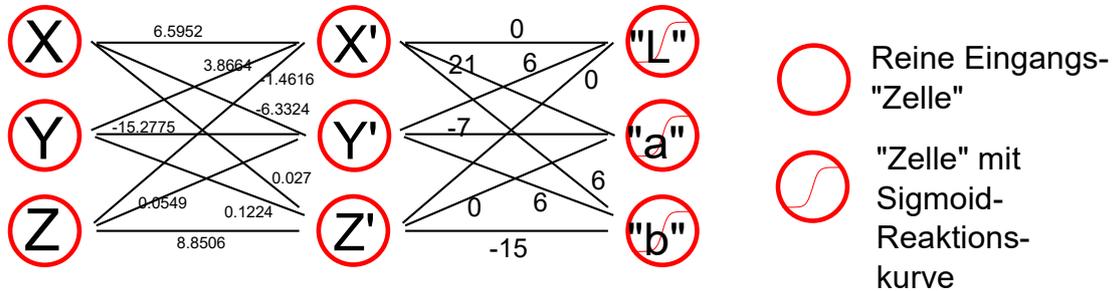


Abbildung 40: Architektur eines Neuronales Netzes analog CIE-Lab allerdings mit den modifizierten XYZ-Werten von CIECAM02-UCS und Sigmoidfunktionen statt Kubikwurzeln

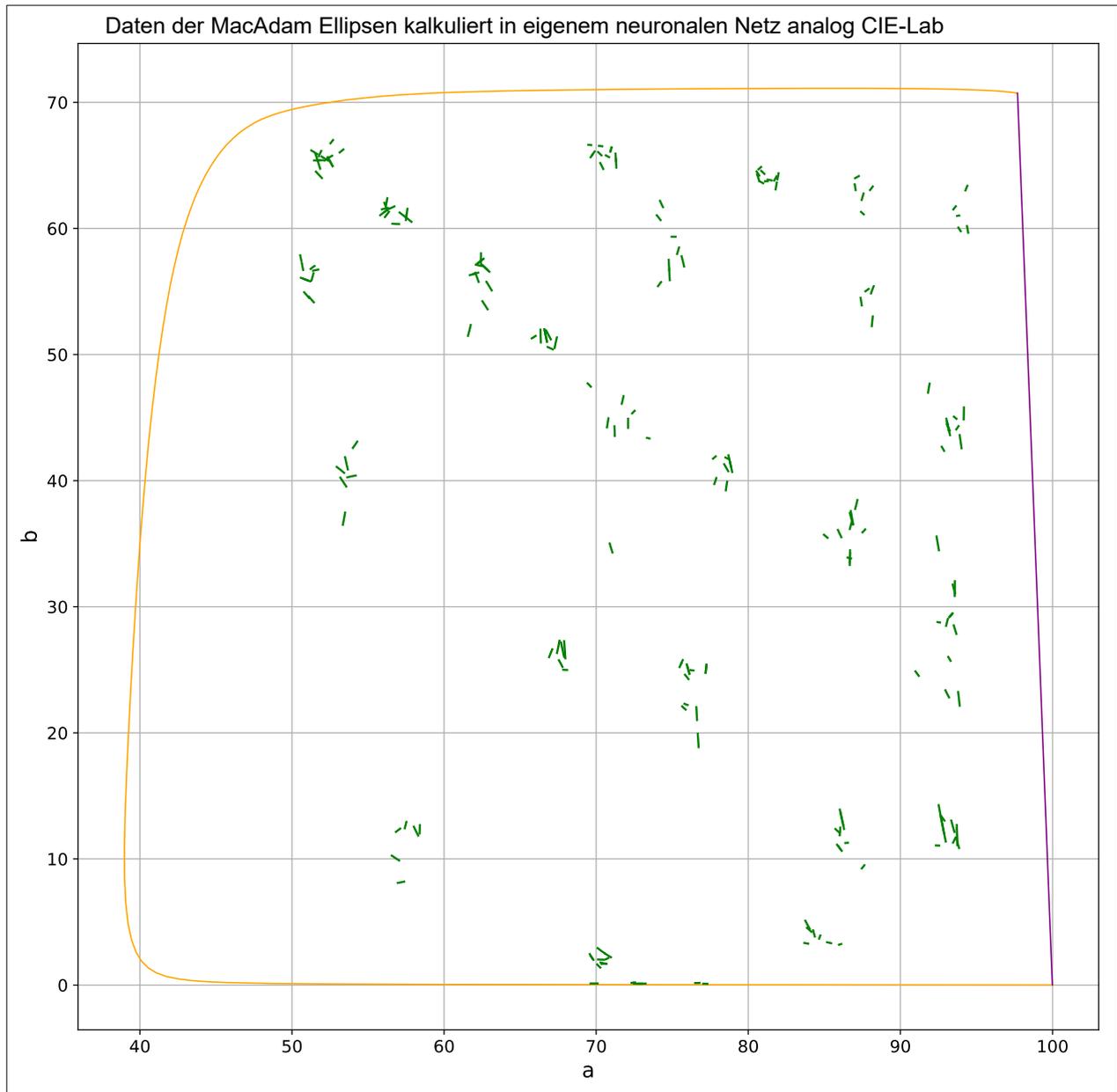


Abbildung 41: Werte für die MacAdam-Ellipsen eingetragen in ein Diagramm, das aus MatrixSigmoid-Berechnungen analog zu CIE-Lab gewonnen wurde.

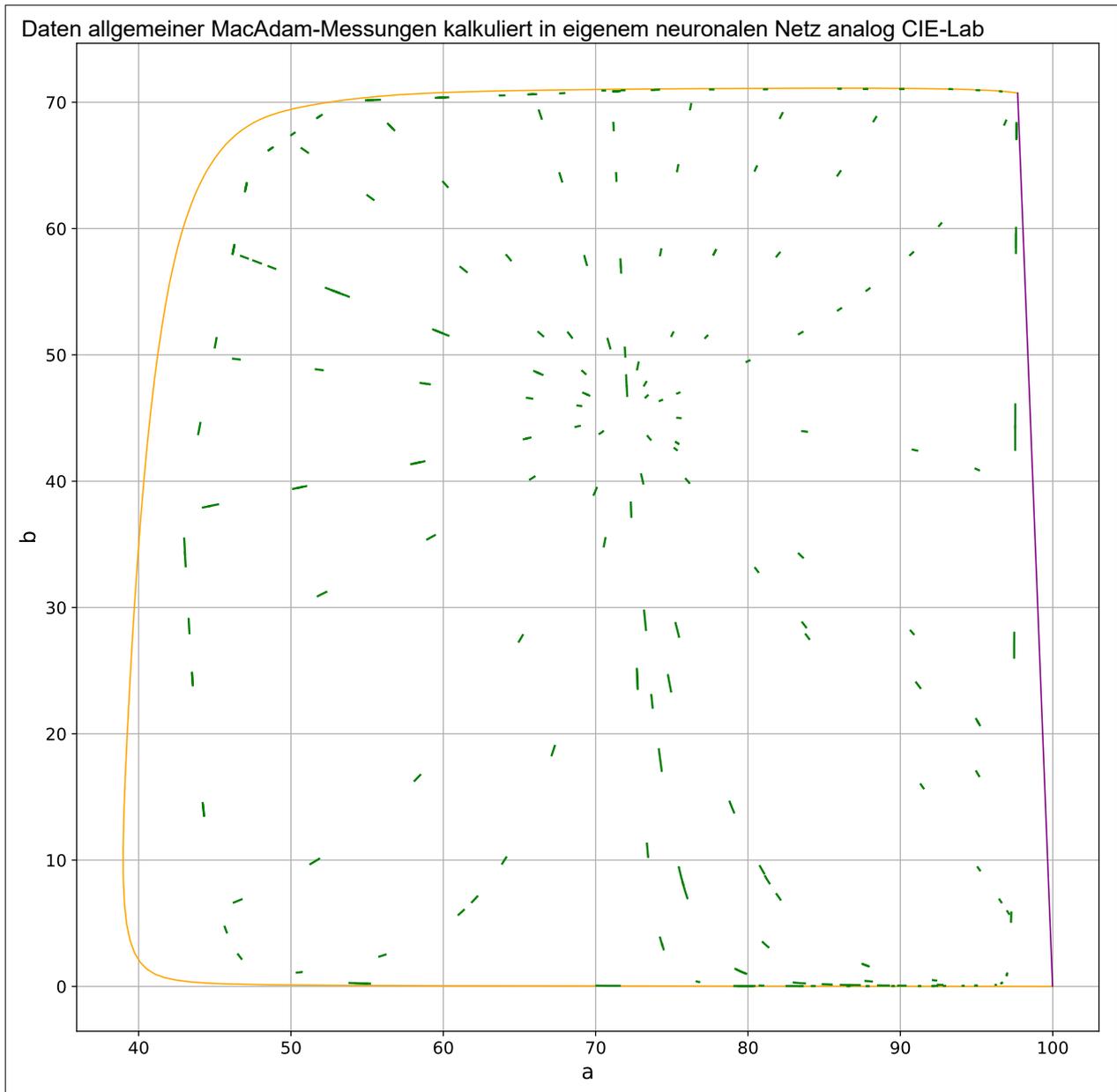


Abbildung 42: Werte für die allgemeinen Messwerte von MacAdam, eingetragen in ein Diagramm, das aus MatrixSigmoid-Berechnungen analog zu CIE-Lab gewonnen wurde.

10 Neuronale Optimierung

Neuronale Netze und Deep learning sind nicht nur Schlagworte, sondern leistungsfähige mathematische Verfahren, für die es mittlerweile auch gute Softwaretools gibt. Wenn wir nun wie oben gezeigt, das Problem des gleichabständigen Farbsehens als Neuronales Netz beschreiben können, können wir auf diese Verfahren und Tools zurückgreifen. Als klassisches Beispiel zeigt Tariq Rashid in seinem Buch [Rashid 2017] wie man Neuronale Netze und Deep learning nutzt, um einen großen Datensatz aus Paaren von Bildern von handgeschriebenen Ziffern und deren eindeutigem Zahlwert zunächst zu verknüpfen. Das Netz wird mit diesen Verknüpfungen trainiert und kann nach diesem Training gute Vorhersagen machen, welches Bild einer geschriebenen Ziffer zu welcher tatsächlichen Ziffer gehört. Dies wird für das optische Erkennen von Handschriften gebraucht, sogenannte Optical Character Recognition (OCR). Zunächst werden die Daten der Bilder mit den Daten der Zahlwerte über ein Neuronales Netz verknüpft. Eingangswert sind dann wenn wir uns am Beispiel unseres MatrixSigmoid-Farbmodells orientieren, eben nicht die drei XYZ-Werte, sondern alle 256 Grauwerte der Bildpunkte. Diese werden analog wie bei unserem Beispiel über Schichten von „Neuronen“ miteinander hin zu zehn Ausgabe-„Neuronen“ verbunden. Nun werden in einem Trainingsprogramm mittels mathematischer Verfahren die besten Gewichtungsfaktoren gesucht. Das geschieht in vielen Durchläufen, in denen die Gewichtungsfaktoren abhängig vom Ergebnis immer ein klein wenig in die richtige Richtung verändert werden. Ziel des Prozesses ist es Gewichtungsfaktoren zu finden, so dass bei Eingabe eines Bildes zuverlässig der richtige Zahlwert ermittelt wird⁶¹. Im Fall der Messwerte von MacAdam haben wir nun das Problem, dass zwar die Farbabstände für die Farbschwelle, also sozusagen die erste Ableitung, oft auch als Deltas bezeichnet, bekannt sind, nicht jedoch der gewünschte Farbraum, in dem alle diese Farbabstände gleich lang sind. In diesem Fall kommt uns ein gängiges Optimierungsverfahren für Neuronale Netze namens Backpropagation sehr entgegen, welches über Ab-

leitungen versucht, das aktuelle lokale Optimum zu finden⁶². Man muss also nicht mehr ins Blaue hinein herumprobieren, sondern man nimmt zunächst zufällig Werte für die Gewichtungsfaktoren an und erhält mittels Backpropagation zuverlässig das nächste lokale Optimum. Dies spart enorm Rechenaufwand und somit Zeit. Nachteil ist, dass man auch nach vielen Versuchen nie ganz sicher sein kann, wie nahe man dem absoluten Optimum ist, da man immer nur lokale Optima finden kann. Da wir nun über die MacAdam-Experimente bereits Ableitungen, also Deltas haben, müssen wir sie nicht erst mühsam berechnen, sondern können sie direkt verwenden, um auf den Farbraum zu schließen, in dem möglichst alle Deltas gleich lang sind. Nachteilig daran ist, dass gängige Softwaretools leider nicht verwendet werden können, sondern für unseren aktuellen Fall die Software umgeschrieben oder teilweise neu entwickelt werden muss, wovon die langen Listings im Anhang Zeugnis geben.

Spannend ist die Frage, wie genau man von einem Delta der Farbe auf die absolute Lage der Farbe schließen kann. Wenn alle Koordinaten des neuen zu erzeugenden Koordinatensystems verkleinert werden, dann sollte eigentlich auch das Delta kleiner werden und umgekehrt. Bei einer Sigmoidfunktion ist es aber genau umgekehrt: wenn das Delta zu groß ist, muss der absolute Wert vergrößert werden und umgekehrt. Ein Versuch ins Ungewisse sollte durchaus erhellend sein. Sinnvoller ist folgendes Vorgehen, wie in Abbildung 43 gezeigt: Je MacAdam-Messung gibt es drei Ortsvektoren. Einmal die Farbe im Mittelpunkt und die Farben, die die Abweichung in beide Richtungen repräsentieren. Aus dem Vektor, der die Farbe im Mittelpunkt repräsentiert und der ersten Farbe am Rand, lässt sich durch einfache Vektorrechnung der Differenzvektor errechnen. Die zweite Randfarbe lässt sich errechnen, in dem man den Ortsvektor der Mittelfarbe mit dem Differenzvektor mit umgekehrtem Vorzeichen addiert. Wenn wir nun einen gleichabständigen Farbraum mit Einheit 1 MacAdam-Abstand hätten, sollte der ideale Differenzvektor die Länge 0,5 haben. Man muss also nur den realen Differenzvektor so strecken, dass er 0,5 lang wird und dann den idealen Ort berechnen, in dem vom Ortsvektor der Mittelfarbe der ideale Differenzvektor abgezogen wird. Nun kann man dem neuronalen Netz den Ide-

61 [Rashid 2017], Seite 133ff

62 [Rashid 2017], Seite 61ff

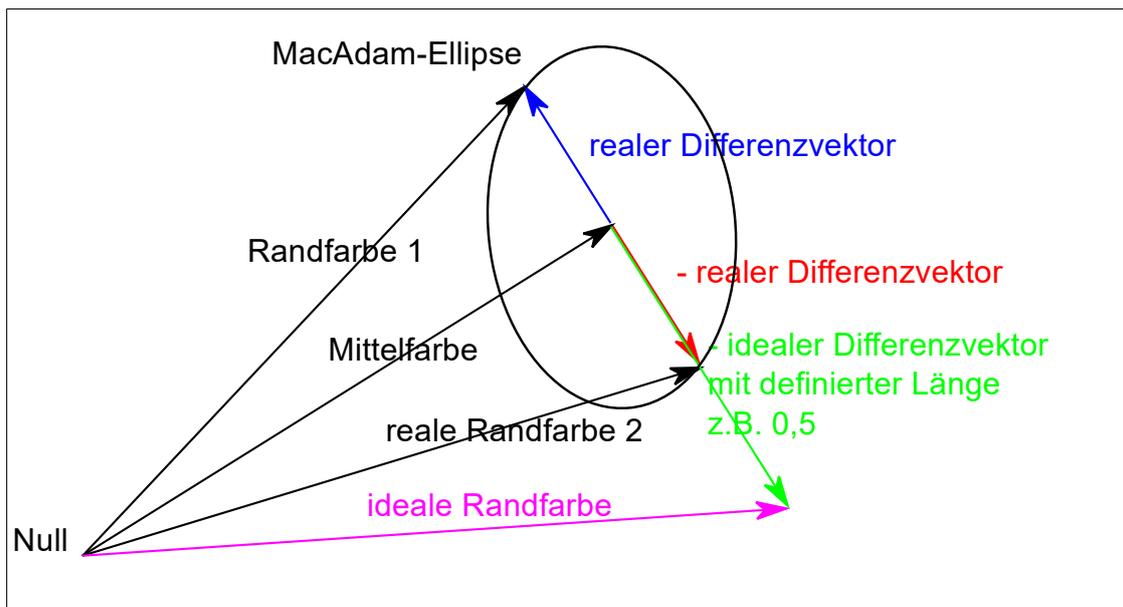


Abbildung 43: Skizze zur Errechnung des idealen Farbortes der zweiten Randfarbe im Gegensatz zum realen Farbort.

alen Ort als Zielwert vorgeben, von dem der reale Ort abweicht. Anders als bei üblichen neuronalen Netzen, ändert sich der Zielwert mit den Näherungsschritten, so dass er für den jeweiligen Durchlauf immer aktuell neu berechnet werden muss, was eine etwas abweichende Programmierung zu üblichen neuronalen Netzen verlangt. Dies resultiert aber aus der Tatsache, dass der ideale angestrebte Farbraum eben unbekannt ist. Es sind nur Farbabweichungen im realen CIE-XYZ-Farbraum bekannt, die in einem idealen Farbraum die Distanz 0,5 oder 1 haben sollen.

Nach einigen Berechnungen stellt sich heraus, dass es der Vereinfachung dient, wenn man nur die Endwerte zwischen 0 und 1 zulässt, da die Sigmoidfunktion nur solche Werte liefern kann, und erst im Nachgang eine Multiplikation mit hundert vornimmt, um Prozentangaben zu erhalten. Dabei hat sich wiederum herausgestellt, dass der Abstand von Mittelfarbe zu Randfarbe gemäß Abbildung 43 0,5 % betragen sollte. Bei der Programmierung hat es sich wiederum als sinnvoll erwiesen, zu unterscheiden, ob man die Ideale Farbe an Randfarbe1 festmacht oder an Randfarbe2. Programmiertechnisch lässt sich das durch Wechsel eines einzigen Vorzeichens bewerkstelligen. Wenn die Gesamt-Werte in einem Näherungsdurchlauf langsam immer besser werden, dann kommt der Punkt, wo es wieder schlech-

ter wird. Das rührt daher, dass das Verfahren nur das nächstliegende relative Optimum finden kann, wie das Tariq Rashid sehr schön im Kapitel 1 seines Buches unter der Überschrift „Wie aktualisieren wir eigentlich die Gewichte?“ beschreibt⁶³. Wenn dann das Vorzeichen entsprechend geändert wird, kann es wieder besser werden. Wenn es mit beiden Vorzeichen keine Verbesserung mehr gibt, scheint ein Lokales Optimum erreicht. Wenn dieses Optimum nicht befriedigt, muss die Berechnung mit neuen Zufallszahlen beginnen.

Üblicherweise braucht man zur Anwendung neuronaler Netze große Datensätze, bei denen die Eingangsgröße mit der gewünschten Ausgangsgröße verknüpft abgespeichert werden. Im Buch von Tariq Rashid sind das Bilder von handgeschriebenen Zahlen, die mit dem Wert der abgebildeten Zahl verknüpft sind. Alle Datensätze werden nun aufgeteilt. Der größte Teil wird zum Training des neuronalen Netzes verwendet. Der kleinere Teil wird später verwendet, um zu prüfen, wie gut das neuronale Netz arbeitet. In unserem Fall wurden wegen der relativ geringen Datenmenge alle Messwerte von MacAdam zum Training verwendet und dann nur eine Teilmenge zur Überprüfung der Güte des neuronalen Netzes verwendet.⁶⁴ Da die MacAdam-Ver-

63 [Rashid 2017], Seite 70ff

64 [Rashid 2017], Seite 134

suche offensichtlich den ganzen Farbraum, wenn auch nur in einer Helligkeitsebene, bedecken, geht jeder Bereich des Farbraumes beim Training mit in das neuronale Netz ein. Zusätzliche Messwerte dürften deshalb bei der Bewertung der Güte keine überraschenden Abweichungen zu Tage fördern.

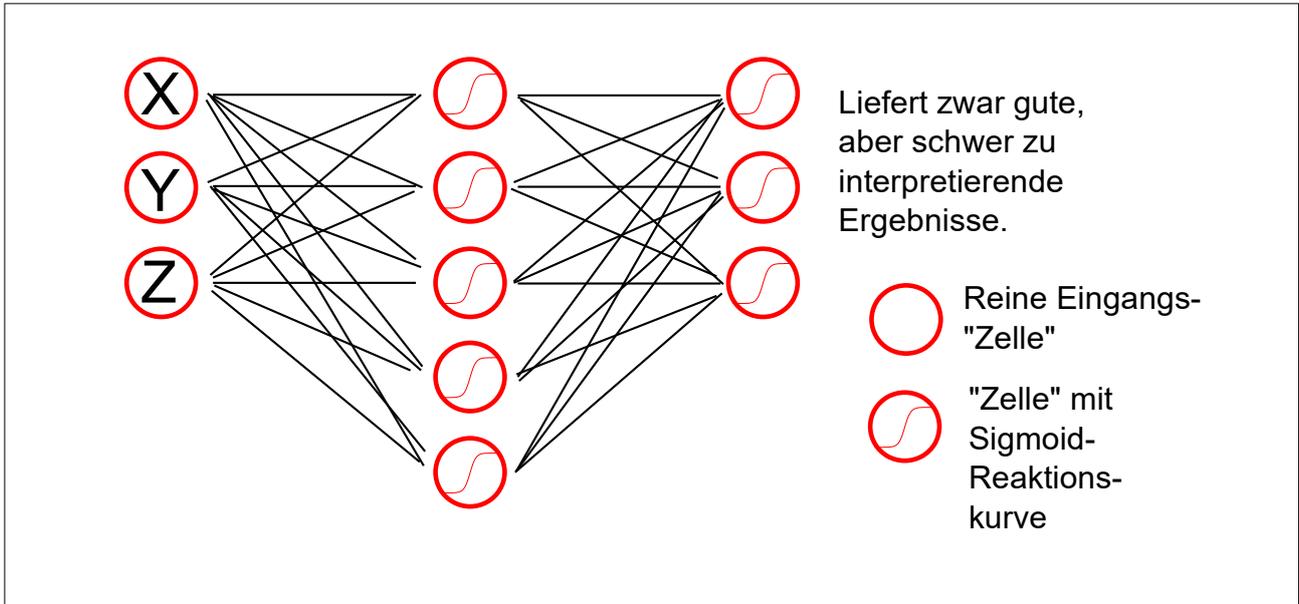


Abbildung 44: Architektur eines Neuronalen 3-5-3-Netzes

11 Architekturen Neuronaler Netze

11.1 Zwei- oder Dreidimensionales Ergebnis?

Die in Abbildung 40 gezeigte Architektur stellt nur die zunächst naheliegende Möglichkeit dar. Letzt-

lich muss bei Neuronalen Netzen ausprobiert werden, welche Architektur die geeignetste ist. Zunächst wurde die Anzahl der mittleren „Neuronen“ erhöht, wie in Abbildung 44 gezeigt. Dieses lieferte auch recht schnell gute Ergebnisse, nämlich ein Verhältnis von 7.2226 für das Verhältnis von minimaler zu maximaler MacAdam-Schwelle für die Allgemeinen Daten von MacAdam. Das ist sogar etwas besser als CIECAM02-UCS, das in der Tabelle aus Kapitel 9 einen Wert von 8.20 liefert. Bei der Betrachtung des Grafen in Abbildung 45 fallen allerdings seltsame „Schlingen“ in den gestrichelten Li-

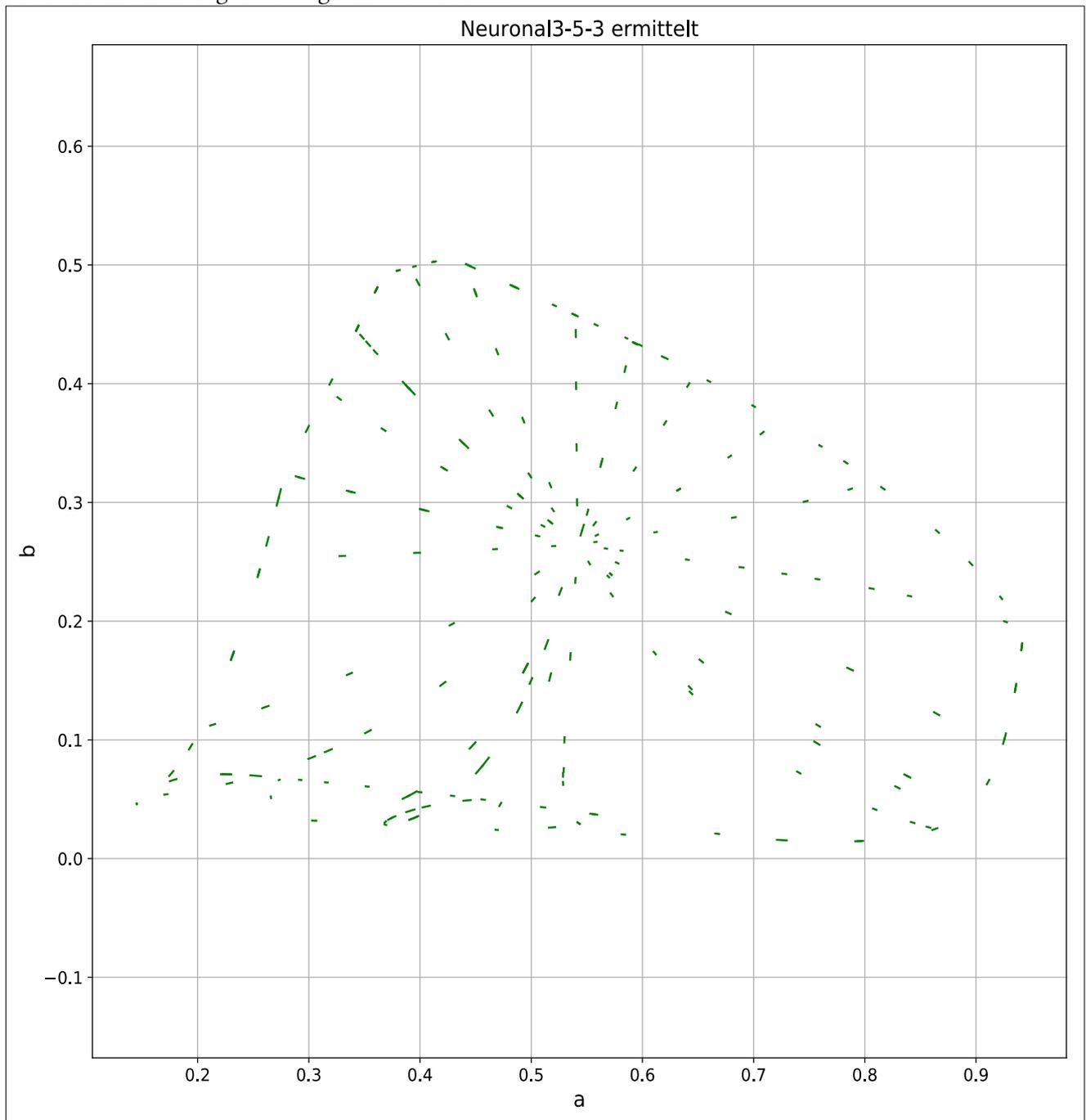


Abbildung 45: a-b-Ebene Allgemeine MacAdam-Messwerte für Neuronales 3-5-3 Netzwerk

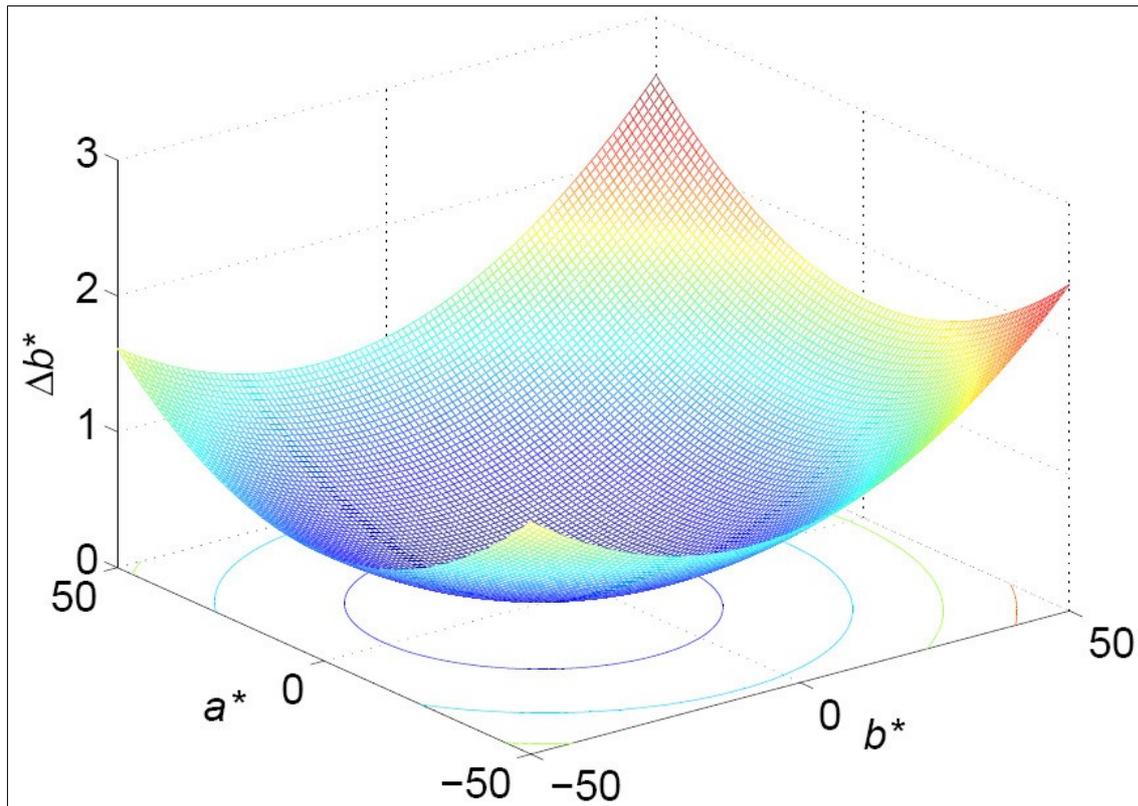


Abbildung 46: Blau-Gelb-Differenz b^* für die Farbschwellen als Funktion der Rot-Grün-Buntheit a^* und der Blau-Gelb-Buntheit b^* , aus [Kittelmann 2010], Seite 52

nien in der unteren Waagerechten auf. Wir sehen auf die Ebene, die a und b bilden. Wir haben aber noch eine dritte Koordinate L . Diese sieht man in der Abbildung nicht, sie ist aber vorhanden. Sie gibt dem Graf sozusagen die Möglichkeit, sich in die dritte Dimension zu wölben. In den Nachfolgenden Abbildungen 47 und 48 ist das nur nach längerer Betrachtung zu erkennen. Insbesondere bei der Abbildung 48, der L - b -Ebene, kann man sehr gut sehen, wie der ganze Graf seitlich gewölbt ist. Dies führte im weiteren Verlauf zu der Entscheidung, nicht mehr drei, sondern nur noch zwei Ausgabeneuronen zuzulassen, um eine ebene Fläche gleichsam zu erzwingen. Die so erreichten Güterwerte wurden auch mit großem Aufwand nicht besser als CIECAM02-UCS. Es ist also zu bezweifeln, dass die Beschränkung auf die Ebene eine gute Entscheidung war.

Manchmal führen vermeintliche Irrwege doch in die richtige Richtung. Wenn man fordert, dass alle Farben einer Helligkeit in einer Ebene liegen müssen, dann hat das unweigerlich zur Folge, dass um so mehr Nuancen wahrgenommen werden können, je gesättigter eine Farbe ist. Das widerspricht dem

Erfahrungswissen des Autors aus vielen Jahren in der Druckindustrie. Auch der aktuelle MedienStandard Druck [MedienStandard 2018] lässt zur Quantifizierung von Farbabständen CIEDE2000 zu. Dieses geht gerade nicht davon aus, dass das Auge in gesättigten Bereichen besonders empfindlich wäre. Auch eine Arbeit von Philipp Kittelmann aus dem Jahr 2010 [Kittelmann 2010] weist mit wissenschaftlicher Gründlichkeit in diese Richtung. Kittelmann untersucht wie weit Farbschwellen von anderen Parametern abhängen. Er kommt zu dem Ergebnis, dass die Farbschwellen sowohl von der Helligkeit als auch von der Sättigung abhängen⁶⁵. Die Grafen in seiner Arbeit, welche die entsprechenden Abhängigkeiten darstellen, erinnern stark an Teile von Kugeloberflächen, siehe Abbildung 46.⁶⁶

Es ist also wahrscheinlich, dass einheitlich lange Farbschwellen bei gleicher Helligkeit gar nicht in einer Ebene dargestellt werden können, sondern nur auf einer gekrümmten Fläche.

65 [Kittelmann 2010], Seite 50ff

66 [Kittelmann 2010], Seite 52

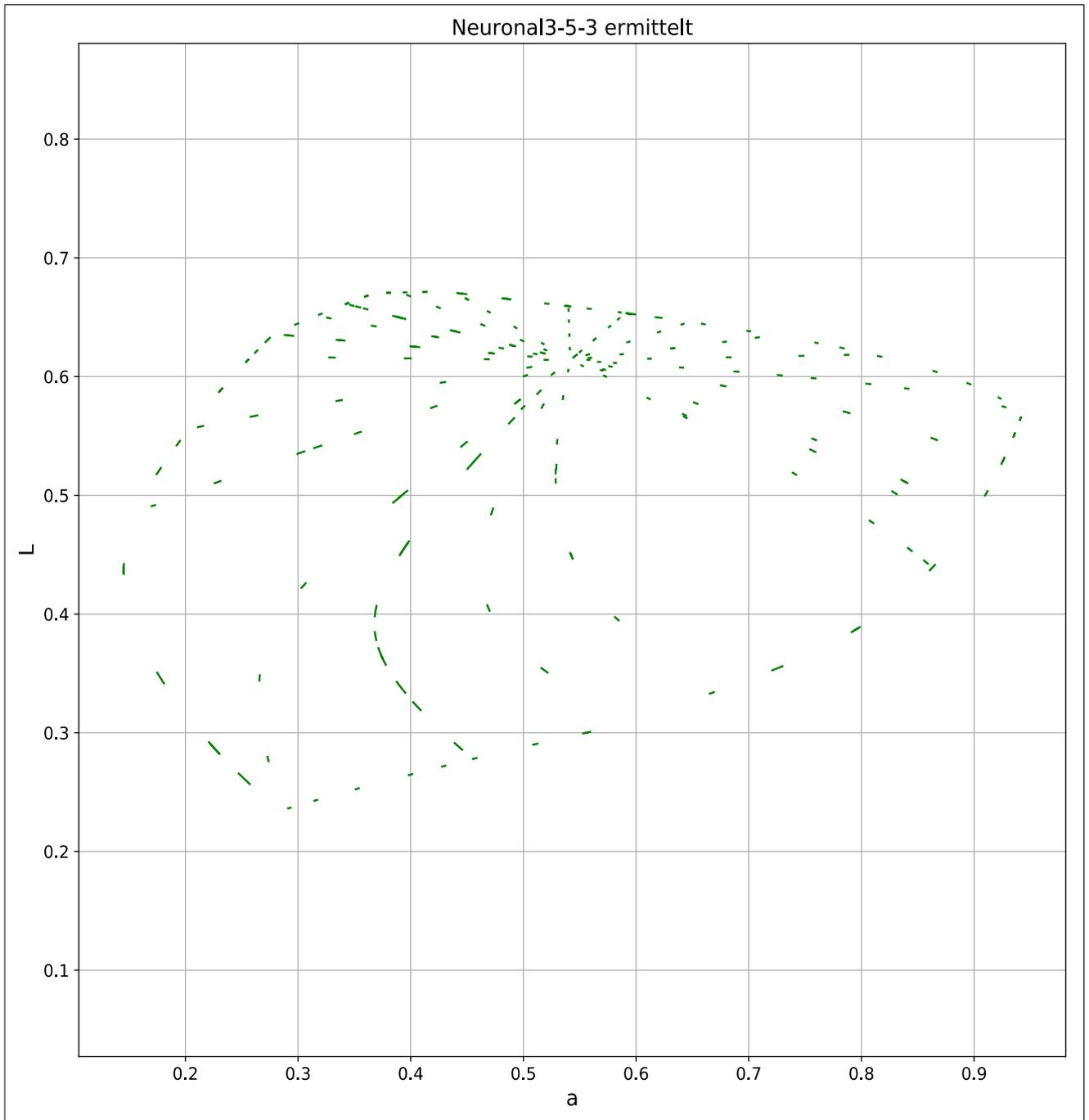


Abbildung 47: L-a-Ebene Allgemeine MacAdam-Messwerte für Neuronales 3-5-3 Netzwerk

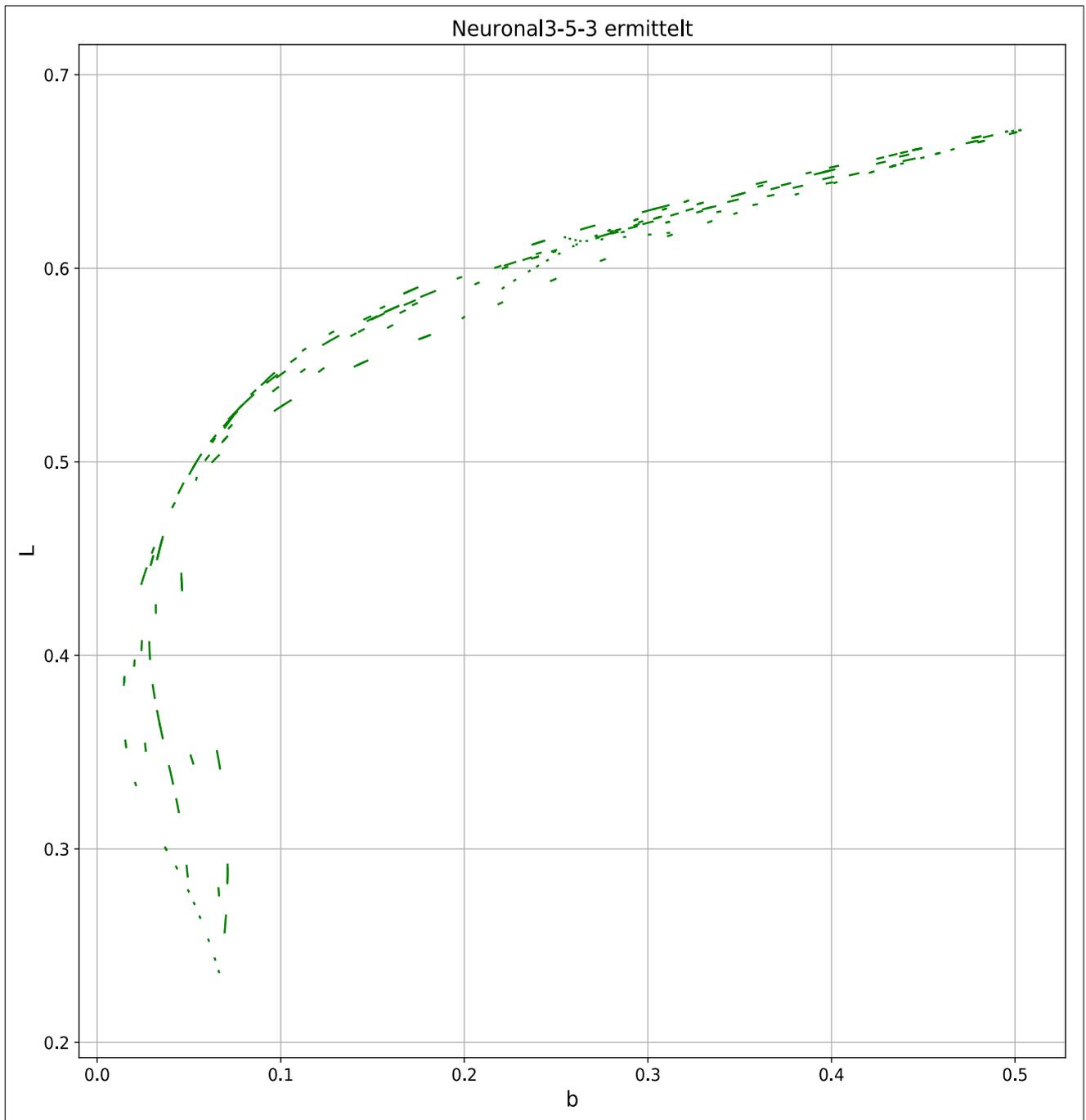


Abbildung 48: L-b-Ebene Allgemeine MacAdam-Messwerte für Neuronales 3-5-3 Netzwerk

Die mathematische Lösung wäre also, die Farbebene sich um eine virtuelle dritte Dimension krümmen zu lassen. Wenn man nun alle wahrnehmbaren Farben in ein System packen will, muss man das in einem vierdimensionalen Raum machen. Ein solcher sollte gemäß Ockhams Rasiermesser erst untersucht werden, wenn eine Dreidimensionale Lösung nicht den erhofften Erfolg bringt. Bemerkenswert ist, dass die gekrümmte Ebene einem Vor-

schlag von MacAdam ähnelt, der die von ihm gewählte Helligkeitsebene ebenfalls krümmt, um einheitliche Farbabstände zu erhalten, wie das in Abbildung 49 dargestellt ist⁶⁷. Noch frappierender ist die Ähnlichkeit zu einem Vorschlag von Walter S. Stiles, der nach MacAdam ähnliche Experimente durchgeführt hat und ebenfalls eine Krümmung der

67 [Wyszecki 1982], Seite 668

Ebene gleicher Helligkeit vorschlägt, wie in Abbildung 50 gezeigt ist.⁶⁸

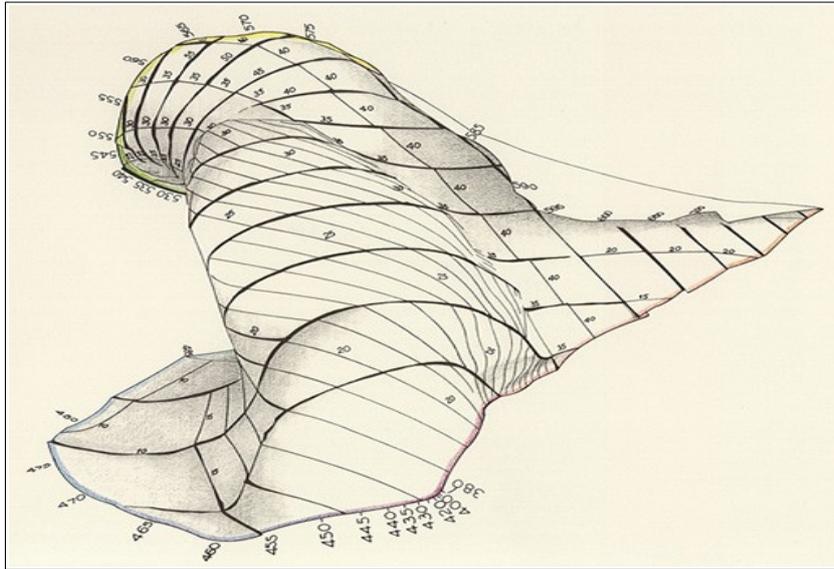


Abbildung 49: Gekrümmte Fläche einheitlicher Helligkeit nach MacAdam. Aus https://www.colorsystem.com/?page_id=907&lang=en gelesen am 2.10.2019 analog zur Abbildung in [Wyszecki 1982], Seite 668

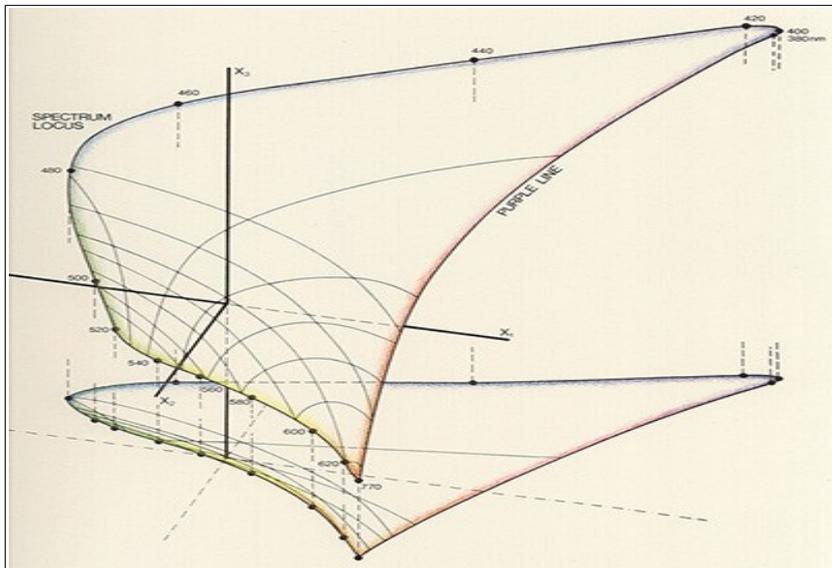


Abbildung 50: Gekrümmte Fläche einheitlicher Helligkeit nach Stiles. Aus https://www.colorsystem.com/?page_id=915 gelesen am 2.10.2019 analog zur Abbildung in [Wyszecki 1982], Seite 669

68 [Wyszecki 1982], Seite 669

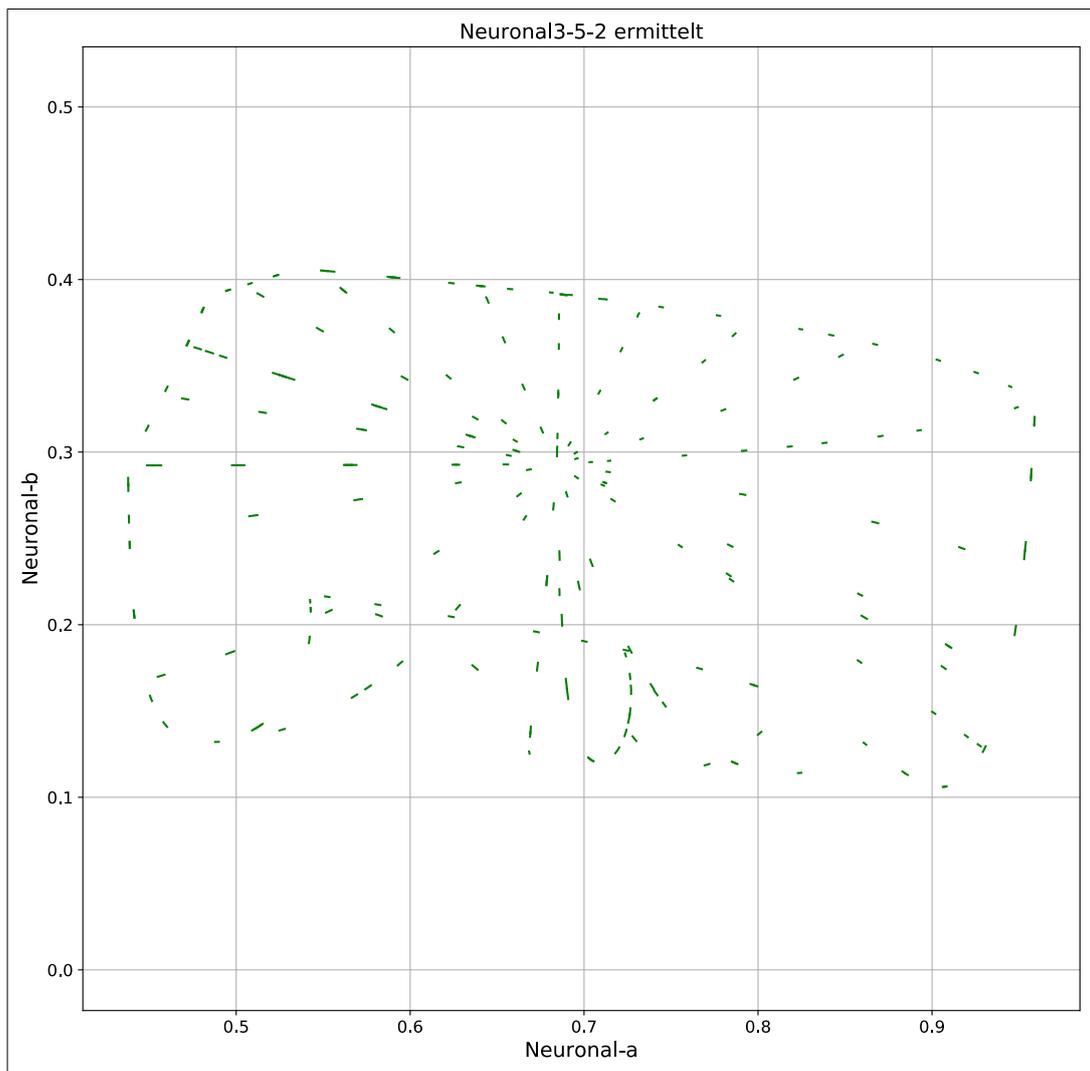


Abbildung 51: Werte für die allgemeinen Messwerte von MacAdam, eingetragen in ein Diagramm, nach Berechnungen eines Neuronalen 3-5-2-Netzes.

11.2 Unterschiedliche Architekturen und ihre Ergebnisse

Die weiteren zweidimensionalen Architekturen oder genauer ausgedrückt, diejenigen Architekturen, die nur zwei Ausgabeneuronen haben, sollen hier kurz mit ihrer Architektur und ihren Ergebnissen dargestellt werden. Die Güterwerte beziehen sich auf die reinen MacAdam-Messergebnisse für die Ellipsen und sind trotz großem Aufwand allesamt schlechter als CIECAM02-UCS.

Aus Kapitel 6.5 wissen wir bereits, dass der Farbraum CIECAM02 für die Messwerte der MacAdam-Ellipsen ein Verhältnis von maximaler zu minimaler Strichlänge von 3.69 liefert. Die zweidimensionalen Ausgaben über neuronale Netze haben in den eigenen Versuchen bei einer 3-5-2-Architektur 4,83 und bei einer 3-3-4-2-Architektur 4,8. Also sind beide zweidimensionale Ergebnisse schlechter, als die Ergebnisse mit CIECAM02. Siehe dazu die Abbildungen 51 bis 56.

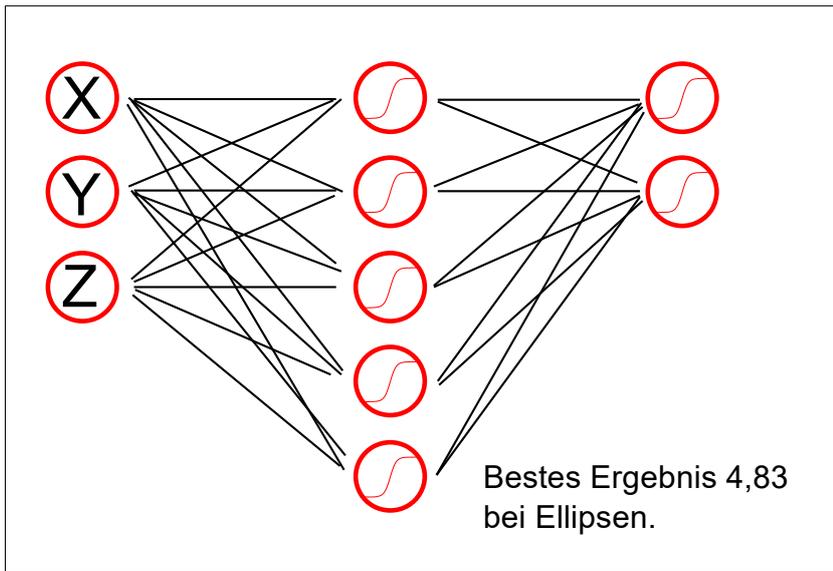


Abbildung 52: Architektur eines Neuronalen 3-5-2-Netzes

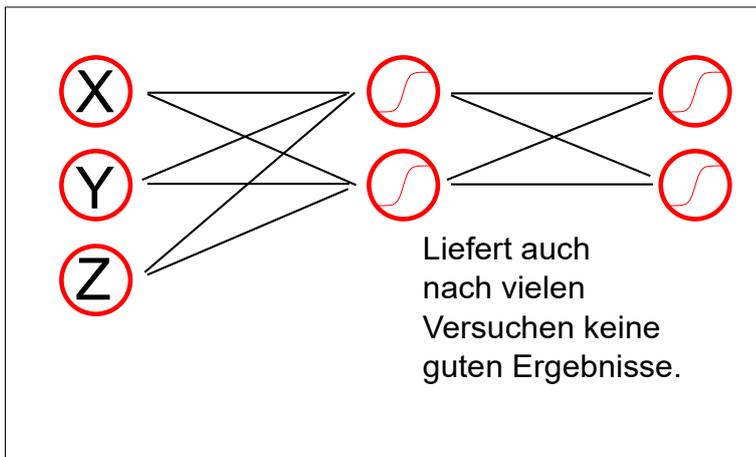


Abbildung 53: Architektur eines Neuronalen 3-2-2-Netzes

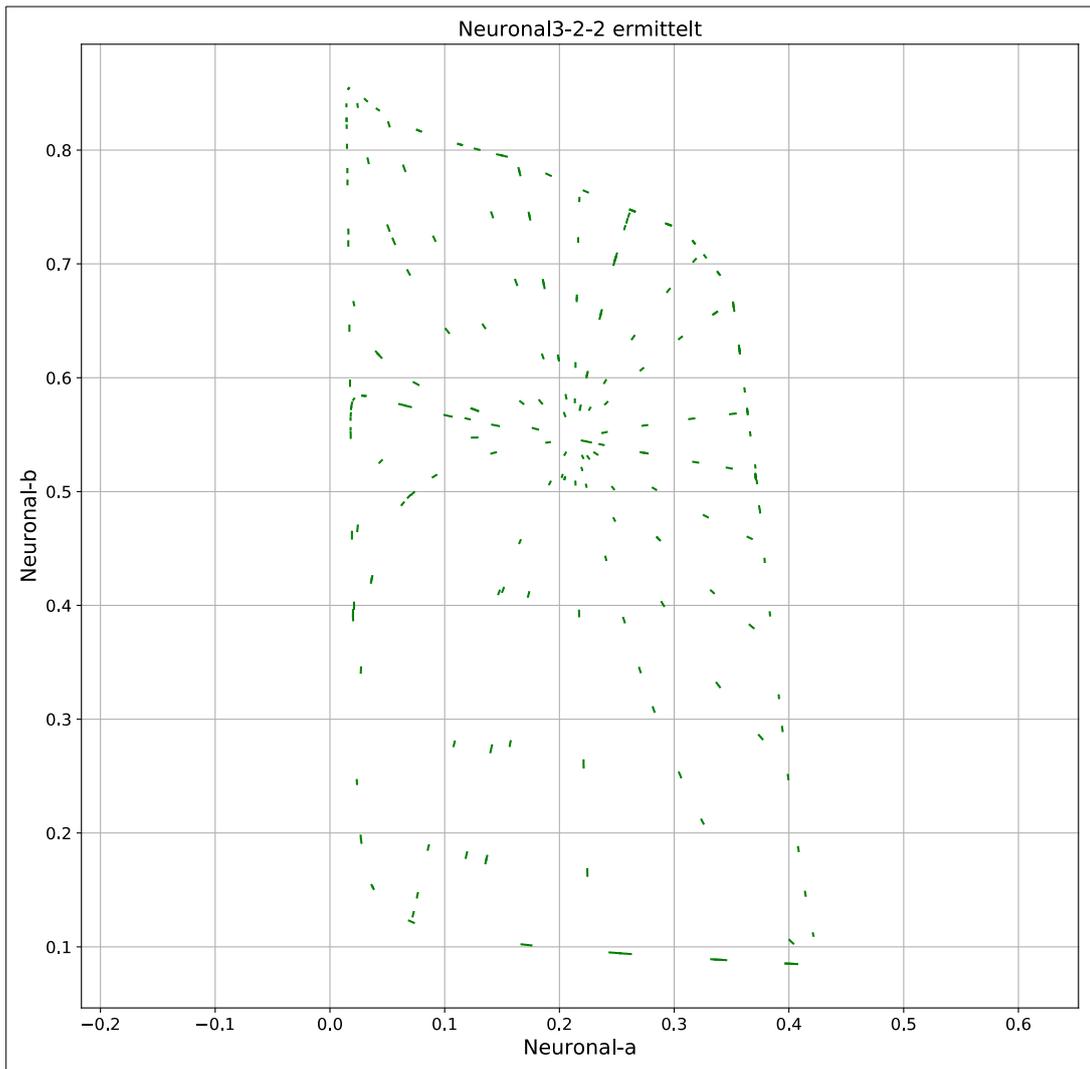


Abbildung 54: Werte für die allgemeinen Messwerte von MacAdam, eingetragen in ein Diagramm, nach Berechnungen eines Neuronalen 3-2-2-Netzes.

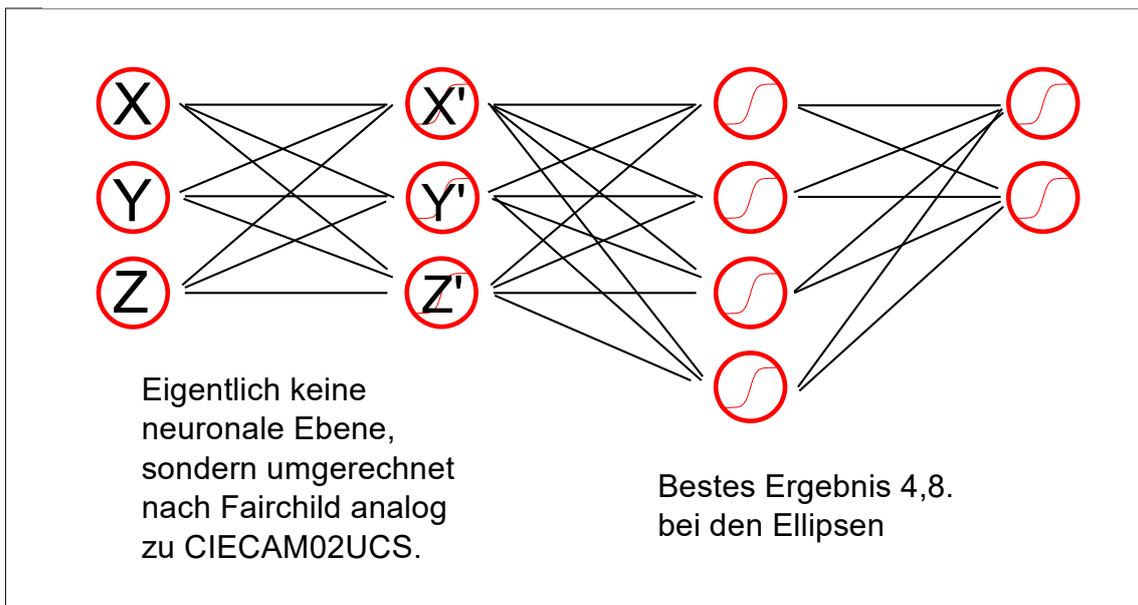


Abbildung 55: Architektur eines Neuronalen 3-3-4-2-Netzes

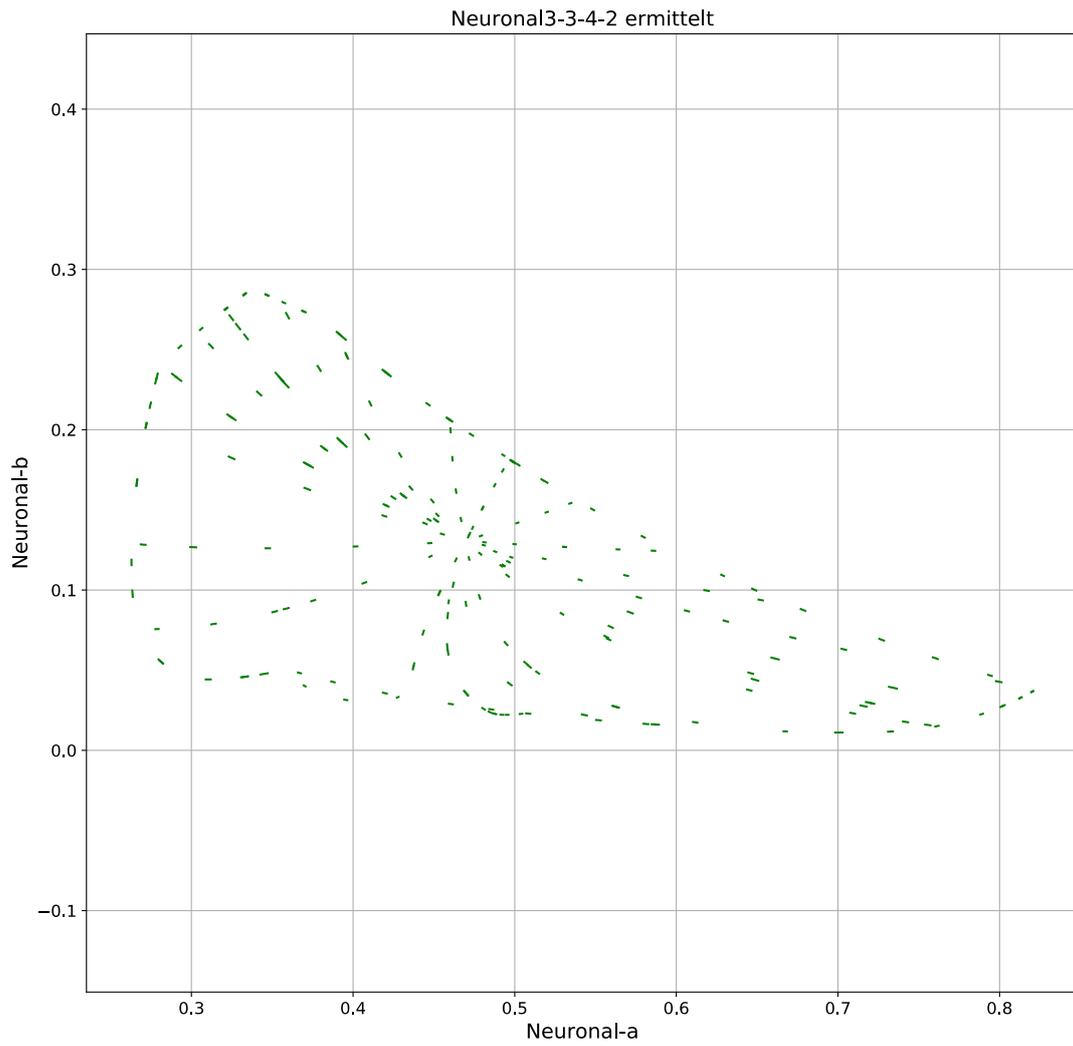


Abbildung 56: Werte für die allgemeinen Messwerte von MacAdam, eingetragen in ein Diagramm, nach Berechnungen eines Neuronalen 3-3-4-2-Netzes.

12 Fazit

Das eigentliche Ziel dieser Arbeit, Entwicklung eines gleichabständigen Farbraumes mittels neuronaler Netze, basierend auf klassischen Messergebnissen, wurde in dieser Arbeit nicht erreicht. Es wurde aber gezeigt, warum sich allein mit Hilfe der MacAdam-Untersuchung kein vollständiger gleichabständiger Farbraum ermitteln lässt. MacAdam begrenzt sich bei seinen Versuchen nämlich leider ganz bewusst auf eine Helligkeitsebene. Von dieser Helligkeitsebene auf die Farbeigenschaften anderer Helligkeitsebenen zu schließen, ist gemäß der Arbeit von Philipp Kittelmann⁶⁹ nicht zulässig. Aus der Arbeit von Kittelmann und der vorliegenden lässt sich mit einigem Recht schließen, dass sich Farben mit einheitlicher Helligkeit nicht auf einer Ebene darstellen lassen, wenn man Farbdifferenzen mit gleichen Abständen in allen Bereichen darstellen will. Die genaue spektrale Verteilung aller einzelnen von MacAdam verwendeten Filter ist zwar nicht mehr hinreichend genau nachvollziehbar. Aber die Messwerte von MacAdam konnten immerhin modellhaft als Gütekriterium für die Gleichmäßigkeit von Farbabständen verwendet werden. Interessant dabei ist, dass die Abweichungsverhältnisse des derzeit für gut befundenen CIECAM02-UCS denen ähneln, die mittels neuronaler Netze bzw. Deep Learnings ermittelt wurden. Es liegt die Vermutung nahe, dass die Abweichungen auf die selben Messabweichungen der CIE-XYZ-Normvalenzen, der prinzipiellen Beschränkung von MacAdam auf eine Helligkeitsebene und der Messfehler von MacAdam beruhen. Mit dieser Arbeit wird aber belegt, dass mit modernen Experimenten mit genaueren und umfassenderen Daten und anschließender Verrechnung mittels Deep Learning ein vielversprechender Weg besteht, ein so dringend benötigtes gleichabständiges Farbsystem zu erforschen, welches die Wahrnehmung der Farbabstände eines normalsichtigen Beobachters zuverlässig und einheitlich beschreibt. Erfolgversprechende Verrechnungsmethoden und ihre Schwierigkeiten wurden in dieser Arbeit aufgezeigt.

13 Ausblick

An der Hochschule Mittweida steht eine spektral programmierbare Lichtquelle zur Verfügung. Dabei fällt annähernd weißes Licht auf einen DLP-Chip. Die einzelnen Spiegel auf diesem Chip können so eingestellt werden, dass sehr schmalbandige Bereiche des Spektrums mehr oder weniger reflektiert werden. Durch gezielte Ansteuerung kann in weiten Bereichen Licht beliebiger Zusammensetzung erzeugt werden. Da es sich um einen Videochip handelt, können die Spiegel sehr schnell geschaltet werden. Denkbar wäre also ein Versuchsaufbau, der ein bestimmtes Licht in etwa 0,5s-Abstand mit einem Licht pulsieren lässt, das in bestimmten Grenzen von den Probanden verändert werden kann. Solange kein Pulsieren wahrgenommen wird, soll der Proband die Änderung verstärken, so lange bis erstmalig das Pulsieren gesehen wird. Die Emissionskurven können sehr genau aufgezeichnet werden. Die Variationen müssen natürlich geschickt gewählt werden und auch die Bedienelemente für den Versuch müssen möglichst einfach gehalten werden. Realistisch sollte die einzelne Messung in 10s erledigt sein. Es können also knapp 400 Messungen pro Stunde gemacht werden. Auch wenn diese Messungen wahrscheinlich nicht so penibel wie die von MacAdam sein werden, können durch die mögliche große Anzahl an Versuchen und die größere Variationsbreite an Probanden trotzdem genauere Werte gemittelt werden. Insbesondere ließe sich unmittelbar von sehr unterschiedlichen spektralen Zusammensetzungen mit unterschiedlichsten Helligkeiten ausgehen, während MacAdam hier doch vergleichsweise begrenzt war. Da mit der vorliegenden Arbeit nachgewiesen wurde, dass sich solche Differenzmessungen von Farben zu kompletten Farbsystemen verrechnen lassen, wäre das den Aufwand bestimmt wert.

69 [Kittelmann 2010], Seite 84

Literaturverzeichnis

[Asano 2015]

Yuta Asano: Individual Colorimetric Observers for Personalized Color Imaging, PhD Dissertation, Rochester Institute of Technology, ergänzt durch: http://www.rit.edu/cos/colorscience/re_AsanoObserverFunctions.php

[CIE 1926]

CIE. (1926). Commission Internationale de l'Éclairage Proceedings, 1924. Cambridge: Cambridge University Press.

[CIE 1932]

CIE. (1932). Commission Internationale de l'Éclairage Proceedings, 1931. Cambridge: Cambridge University Press.

[cvrl]

<http://www.cvrl.org/> The Colour & Vision Research laboratory database originated at the University of California at San Diego in 1995. In 2001, the database and CVRL moved to the Institute of Ophthalmology at University College London, where they continue today.

[Dartnall 1980]

J. K. Bowmaker, H. J. A. Dartnall: Visual pigments of rods and cones in a human retina. In: The Journal of Physiology. Bd. 298, Januar 1980, ISSN 0022-3751, S. 501–511

[Davis 1931]

U.S. Department of Commerce, Bureau of Standards, Miscellaneous Publication, No. 114; Filters for the Reproduction of Sunlight and Daylight and the Determination of Color Temperature, Raymond Davis, K.S. Gibson, January 21, Washington 1931

[DfwG-Report]

DfwG Report, Informationen für Mitglieder der Deutschen farbwissenschaftlichen Gesellschaft e.V., Veröffentlicht in der Deutschen Nationalbibliothek unter ISSN 1860-2835, Typisch 3 Ausgaben pro Jahr

[ECI 1999]

ECI (European Color Initiative) Richtlinien: Richtlinien für die medienneutrale Farbdatenverarbeitung gemäß des ICC-Standards, ECI, 14.11.1999, Version 1.1

[Fairchild 2016]

Yuta Asano, Mark D. Fairchild , Laurent Blondé: Individual Colorimetric Observer Model, Published: February 10, 2016 <https://doi.org/10.1371/journal.pone.0145671>

[Fairman 1997]

Hugh S. Fairman, Michael H. Brill, Henry Hemmendinger: How the CIE 1931 color-matching functions were derived from Wright-Guild data, First published: February 1997

[Kamm 1987]

Kamm, Prof. Dr. Gunther: Farbmatrik mit Blick auf Repro und Druck ; Zeitschrift „Der Druckspiegel“, Ausgabe 10/1987

[Kittelmann 2010]

Diplom-Ingenieur Philipp Kittelmann: Visuelle Beurteilung von kleinen und großen Farbunterschieden und Beschreibung mit Farbabstandsformeln, Fakultät IV – Elektrotechnik und Informatik der Technischen Universität Berlin zur Erlangung des akademischen Grades Doktor der Ingenieurwissenschaften, Berlin 2010

[Kodak 1928]

WRATTENLIGHT FILTERS, NINTH EDITION, REVISED, Eastman Kodak Company, ROCHESTER, NEW YORK 1928

[Lippe 2005]

Lippe, Wolfram-Manfred, Soft-Computing. Mit Neuronalen Netzen, Fuzzy-Logic und Evolutionären Algorithmen, Verlag: Springer, Berlin (2006)

[Luo 2000]

M. R. Luo,* G. Cui, B. Rigg: The Development of the CIE 2000 Colour-Difference Formula: CIEDE2000, Colour & Imaging Institute, University of Derby, UK, aus COLOR research and application, Volume 26, Number 5, October 2001, Seite 340 bis 350

[MacAdam 1942]

David L. MacAdam, Visual Sensitivities to Color Differences in Daylight, Journal of the OPTICAL SOCIETY of AMERICA, VOLUME 32, MAY, 1942 NUMBER 5, Seite 247f

[MacEvoy 2015]

<https://www.handprint.com/HP/WCL/color2.html> Last revised 08.1.2015 • © 2015 Bruce MacEvoy, gelesen am 4.7.2017. Darstellungen dieser Web-Site werden häufig

verwendet, wegen der nicht immer klaren Quellenlage dort, allerdings in der Regel geprüft an [Wyszecki 1982]

[MedienStandard 2018]

MedienStandard Druck 2018 – Technische Richtlinien, für Daten, Prüfdruck und Auflagendruck (PDF), © 2016–2018 Bundesverband Druck und Medien e. V. (bvdn)

[Moroney 2002]

Nathan Moroney, Mark D. Fairchild, Robert W.G. Hunt, Changjun Li, M. Ronnier Luo and Todd Newman :The CIECAM02 Color Appearance Model, <https://scholarworks.rit.edu/other/143>, IS&T/SID Tenth Color Imaging Conference

[Peed 1920]

Transmission of Wratten filters, detailed numerical information compiled by Allie C Peed Jr for the Eastman Kodak Company, vermutlich um 1920

[Richter 1996]

Richter, Klaus: Computergrafik und Farbmeterik. Farbsysteme, PostScript, geräteunabhängige CIE-Farben. - Taschenbuch 1996, ISBN: 9783800717750 [ED: Softcover], [PU: Technische Akademie Wuppertal - VDE-Verlag Berlin 1996], 15,2*21,2 cm, 288 Seiten,

[Rashid 2017]

Tariq Rashid, Frank Langenau (Übersetzung): Neuronale Netze selbst programmieren, Ein verständlicher Einstieg mit Python, April 2017, O'Reilly, ISBN Print: 978-3-96009-043-4

[Safdar 2017]

Muhammad Safdar, Guihua Cui, Youn Jin Kim, Ming Ronnier Luo: Perceptually uniform color space for image signals including high dynamic range and wide gamut, Vol. 25, No. 13 | 26 Jun 2017 | OPTICS EXPRESS 15131 Optical Society of America

[Stockman 1993]

Andrew Stockman, Donald I. A. MacLeod, and Nancy E. Johnson: Spectral sensitivities of the human cones, Vol. 10, No. 12/December 1993 Journal of the Optical Society of America, Seite 2491ff.

[Wasilewski Kostova 2009]

Emil Wasilewski, Nataliya Kostova: CIECAM02 Standard Color Appearance Model; Universität Koblenz-Landau, Fachbereich 4, Institut für Computervisualistik, 2009

[Wikipedia] Suchbegriff, Datum

[https://de.wikipedia.org/wiki/\"Suchbegriff\"](https://de.wikipedia.org/wiki/\) und Datum, an dem die Seite gelesen wurde

[Wikipedia en]

[https://en.wikipedia.org/wiki/\"Suchbegriff\"](https://en.wikipedia.org/wiki/\) und Datum, an dem die Seite gelesen wurde auf der englischsprachigen Wikipedia-Seite

[Wright 1928]

Wright, W. D. (1928). A re-determination of the trichromatic coefficients of the spectral colours. Transactions of the Optical Society, 30, 141-164.

[Wyszecki 1982]

Wyszecki, Günther , Stiles, W. S.: Color Science, Concepts and Methods, Quantitative Data Formulae, Second Edition, Repr. of the 1982 ed. 2000 - Wiley & Sons

[Xu 1997]

Jun Xu, Joel Pokorny, and Vivianne C. Smith: Optical density of the human lens; Journal of the Optical Society of America A Vol. 14, Issue 5, pp. 953-960 (1997)

[Zumtobel 2018]

Zumtobel (Firma): Licht-Handbuch für den Praktiker - Ihr kompaktes Nachschlagewerk, 11. Auflage: April 2018

Anlagen

Anlage 1:	Listings für Grafikausgabe	Seite 67
Anlage 2:	Listings für Neuronale Netze	Seite 83

MacAdam in Diagrammen

Einbindung Farb-Bibliothek

die Bibliothek findet man unter: <http://colour-science.org/tutorial/> (<http://colour-science.org/tutorial/>) oder unter Github <https://github.com/colour-science/colour> (<https://github.com/colour-science/colour>) Manual und Source-Code <https://colour.readthedocs.io/en/latest/manual.html> (<https://colour.readthedocs.io/en/latest/manual.html>) DIN99 Formeln, wahrscheinlich noch nicht implementiert <https://github.com/colour-science/colour/blob/develop/colour/models/din99.py> (<https://github.com/colour-science/colour/blob/develop/colour/models/din99.py>)

```
In [1]: %matplotlib inline
import numpy as np
import scipy.special
import matplotlib
import colour
from pprint import pprint
import colour.plotting
from colour.plotting import *

colour.filter_warnings(True, False)
colour.plotting_defaults()

import colour.colorimetry as colorimetry
import colour.colorimetry.dataset as dataset

cmfs = colour.STANDARD_OBSERVERS_CMFS['CIE 1931 2 Degree Standard Observer']
illuminant = colour.ILLUMINANTS_RELATIVE_SPDS['A']

D:\Programme\anaconda3\lib\site-packages\colour\utilities\verbose.py:141: ColourWarning: "x" independent
variable is not uniform, unpredictable results may occur!
  warn(*args, **kwargs)
D:\Programme\anaconda3\lib\site-packages\colour\utilities\verbose.py:141: ColourWarning: "colour.filter_w
arnings" object access is deprecated and will change to "colour.utilities.filter_warnings" in a future re
lease.
  warn(*args, **kwargs)
```

Einlesen der Daten für die Filter

Die CIE-XYZ, CIE-Yxy und weitere Daten für die Wratten-Filter und einige Spezialfilter werden eingelesen

```
In [2]: import csv
lst = []
with open('MacAdamTableI.csv', newline='') as f:
    reader = csv.reader(f)
    for row in reader:
        lst.append(row)
        print(row)
MacAdamTableI = lst

['Filternummer', 'Bestandteile', 'Helligkeit', 'X', 'Y', 'Z', 'm', 'x', 'y']
['0', '', '', '', '', '', '', '', '', '']
['0', '', '', '', '', '', '', '', '', '']
['4', '34,38', '0.07', '0.0164', '0.0028', '0.08855', '0.10775', '0.152', '0.026']
['0', '', '', '', '', '', '', '', '', '']
['6', '65,8', '1.25', '0.00059', '0.00265', '0.00066', '0.0039', '0.15', '0.68']
['0', '', '', '', '', '', '', '', '', '']
['8', '75', '0.42', '0.00071', '0.0028', '0.00575', '0.00911', '0.077', '0.302']
['0', '', '', '', '', '', '', '', '', '']
['0', '', '', '', '', '', '', '', '', '']
['0', '', '', '', '', '', '', '', '', '']
['0', '', '', '', '', '', '', '', '', '']
['0', '', '', '', '', '', '', '', '', '']
['14', 'x1', '0.24', '0.00044', '0.00276', '0.00023', '0.00343', '0.128', '0.805']
['15', '3(63)', '', '0.00027', '0.0028', '0.00076', '0.00383', '0.07', '0.736']
['16', '3(74)', '0.1', '0.00062', '0.0027', '0.00011', '0.00353', '0.18', '0.785']
['17', '2(73)', '0.5', '0.00256', '0.00272', '0', '0.00528', '0.485', '0.514']
['0', '', '', '', '', '', '', '', '', '']
['0', '', '', '', '', '', '', '', '', '']
...

```

Die allgemeinen Messdaten von MacAdam werden eingelesen

Die Daten, die nicht speziell für die Ellipsen werden eingelesen. Vorzugsweise sind das weit entfernte Farborte im CIE-Yxy-Farbraum, die ausgemischt und auf Farbschwellen untersucht werden.

```
In [3]: """          row = [float(i) for i in row]"""
import csv
lst = []
with open('MacAdamTableII.csv', newline='') as f:
    reader = csv.reader(f)
```

```

for row in reader:
    lst.append(row)
    print(row)
MacAdamTableII = lst

```

```

['0° Filternummer', '90° Filternummer', '?° Average setting', '??° Standard Deviation', 'x', 'y', "Distance from 0° point=", '?s Standard Deviation', '?y/?x', 'Prüfung ob X-Filter in A oder B enthalten']
['4', '23', '83.8', '0.239', '0.286', '0.286', '0.294', '0.00271', '1.94', 'FALSE']
['4', '23', '79.1', '0.256', '0.259', '0.233', '0.232', '0.00370', '1.94', 'FALSE']
['4', '23', '79.1', '0.230', '0.259', '0.233', '0.232', '0.00332', '1.94', 'FALSE']
['4', '23', '74.1', '0.247', '0.230', '0.177', '0.171', '0.00290', '1.94', 'FALSE']
['4', '23', '69.0', '0.382', '0.208', '0.135', '0.122', '0.00312', '1.94', 'FALSE']
['4', '23', '68.5', '0.304', '0.206', '0.131', '0.117', '0.00242', '1.94', 'FALSE']
['4', '23', '58.4', '0.335', '0.180', '0.081', '0.062', '0.00150', '1.94', 'FALSE']
['4', '23', '42.1', '0.477', '0.162', '0.045', '0.022', '0.00067', '1.94', 'FALSE']
['4', '23', '25.1', '0.563', '0.155', '0.031', '0.006', '0.00028', '1.94', 'FALSE']
['100', '23', '43.7', '0.823', '0.259', '0.254', '0.056', '0.00213', '2.21', 'TRUE']
['66', '23', '73.9', '0.284', '0.268', '0.280', '0.229', '0.00163', '1.16', 'FALSE']
['66', '23', '60.9', '0.314', '0.216', '0.219', '0.139', '0.00184', '1.16', 'FALSE']
['66', '23', '52.0', '0.334', '0.185', '0.183', '0.102', '0.00156', '1.16', 'FALSE']
['66', '23', '47.2', '0.430', '0.169', '0.165', '0.080', '0.00176', '1.16', 'FALSE']
['66', '23', '45.1', '0.423', '0.166', '0.161', '0.072', '0.00154', '1.16', 'FALSE']
['66', '23', '32.8', '0.507', '0.142', '0.133', '0.035', '0.00116', '1.16', 'FALSE']
['66', '23', '16.7', '0.642', '0.125', '0.113', '0.008', '0.00064', '1.16', 'FALSE']

```

```

In [4]: FarbeM = np.zeros((3)).astype(float)
Tripel = np.zeros((3,3)).astype(float)
#Deltas = np.zeros((0,3,3)).astype(float)
Deltas = np.zeros((0,3)).astype(float)
for row in MacAdamTableII[1:]:
    # if row[8]=='FALSE':
    if row:
        xySollfarbe = np.array([float(row[4]),float(row[5])])
        Filter0 = MacAdamTableI[int(row[0])-1]
        Filter90 = MacAdamTableI[int(row[1])-1]
        FarbeV = np.array(Filter0[3:6]).astype(float)
        FarbeU = np.array(Filter90[3:6]).astype(float)
        Winkel = np.array([float(row[2]) - float(row[3]), float(row[2]), float(row[2]) + float(row[3])])
        for n in range(0, 3):
            U = (np.sin(float(Winkel[n])/180*np.pi))**2
            V = 1 - U
            FarbeM = FarbeV*V + FarbeU*U
            Tripel[n] = FarbeM
        # Deltas = np.append(Deltas, [np.transpose(Tripel)], axis=0)
        Deltas = np.append(Deltas, [FarbeM], axis=0)
#Deltas = np.delete(Deltas, (0), axis=0)
print(Deltas)

```

```

[[0.00295129 0.00293824 0.00446102]
 [0.00293886 0.00293837 0.00438329]
 [0.00292689 0.00293849 0.00430846]
 ...
 [0.00774377 0.00279949 0.0004905 ]
 [0.00773471 0.00279957 0.00040785]
 [0.00772648 0.00279965 0.00033275]]

```

Ausschließlich die Messwerte für die Ellipsen nach MacAdam

Nur die Daten für die Ellipsen werden eingelesen

```

In [3]: """ row = [float(i) for i in row]"""
import csv
lst = []
with open('MacAdamTableIII.csv', newline='') as f:
    reader = csv.reader(f)
    for row in reader:
        lst.append(row)
        print(row)
MacAdamTableIII = lst

```

```

['0° Filternummer', '90° Filternummer', '?° Average setting', '??° Standard Deviation', 'x', 'y', '?y/?x', '?s Standard Deviation', 'Prüfung ob X-Filter in A oder B enthalten']
['80', '16', '63.1', '0.281', '0.258', '0.450', '-5.18', '0.00388', 'FALSE']
['84', '35', '57.1', '0.445', '0.258', '0.450', '-500.00', '0.00424', 'TRUE']
['23', '14', '44.1', '0.586', '0.258', '0.450', '-2.72', '0.00413', 'TRUE']
['64', '33', '65.0', '0.292', '0.258', '0.450', '1.66', '0.00323', 'FALSE']

```

```
[ '90', '28', '48.8', '0.241', '0.258', '0.450', '-0.47', '0.00188', 'TRUE' ]
[ '90', '28', '49.1', '0.240', '0.258', '0.450', '-0.47', '0.00186', 'TRUE' ]
[ '74', '57', '66.5', '0.320', '0.258', '0.450', '3.26', '0.00398', 'FALSE' ]
[ '23', '14', '43.8', '0.563', '0.258', '0.450', '-2.72', '0.00397', 'TRUE' ]
[ '60', '44', '32.4', '0.548', '0.258', '0.450', '-1.02', '0.00237', 'TRUE' ]
[ '54', '17', '48.0', '0.304', '0.258', '0.450', '0.26', '0.00234', 'FALSE' ]
[ '54', '17', '48.1', '0.317', '0.258', '0.450', '0.26', '0.00244', 'FALSE' ]
[ '80', '36', '64.0', '0.315', '0.258', '0.450', '-6.33', '0.00430', 'FALSE' ]
[ '106', '104', '52.7', '0.874', '0.441', '0.198', '0.58', '0.00270', 'FALSE' ]
[ '119', '73', '50.7', '0.400', '0.441', '0.198', '3.24', '0.00123', 'FALSE' ]
[ '25', '81', '51.0', '0.452', '0.441', '0.198', '7.35', '0.00108', 'FALSE' ]
[ '79', '38', '39.7', '0.352', '0.441', '0.198', '-0.06', '0.00137', 'FALSE' ]
[ '79', '38', '40.0', '0.306', '0.441', '0.198', '-0.06', '0.00119', 'FALSE' ]
```

Hier werden einfach die XYZ-Tripels nacheinander weg in ein Array geschrieben. Jeweils drei gehören zusammen, um einen Strich bei MacAdam zu machen. Die untere Zeile dient dazu, die drei zusammengehörigen Tripels wieder zusammen zu fassen und später zu transponieren.

```
In [4]: FarbeM = np.zeros((3)).astype(float)
Tripel = np.zeros((3,3)).astype(float)
Deltas = np.zeros((0,3)).astype(float)
for row in MacAdamTableIII[1:]:
#   if row[8]=='FALSE':
    if row:
        xySollfarbe = np.array([float(row[4]),float(row[5])])
        Filter0 = MacAdamTableI[int(row[0])-1]
        Filter90 = MacAdamTableI[int(row[1])-1]
        FarbeV = np.array(Filter0[3:6]).astype(float)
        FarbeU = np.array(Filter90[3:6]).astype(float)
        Winkel = np.array([float(row[2]) - float(row[3]), float(row[2]), float(row[2]) + float(row[3])])
        for n in range(0, 3):
            U = (np.sin(float(Winkel[n])/180*np.pi))**2
            V = 1 - U
            FarbeM = FarbeV*V + FarbeU*U
            xy = colour.XYZ_to_xy(FarbeM)
            Tripel[n] = FarbeM
#   Striche = np.append(Striche, [np.transpose(Tripel)], axis=0)
        Deltas = np.append(Deltas, [FarbeM], axis=0)
#Deltas = Deltas.T
print(Deltas)
```

```
[[0.00168004 0.00278347 0.00221547]
 [0.00165986 0.00278188 0.00217539]
 [0.00163983 0.0027803 0.00213561]
 ...
 [0.003619 0.002969 0.00114717]
 [0.0036167 0.0029667 0.00110924]
 [0.00361443 0.00296443 0.00107173]]
```

Über die Bedienung oben können die MacAdam-x-Filter aus- oder eingeblendet werden

Klasse zum Zeichnen der Diagramme

und zur Erzeugung eines Diagrammes über die Verteilung der jeweiligen MacAdam-Schwellen in Bezug auf den Nullpunkt

```
In [12]: class MacAdamGraf():
    def __init__(self, Titel, Xachse, Yachse, Xindex, Yindex, Null, Spektralkurve, Striche):
        self.Titel = Titel
        self.Xachse = Xachse
        self.Yachse = Yachse
        self.Xindex = Xindex
        self.Yindex = Yindex
#
        self.Null = np.asarray([0, 0, 0])
        self.Null = Null
        self.Spektralk = Spektralkurve
        self.Striche = Striche

    def zeichnen(self):
        cmfs = colour.STANDARD_OBSERVERS_CMFS['CIE 1931 2 Degree Standard Observer']
        illuminant = colour.ILLUMINANTS['CIE 1931 2 Degree Standard Observer']['C']
        #illuminant = colour.ILLUMINANTS_RELATIVE_SPDS['C']

        #Graf wird initialisiert
        %pylab inline
        %config InlineBackend.figure_formats=['svg']
        pylab.rcParams['figure.figsize'] = (10, 10)
        pylab.xlabel(self.Xachse)
        pylab.ylabel(self.Yachse)
        pylab.title(self.Titel)
        pylab.grid(True)
```

```

#Es wird versucht, den Spektralkurvenzug zu zeichnen
pylab.plot(self.Spektralk[...], self.Xindex, self.Spektralk[...], self.Yindex, color='orange', line

#Es wird versucht, die Purpurgerade zu zeichnen
pylab.plot(
    (self.Spektralk[-1][self.Xindex], self.Spektralk[0][self.Xindex]), (self.Spektralk[-1][self.Yin
    color='purple',
    linewidth=1)

#alle X,Y und Z werden wieder in 3 x 3 - Matrizen aufgeteilt und dann Transponiert
self.Striche = self.Striche.reshape(int(self.Striche.shape[0]/3), 3, self.Striche.shape[1])

#die MacAdam-Striche werden gezeichnet und ihre Länge wird errechnet
LaengenFeld = np.array([[ ],[ ]])
for Koordinaten in self.Striche:
    Laenge = np.sum((Koordinaten[0]-Koordinaten[2])**2)**.5
    Abstand = np.sum((Koordinaten[1] - self.Null)**2)**.5
    LaengenFeld = np.concatenate((LaengenFeld, np.array([[Laenge], [Abstand]])), axis=1)
    Koordinaten = Koordinaten.transpose()
    pylab.plot(Koordinaten[self.Xindex],Koordinaten[self.Yindex], color='green')
plt.figure(figsize=(5,5))
pylab.xlabel("Abstand von Null")
pylab.ylabel("Längen")
pylab.title("Längenverteilung")
pylab.plot(LaengenFeld[1], LaengenFeld[0], 'bo', color='red')
print("Durchschnitt: ", np.average(LaengenFeld[0]),
      ",\nMedian: ", np.median(LaengenFeld[0]),
      ",\nMinimum: ", np.amin(LaengenFeld[0]),
      ",\nMaximum: ", np.amax(LaengenFeld[0]),
      ",\nVerhältnis: ", np.amax(LaengenFeld[0])/np.amin(LaengenFeld[0]))
pass

```

Klasse erzeugen und entsprechende Werte übergeben

```

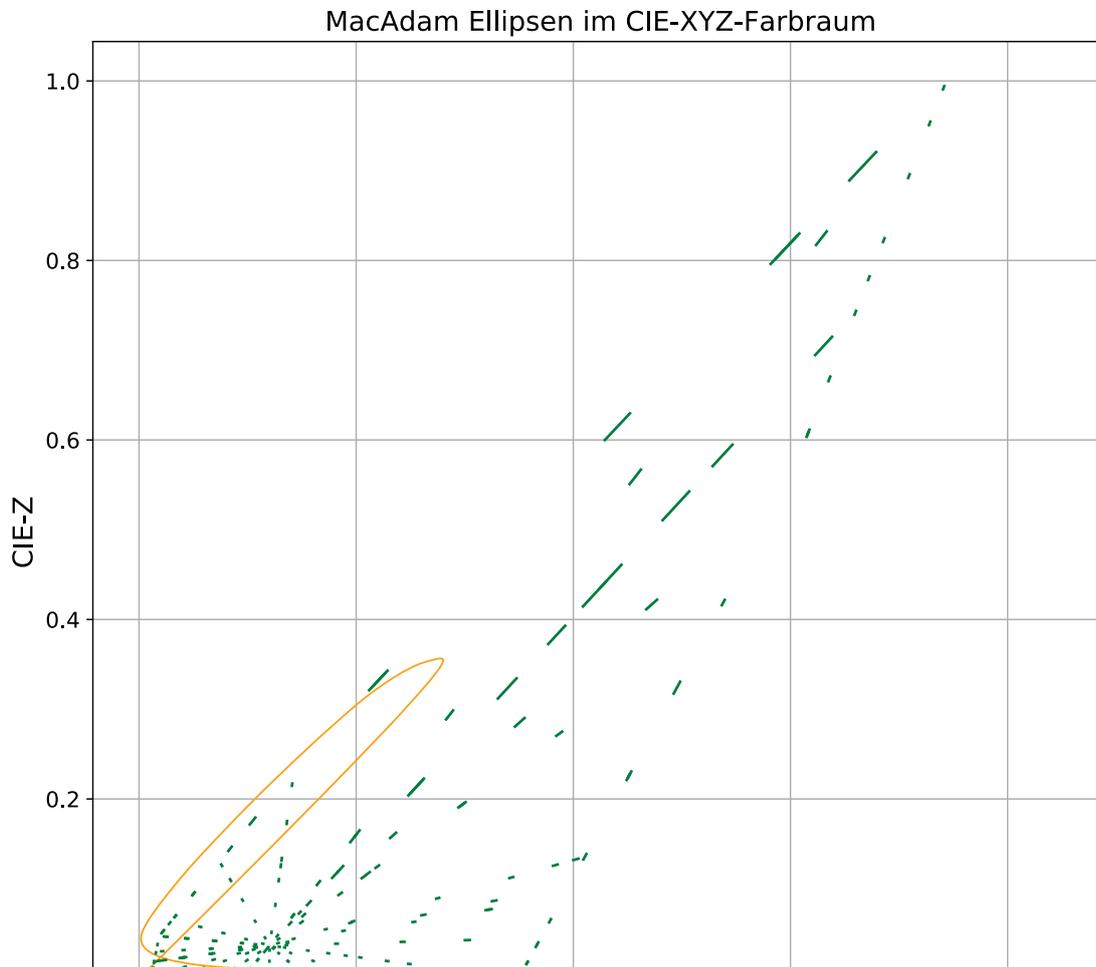
In [33]: GrafXYZ = MacAdamGraf("MacAdam Ellipsen im CIE-XYZ-Farbraum", "CIE-X", "CIE-Z", 0, 2, [0, 0, 0], cmfs.value
GrafXYZ.zeichnen()

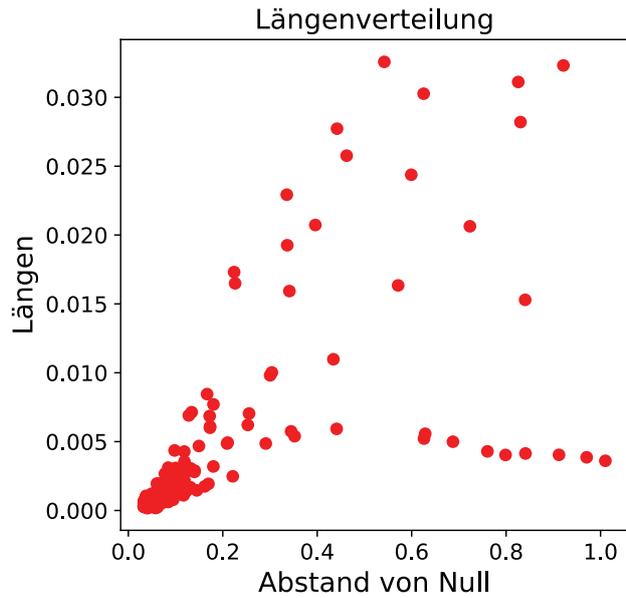
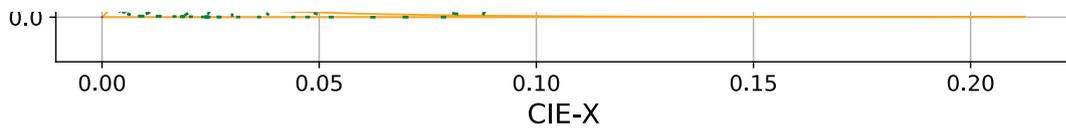
```

```

Populating the interactive namespace from numpy and matplotlib
Durchschnitt: 0.0032246154930871463 ,
Median: 0.0007433231897014183 ,
Minimum: 0.00018009158031908872 ,
Maximum: 0.03257281310592892 ,
Verhältnis: 180.8680508451087

```



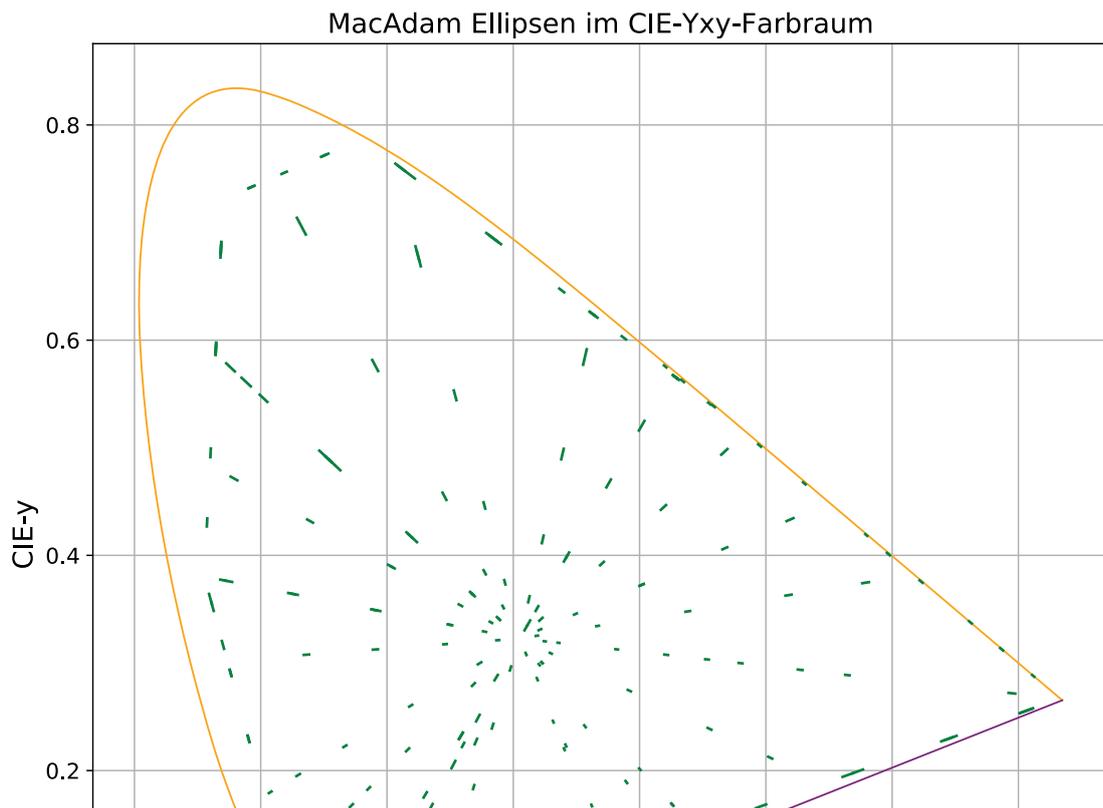


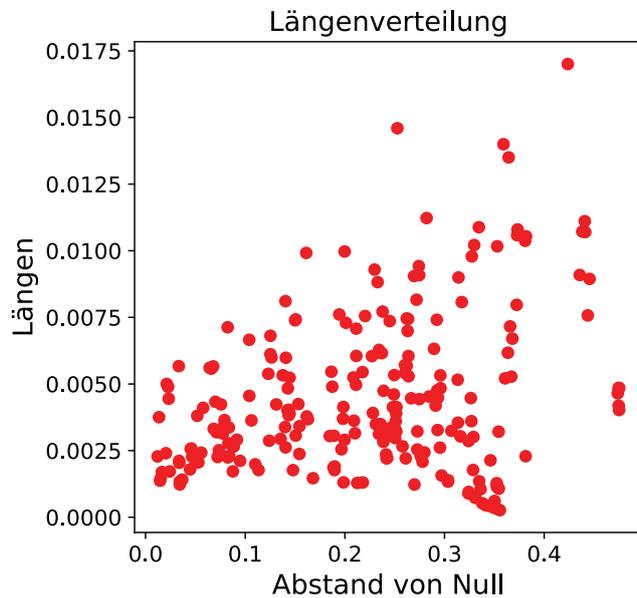
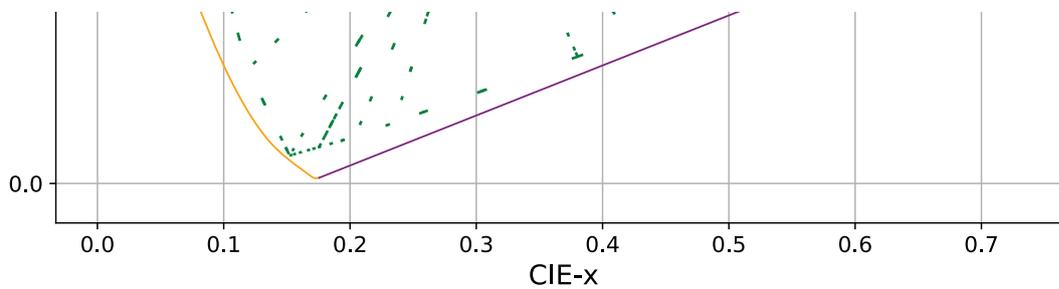
CIE-Yxy-Diagramm

Die Eingangsgrößen können skaliert werden, wie man will, der Graf bleibt unverändert

```
In [14]: GrafXYZ = MacAdamGraf("MacAdam Ellipsen im CIE-Yxy-Farbraum",
                                "CIE-x", "CIE-y", 0, 1,
                                [0.333, 0.333],
                                colour.XYZ_to_xy(cmfs.values/100),
                                colour.XYZ_to_xy(Deltas))
GrafXYZ.zeichnen()
```

```
Populating the interactive namespace from numpy and matplotlib
Durchschnitt: 0.00452685640200878 ,
Median: 0.00379356229228297 ,
Minimum: 0.0002649604576282501 ,
Maximum: 0.017005864327405997 ,
Verhältnis: 64.18265004382613
```





Leicht modifizierte Lab-Funktion

Die normale Lab-Funktion aus der colour-Bibliothek wurde leicht verändert, damit für L, a oder b die Werte niemals größer als 1 werden, egal wie man die Eingangsgrößen (Deltas) skaliert

```
In [15]: # -*- coding: utf-8 -*-
"""
CIE L*a*b* Colourspace
=====

Defines the *CIE L*a*b** colourspace transformations:

- :func:`colour.XYZ_to_Lab`
- :func:`colour.Lab_to_XYZ`
- :func:`colour.Lab_to_LCHab`
- :func:`colour.LCHab_to_Lab`

See Also
-----
`CIE L*a*b* Colourspace Jupyter Notebook
<http://nbviewer.jupyter.org/github/colour-science/colour-notebooks/blob/master/notebooks/models/cie\_lab.ipynb>`_

References
-----
- :cite:`CIETC1-482004m` : CIE TC 1-48. (2004). CIE 1976 uniform colour
spaces. In CIE 015:2004 Colorimetry, 3rd Edition (p. 24).
ISBN:978-3-901-90633-6
"""

from __future__ import division, unicode_literals

import numpy as np

from colour.algebra import cartesian_to_polar, polar_to_cartesian
from colour.colorimetry import ILLUMINANTS
from colour.constants import CIE_E, CIE_K
from colour.models import xy_to_xyY, xyY_to_XYZ
from colour.utilities import tsplit, tstack

__author__ = 'Colour Developers'
__copyright__ = 'Copyright (C) 2013-2018 - Colour Developers'
__license__ = 'New BSD License - http://opensource.org/licenses/BSD-3-Clause'
__maintainer__ = 'Colour Developers'
__email__ = 'colour-science@googlegroups.com'
__status__ = 'Production'
```

```

__all__ = ['XYZ_to_Lab', 'Lab_to_XYZ', 'Lab_to_LCHab', 'LCHab_to_Lab']

def XYZ_to_Lab(
    XYZ,
    illuminant=ILLUMINANTS['CIE 1931 2 Degree Standard Observer']['C']):
    """
    Converts from *CIE XYZ* tristimulus values to *CIE L*a*b** colourspace.

    Parameters
    -----
    XYZ : array_like
        *CIE XYZ* tristimulus values.
    illuminant : array_like, optional
        Reference *illuminant* *xy* chromaticity coordinates or *CIE xyY*
        colourspace array.

    Returns
    -----
    ndarray
        *CIE L*a*b** colourspace array.

    Notes
    -----
    - Input *CIE XYZ* tristimulus values are normalised to domain [0, 1].
    - Input *illuminant* *xy* chromaticity coordinates or *CIE xyY*
      colourspace array are normalised to domain [0, 1].
    - Output *Lightness* :math:`L^*` is normalised to range [0, 100].

    References
    -----
    - :cite:`CIETC1-482004m`

    Examples
    -----
    >>> XYZ = np.array([0.07049534, 0.10080000, 0.09558313])
    >>> XYZ_to_Lab(XYZ) # doctest: +ELLIPSIS
    array([ 37.9856291..., -23.6290768..., -4.4174661...])
    """
    XYZ = np.asarray(XYZ)
    XYZ_r = xyY_to_XYZ(xy_to_xyY(illuminant))

    XYZ_f = XYZ / XYZ_r

    #Eigene Anpassung
    #alle Tripel, die einen Wert größer 1 enthalten werden verworfen. Das macht leider
    #eventuell die Matrizen kleiner, ohne, dass man es merkt
    #np.putmask(XYZ_f, XYZ_f > 1, [1])

    mask = (XYZ_f[:, 0] < 0.9) & (XYZ_f[:, 1] < 0.9) & (XYZ_f[:, 2] < 1)
    XYZ_f = XYZ_f[mask, :]

    XYZ_f = np.where(XYZ_f > CIE_E,
                    np.power(XYZ_f, 1 / 3), (CIE_K * XYZ_f + 16) / 116)

    X_f, Y_f, Z_f = tsplit(XYZ_f)

    L = 116 * Y_f - 16
    a = 500 * (X_f - Y_f)
    b = 200 * (Y_f - Z_f)

    Lab = tstack((L, a, b))

    return Lab

def Lab_to_XYZ(
    Lab,
    illuminant=colour.ILLUMINANTS['CIE 1931 2 Degree Standard Observer']['C']):
    """
    Converts from *CIE L*a*b** colourspace to *CIE XYZ* tristimulus values.

    Parameters
    -----
    Lab : array_like
        *CIE L*a*b** colourspace array.
    illuminant : array_like, optional
        Reference *illuminant* *xy* chromaticity coordinates or *CIE xyY*
        colourspace array.

    Returns
    -----
    ndarray
        *CIE XYZ* tristimulus values.

    Notes
    -----

```

```

NOTES
-----
- Input *Lightness* :math:`L^*` is normalised to domain [0, 100].
- Input *illuminant* *xy* chromaticity coordinates or *CIE xyY*
  colourspace array are normalised to domain [0, 1].
- Output *CIE XYZ* tristimulus values are normalised to range [0, 1].

References
-----
- :cite:`CIETC1-482004m`

Examples
-----
>>> Lab = np.array([37.98562910, -23.62907688, -4.41746615])
>>> Lab_to_XYZ(Lab) # doctest: +ELLIPSIS
array([ 0.0704953...,  0.1008      ,  0.0955831...])
"""

L, a, b = tsplit(Lab)
XYZ_r = xyY_to_XYZ(xy_to_xyY(illuminant))

f_y = (L + 16) / 116
f_x = a / 500 + f_y
f_z = f_y - b / 200

x_r = np.where(f_x ** 3 > CIE_E, f_x ** 3, (116 * f_x - 16) / CIE_K)
y_r = np.where(L > CIE_K * CIE_E, ((L + 16) / 116) ** 3, L / CIE_K)
z_r = np.where(f_z ** 3 > CIE_E, f_z ** 3, (116 * f_z - 16) / CIE_K)

XYZ = tstack((x_r, y_r, z_r)) * XYZ_r

return XYZ

def Lab_to_LCHab(Lab):
    """
    Converts from *CIE L*a*b** colourspace to *CIE L*C*Hab* colourspace.

    Parameters
    -----
    Lab : array_like
        *CIE L*a*b** colourspace array.

    Returns
    -----
    ndarray
        *CIE L*C*Hab* colourspace array.

    Notes
    -----
    - *Lightness* :math:`L^*` is normalised to domain [0, 100].

    References
    -----
    - :cite:`CIETC1-482004m`

    Examples
    -----
    >>> Lab = np.array([37.98562910, -23.62907688, -4.41746615])
    >>> Lab_to_LCHab(Lab) # doctest: +ELLIPSIS
    array([ 37.9856291...,  24.0384542...,  190.5892337...])
    """

    L, a, b = tsplit(Lab)

    C, H = tsplit(cartesian_to_polar(tstack((a, b))))

    LCHab = tstack((L, C, np.degrees(H) % 360))

    return LCHab

def LCHab_to_Lab(LCHab):
    """
    Converts from *CIE L*C*Hab* colourspace to *CIE L*a*b** colourspace.

    Parameters
    -----
    LCHab : array_like
        *CIE L*C*Hab* colourspace array.

    Returns
    -----
    ndarray
        *CIE L*a*b** colourspace array.

    Notes
    -----
    - *Lightness* :math:`L^*` is normalised to domain [0, 100].

    References
    -----

```

```

-----
- :cite:`CIETC1-482004m`

Examples
-----
>>> LCHab = np.array([37.98562910, 24.03845422, 190.58923377])
>>> LCHab_to_Lab(LCHab) # doctest: +ELLIPSIS
array([ 37.9856291..., -23.6290768..., -4.4174661...])
"""

L, C, H = tsplit(LCHab)

a, b = tsplit(polar_to_cartesian(tstack((C, np.radians(H)))))

Lab = tstack((L, a, b))

return Lab

```

CIE-Lab-Diagramm

Laut der Arbeit von MacAdam hat die beobachtete Fläche doppelt so viel Licht in cd/m^2 wie die Umgebung. Fraglich ist, ob die Beobachter das Weiß der Umgebung als Weiß empfunden haben oder als Grau. Wenn man die Eingangsgrößen (Deltas) mit 2 skaliert, dann erhält man Werte für L um die 10, was also sehr dunkel und experimentell schwer zu unterscheiden wäre. Mit einem Faktor von 13 für die Eingangsgrößen erhält man einen L-Wert von 25, womit sich ordentlich experimentieren lässt. Zudem tritt bei diesem Faktor noch kein einziges Mal der unerwünschte Effekt auf, dass L-, a- oder b-Werte größer als 1 werden. Bei Faktoren zwischen 3 und 13 bleibt Das Verhältnis von kleinster zu größter Abweichung konstant bei 19. Im Farbraum DIN 99 erhält man sogar eine Helligkeit L von 32.

```

In [16]: cmfs = colour.STANDARD_OBSERVERS_CMFS['CIE 1931 2 Degree Standard Observer']
illuminant = colour.ILLUMINANTS_RELATIVE_SPDS['C']

#Der erste Faktor im array ist der angestrebte Helligkeitswert
Null = [21.6, 0, 0]
HelligkeitsY = Lab_to_XYZ(np.asarray(Null))[0]

#Speziell bei Lab müssen die Spektralkurven auf eine bestimmte Helligkeit normiert werden

#für jeden Spektralwert muss ein eigener Faktor errechnet werden, mit dem dann
#alle drei Wert multipliziert werden müssen
Spektralkurven = cmfs.values / 100
Lfaktoren = HelligkeitsY / Spektralkurven[..., 1]
#Aus dem eindimensionalen Feld mit den Helligkeitsfaktoren muss eines mit Tiefe 3
#und jeweils drei gleich Werten gemacht werden
Spektralkurven = Spektralkurven * np.array([Lfaktoren,Lfaktoren,Lfaktoren]).T

Spektralkurven = XYZ_to_Lab(Spektralkurven)

#Die eigentlichen MacAdam-Werte bezogen auf vorgegebene Lab-Helligkeit Lfaktoren wird neu besetzt
Lfaktoren = HelligkeitsY / Deltas[..., 1]
DeltasGew = Deltas * np.array([Lfaktoren,Lfaktoren,Lfaktoren]).T

GrafLab = MacAdamGraf('MacAdam Ellipses calculated in CIE-Lab',
                    'CIE-a', 'CIE-b', 1, 2,
                    Null,
                    Spektralkurven,
                    XYZ_to_Lab(DeltasGew))

GrafLab.zeichnen()

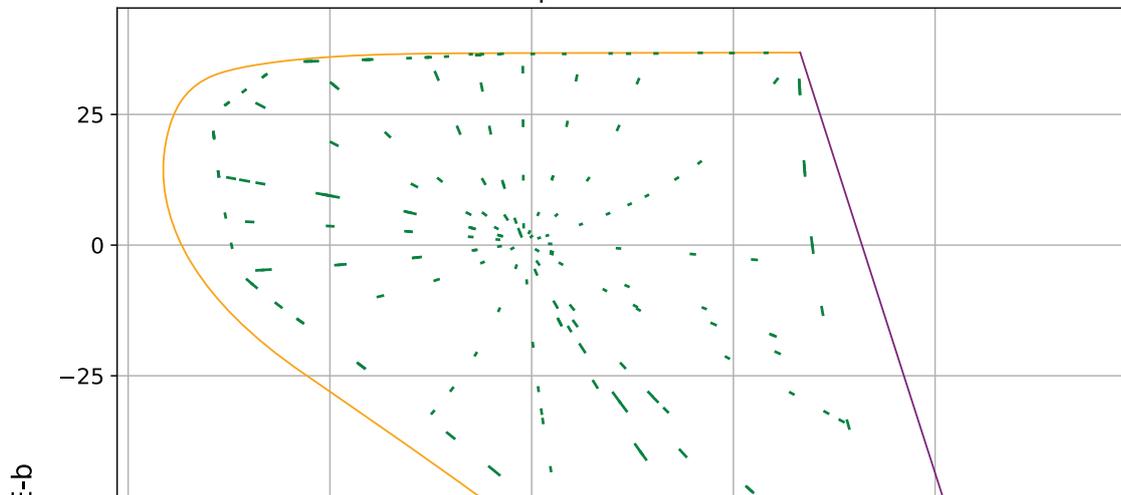
```

```

Populating the interactive namespace from numpy and matplotlib
Durchschnitt: 1.2108497624526364 ,
Median: 0.8827298782691653 ,
Minimum: 0.2598703162786763 ,
Maximum: 5.243619099039267 ,
Verhältnis: 20.17783013515166

```

MacAdam Ellipses calculated in CIE-Lab




```

- Input *Lightness* :math:`L` is normalised to domain [0, 100].
References
-----
- :cite:`ASTMInternational2007`
Examples
-----
>>> import numpy as np
>>> Lab = np.array([37.98562910, -23.62907688, -4.41746615])
>>> Lab_to_DIN99(Lab) # doctest: +ELLIPSIS
array([ 49.6010164..., -16.2314573...,  1.0761812...])
"""

L, a, b = tsplit(Lab)

cos_16 = np.cos(np.radians(16))
sin_16 = np.sin(np.radians(16))

e = cos_16 * a + sin_16 * b
f = 0.7 * (-sin_16 * a + cos_16 * b)
G = (e ** 2 + f ** 2) ** 0.5
h_ef = np.arctan2(f, e)

C_99 = (np.log(1 + 0.045 * G)) / (0.045 * k_CH * k_E)
# Hue angle is unused currently.
# h_99 = np.degrees(h_ef)
a_99 = C_99 * np.cos(h_ef)
b_99 = C_99 * np.sin(h_ef)
L_99 = 105.509 * (np.log(1 + 0.0158 * L)) * k_E

return tstack([L_99, a_99, b_99])

def DIN99_to_Lab(Lab_99, k_E=1, k_CH=1):
    """
    Converts from *DIN99* colourspace to *CIE L*a*b** colourspace.
    Parameters
    -----
    Lab_99 : array_like
        *DIN99* colourspace array.
    k_E : numeric, optional
        Parametric factor :math:`K_E` used to compensate for texture and other
        specimen presentation effects.
    k_CH : numeric, optional
        Parametric factor :math:`K_{CH}` used to compensate for texture and
        other specimen presentation effects.
    Returns
    -----
    ndarray
        *CIE L*a*b** colourspace array.
    Notes
    ----
    - Input *Lightness* :math:`L` is normalised to domain [0, 100].
    References
    -----
    - :cite:`ASTMInternational2007`
    Examples
    -----
    >>> import numpy as np
    >>> Lab_99 = np.array([49.60101649, -16.2314573, 1.07618123])
    >>> DIN99_to_Lab(Lab_99) # doctest: +ELLIPSIS
    array([ 37.9856291..., -23.6290768..., -4.4174661...])
    """

    L_99, a_99, b_99 = tsplit(Lab_99)

    cos_16 = np.cos(np.radians(16))
    sin_16 = np.sin(np.radians(16))

    h_99 = np.arctan2(b_99, a_99)

    C_99 = np.sqrt(a_99 ** 2 + b_99 ** 2)
    G = (np.exp(0.045 * C_99 * k_CH * k_E) - 1) / 0.045

    e = G * np.cos(h_99)
    f = G * np.sin(h_99)

    a = e * cos_16 - (f / 0.7) * sin_16
    b = e * sin_16 + (f / 0.7) * cos_16
    L = (np.exp(L_99 * k_E / 105.509) - 1) / 0.0158

    return tstack([L, a, b])

```

CIE-Lab/DIN99-Diagramm

Für die Berechnung des DIN99-Farbraums gibt es noch keinen offiziellen Teil der colour-Bibliothek, aber einen recht gelungenen Vorschlag. Mit diesem wurden die Berechnungen ausgeführt. Der Farbraum sieht gleichmäßiger aus, als in CIE-Lab, aber erstaunlicherweise weichen die Längen der Striche nach MacAdam stärker voneinander ab, als im CIE-Lab-Farbraum, wobei es sich nur um wenige Extreme Werte handelt.

```
In [18]: cmfs = colour.STANDARD_OBSERVERS_CMFS['CIE 1931 2 Degree Standard Observer']
illuminant = colour.ILLUMINANTS_RELATIVE_SPDS['C']

#Der erste Faktor im array ist der angestrebte Helligkeitswert
Null = [31, 0, 0]
HelligkeitsY = Lab_to_XYZ(DIN99_to_Lab(np.asarray(Null)))[0]

#Speziell bei Lab müssen die Spektralkurven auf eine bestimmte Helligkeit normiert werden

#für jeden Spektralwert muss ein eigener Faktor errechnet werden, mit dem dann
#alle drei Wert multipliziert werden müssen
Spektralkurven = cmfs.values / 100
Lfaktoren = HelligkeitsY / Spektralkurven[... , 1]
#Aus dem eindimensionalen Feld mit den Helligkeitsfaktoren muss eines mit Tiefe 3 gemacht werden
Spektralkurven = Spektralkurven * np.array([Lfaktoren,Lfaktoren,Lfaktoren]).T

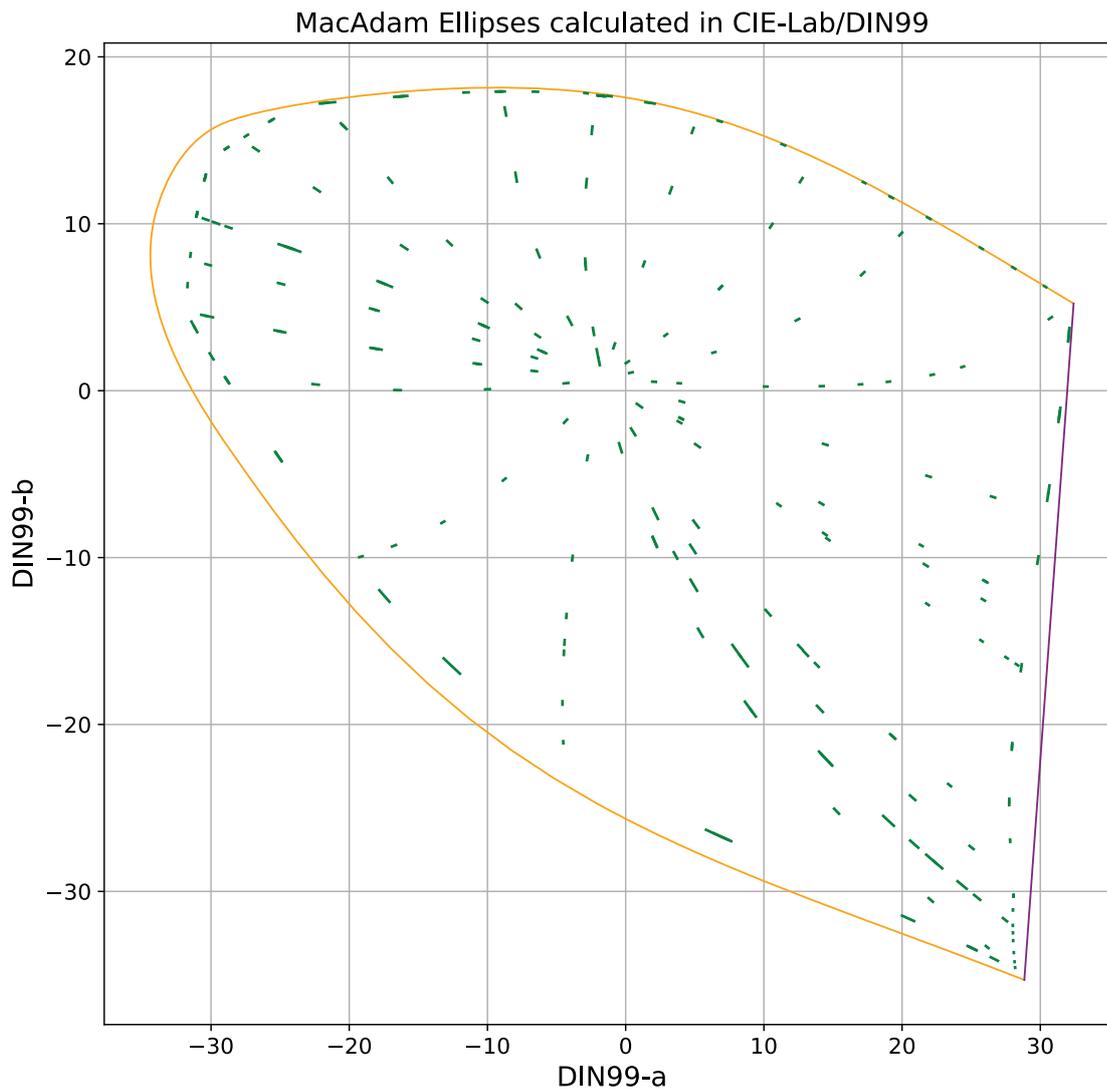
Spektralkurven = Lab_to_DIN99(XYZ_to_Lab(Spektralkurven))

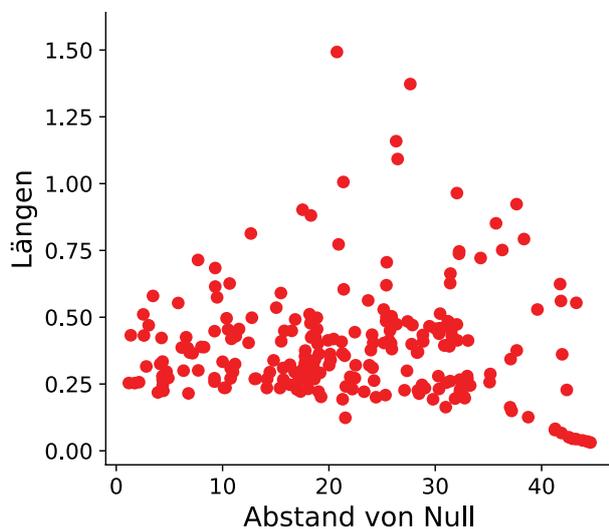
#Die eigentlichen MacAdam-Werte bezogen auf vorgegebene Lab-Helligkeit Lfaktoren wird neu besetzt
Lfaktoren = HelligkeitsY / Deltas[... , 1]
DeltasGew = Deltas * np.array([Lfaktoren,Lfaktoren,Lfaktoren]).T

GrafLab = MacAdamGraf('MacAdam Ellipses calculated in CIE-Lab/DIN99',
                      'DIN99-a', 'DIN99-b', 1, 2,
                      Null,
                      Spektralkurven,
                      Lab_to_DIN99(XYZ_to_Lab(DeltasGew)))

GrafLab.zeichnen()
```

```
Populating the interactive namespace from numpy and matplotlib
Durchschnitt: 0.39725150686373234 ,
Median: 0.36157864083613744 ,
Minimum: 0.03106396715404213 ,
Maximum: 1.7009343737702491 ,
Verhältnis: 54.75586441794569
```





CIECAM02UCS

Um diesen Farbraum nutzen zu können, müssen diverse Vorberechnungen und Annahmen gemacht werden.

```
In [28]: cmfs = colour.STANDARD_OBSERVERS_CMFS['CIE 1931 2 Degree Standard Observer']
illuminant = colour.ILLUMINANTS_RELATIVE_SPDS['C']
#so kommt notfalls schon mal was sinnvolles raus, schlecht ist, dass die CIECAM-Funktion
#nur einzelne Tripel verarbeiten kann, keine größeren Felder
#Konstanten, die für die Berechnung notwendig sind
#MacAdam gibt für das Umgebungslicht Normlich C an, also XYZ: 0,309 0,319 0,372
XYZ_w = np.array([30.9, 31.9, 37.2])
#Beleuchtungsumgebung laut MacAdam
L_A = 24
#Weiß der Probe in Relation zum Weiß der Umgebung
Y_b = 10
surround = colour.CIECAM02_VIEWING_CONDITIONS['Dark']

#Der erste Faktor im array ist der angestrebte Helligkeitswert
specification = colour.CAM02UCS_to_JMh_CIECAM02([21.6, 0, 0])
Null = colour.CIECAM02_to_XYZ(specification, XYZ_w, L_A, Y_b, surround)
HelligkeitsY = Null[0]

#für jeden Spektralwert muss ein eigener Faktor errechnet werden, mit dem dann
#alle drei Werte multipliziert werden müssen
Spektralkurven = cmfs.values
Lfaktoren = HelligkeitsY / Spektralkurven[..., 1]
#Aus dem eindimensionalen Feld mit den Helligkeitsfaktoren muss eines mit Tiefe 3
#und jeweils drei gleich Werten gemacht werden
Spektralkurven = Spektralkurven * np.array([Lfaktoren, Lfaktoren, Lfaktoren]).T

CAMDaten = []
for row in Spektralkurven:
    specification = colour.XYZ_to_CIECAM02(row, XYZ_w, L_A, Y_b, surround, True)
    JMh = (specification.J, specification.M, specification.h)
    row = colour.JMh_CIECAM02_to_CAM02UCS(JMh)
    CAMDaten.append(row)

SpektralkurvenCAM = np.array(CAMDaten)

#Die eigentlichen MacAdam-Werte bezogen auf vorgegebene Lab-Helligkeit Lfaktoren wird neu besetzt
XYZ = Deltas * 10

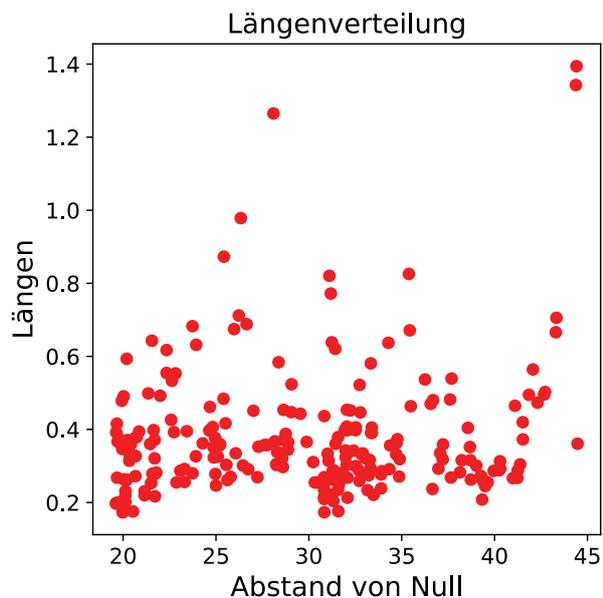
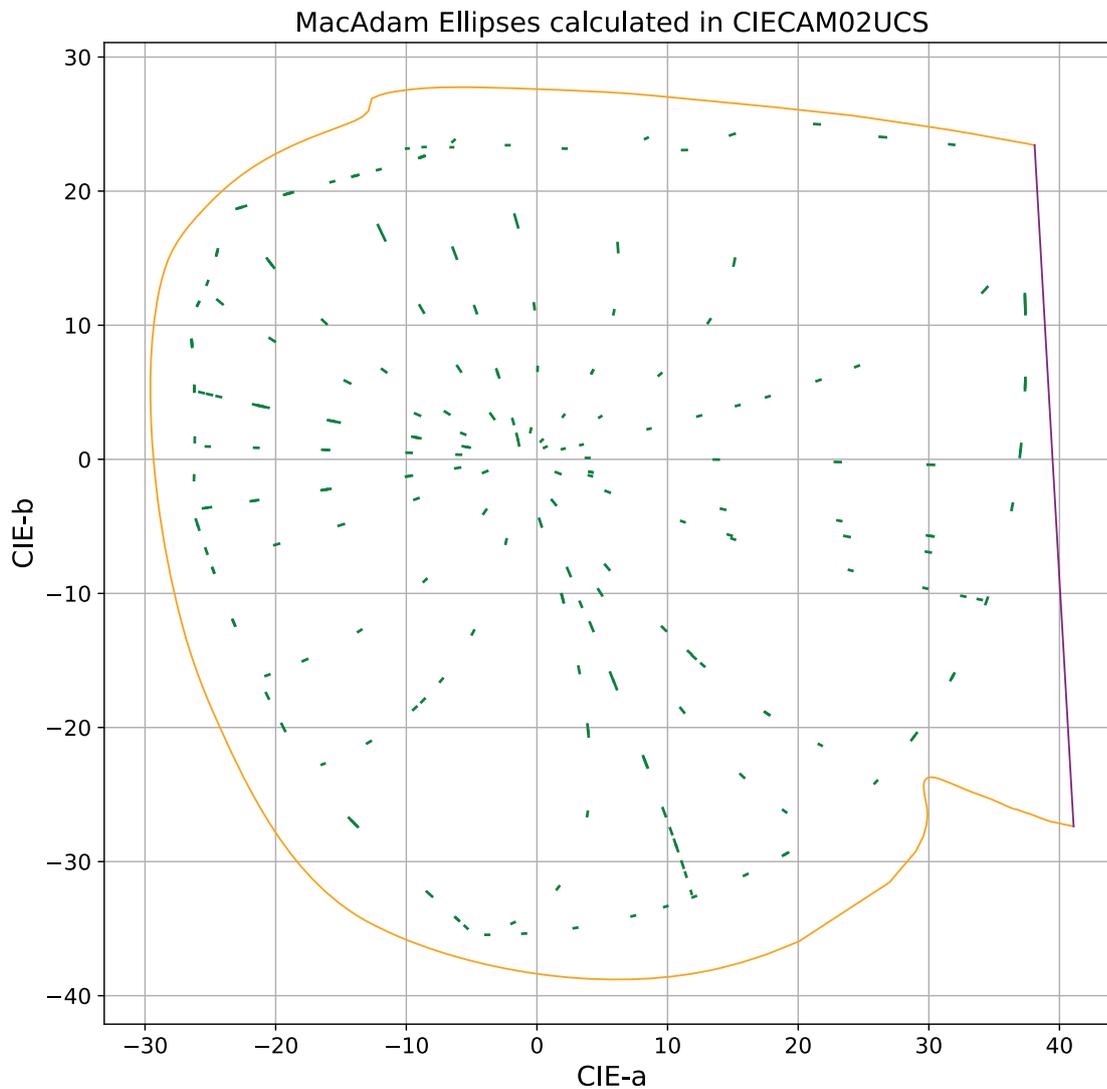
CAMDaten = []
for row in XYZ:
    specification = colour.XYZ_to_CIECAM02(row, XYZ_w, L_A, Y_b, surround, True)
    JMh = (specification.J, specification.M, specification.h)
    row = colour.JMh_CIECAM02_to_CAM02UCS(JMh)
    CAMDaten.append(row)
DeltasGew = np.array(CAMDaten)

GrafLab = MacAdamGraf('MacAdam Ellipses calculated in CIECAM02UCS',
                    'CIE-a', 'CIE-b', 1, 2,
                    Null,
                    SpektralkurvenCAM,
                    DeltasGew)

GrafLab.zeichnen()
```

```
Populating the interactive namespace from numpy and matplotlib
Durchschnitt: 0.38048947582294534 ,
Median: 0.3351623892148195 ,
Minimum: 0.17298491517442677 ,
Maximum: 1.3942703671661232 ,
```

Verhältnis: 8.060069086141016



Eigenes Sigmoid-basierendes Farbsystem

Der Aufbau ist wie in einem gängigen Neuronalen Netz, allerdings sind die Werte für die Knoten empirisch ermittelt oder in einfacher Analogie zum herkömmlichen Lab. Mit dem wichtigen Unterschied, dass die Übertragungsfunktion aus keiner Kubikwurzel gebildet wird, sondern aus einer Sigmoid-Funktion, wie das bei Neuronalen Netzen üblich ist.

```
In [20]: cmfs = colour.STANDARD_OBSERVERS_CMFS['CIE 1931 2 Degree Standard Observer']
         illuminant = colour.ILLUMINANTS_RELATIVE_SPDS['C']
```

```

def XYZ_to_sigmoLab(
    XYZ,
    illuminant=colour.ILLUMINANTS['CIE 1931 2 Degree Standard Observer']['C']):

    XYZ = np.asarray(XYZ)
    XYZ_r = colour.xyY_to_XYZ(colour.xy_to_xyY(illuminant))

    XYZ_f = XYZ / XYZ_r
    XYZ_fair = []
    #Umrechnung analog CAM02 mit weniger Anteil von L in S
    for row in XYZ_f:
        row = np.dot(colour.CAT02_CAT, row)
        XYZ_fair.append(row)
    XYZ_f = np.array(XYZ_fair)

    sigmoid = lambda X: scipy.special.expit(X)
    XYZ_f = sigmoid(XYZ_f * 9) - .5

    X_f, Y_f, Z_f = tsplit(XYZ_f)

#    Y_f = sigmoid(Y_f*9) - .5
#    X_f = sigmoid(X_f*9) - .5
#    Z_f = sigmoid(Z_f*9) - .5

    L = sigmoid(Y_f * 6)
    a = (sigmoid((3*X_f - 1*Y_f)*7))
    b = (sigmoid((1*X_f + 1*Y_f - 2.5*Z_f)*6))
    #a = 200 * ((X_f - .5*Y_f)*4)
    #b = 300 * ((X_f + Y_f) - 1.5*Z_f)

    sigmoLab = tstack((L, a, b)) * 100

    return sigmoLab

#Der erste Faktor im array ist der angestrebte Helligkeitswert
# Umkehrfunktion y = sigmoid(x) ist x = ln(y/(1-y))
#Hellwert = 50
#HelligkeitsY = np.log((100/Hellwert) - 1)/(7)
Null = [21, 0, 0]
HelligkeitsY = Lab_to_XYZ(np.asarray(Null))[0]
#HelligkeitsY = 50 [21, 19, 45]
Null = [21, 73, 45]

#Speziell bei Lab müssen die Spektralkurven auf eine bestimmte Helligkeit normiert werden

#für jeden Spektralwert muss ein eigener Faktor errechnet werden, mit dem dann
#alle drei Wert multipliziert werden müssen
Spektralkurven = cmfs.values/100
Lfaktoren = HelligkeitsY / Spektralkurven[... , 1]
#Aus dem eindimensionalen Feld mit den Helligkeitsfaktoren muss eines mit Tiefe 3 gemacht werden
Spektralkurven = Spektralkurven * np.array([Lfaktoren,Lfaktoren,Lfaktoren]).T

Spektralkurven = XYZ_to_sigmoLab(Spektralkurven)

#Die eigentlichen MacAdam-Werte bezogen auf vorgegebene Lab-Helligkeit Lfaktoren wird neu besetzt
Lfaktoren = HelligkeitsY / Deltas[... , 1]
Deltas = Deltas * np.array([Lfaktoren,Lfaktoren,Lfaktoren]).T

GrafLab = MacAdamGraf('Eigene Kurven mit Sigmoid-Funktionen',
    'a', 'b', 1, 2,
    Null,
    Spektralkurven,
    XYZ_to_sigmoLab(Deltas))
GrafLab.zeichnen()

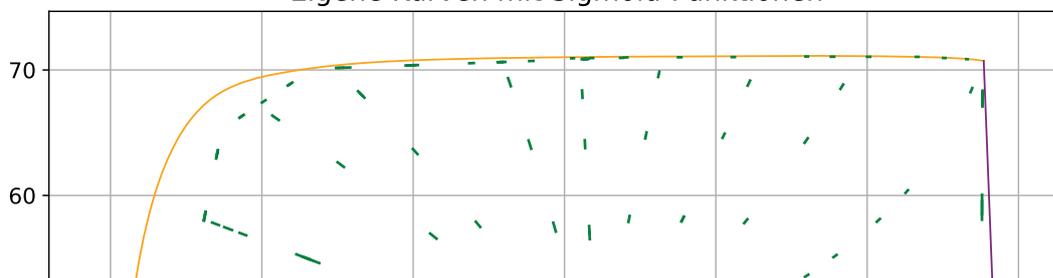
```

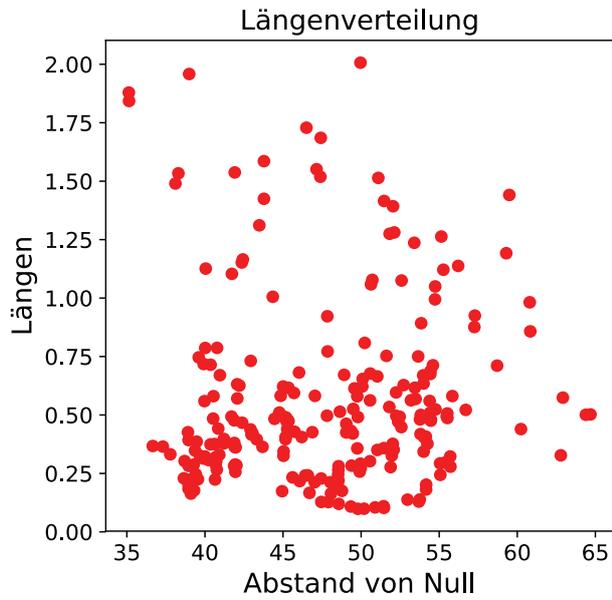
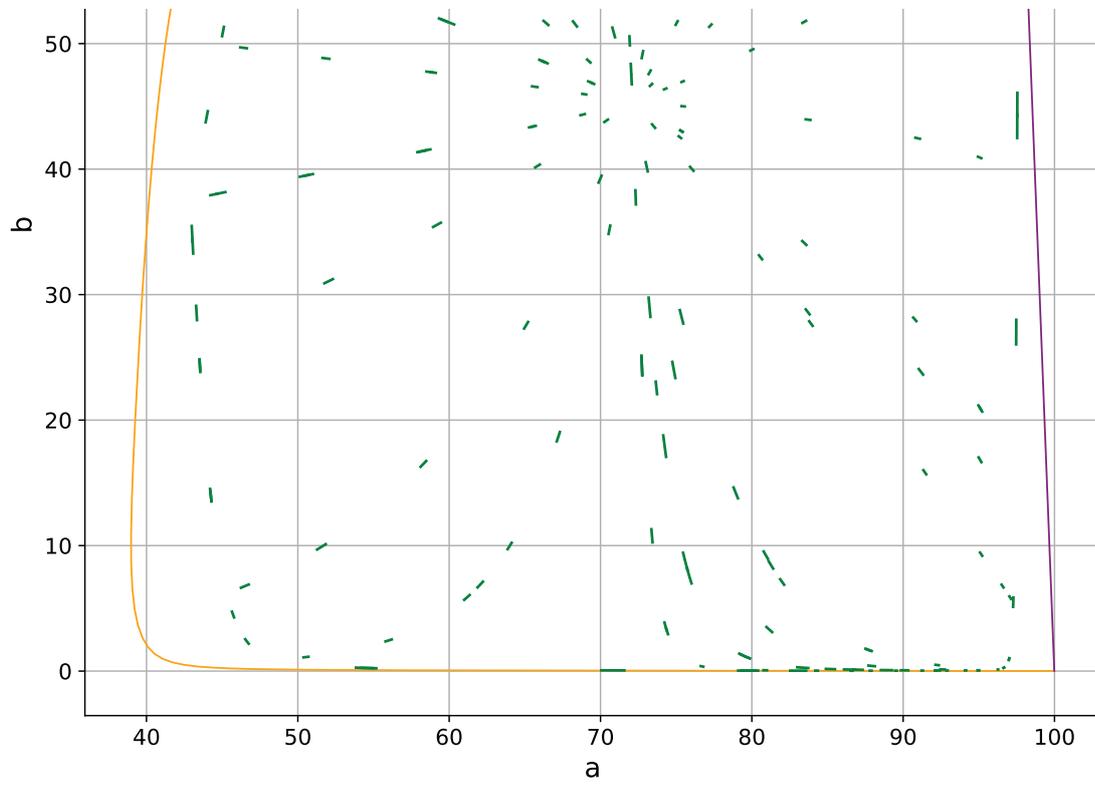
```

Populating the interactive namespace from numpy and matplotlib
Durchschnitt: 0.5695539014323112 ,
Median: 0.4688245718834063 ,
Minimum: 0.09850254402102261 ,
Maximum: 2.006870765041408 ,
Verhältnis: 20.37379628096811

```

Eigene Kurven mit Sigmoid-Funktionen





In []:

```
In [1]: # python notebook for Make Your Own Neural Network
# code for a 3-layer neural network, and code for learning the MNIST dataset
# (c) Tariq Rashid, 2016
# license is GPLv2
# bearbeitet für die Ermittlung von Farbsystemen aus MacAdam Farbschwellen
# erkennbar an den deutschsprachigen Kommentaren und Variablennamen
# von Christian Greim 2018
```

```
In [2]: import numpy
# scipy.special for the sigmoid function expit()
import scipy.special
# library for plotting arrays
import matplotlib.pyplot
# ensure the plots are inside this notebook, not an external window
%matplotlib inline
```

```
In [3]: import numpy as np
import matplotlib
import colour
from pprint import pprint
import colour.plotting
from colour.plotting import *

colour.filter_warnings(True, False)
colour.plotting_defaults()

import colour.colorimetry as colorimetry
import colour.colorimetry.dataset as dataset

cmfs = colour.STANDARD_OBSERVERS_CMFS['CIE 1931 2 Degree Standard Observer']
illuminant = colour.ILLUMINANTS_RELATIVE_SPDS['A']
```

```
D:\Programme\anaconda3\lib\site-packages\colour\utilities\verbose.py:141: ColourWarning: "x" independent
variable is not uniform, unpredictable results may occur!
  warn(*args, **kwargs)
D:\Programme\anaconda3\lib\site-packages\colour\utilities\verbose.py:141: ColourWarning: "colour.filter_w
arnings" object access is deprecated and will change to "colour.utilities.filter_warnings" in a future re
lease.
  warn(*args, **kwargs)
```

```
In [4]: import csv
lst = []
with open('MacAdamTableI.csv', newline='') as f:
    reader = csv.reader(f)
    for row in reader:
        lst.append(row)
#     print(row)
MacAdamTableI = lst
```

Hier werden die allgemeinen Daten von MacAdam eingelesen ohne die Ellipsen

Hier werden einfach die XYZ-Tripels nacheinander weg in ein Array geschrieben. Jeweils drei gehören zusammen, um einen Strich bei MacAdam zu machen. Die untere Zeile dient dazu, die drei zusammengehörigen Tripels wieder zusammen zu fassen und später zu transponieren.

```
In [5]: """         row = [float(i) for i in row]"""
import csv
lst = []
with open('MacAdamTableII.csv', newline='') as f:
    reader = csv.reader(f)
    for row in reader:
        lst.append(row)
#     print(row)
MacAdamTableII = lst
```

```
In [6]: FarbeM = np.zeros((3)).astype(float)
Tripel = np.zeros((3,3)).astype(float)
#Deltas = np.zeros((0,3,3)).astype(float)
Deltas = np.zeros((0,3)).astype(float)
for row in MacAdamTableII[1:]:
#     if row[8]=='FALSE':
    if row:
        xySollfarbe = np.array([float(row[4]),float(row[5])])
```

```

Filter0 = MacAdamTableI[int(row[0])-1]
Filter90 = MacAdamTableI[int(row[1])-1]
FarbeV = np.array(Filter0[3:6]).astype(float)
FarbeU = np.array(Filter90[3:6]).astype(float)
Winkel = np.array([float(row[2]) - float(row[3]), float(row[2]), float(row[2]) + float(row[3])])
for n in range(0, 3):
    U = (np.sin(float(Winkel[n])/180*np.pi)**2
    V = 1 - U
    FarbeM = FarbeV*V + FarbeU*U
    Tripel[n] = FarbeM
#     Deltas = np.append(Deltas, [np.transpose(Tripel)], axis=0)
    Deltas = np.append(Deltas, [FarbeM], axis=0)
#Deltas = np.delete(Deltas, (0), axis=0)
#print(Deltas)

```

Im Folgenden werden die Daten für die Ellipsen eingelesen

Hier werden einfach die XYZ-Tripels nacheinander weg in ein Array geschrieben. Jeweils drei gehören zusammen, um einen Strich bei MacAdam zu machen. Die untere Zeile dient dazu, die drei zusammengehörigen Tripels wieder zusammen zu fassen und später zu transponieren.

```

In [513]: """         row = [float(i) for i in row]"""
import csv
lst = []
with open('MacAdamTableIII.csv', newline='') as f:
    reader = csv.reader(f)
    for row in reader:
        lst.append(row)
        print(row)
MacAdamTableIII = lst

['0° Filternummer', '90° Filternummer', '?° Average setting', '??° Standard Deviation', 'x', 'y', '?y/?x',
 '?s Standard Deviation', 'Prüfung ob X-Filter in A oder B enthalten']
['80', '16', '63.1', '0.281', '0.258', '0.450', '-5.18', '0.00388', 'FALSE']
['84', '35', '57.1', '0.445', '0.258', '0.450', '-500.00', '0.00424', 'TRUE']
['23', '14', '44.1', '0.586', '0.258', '0.450', '-2.72', '0.00413', 'TRUE']
['64', '33', '65.0', '0.292', '0.258', '0.450', '1.66', '0.00323', 'FALSE']
['90', '28', '48.8', '0.241', '0.258', '0.450', '-0.47', '0.00188', 'TRUE']
['90', '28', '49.1', '0.240', '0.258', '0.450', '-0.47', '0.00186', 'TRUE']
['74', '57', '66.5', '0.320', '0.258', '0.450', '3.26', '0.00398', 'FALSE']
['23', '14', '43.8', '0.563', '0.258', '0.450', '-2.72', '0.00397', 'TRUE']
['60', '44', '32.4', '0.548', '0.258', '0.450', '-1.02', '0.00237', 'TRUE']
['54', '17', '48.0', '0.304', '0.258', '0.450', '0.26', '0.00234', 'FALSE']
['54', '17', '48.1', '0.317', '0.258', '0.450', '0.26', '0.00244', 'FALSE']
['80', '36', '64.0', '0.315', '0.258', '0.450', '-6.33', '0.00430', 'FALSE']
['106', '104', '52.7', '0.874', '0.441', '0.198', '0.58', '0.00270', 'FALSE']
['119', '73', '50.7', '0.400', '0.441', '0.198', '3.24', '0.00123', 'FALSE']
['25', '81', '51.0', '0.452', '0.441', '0.198', '7.35', '0.00108', 'FALSE']
['79', '38', '39.7', '0.352', '0.441', '0.198', '-0.06', '0.00137', 'FALSE']
['79', '38', '40.0', '0.306', '0.441', '0.198', '-0.06', '0.00119', 'FALSE']

```

```

In [514]: FarbeM = np.zeros((3)).astype(float)
Tripel = np.zeros((3,3)).astype(float)
Deltas = np.zeros((0,3)).astype(float)
for row in MacAdamTableIII[1:]:
#     if row[8]=='FALSE':
    if row:
        xySollfarbe = np.array([float(row[4]),float(row[5])])
        Filter0 = MacAdamTableI[int(row[0])-1]
        Filter90 = MacAdamTableI[int(row[1])-1]
        FarbeV = np.array(Filter0[3:6]).astype(float)
        FarbeU = np.array(Filter90[3:6]).astype(float)
        Winkel = np.array([float(row[2]) - float(row[3]), float(row[2]), float(row[2]) + float(row[3])])
        for n in range(0, 3):
            U = (np.sin(float(Winkel[n])/180*np.pi)**2
            V = 1 - U
            FarbeM = FarbeV*V + FarbeU*U
            xy = colour.XYZ_to_xy(FarbeM)

```

```

    Tripel[n] = FarbeM
#     Striche = np.append(Striche, [np.transpose(Tripel)], axis=0)
    Deltas = np.append(Deltas, [FarbeM], axis=0)
#Deltas = Deltas.T
print(Deltas)

```

```

[[0.00168004 0.00278347 0.00221547]
 [0.00165986 0.00278188 0.00217539]
 [0.00163983 0.0027803  0.00213561]
 ...
 [0.003619  0.002969  0.00114717]
 [0.0036167 0.0029667 0.00110924]
 [0.00361443 0.00296443 0.00107173]]

```

```

In [9]: class MacAdamGraf():
    def __init__(self, Titel, Xachse, Yachse, Xindex, Yindex, Null, Spektralkurve, Striche):
        self.Titel = Titel
        self.Xachse = Xachse
        self.Yachse = Yachse
        self.Xindex = Xindex
        self.Yindex = Yindex
        self.Null = Null
        self.Spektralk = Spektralkurve
        self.Striche = Striche

    def zeichnen(self):
        cmfs = colour.STANDARD_OBSERVERS_CMFS['CIE 1931 2 Degree Standard Observer']
        illuminant = colour.ILLUMINANTS['CIE 1931 2 Degree Standard Observer']['C']
        #illuminant = colour.ILLUMINANTS_RELATIVE_SPDS['C']

        #Graf wird initialisiert
        %pylab inline
        %config InlineBackend.figure_formats=['svg']
        pylab.rcParams['figure.figsize'] = (10, 10)
        pylab.xlabel(self.Xachse)
        pylab.ylabel(self.Yachse)
        pylab.title(self.Titel)
        pylab.axis('equal')
        pylab.grid(True)

```

```

#Es wird versucht, den Spektralkurvenzug zu zeichnen
# pylab.plot(self.Spektralk[...], self.Xindex, self.Spektralk[...], self.Yindex, color='orange', li

#Es wird versucht, die Purpurgerade zu zeichnen
#pylab.plot(
# (self.Spektralk[-1][self.Xindex], self.Spektralk[0][self.Xindex]), (self.Spektralk[-1][self.Yi
# color='purple',
# linewidth=1)

#alle X,Y und Z werden wieder in 3 x 3 - Matrizen aufgeteilt
self.Striche = self.Striche.reshape(int(self.Striche.shape[0]/3), 3, self.Striche.shape[1])

#die MacAdam-Striche werden gezeichnet und ihre Länge wird errechnet
LaengenFeld = np.array([[ ],[ ]])
for Koordinaten in self.Striche:
    Laenge = np.sum((Koordinaten[0]-Koordinaten[1])**2)**.5
    print(Koordinaten[1], self.Null)
    Abstand = np.sum((Koordinaten[1] - self.Null)**2)**.5
    LaengenFeld = np.concatenate((LaengenFeld, np.array([[Laenge], [Abstand]])), axis=1)
    Koordinaten = Koordinaten.transpose()
    pylab.plot(Koordinaten[self.Xindex],Koordinaten[self.Yindex], color='green')
guete = np.amax(LaengenFeld[0])/np.amin(LaengenFeld[0])
plt.figure(figsize=(5,5))
pylab.xlabel("Abstand von Null")
pylab.ylabel("Längen")
pylab.title("Längenverteilung")
pylab.plot(LaengenFeld[1], LaengenFeld[0], 'bo', color='red')
print("Verhältnis:", guete,
      ",\nDurchschnitt: ", np.average(LaengenFeld[0]),
      ",\nMedian: ", np.median(LaengenFeld[0]),
      ",\nMinimum: ", np.amin(LaengenFeld[0]),
      ",\nMaximum: ", np.amax(LaengenFeld[0]))
plt.show()
return(guete)

```

```

In [10]: GrafXYZ = MacAdamGraf("MacAdam Ellipsen im CIE-XYZ-Farbraum", "CIE-X", "CIE-Z", 0,2, [0, 0, 0], cmfs.values
GrafXYZ.zeichnen()

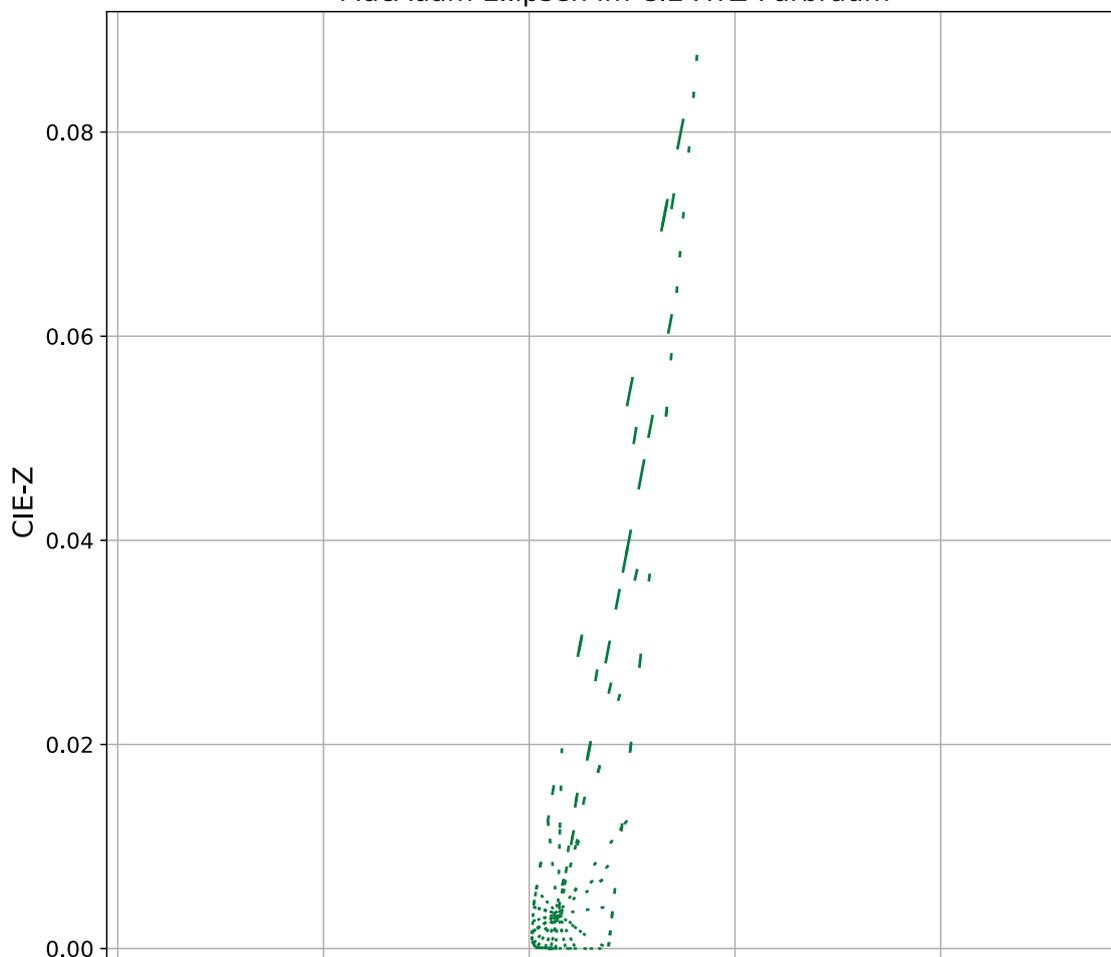
```

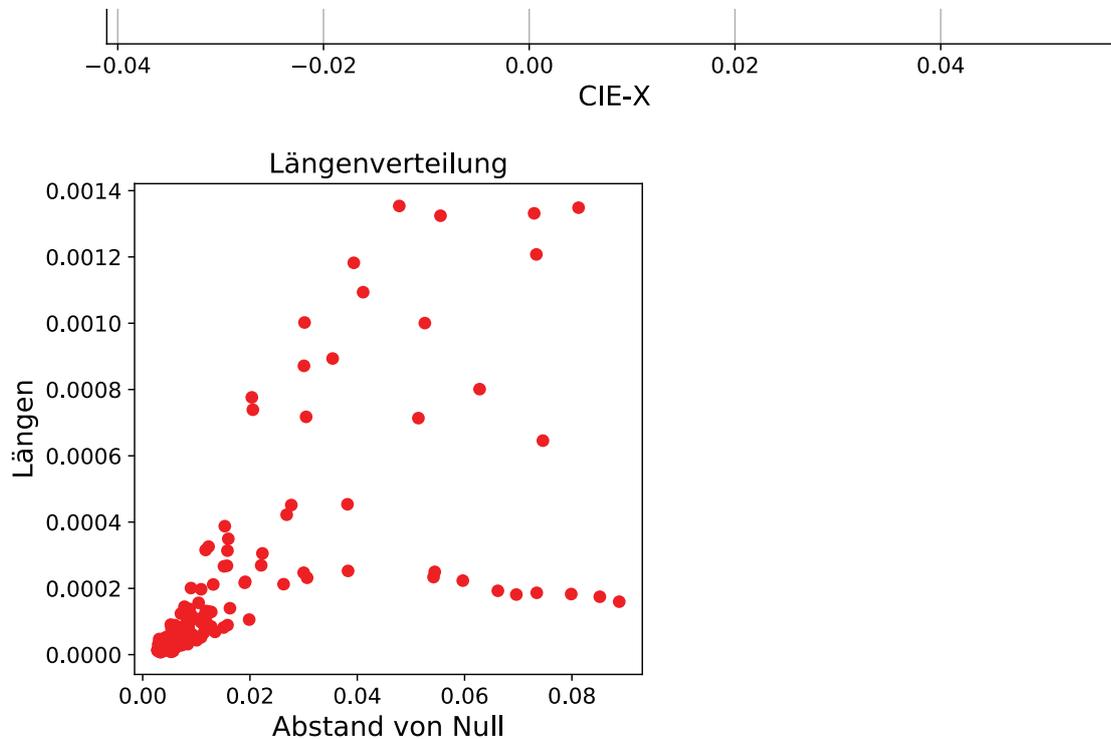
```

Populating the interactive namespace from numpy and matplotlib
Verhältnis: 177.91576320022898 ,
Durchschnitt: 0.00014078479731521052 ,
Median: 3.403118397587086e-05 ,
Minimum: 7.609312382282283e-06 ,
Maximum: 0.0013538166199227048

```

MacAdam Ellipsen im CIE-XYZ-Farbraum





Out[10]: 177.91576320022898

```
In [14]: # umgeschrieben für MacAdam
# neural network class definition
class neuralNetwork:

    # initialise the neural network
    def __init__(self, inputnodes, hiddennodes, outputnodes):
        # set number of nodes in each input, hidden, output layer
        self.inodes = inputnodes
        self.hnodes = hiddennodes
        self.onodes = outputnodes

        # link weight matrices, wih and who
        # weights inside the arrays are w_i_j, where link is from node i to node j in the next layer
        # w11 w21
        # w12 w22 etc
        tol = 15
        muu = 0.5
        muo = 2
        self.wih = np.array([[random.uniform(-tol, tol), random.uniform(-tol, tol), random.uniform(-tol, to
            [random.uniform(-tol, tol), random.uniform(-tol, tol), random.uniform(-tol, to
            [random.uniform(-tol, tol), random.uniform(-tol, tol), random.uniform(-tol, to
        self.who = np.array([[random.uniform(-tol, tol), random.uniform(-tol, tol), random.uniform(-tol, to
            [random.uniform(-tol, tol), random.uniform(-tol, tol), random.uniform(-tol, to
        self.wih.astype(np.float64)
        self.who.astype(np.float64)

        # activation function is the sigmoid function
        self.activation_function = lambda x: scipy.special.expit(x)

    pass

    # train the neural network
    def train(self, inputs_list, targets_list, learningrate):
        # learning rate
        self.lr = learningrate

        # convert inputs list to 2d array
        inputs = numpy.array(inputs_list, ndmin=2).T
        targets = numpy.array(targets_list, ndmin=2).T

        # calculate signals into hidden layer
        hidden_inputs = numpy.dot(self.wih, inputs)
        # calculate the signals emerging from hidden layer
        hidden_outputs = self.activation_function(hidden_inputs)

        # calculate signals into final output layer
        final_inputs = numpy.dot(self.who, hidden_outputs)
        # calculate the signals emerging from final output layer
        final_outputs = self.activation_function(final_inputs)

        # output layer error is the (target - actual)
```

```

# da nur eine Ableitung gesucht wird, muss das Vorzeichen genau anders herum sein, wie bei
# einer absoluten Größe
output_errors = final_outputs - targets[1:3]
# hidden layer error is the output_errors, split by weights, recombined at hidden nodes
hidden_errors = numpy.dot(self.who.T, output_errors)
#
    print(output_errors, final_outputs, hidden_outputs)

# update the weights for the links between the hidden and output layers
self.who += self.lr * numpy.dot((output_errors * final_outputs * (1.0 - final_outputs)), numpy.transp

# update the weights for the links between the input and hidden layers
self.wih += self.lr * numpy.dot((hidden_errors * hidden_outputs * (1.0 - hidden_outputs)), numpy.transp

pass

# query the neural network
def query(self, inputs_list):
    # convert inputs list to 2d array
    #inputs = numpy.array(inputs_list, ndmin=2).T wird bei einem Eindimensionalen Eingabevektor nicht g
    inputs = numpy.array(inputs_list, ndmin=2).T
    # calculate signals into hidden layer
    hidden_inputs = numpy.dot(self.wih, inputs)
    # calculate the signals emerging from hidden layer
    hidden_outputs = self.activation_function(hidden_inputs)

    # calculate signals into final output layer
    final_inputs = numpy.dot(self.who, hidden_outputs)
    # calculate the signals emerging from final output layer
    final_outputs = self.activation_function(final_inputs)

    return final_outputs

```

```

In [17]: # Ab hier wird der Graf gezeichnet
from IPython.display import clear_output
from colour.utilities import tsplit, tstack
cmfs = colour.STANDARD_OBSERVERS_CMFS['CIE 1931 2 Degree Standard Observer']
illuminant = colour.ILLUMINANTS['CIE 1931 2 Degree Standard Observer']['C']

def XYZ_to_sigmoidLab(
    XYZ,
    illuminant=colour.ILLUMINANTS['CIE 1931 2 Degree Standard Observer']['C']):

    XYZ = np.asarray(XYZ)
    XYZ_r = colour.xyY_to_XYZ(colour.xy_to_xyY(illuminant))

    XYZ_f = XYZ / XYZ_r

    sigmoid = lambda X: scipy.special.expit(X)
    # Variante der durch probieren gefunden Faktoren siehe MacAdamStricheVerrObjektePunktvert
    # so umgesetzt, dass man mit zwei dot-Operationen zum selben Ergebniss kommt.
    # eine addierte Konstante, in diesem Fall - 0.5 ändert die Ableitung nicht, darf als verwendet werden
    # Umrechnung analog CAM02 mit Faktor 9 Multipliziert
    ih = np.array([[ 5.73023365,  2.77487545, -1.39854523],
                   [-5.94844201, 16.192547,  0.30385909],
                   [ 0.29041382, -0.1930856, 10.8518794 ],
                   [ 2.41411996,  5.75433054, -6.6977864 ],
                   [ 5.18921823,  5.27202788, -1.60201707]])

    ho = np.array([[ -0.62661618,  5.4519609, -2.01235018, -0.93430437,  3.26259839],
                   [ 20.48459487, -7.93787321, -2.15241145,  1.12826774, -1.67940661],
                   [ 6.33228589,  6.59367038, -14.68664308, -9.85619169, -8.57344153]])

    sigmoidLab = []
    if XYZ_f.ndim > 1:
        for row in XYZ_f:
            row = sigmoid(np.dot(ih, row))-0.5
            row = sigmoid(np.dot(ho, row))
            sigmoidLab.append(row)
    else:
        sigmoidLab = sigmoid(np.dot(ih, XYZ_f))-0.5
        sigmoidLab = sigmoid(np.dot(ho, sigmoidLab))
    sigmoidLab = np.array(sigmoidLab)

    return sigmoidLab

def XYZ_to_ownNeuro(
    XYZ,
    illuminant=colour.ILLUMINANTS['CIE 1931 2 Degree Standard Observer']['C']):

    XYZ = np.asarray(XYZ)
    XYZ_r = colour.xyY_to_XYZ(colour.xy_to_xyY(illuminant))

    XYZ_f = XYZ / XYZ_r
    # vorab bereits die XYZ in CIECAM02-XYZ umrechnen

    sigmoid = lambda X: scipy.special.expit(X)
    # Variante bei der n.query vom Neuronalen Netz direkt angesprochen wird.

```

```

ownNeuro = []
if XYZ_f.ndim > 1:
    for row in XYZ_f:
        row = n.query(row)
        ownNeuro.append(row)
else:
    ownNeuro = n.query(XYZ_f)
ownNeuro = np.array(ownNeuro)
return ownNeuro
#Der erste Faktor im array ist der angestrebte Helligkeitswert
# Umkehrfunktion  $y = \text{sigmoid}(x)$  ist  $x = \ln(y/(1-y))$ 
# CIE-XYZ von Normlicht C ist 0,309 0,319 0,372
Null = [21, 0, 0]
HelligkeitsY = colour.Lab_to_XYZ(np.asarray(Null))[0]
#HelligkeitsY = 50 [21, 19, 45]
Null = np.array([0.309, 0.319, 0.372]) / 0.319 * HelligkeitsY

#Speziell bei Lab müssen die Spektralkurven auf eine bestimmte Helligkeit normiert werden

#für jeden Spektralwert muss ein eigener Faktor errechnet werden, mit dem dann
#alle drei Wert multipliziert werden müssen
Spektralkurven = cmfs.values/100
Lfaktoren = HelligkeitsY / Spektralkurven[... , 1]
#Aus dem eindimensionalen Feld mit den Helligkeitsfaktoren muss eines mit Tiefe 3 gemacht werden
Spektralkurven = Spektralkurven * np.array([Lfaktoren,Lfaktoren,Lfaktoren]).T
Spektralkurven = XYZ_to_sigmoLab(Spektralkurven)

#Die eigentlichen MacAdam-Werte bezogen auf vorgegebene Lab-Helligkeit Lfaktoren wird neu besetzt
Lfaktoren = HelligkeitsY / Deltas[... , 1]
DeltasGew = Deltas * np.array([Lfaktoren,Lfaktoren,Lfaktoren]).T

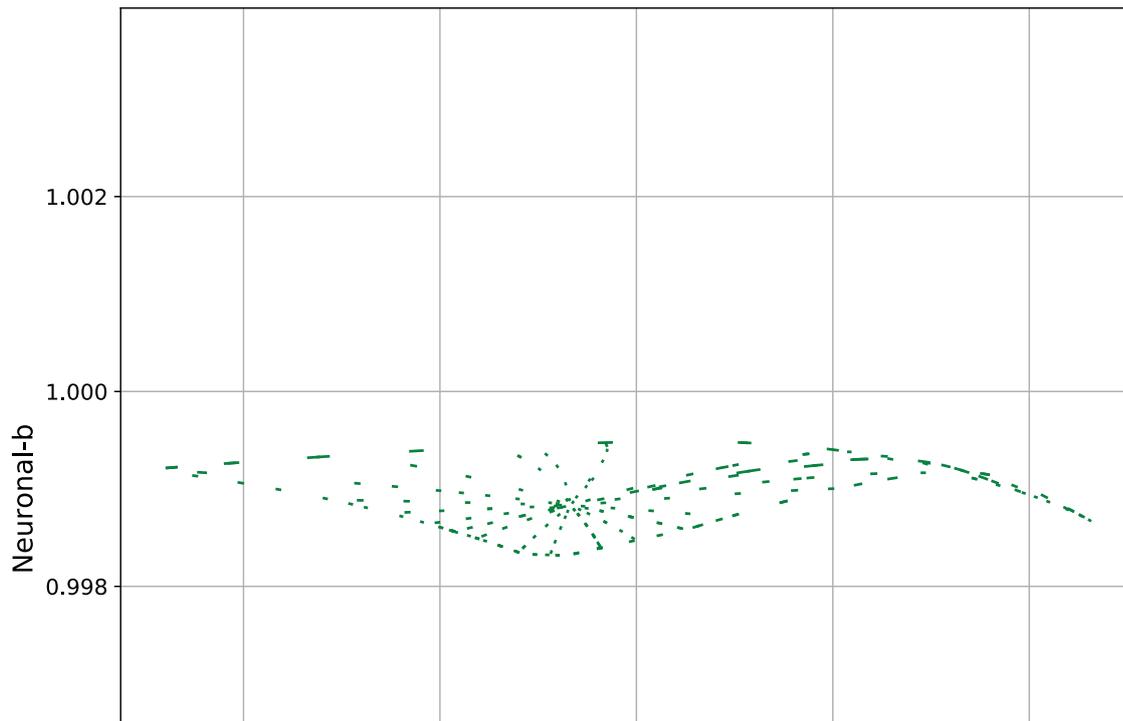
print( XYZ_to_ownNeuro(Null))
GrafABC = MacAdamGraf('Neuronal3-2-2 ermittelt',
                    'Neuronal-a', 'Neuronal-b', 0, 1,
                    XYZ_to_ownNeuro(Null).reshape(1,2)[0],
                    Spektralkurven,
                    XYZ_to_ownNeuro(DeltasGew))
# print dient nur dazu, den Übergabewert auszugeben. .zeichnen gibt den Graf aus.
GrafABC zeichnen()
GrafLab = MacAdamGraf('Neuronal3-5-3 ermittelt',
                    'b', 'L', 1, 2,
                    XYZ_to_sigmoLab(Null),
                    Spektralkurven,
                    XYZ_to_sigmoLab(DeltasGew))

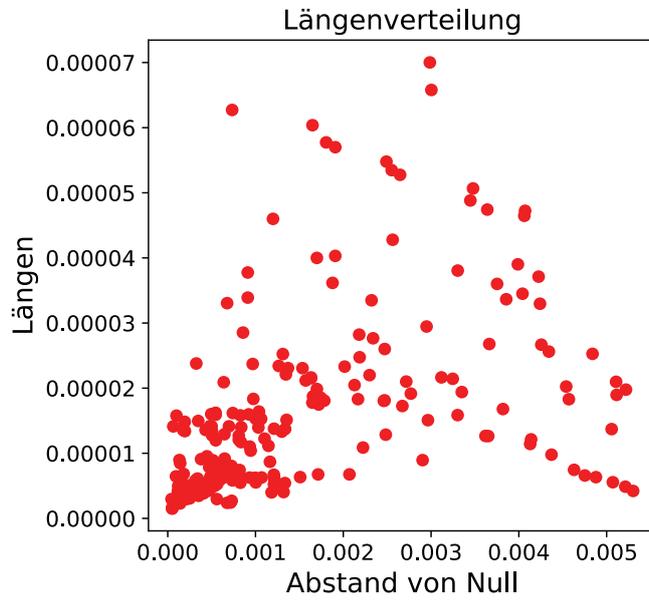
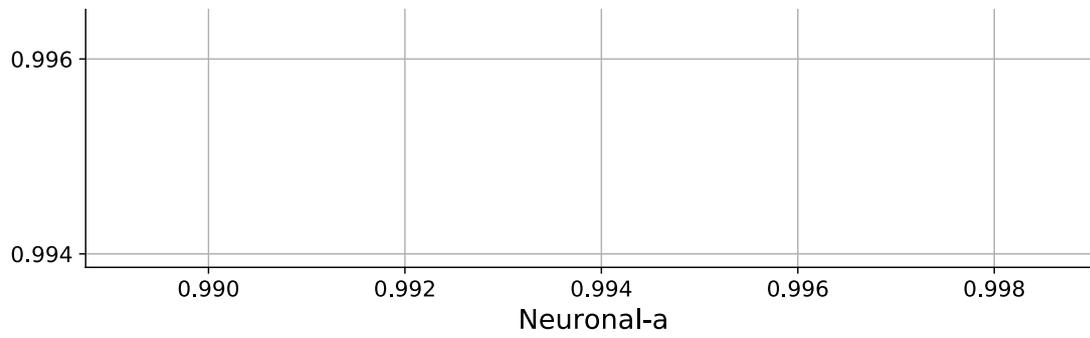
GrafLab zeichnen()

[[0.99331565]
 [0.99884056]]
Populating the interactive namespace from numpy and matplotlib
Verhältnis: 46.42783504407383 ,
Durchschnitt: 1.5791574770125898e-05 ,
Median: 1.2444092686512492e-05 ,
Minimum: 1.5081320382604452e-06 ,
Maximum: 7.00193054970388e-05

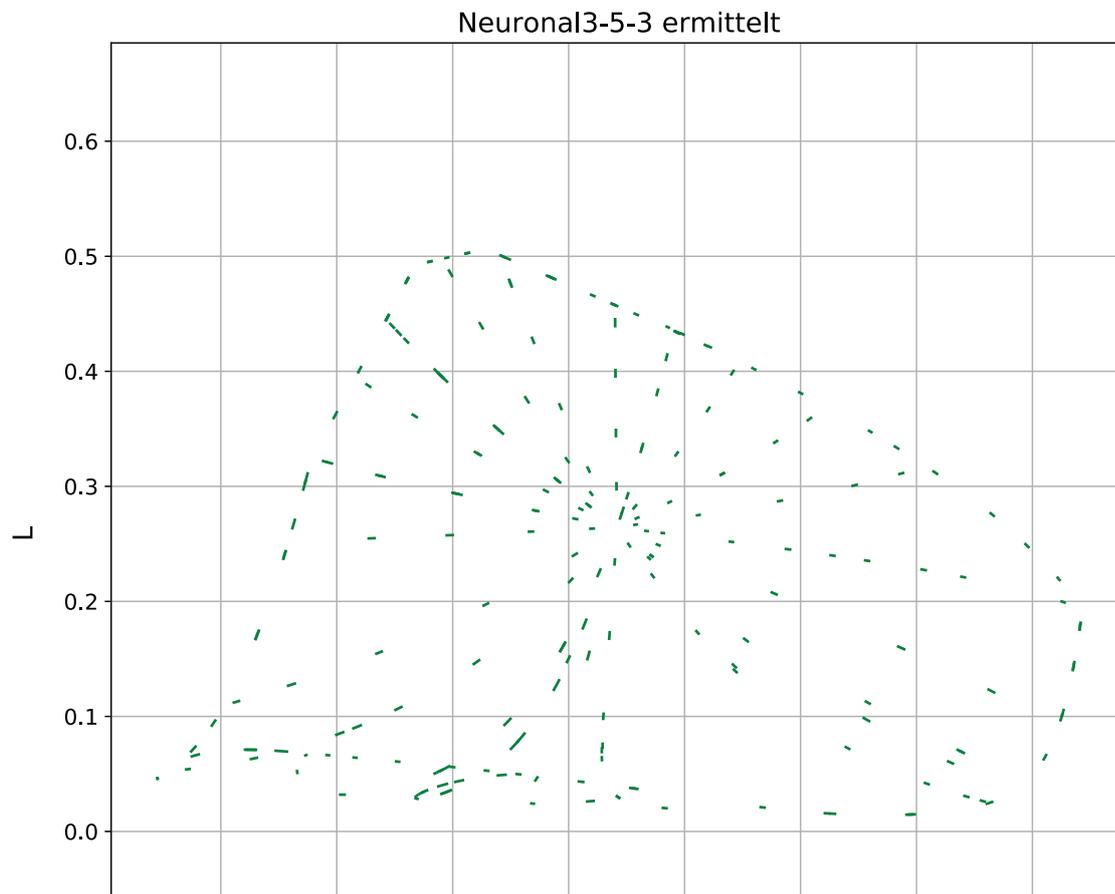
```

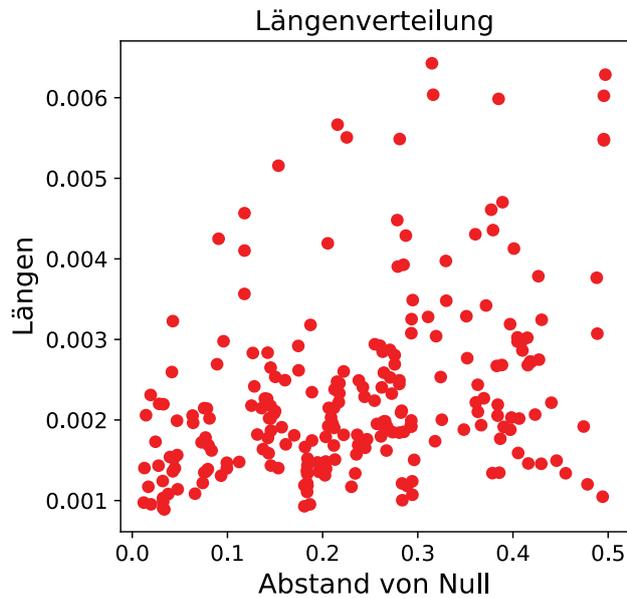
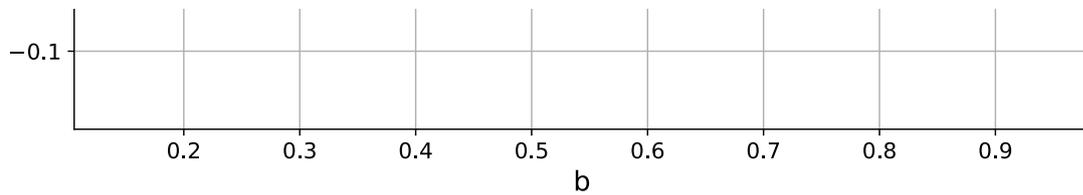
Neuronal3-2-2 ermittelt





Populating the interactive namespace from numpy and matplotlib
Verhältnis: 7.222619531444553 ,
Durchschnitt: 0.002333463151399638 ,
Median: 0.0020311351248704364 ,
Minimum: 0.0008899187643448735 ,
Maximum: 0.006427544648756286





Out[17]: 7.222619531444553

In [628]: `print(guetemin)`

6.031701891152638

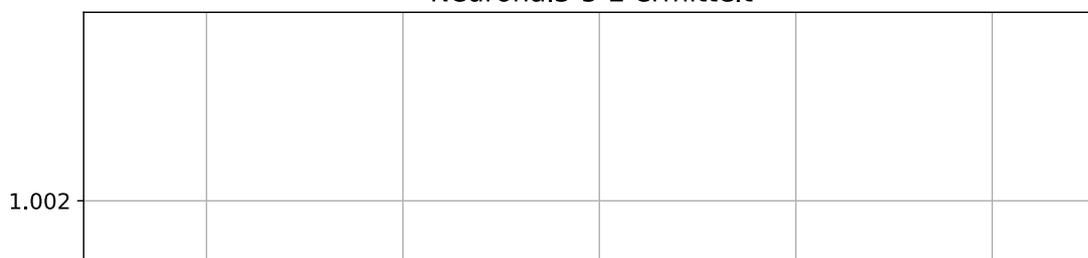
```
In [16]: # umgeschrieben für MacAdam
# number of input, hidden and output nodes
guete = 1000
guetemin = 100
while guete > 100:
    clear_output(wait=True)
    input_nodes = 3
    hidden_nodes = 5
    output_nodes = 2

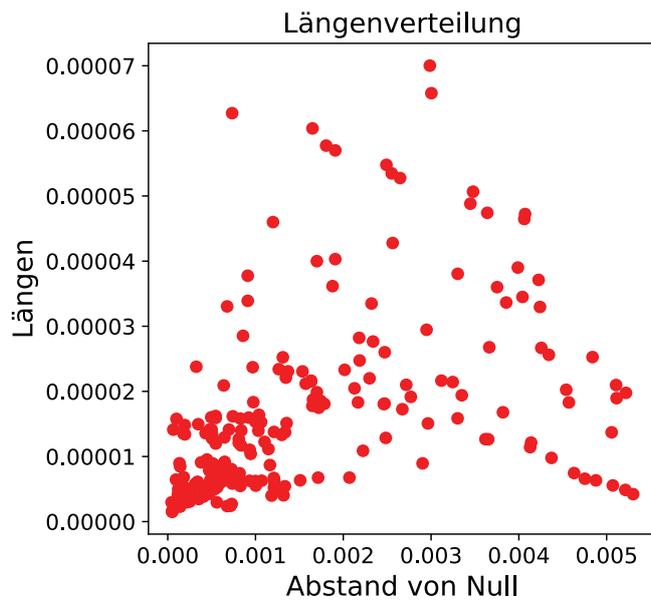
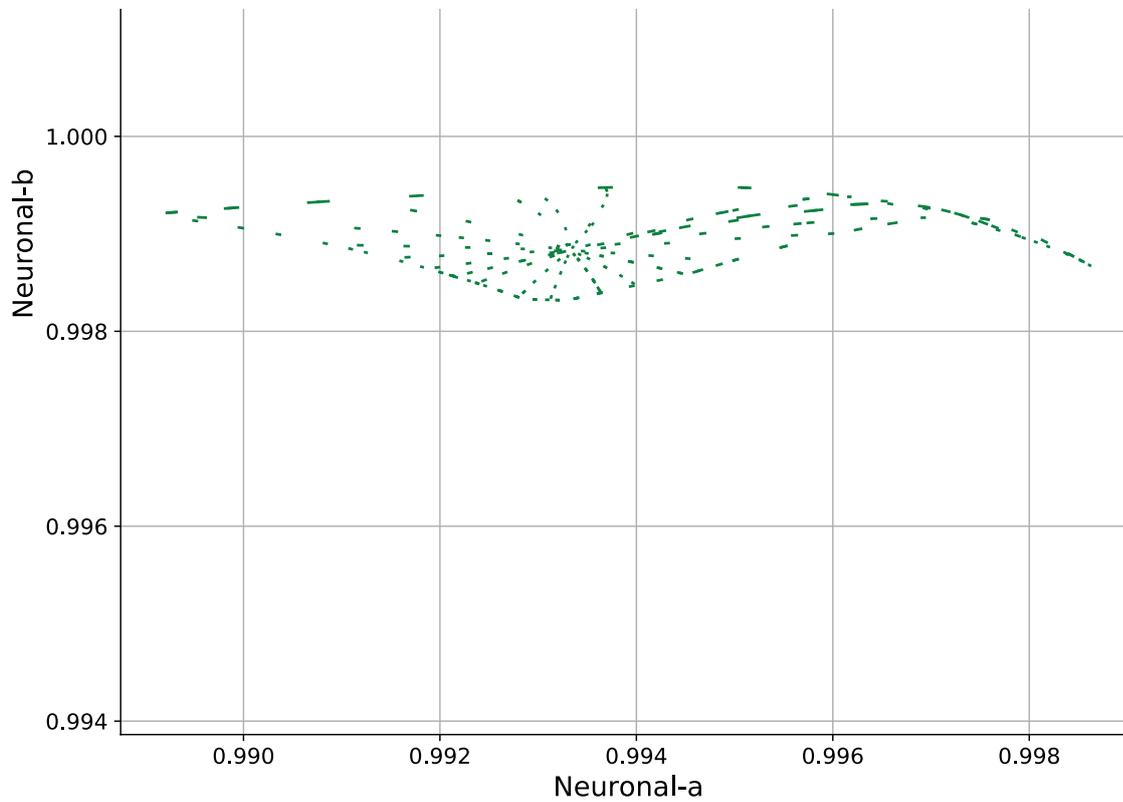
    # create instance of neural network
    n = neuralNetwork(input_nodes,hidden_nodes,output_nodes)
    GrafABC = MacAdamGraf('Neuronal3-5-2 ermittelt',
                          'Neuronal-a', 'Neuronal-b', 0, 1,
                          XYZ_to_ownNeuro(Null).reshape(1,2)[0],
                          Spektralkurven,
                          XYZ_to_ownNeuro(DeltasGew))

    # print dient nur dazu, den Übergabewert auszugeben. .zeichnen gibt den Graf aus.
    guete = GrafABC.zeichnen()
    if guete < guetemin:
        guetemin = guete
        sicherwih = n.wih
        sicherwho = n.who
```

Populating the interactive namespace from numpy and matplotlib
Verhältnis: 46.42783504407383 ,
Durchschnitt: 1.5791574770125898e-05 ,
Median: 1.2444092686512492e-05 ,
Minimum: 1.5081320382604452e-06 ,
Maximum: 7.00193054970388e-05

Neuronal3-5-2 ermittelt





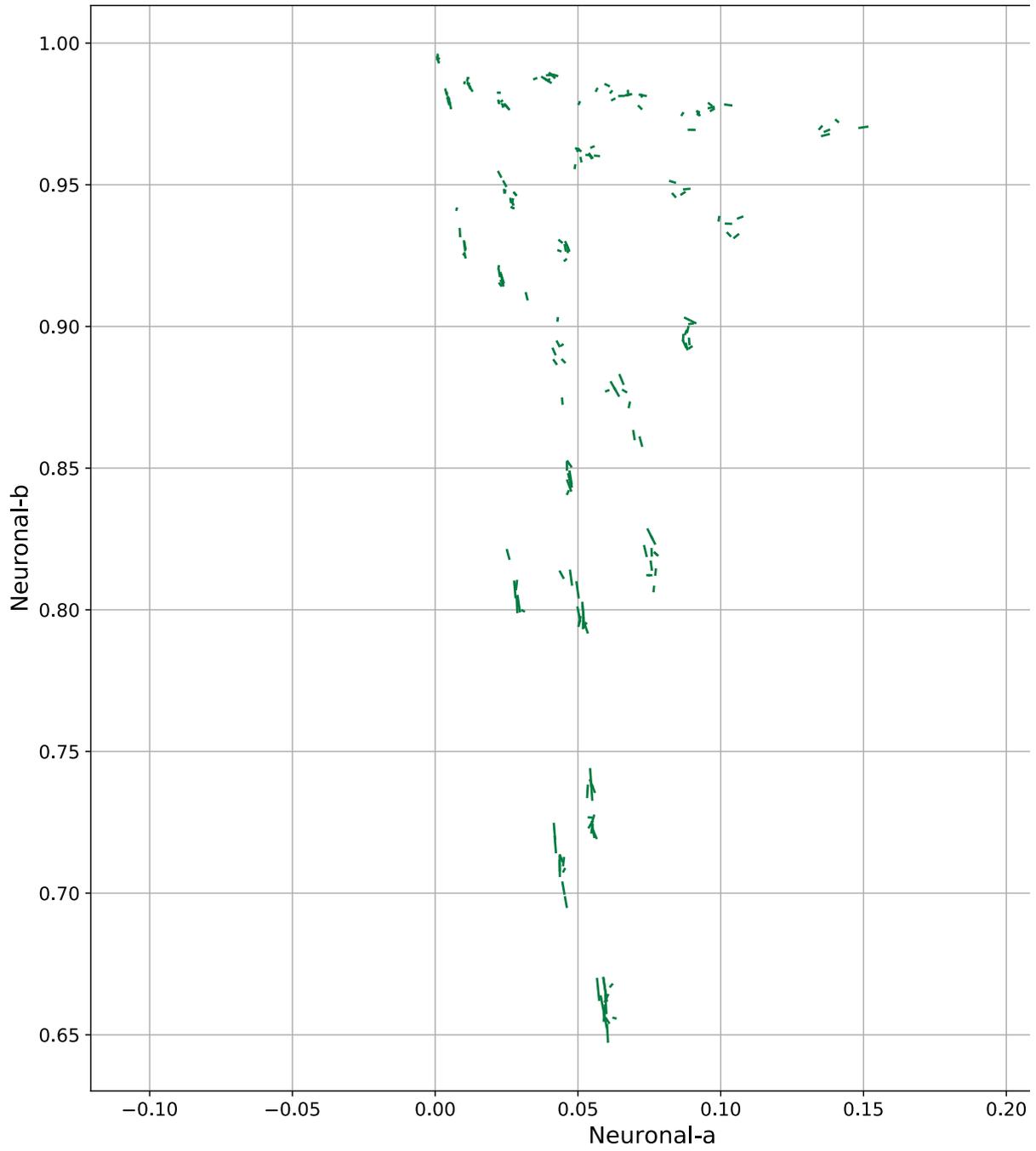
```
In [522]: #initialisierung mit nur einem Versuch
input_nodes = 3
hidden_nodes = 3
output_nodes = 2

# learning rate
learning_rate = 0.1

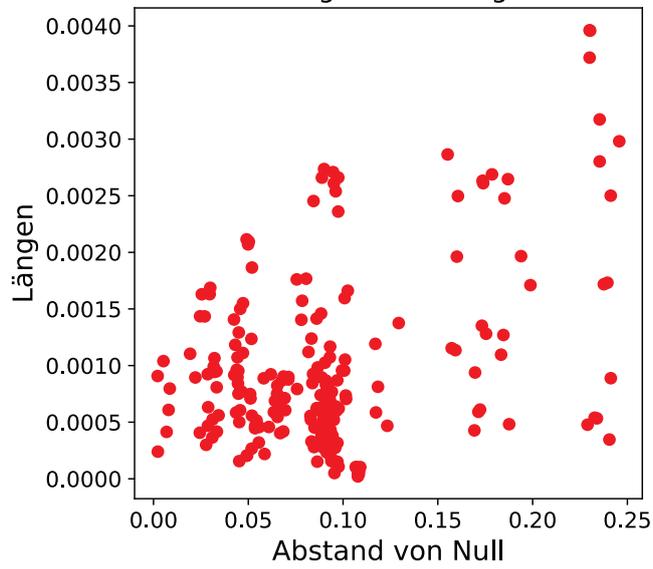
# create instance of neural network
n = neuralNetwork(input_nodes,hidden_nodes,output_nodes)
GrafABC = MacAdamGraf('Neuronal3-5-3 ermittelt',
                      'Neuronal-a', 'Neuronal-b', 0, 1,
                      XYZ_to_ownNeuro(Null).reshape(1,2)[0],
                      Spektralkurven,
                      XYZ_to_ownNeuro(DeltasGew))
# print dient nur dazu, den Übergabewert auszugeben. .zeichnen gibt den Graf aus.
guete = GrafABC.zeichnen()
```

```
Populating the interactive namespace from numpy and matplotlib
Verhältnis: 187.22036422793576 ,
Durchschnitt: 0.0009786054222889148 ,
Median: 0.000751150973550816 ,
Minimum: 2.1161175122266223e-05 ,
Maximum: 0.003961802913881815
```

Neuronal3-5-3 ermittelt



Längenverteilung



```

In [659]: # umgeschrieben für MacAdam
# train the neural network

# epochs is the number of times the training data set is used for training
# Vorsicht, 10000 Epochen dauern eine ganze Weile
epochs = 10
Richtung = 1
learning_rate = 1
# Y als Helligkeitswert an die erste Stelle
for e in range(epochs):
    DeltasT = DeltasGew.T
# Helligkeit Y wird an die erste Stelle gesetzt
    training_data_list = numpy.asfarray([DeltasT[1], DeltasT[0], DeltasT[2]].T)
# alle X, Y und Z werden wieder in 3 x 3 - Matrizen aufgeteilt
    training_data_list = training_data_list.reshape(int(training_data_list.shape[0]/3), 3,3)

# go through all records in the training data set
for record in training_data_list:
    # Der untere Wert für die MacAdam-Striche wird direkt verwendet
    record = record[:,Richtung]
    record[0][1:3] = n.query(record[0]).reshape((2,))
    record[1][1:3] = n.query(record[1]).reshape((2,))
    inputs = numpy.asfarray(record[2])
    # Das Ziel muss relativ kompliziert berechnet werden, siehe Arbeit und Notiz unten
    Dif = record[0]-record[1]
    # Länge des Differenzvektors von MacAdam-StricheMitte zu MacAdam-StricheOben
    LaengeDif = np.sum(Dif[1:3]**2)**.5
    # Soll Differenz eines MacAdam-Striches ist 0.5 % ist 0.005. Ein halber wird berechnet
    targets = numpy.asfarray(record[1] - Dif * 0.0013 / (LaengeDif + 0.0000000001))
# errors = record[2]-targets
# print(Dif, ' ',inputs, ' ',targets)
    n.train(inputs, targets, learning_rate)
pass
pass
GrafABC = MacAdamGraf('Neuronal3-5-2 ermittelt',
                      'Neuronal-a', 'Neuronal-b', 0, 1,
                      XYZ_to_ownNeuro(Null).reshape(1,2)[0],
                      Spektralkurven,
                      XYZ_to_ownNeuro(DeltasGew))
# print dient nur dazu, den Übergabewert auszugeben. .zeichnen gibt den Graf aus.
guete = GrafABC.zeichnen()

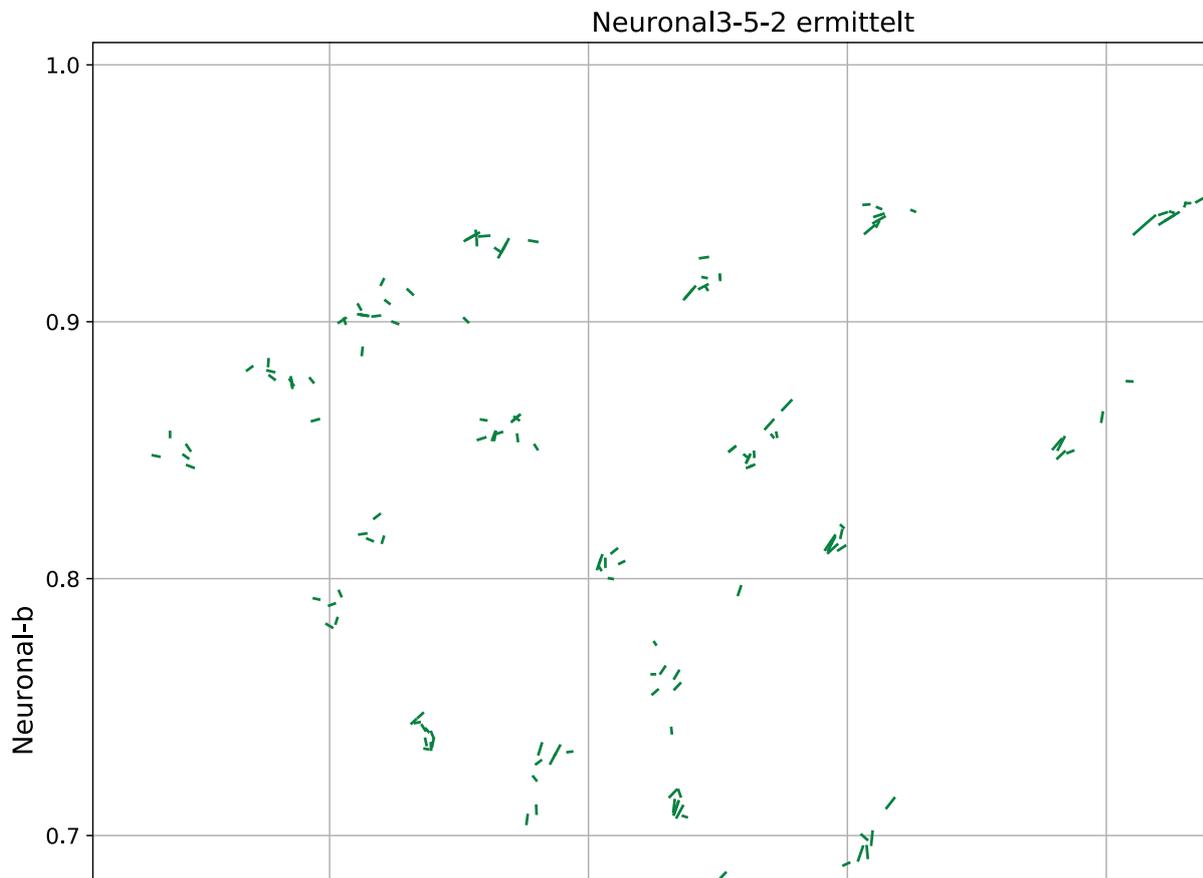
```

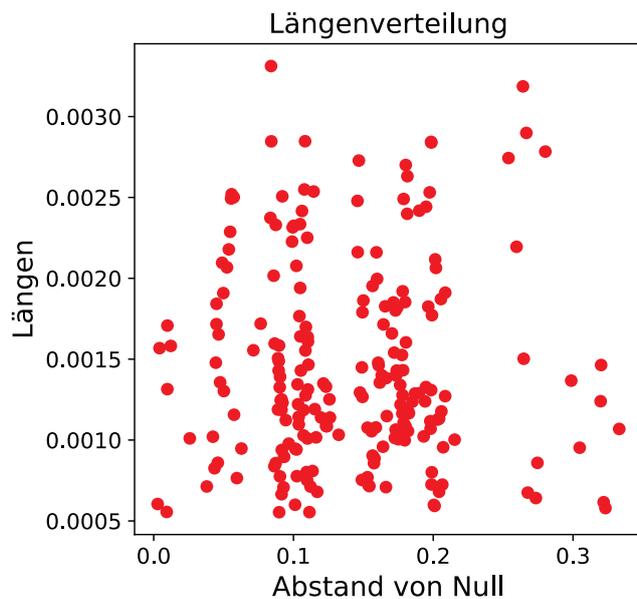
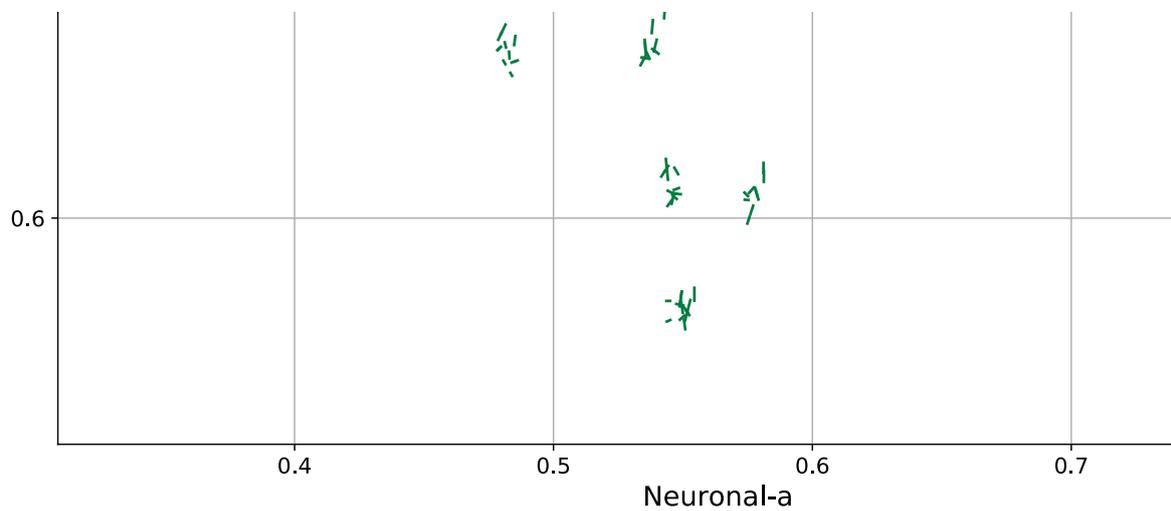
Populating the interactive namespace from numpy and matplotlib

```

Verhältnis: 5.981109694816019 ,
Durchschnitt: 0.0014610033631597835 ,
Median: 0.001313923547556324 ,
Minimum: 0.0005537787210578601 ,
Maximum: 0.0033122112773019833

```





```
In [629]: print (sicherwih, "\n", sicherwho)
```

```
[[ 14.39019108 -4.15324537 -3.03877026]
 [-13.29363919  8.42441535 -7.02333693]
 [-7.25041206  8.95232115 -0.87799647]]
[[ -6.17868211 -4.46975342  4.62994242]
 [ 2.61450341 -10.48461674  10.80090746]]
```

```
In [630]: n.wih = np.array([[ 14.39019108, -4.15324537, -3.03877026],
 [-13.29363919,  8.42441535, -7.02333693],
 [-7.25041206,  8.95232115, -0.87799647]])
n.who = np.array([[ -6.17868211, -4.46975342,  4.62994242],
 [ 2.61450341, -10.48461674,  10.80090746]])
```

Je MacAdam-Messung gibt es drei Ortsvektoren. Einmal die Farbe im Mittelpunkt und die Farben, die die Abweichung in beide Richtungen repräsentieren. Aus dem Vektor, der die Farbe Mittelpunkt repräsentiert und der ersten Farbe am Randfand, lässt sich durch einfache Vektorrechnung der Differenzvektor errechnen. Die zweite Randfarbe lässt sich errechnen, in dem man den Ortsvektor der Mittelfarbe mit dem Differenzvektor mit umgekehrtem Vorzeichen addiert. Wenn wir nun einen gleichabständigen Farbraum mit Einheit 1 MacAdam-Abstand hätten, sollte der ideale Differenzvektor die Länge 0,5 haben bzw. ohne Multiplikation mit 100, die Differenz 0.005. Man muss also nur den realen Differenzvektor so strecken, dass er 0,5 lang wird und dann den idealen Ort berechnen, in dem vom Ortsvektor der Mittelfarbe der ideale Differenzvektor abgezogen wird. Nun kann man dem neuronalen Netz den Idealen Ort als Zielwert vorgeben, von dem der reale Ort abweicht. Anders als bei üblichen neuronalen Netzen, ändert sich der Zielwert mit den Näherungsschritten, so dass er für den jeweiligen Durchlauf immer aktuell neu berechnet werden muss, was eine etwas abweichende Programmierung zu üblichen neuronalen Netzen verlangt. Dies resultiert aber aus der Tatsache, dass der idealen angestrebte Farbraum eben unbekannt ist. Es sind nur Farbabweichungen im realen CIE-XYZ-Farbraum bekannt, die in einem idealen Farbraum die Distanz 0,5 oder 1 haben sollen.

```
In [5500]: #[[ 6.5952,  3.8664, -1.4616],
# [-6.3324, 15.2775,  0.0549],
# [ 0.027,  0.122,  8.8506]]
#[[0.0001, 6, 0.0001],
# [21, -7, 0.0001],
# [6. 6.0001. -1511]
```

```
print(n.wih)
```

```
[[ 6.61278277e+00  3.86152518e+00 -1.51607176e+00]
 [-6.34330663e+00  1.52842047e+01 -9.06854700e-03]
 [ 1.66680735e-01  1.80138362e-01  9.46448520e+00]
 [-7.71414535e+00  1.01193830e+01  4.21850666e+00]]
```

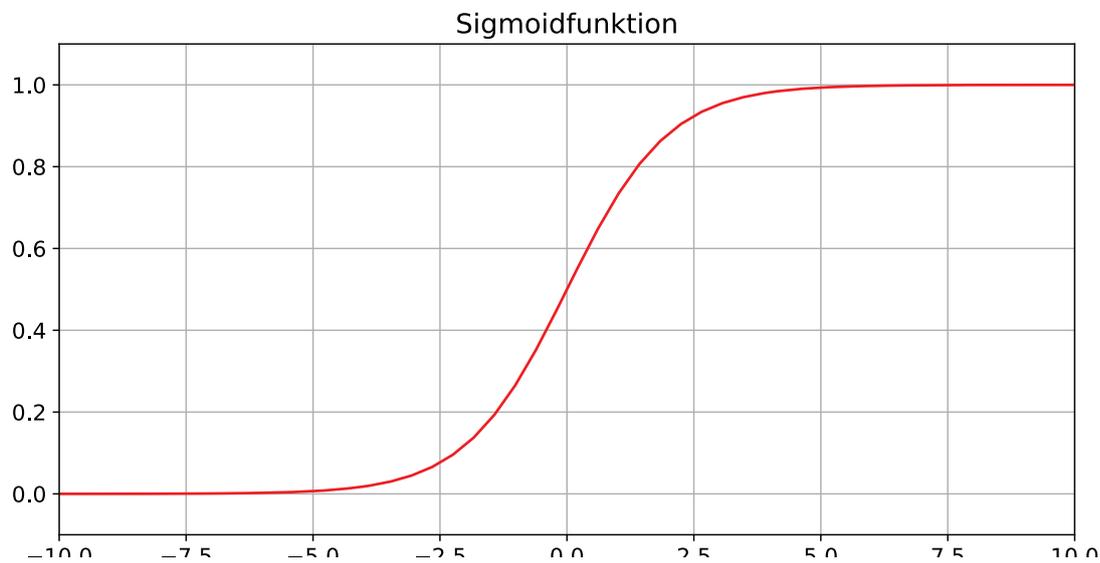
```
In [5501]: print(n.who)
```

```
[[ -4.31909832e-03  6.00231115e+00  1.52660395e-02  6.01437213e+00]
 [ 2.09954074e+01 -7.00925315e+00 -1.56466120e-02 -1.70113656e+01]
 [ 6.00336807e+00  5.99832208e+00 -1.50021298e+01 -1.70035413e+01]]
```

```
In [15]: %pylab inline
%config InlineBackend.figure_formats=['svg']
pylab.rcParams['figure.figsize'] = (10, 5)
#pylab.xlabel(self.Xachse)
#pylab.ylabel(self.Yachse)
#pylab.title(self.Titel) n.activation_function
pylab.grid(True)

X = np.linspace(-10, 10, endpoint=True)
mist = lambda X: scipy.special.expit(X)
F = mist(X)
plt.plot(X,F)
pylab.title("Sigmoidfunktion")
startx, endx = -10, 10
starty, endy = -0.1, 1.1
plt.axis([startx, endx, starty, endy])
plt.show()
```

Populating the interactive namespace from numpy and matplotlib



Transformationsmatrix weitgehend Unabhängig

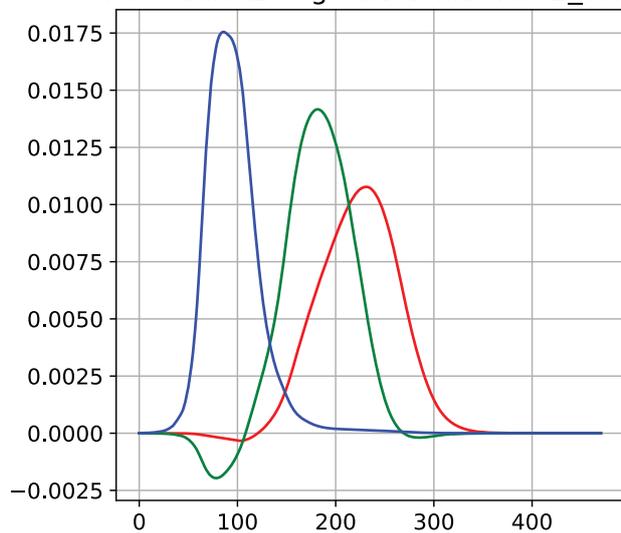
Die Transformationsmatrix ist diejenige, die als erstes bei CIECAM02 verwendet wird. Eigentlich sollten nun bei den Normvalenzen alle negativen Beiträge verschwinden. Anscheinend tun sie das nicht und die negativen Beiträge werden später herausgerechnet. Anscheinend sind die Beiträge so verschwindend, dass sie später herausgerechnet werden können.

```
In [18]: Valenzen = cmfs.values/100
ValenzenFair = []
print(colour.CAT02_CAT)
for row in Valenzen:
    #row = colour.utilities.dot_vector(colour.CAT02_CAT, row)
    row = np.dot(colour.CAT02_CAT, row)
    ValenzenFair.append(row)
ValenzenFair = np.array(ValenzenFair)
#print(ValenzenFair)
%pylab inline
%config InlineBackend.figure_formats=['svg']
pylab.rcParams['figure.figsize'] = (5, 5)
#pylab.xlabel(self.Xachse)
#pylab.ylabel(self.Yachse)
#pylab.title(self.Titel) n.activation_function
pylab.title("Normvalenzen gefaltet mit CAT02_CAT")
pylab.grid(True)
plt.plot(ValenzenFair.T[0])
plt.plot(ValenzenFair.T[1])
plt.plot(ValenzenFair.T[2])
#startx, endx = 0, 2
#starty, endy = 0, 0.1
#plt.axis([startx, endx, starty, endy])
plt.show()
pylab.title("Normvalenzen CIE-XYZ")
pylab.grid(True)
plt.plot(Valenzen.T[0])
plt.plot(Valenzen.T[1])
plt.plot(Valenzen.T[2])
plt.show()
```

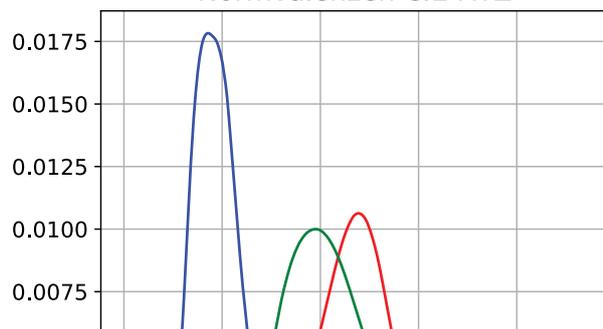
```
[[ 0.7328  0.4296 -0.1624]
 [-0.7036  1.6975  0.0061]
 [ 0.003   0.0136  0.9834]]
```

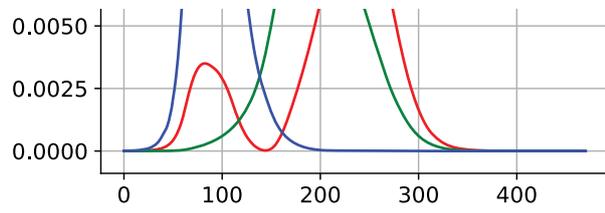
Populating the interactive namespace from numpy and matplotlib

Normvalenzen gefaltet mit CAT02_CAT



Normvalenzen CIE-XYZ





In []:

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 12.2.2020

Christian Greim